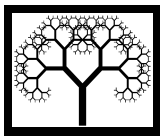
Paper 57



©2002, Civil-Comp Ltd., Stirling, Scotland
Proceedings of the Sixth International Conference on
Computational Structures Technology,
B.H.V. Topping and Z. Bittnar (Editors),
Civil-Comp Press, Stirling, Scotland.

3D Simulation of Concrete Cracking: Probabilistic Formulation in a Parallel Environment

C.N.M. Paz†, L.F. Martha†, E.M.R. Fairbairn‡, J.L.D. Alves‡
N.F.F. Ebecken‡ and A.L.G.A. Coutinho‡
† Department of Civil Engineering, Technology Group on Computer Graphics
Pontifical Catholic University of Rio de Janeiro, Brazil
‡ Program of Civil Engineering
Federal University of Rio de Janeiro, Brazil

Abstract

This work presents a probabilistic crack approach based on the Monte Carlo method, implemented in a 3D fully parallelized finite element code (Paz, [1]). The cracking scheme used is the discrete crack approach. In this approach the heterogeneity of the material is taken into account by considering the properties to vary spatially following a normal distribution determined by the mean and the standard deviation of the considered material properties.
Fracturing is modeled by 3D interface elements generated in a previously defined region within the mesh (Paz [1]). The interface elements are triangular base prisms connecting adjacent faces of neighboring tetrahedra. These elements simulate crack opening through relative displacements between the triangular faces.

Keywords: Parallel processing, high performance computing, discrete cracking concrete, probabilistic crack approach, material heterogeneity, size effects, tensile strength domain, Monte Carlo method, non linear analysis, finite elements.

1 Introduction: probabilistic model

Concrete cracking depends on several relevant factors such as water/cement ratio, casting and curing, loading conditions, etc. Due to the random distribution of constituents and initial defects, 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 developed by Rossi and co-workers [2,3,4 and 5] takes this stochastic process into account by assigning in finite elements analysis, randomly distributed material properties (tensile strength, Young's modulus) to both the solid elements and the contact elements. 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 material behavior

in concrete is assumed to obey a perfect elastic brittle behavior, so that the random distribution of local cracking energies can be replaced by a random distribution of local strengths. Therefore, solid elements are elastic, while interface elements are considered elastic-brittle.

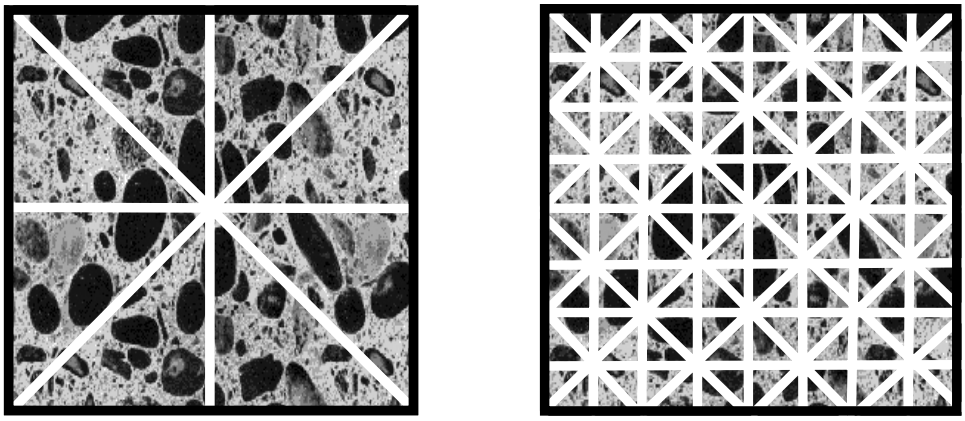


Figure 1: This stochastic process into account by assigning in finite elements analysis, randomly distributed material properties.

Thus, the present probabilistic model involves a number of mechanic properties of the material to be determined, which constitutes the modeling data. From a large number of direct tensile tests it was found that a normal law describes rather well the experimental distribution (Rossi *et al.* [3]). These characteristics are: $f_{ct,\mu}$ and E_μ, the means of the tensile strength and of the Young's modulus respectively; $f_{ct,\sigma}$ and E_σ, the standard deviations of the tensile strength and of the Young's modulus respectively. The following analytical expressions were proposed:

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

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

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 average Young's modulus that does not exhibit significant volume effects. For cylinder specimens, whose dimensions are 160 mm in diameter and 320 mm high constants *a, b* and *c* are related to the compressive strength f_c given by the relations,

$$\begin{aligned} a &= 0.25 - 3{,}6\times10^{-3}(f_c) + 1.3\times10^{-5}(f_c)^2 \\ b &= 4.5\times10^{-2} + 4.5\times10^{-3}(f_c) - 1.8\times10^{-5}(f_c)^2 \\ c &= 0.116 + 2.7\times10^{-3}(f_c) - 3.4\times10^{-6}(f_c)^2 \end{aligned} \tag{3}$$

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.

Equations (1) to (3) show that the smaller the scale of observation, 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.

Let m_v the number of tetrahedra and m_i the number of interface elements in a given mesh. Also let the set of all Young's modulus of the solid elements be denoted by $\mathbf{E}$ and $\mathbf{f}_{ct}$ the at of all tensile strength of interface elements

According to Rossi *et al*. [3] these individual local tensile strengths and Young's modulus are represented by normal distributions having the densities:

$$g_f(f_{ct}) = \frac{1}{f_{ct,\sigma}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{f_{ct,\mu}}{f_{ct,\sigma}}\right)^2\right] \tag{4}$$

$$g_E(E) = \frac{1}{E_\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{E - E_\mu}{E_\sigma}\right)^2\right] \tag{5}$$

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, and 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 m_i values $f_{ct,ii}$, each value corresponding to an interface element, and m_v values E_{iv}, each value corresponding to a volume element, by using a standard routine for generation of random numbers for a given normal distribution (Press *et al*., [6]).

The problem with this approach is that these statistical moments are not known a priori for the characteristic volume of the finite elements used in the analysis. However, some methods have been proposed to determine these parameters by means of inverse analysis using neural networks [1, 7, 8, 9,10].

The solution for this probabilistic approach is obtained by means of a Monte Carlo simulation. As depicted in figure 2, in a Monte Carlo simulation a number of *n* samples are generated for a given normal distribution and some characteristic responses of the structure. A characteristic response may be for example a stress crack-width $\sigma - w$ curve. Let the *j*th samples correspond to the *j*th $\sigma - w$ curve. This *j*th $\sigma - w$ curve is composed of discrete values, σ_k^j and w_k^j, where the superscript *j* indicates the sample and the subscript *k* the discrete value of the $\sigma - w$ curve. The discrete w values are assumed known and the response is defined exclusively by the values of σ_k^j . The mean curve composed by pairs σ_k^{mean}, w_k then simply reads

$$\sigma_k^{mean,j} = \frac{1}{j}\sum_{l=1}^{j}\sigma_k^l \tag{6}$$

The Monte Carlo simulation is stopped when

$$\left|\sigma_k^{mean,j} - \sigma_k^{mean,j-1}\right| \le tol \tag{7}$$

where *tol* is the prescribed tolerance to check the convergence of the procedure. Where convergence is reached the 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 accuracy required by the analysis. It also depends on the heterogeneity of the material represented by the standard deviation. The more heterogeneous is the material the greater is the number of samples necessary to obtain a converged solution. Our experience in this field indicates that 25 to 50 samples are sufficient to obtain a converged $\sigma - w$ curve.

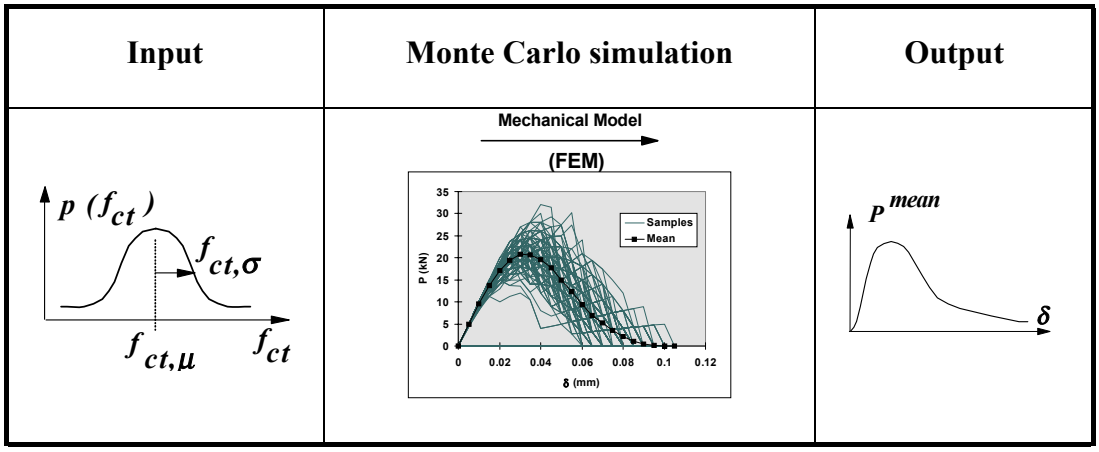


Figure 2: Monte Carlo simulation.

2 Discrete cracking: 3D Interface elements

The finite element cracking model is a discrete model for which volume elements are always elastic and cracking occurs in elastic-brittle (almost rigid brittle) contact elements placed between two neighbor surfaces of the volume elements. The 3D interface elements (Paz [1]) depicted in Figure 3 (a) can be thought as triangular base prisms connecting adjacent faces of neighboring tetrahedra.

These elements are formulated to represent relative displacements between the triangular faces to simulate crack opening.

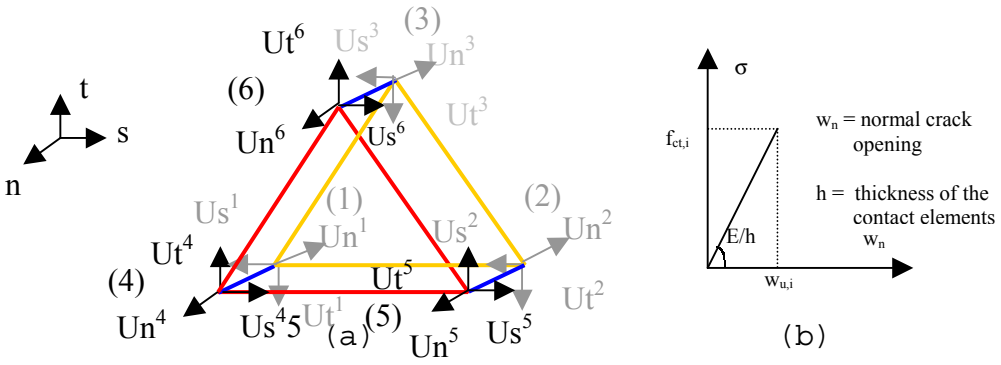


Figure 3: (a) An interface element and its degrees of freedom in a local system; (b) Elastic-brittle contact law.

The constitutive law of the 3D interface element is defined by equation (8) for a non cracked elastic state, characterized by $\sigma_n < f_{ct,i}$. When the tensile strength is exceeded, the elements reach a cracked stage and module E_c and G_c are set to zero as shown in figure 3b.

$$\Delta\boldsymbol{\sigma} = \mathbf{D}_{cr}\ \Delta\mathbf{w} = \begin{Bmatrix} \Delta\sigma_n \\ \Delta\sigma_s \\ \Delta\sigma_t \end{Bmatrix} = \begin{bmatrix} E_{c/h} & 0 & 0 \\ 0 & G_c/h & 0 \\ 0 & 0 & G_c/h \end{bmatrix} \begin{Bmatrix} \Delta w_n \\ \Delta w_s \\ \Delta w_t \end{Bmatrix} \tag{8}$$

In equation (8) the subscripts *n, s* and *t* indicate the directions normal and transversal to the crack plane respectively, *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 the longitudinal (Young's) and the transversal modulus respectively.

Equation (8) and figure 3(b) define the elasto-brittle constituve behavior. However, it can be considered rigid-fragile, since thickness *h* of the interface element is considered very small (less than a value h_{lim}). In this way the modulus E_c and G_c in equation (8) do not have a physical meaning and the terms E_c/h and G_c/h tend to infinity.

The kinematic relation for the interface element is given by:

$$\Delta\mathbf{w} = \mathbf{B}\,\Delta\mathbf{a}_l^e \tag{9}$$

The stiffness matrix for the interface element is by:

$$\mathbf{K}_{\text{Intf}}^{\text{e}} = \int_{\Omega} \mathbf{B}^T\ \mathbf{D}_{cr}\mathbf{B}\ \mathrm{d}\Omega \tag{10}$$

3 3D 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. The interface elements are automatically generated (Paz [1]) contiguous to the faces of the selected tetrahedra.

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 neighbor 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 figure 4a) and, for the other elements that share this node, it is introduced a new node numbering (see figure 4b) 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 figure 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.

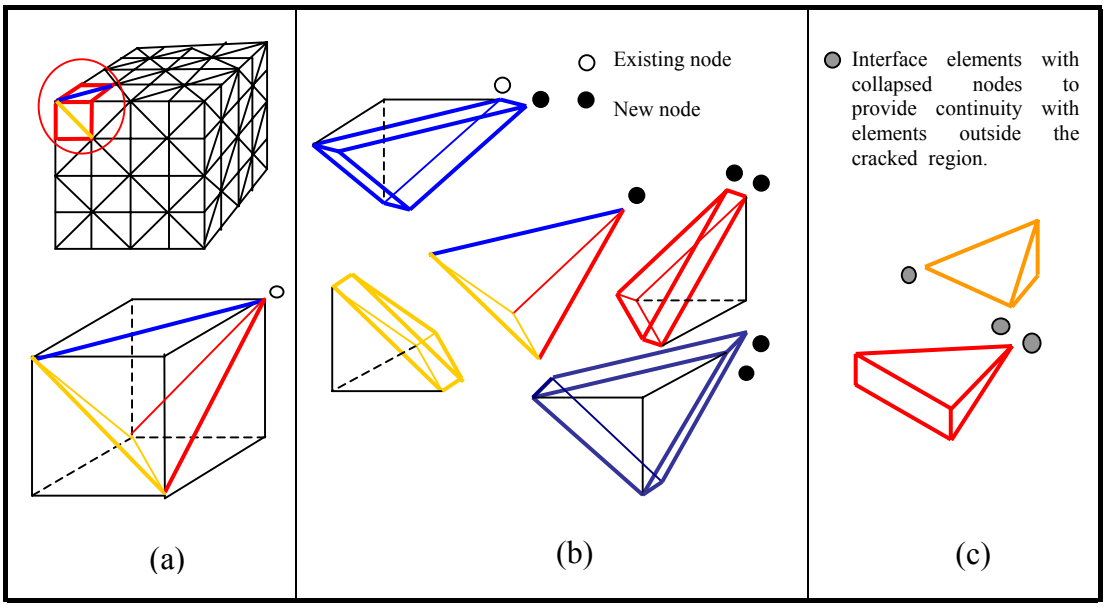


Figure 4: 3D Interface Elements mesh generation.

4 Implementation code strategies, parallel vector processor (PVP)

4.1 Solution of equilibrium equations and the inexact Newton Method

Traditional finite element technology for non linear problems involves the repeated solution of systems of sparse linear equations by a direct solution method, that is, some variant of Gauss elimination. The updating and factorization of the sparse global stiffness matrix can result in extremely large storage requirements and a very large number of floating point operations.

In this paper we employ an Inexact Newton method (Kelley [11] , Coutinho[12]), to solve large-scale three-dimensional incremental elastic-brittle problems. In the Inexact Newton Method, at each non linear iteration, a linear system of finite element equations is approximately solved by the preconditioned conjugate gradient method (PCG).

The implementation of global matrix-vector products are easily parallelized in different computer architectures, performing element level products followed by global assembly. This type of implementation is often referred to element-by-element (EBE) schemes. Matrix-vector products computed by EBE schemes are memory intensive, requiring more operations than the product with the assembled matrix, because element matrices have many overlapping non-zero entries. However, particularly for large scale non linear problems EBE methods have been very successful, because they handle large sparse matrices in a simple and straightforward manner. Besides, efficient preconditioners may be derived keeping the same data structure. For a recent review of such topics see Hughes [13].

When solving iteratively the finite element system of linear equations, it is straightforward to employ inexact versions of the standard Newton-like methods. In this case, tolerances for the inner iterative driver may be adaptively selected to minimize computational effort towards the solution, giving rise to the following algorithm:

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.

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 $\mathbf{F}_{\text{int}}^{i}$ is the sum of the solid elements internal forces $\left(\mathbf{F}_{\text{int}}^{i} \right)_{Tetra}$ plus the interface element internal forces vector $\left(\mathbf{F}_{\text{int}}^{i} \right)_{Intf}$. The total stiffness matrix is the sum of the continuum matrix $\mathbf{K}_{Tetra}$ plus the interface matrix $\mathbf{K}_{Intf}^{i}$ updated at each non linear iteration.

We adopted a simple nodal block-diagonal preconditioner. Therefore, the most expensive computational kernel in the linear solver is the matrix-vector product. According to the above algorithm, an approximate solution is obtained when the Inexact Newton termination criterion is satisfied, that is, when,

$$\| \mathbf{A}^{i} \ \Delta\mathbf{u} - \Psi^{i} \| \le \eta_i \| \Psi^{i} \| \qquad (11)$$

We selected η_i as suggested by Kelley [11], based on a measure of how far the nonlinear iteration is from the solution, that is,

$$\eta_i^{A} = \min\left(\eta_{\max}, \gamma \frac{\| \boldsymbol{\psi}^{i} \|^2}{\| \boldsymbol{\psi}^{o} \|^2} \right) \ , \qquad 0 < \gamma < 1 \qquad (12)$$

If η_i^{A} is uniformly limited away from 1, and taking $\eta_i = \max\left(\eta_{\min}, \eta_i^{A}\right)$ Kelley [11] has shown general convergence properties when Equation (12) is used. To

avoid that η_i^A be too small when the nonlinear iteration is away from the solution, Kelley also suggests the following modification,

$$\eta_i^B = \begin{cases} \min\left(\eta_{\max}, \eta_i^A\right) & ,if\ \gamma\,\eta_{i-1}^2 < 0.1 \\ \min\left(\eta_{\max}, \max\left(\eta_i^A, \gamma\,\eta_{i-1}^2\right)\right) & ,if\ \gamma\,\eta_{i-1}^2 \geq 0.1 \end{cases} \tag{13}$$

In some cases $\|\boldsymbol{\psi}_i\|$ can be very small, well beyond the required accuracy, resulting in undesired work. To remedy this over solving Kelley [11] proposes to compute η_i^C using;

$$\eta_i^C \left(\eta_{\max}, \max\left(\eta_i^B, 05\ rtol\ \frac{\left\| \mathbf{F}_{ext}^k \right\|}{\left\| \boldsymbol{\psi}^i \right\|} \right) \right) \tag{14}$$

and finally taking $\eta_i = \max\left(\eta_{\min}, \eta_i^C\right)$.

Our experience indicates that selecting $\eta_{\max}, = 0.1\ and\ 10^{-3} \leq \eta_{\min} \leq 10^{-6}$ for *utol* and *rtol* in the usual range, that is, 10^{-3} to 10^{-2}, is enough for practical engineering computations. Typical values for γ and *g* are 0.5 and 0.1, respectively.

4.2 Matrix-vector products element-by-element, EBE

In the element-by-element EBE matrix-vector product, the matrix $\mathbf{A}$ it is never formed. Rather, the product is computed as:

$$\mathbf{A}\,\mathbf{p} = \sum_{e=1}^{Nel} \mathbf{A}_e \mathbf{p} = \sum_{i=1}^{N_{tetra}} \left(\mathbf{K}_{Tetra} \quad \mathbf{p}_i \right) + \sum_{i=1}^{N_{Intf}} \left(\mathbf{K}_{Intf} \quad \mathbf{p}_i \right) \tag{15}$$

where *Nel* is the number of elements in the mesh, N_{tetra} is the number of tetrahedra, N_{intf} is the number of interface elements, $\mathbf{A}_e$ are the element matrices for the tetrahedra and interface; $\mathbf{p}_e$ the components of $\mathbf{p}$ restricted to the degrees of freedom of the element. The arrays of the element stiffness matrices are stored taking into account their symmetry; in the case of tetrahedra 78 coefficients are stored and for the interface element only 18 coefficients are stored, exploring the particular structure of the discrete gradient operator .

Note that, during the nonlinear iterations, only the interface elements stiffness matrices should be updated.

The mesh coloring algorithm Hughes [13] was extended in order to block both solid and interface elements into of disjoint groups thus enabling full vectorization and parallelism of the operations involved in equation (15).

5 Numerical examples

5.1 Uniaxial tension test - Comparasion with experimental data

The experimental results of concrete uniaxial tension tests published by (Li *et al* [14]) were used to illustrate the developments presented in this paper. The specimens are cylinders 101.6 mm in diameter and 203.2 mm high. This specimens had 25.4 mm notches at their midheight on both sides (Figure 5).
Concrete with maximum aggregate diameter of 9.525 mm was used. Its average tensile strength and Young's modulus at the age of 28 days were: $f_{ct} = 4{,}72 \quad MPa$ and $E_c = 42000 \quad MPa$. This model was created by Mesh Generation (MG) modeler (Coelho *et al* [15]) using J-mesh algorithm (Cavalcante *et al* [16]). Interface elements were later included in the model using the algorithm developed in Paz [1].

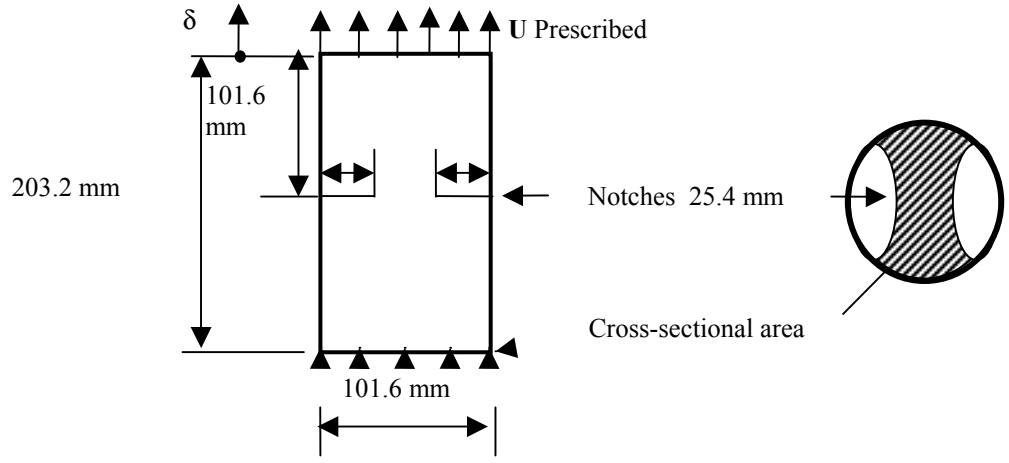


Figure. 5: Uniaxial tension specimen geometry, dimensions, load, boundary conditions and place of the measurement of δ.

The numeric experiments were controlled by a field of uniform displacements applied at the upper end of the test specimen. The boundary conditions restrain the degrees of freedom in the vertical at the lower end and place of the measurement of δ (Figure 5). The mesh of tetrahedra is shown in Figure 6.

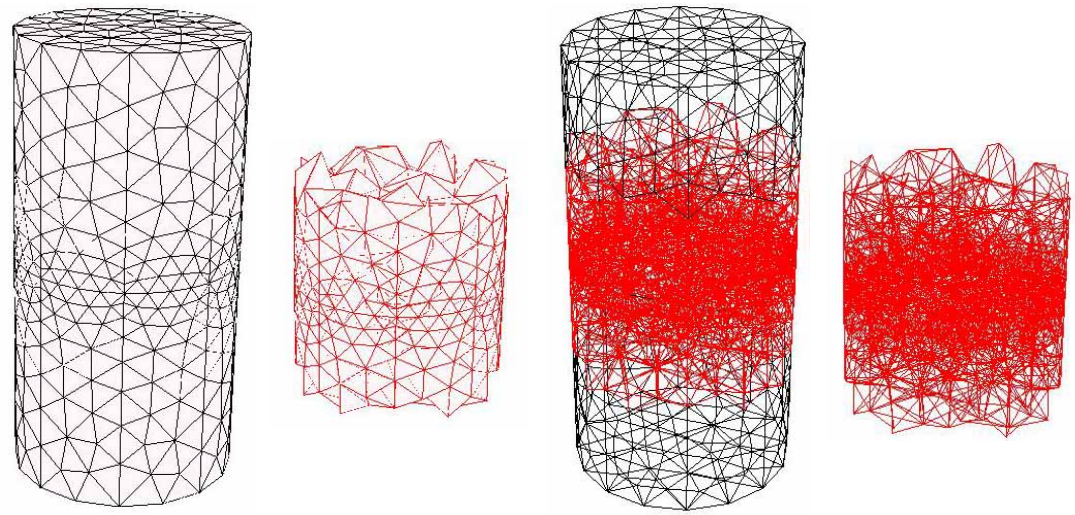


Figure 6: Mesh tetrahedra with interface elements and detail of the mesh of interface elements.

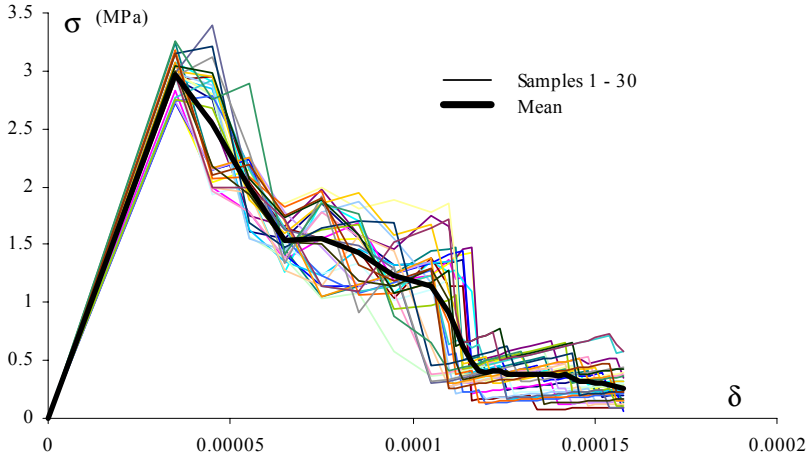


Figure 7: Monte Carlo simulation.

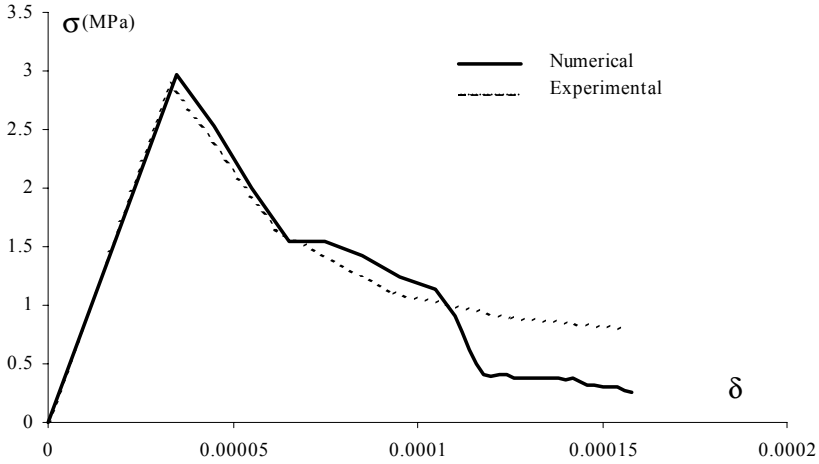


Figure 8: Comparison of experimental and numerical results.

Curves $\sigma - \delta$ for the several samples of the Monte-Carlo simulation are given in figure 7 (place of the measurement of δ (Figure. 5)). The comparison between experimental results by Li *et al* [14] and the converged Monte Carlo (30 samples) $\sigma - \delta$ curve is given in figure 8. With the experience gotten in the works [1, 7, 8, 9 and 10] we are using inverse analysis to introduce factor 2 that multiplies $f_{ct,\mu}$ of Equation (1).

Figure 9 presents the crack configuration for a given sample at a stage corresponding to the softening branch of the $\sigma - \delta$ curve.

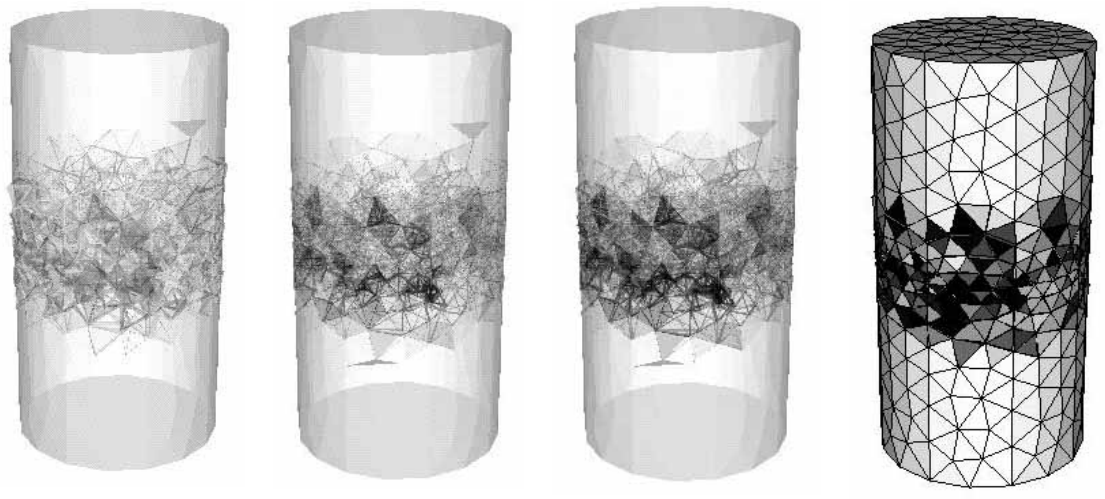


Figure 9: Crack evolution.

5.2 Beam test - Comparasion with experimental data

A notched, plain concrete three-point bending beam test carried out by Amparano *et al* [17]. The geometrical details of the test are shown in figure 10 and figure 11 with the mesh of tetraedra and interface elements. The thickness of the beam is 63.5 mm, which was determined by considering the maximum aggregate size 19 mm. To examine the effect of maximum aggregate content, a volume fraction of 55% of aggregate to total concrete volume was considered. Tests on specimens made with this concrete indicated the following average characteristics again at (28 days) the age at test: $f_{ct} = 3{,}45\,MPa$ and $E_c = 10500\,MPa$. Again this model was created by Mesh Generation MG modeler (Coelho *et al* [15]) using J-mesh algorithm (Cavalcante *et al* [16]). Interface elements were later included in the model using the algorithm developed in Paz [1].

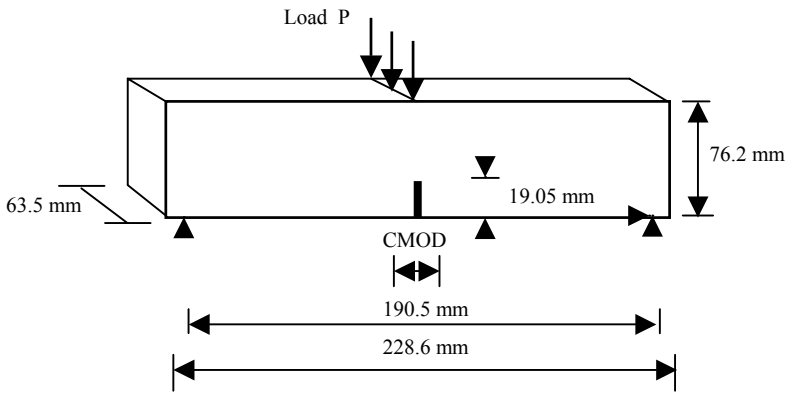


Figure 10: Specimen geometry for the three-point bend beam.

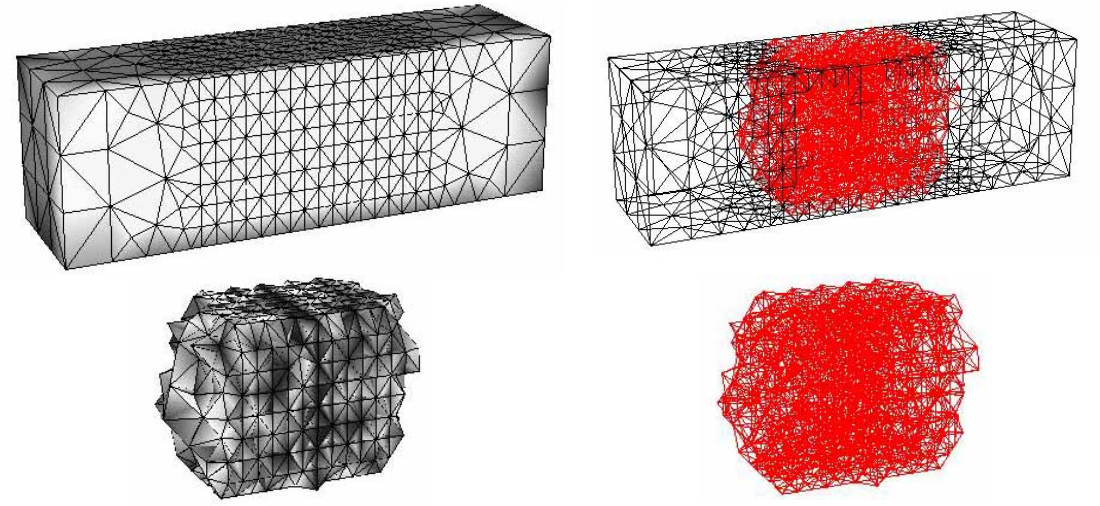


Figure 11: Mesh of tetraedra with interface elements and detail for the mesh of interface elements.

The comparison between experimental results by Amparano *et al* [17] and the converged Monte Carlo (40 samples) Load-CMOD curve is given in figure 12. Typical Load-CMOD (Crack Mouth Opening Displacement) curves, obtained from numerical simulation, and comparison of experimental and numerical results are shown in figure 13.

Figure 14 presents the crack configurations for given sample at a stage corresponding to de softening branch of the Load-CMOD curve.

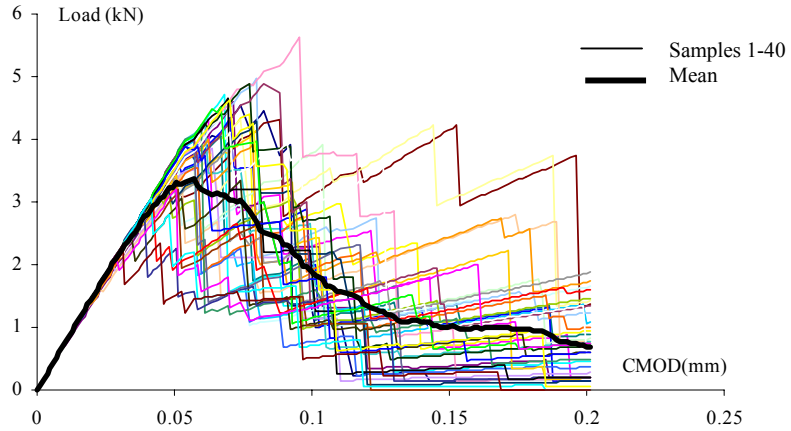


Figure 12: Monte Carlo simulation.

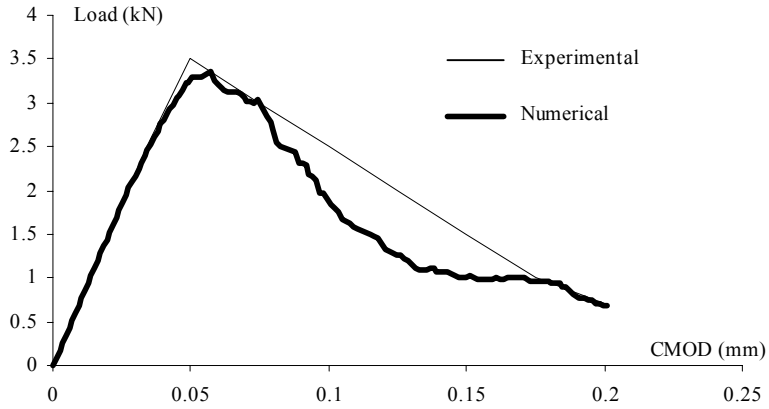


Figure 13: Comparison of experimental and numerical results.

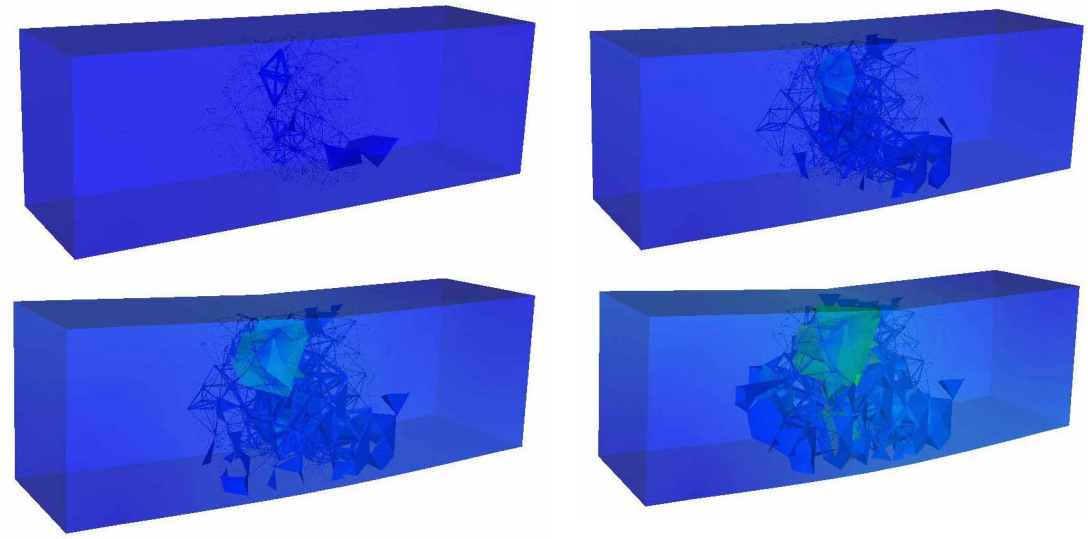


Figure 14: Crack evolution.

6 Computational Performance

Vector performance analysis was obtained using the program summary provided by PERFVIEW's Report as shown in table 1 and figures 15 and 16. The CPU time of the vectorized single processor run for CRAY T90 are 34.87 and 33.99 hours (examples 1 and 2 respectively). This table relates the single CPU utilization to the Mflop/s rates for the three top routines. The routines **Smatv-fint** and **Smatv-tetra** are responsible respectively for the matrix-vector operations on the interface elements and tetrahedra elements, these multiplications are needed in the routine **PCG-block,** the iterative driver implementing the nodal block diagonal preconditioned conjugate gradient method (PCG).

ROUTINES	EXAMPLE 1		EXAMPLE 2	
	Single CPU (%)	Performance (Mflop/s)	Single CPU (%)	Performance (Mflop/s)
smatv-intf	52.80	613.7	52.50	597.3
smatv-tetra	17.32	554.0	25.40	536.8
PCG-block	25.90	82.4	19.10	82.2
Others	3.98	-	3.00	-

Table 1: Performance Analysis -The top 3 subroutines - The 3 examples.

N° Example	N° Elements	N° Eelements tetrahedra	N° Elements Interface
Example 1	11933	4775	7158
Example 2	8750	4331	4419

Table 2: Number of elements the meshes.

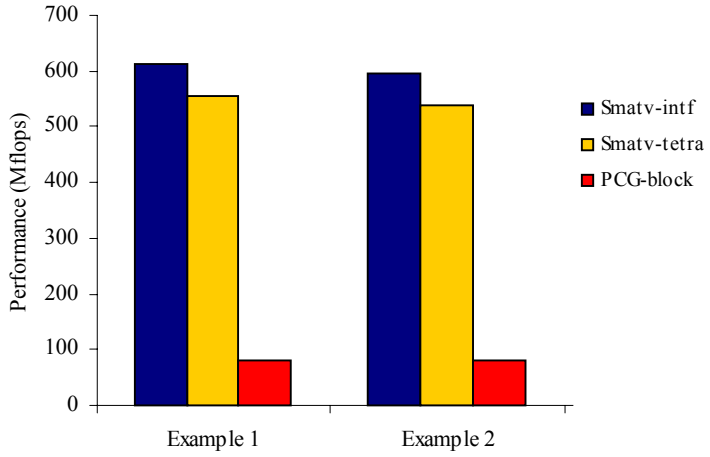


Figure 15: Performance Analysis -The top three subroutines - The 2 examples.

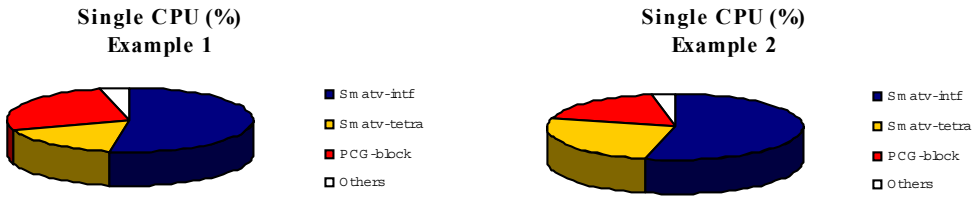


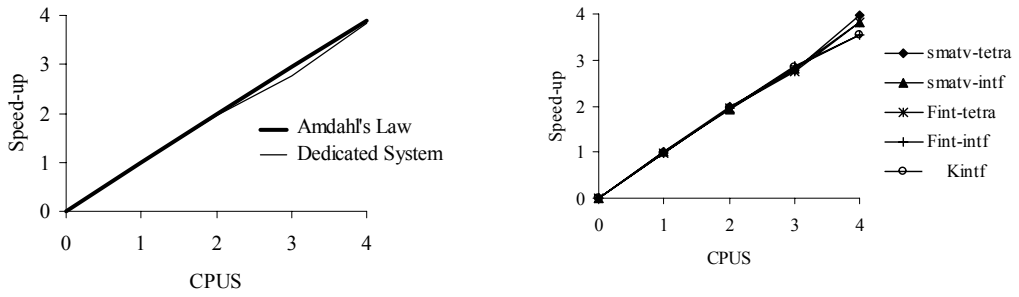
Figure 16: Performance Analysis - Single CPU (%) -The top three subroutines.

The code achieved good vectorization on the CRAY T90 for a 2 examples show in table 1 The top three subroutines are responsible for major CPU utilization in the whole analysis.

The parallel performance is shown in table 4 and the figures 17 (a) and (b) give a summary report provided ATEXPERT, the autotasking performance tool. The top five subroutines assigned for parallel analysis are presented in table 3 for the uniaxial tension test. The routines **Fint-tetra** and **Fint-intf** evaluate respectively the internal force vector of the interface and tetrahedra elements. The routine **Kintf** computes the stiffness matrices for the interface elements.

Routines	% Parallel	Dedicated Speedup	Actual Speedup
Smatv-tetra	98.9	3.96	3.8
Smatv-intf	99.8	3.83	3.8
Fint-tetra	92.5	3.83	3.2
Fint-intf	99.1	3.55	3.5
Kintf	85.8	3.53	2.6

Table 3: Summary of the ATEXPERT's Report for the 5 dominant loops.



(a) (b)

Figure.17: (a) Program Summary of the ATEXPERT's Report;
(b) ATEXPERT's Report -The top 6 subroutines on a Dedicated Speedup.

According to ATEXPERT tool this program appears to be 99.2 percent parallel and 0.8 percent serial. Amdahl's Law predicts the program could expect to achieve a 3.9 times speedup on 4 cpus.

7 Concluding remarks

This paper presented the optimized implementation of Rossi's a probabilistic model for the simulation of cracking in concrete structures. This model in 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 cracks. In adition, the finite elements analysis must be called several times within a Monte Carlo simulation. Therefore, the code needs to be optimized in such a way that the simulation time does not exceed a practical limit. The example presented in this paper shows that the model is capable of simulating the crack opening and the crack pattern.
The code achieved a very good level for both parallel performance and vetorization. The most demanding routines, which implement the matrix-vector-multiply computational kernel for the interface and tetrahedral elements, are "fully" parallelized (~ 99%) and responsible for over 80% of CPU time. The results emphasizes the suitability of the implemented code on the parallel-vector machine, CRAY T90 for 2 CPU's, which presented a flop rate of 614 Mflop/s and a parallel speed-up of 3.8 for 4 CPU's.
Extensive use of element-by-element techniques within the computational kernels comprised in the iterative solution drivers provided a natural way for achieving high Flop rates and good parallel speed-up's. Furthermore, element-by-element techniques, avoid completely the formation and handling of large sparse matrices. Therefore, the computational strategies presented herein provide a natural way to deal with more complex scenarios, particularly those involving three-dimensional problems.

Acknowledgments

The authors are indebted to the Computer Graphics Technology Group TECGRAF/PUC-Rio, High Performance Computing Center NACAD/COPPE/UFRJ, and the Laboratory of Computational Methods in Engineering of the Program of Civil Engineering LAMCE /COPPE/UFRJ. CESUP/UFRGS is gratefully acknowledged for the computer time provided in the CRAY T90. This work was partially supported by CAPES and CNPq grants N^{o} 150039/01-8(NV).

References

[1] C. N. M. Paz, "Development and Implementation Probabilistic Model for 2D and 3D Discrete Cracking Concrete in Parallel Computing", D.Sc. Thesis, Dept, of Civil Engineering, COPPE/UFRJ, Federal University of Rio de Janeiro, Brazil [in Portuguese] 2000.

[2] P. Rossi, X. Wu, F. le Maou, and A. Belloc, “Scale effect on concrete in tension”, Materials and Structures, 27 (172), 437-444, 1994.
[3] P. Rossi, X, Wu, F. le Maou, and Belloc, “Scale effect on concrete in tension “, Materials and Structures, 27 (172), 437-444, 1994.
[4] P. Rossi, F.-J. Ulm, and F. Hachi, “Compressive behavior of concrete: physical mechanisms and modeling”, Journal of Engineering Mechanics ASCE, 122 (11), 1038-1043, 1996.
[5] P. Rossi and F.-J.Ulm, “Size effects in the biaxial behavior of concrete: physical mechanisms and modeling”, Materials and Structures, 30 (198), 210-216, 1997.
[6] W. H. Press, S. Teukolski, W.T. Vetterling and B. Flannery, “Numerical Recipes”, Cambridge University Press,1992.
[7] E.M.R. Fairbairn, N.F.F. Ebecken, E. Goulart and C.N.M. Paz, “Probabilistic modelling of Concrete cracking using Monte Carlo and Neural networks to solve the inverse problem”, Computational Structures Technology 1998 & Engineering Computational Technology 1998 (Edinburgh), Editor: B.H.V. Topping, Advances in Engineering Computational Technology – Neural computing for engineering computations, ISBN 0-948749-55-5, 215-220, 1998.
[8] E. M. R. Fairbairn, C. N. M. Paz, N. F. F. Ebecken, and F-J. Ulm, “Use of neural network for fitting of probabilistic scaling model parameter”, Int. J. Fracture, 95, 315-324, 1999.
[9] E. M. R. Fairbairn, N. F. F. Ebecken, C. N. M. Paz and F-J. Ulm, “Determination of probabilistic parameters of concrete: solving the inverse problem by using artificial neural networks”, Computers and Structures, 78, 497-503, 2000.
[10] E. M. R. Fairbairn, V.J.C. Debeux, C. N. M. Paz and N. F. F. Ebecken, “Applications of probabilistic Aproach to the Analisysis of gravite Dam Centrifuge”, Test, 8th ASCE Specialty Conference on probabilistic Mechanics and Structural Reability. 24-26 July, USA, CD ROMM Proceedings PMC 2000-216, 2000.
[11] C. T. Kelley, “Iterative Methods for Linear and Nonlinear Equations”, Frontiers in applied mathematics, SIAM˙ Society for Industrial and Applied Mathematics, Philadelphia, 1995.
[12] A.L.G.A.Coutinho, M.A.D. Martins, J.L.D. Alves, L. Landau, and A. Moraes, “Edge-based finite element tecniques for nonlinear solid mechanics problems”, Int. J. for Numerical Methods in Engineering, 50 (9), 2050-2068, 2001.
[13] T. J. R. Hughes,”Algorithms for parabolic problems, Element-by-element (EBE) implicit methods. In: The Finite element method analysis”, Chaper 8, p 483-489, New Jersey: Prentice - Hall, 1987.
[14] Q. Li and F. Ansari, “High Concrete in Uniaxial Tension”, ACI Material J. 97 (1), 49- 57, 2000.
[15] L.C.G. Coelho, M. Gattass and L.H. de Figueiredo, “Intersecting and Trimming Parametric Meshes on Finite-Element Shells”, International Journal for Numerical Methods in Engineering; 47 (4), 777-800, 2000.

[16] J.B. Cavalcante Neto, P.A. Wawrzynek, M.T.M. Carvalho, L.F. Martha and A.R. Ingraffea, “An Algorithm for Three-dimensional Mesh Generation forArbitrary Regions with Cracks”, Engineering with Computers; 17 (1), 75-91, 2001.

[17] F.E. Amparano, X.Yunping and R. Young-Sook, “Experimental stydy on the effect of aggregate contents on fracture behavior of concrete”, Engineering Fracture Mechanics 67, 65-84, 2000.

[18] C.O. Moretti, J.B Cavalcante Neto, T.N. Bittencourt and L.F. Martha, “Parallel Environment for thee-Dimensional finite Element Analysis”, Developments in Engineering Computational Technology, Edited by: B.H.V. Topping, Civil-Comp, Press, Edinburgh, UK, ISBN 0-948749-70-90, 283-287, 2000.

[19] C. N. M. Paz, L.F. Martha, E. M. R. Fairbairn, J.L.D. Alves, N. F. F. Ebecken and A.L.G.A.Coutinho “Parallel implementation and Development of a probabilistic Model for 3D Discrete Concrete Cracking”, accepted to WCCM Fifth Congress on Computational Mechanics July 7-12, Vienna, Austria Eds: H.A. Mang, F.G. rammerstorfer, J. Eberhardsteiner. ISBN 3-9501554-0-6, 2002.

[20] C. N. M. Paz, L.F. Martha, E. M. R. Fairbairn, J.L.D. Alves, N. F. F. Ebecken and A.L.G.A.Coutinho “Probabilistic model 3D discrete cracking concrete in parallel computing”, accepted to VECPAR 2002, 5th International meeting, high performance computing for computational science, University of Porto FEUP, Porto, Portugal June 26-28, 2002.