Performance Assessment for a Family of Continuum-Based Shell Elements for Nonlinear Finite Element Analysis in Shared-Memory Systems

José Luis Drummond Alves and Elizabeth Ferreira da Silva

jalves@lamce.ufrj.br; beth@lamce.ufrj.br
LAMCE – Laboratório de Métodos Computacionais em Engenharia, Programa de Engenharia Civil COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Caixa Postal 68552, CEP 21945-900 Rio de Janeiro, RJ – Brazil

Abstract. This work presents an assessment of some proposals for improvements on members of the C^0 quadrilateral continuum-based (CB) shell elements originally developed by Belytschko et al., 1984, 1989, 1992, for explicit type nonlinear finite element analysis (NL-FEA). These elements are extensively available in high performance commercial codes that implement nonlinear finite element analysis, notably for industrial applications where computational models of practical interest may typically comprise hundred thousand elements as a rough measure of problem dimensions. Reportedly, members of this element family perform very differently, considering overall computational performance, accuracy and robustness. In this work, proposals for new members in this family are derived and each one of these members is assessed for solution quality and computational costs involved while addressing several benchmark problems. A special implementation technique, namely the "lamina-by-lamina" approach, is introduced for members of this family. This approach enables improvements and robustness for every element in the family at marginal additional cost. Computational performance was evaluated in shared memory PVP (parallel vector processor) systems, such as T90 and SV1, installed respectively at CESUP/UFRGS and NACAD/COPPE/UFRJ.

1 Introduction

The members of the Belytschko shell element family, with one-point quadrature per lamina, are suitable for large non-linear transient response such as crashworthiness and sheet metal forming analyses, nowadays a routine task in Computer Assisted Engineering and Design (CAE/CAD) production environments. These elements can be found in many commercial explicit codes for transient solid mechanics such as LS-DYNA, Hallquist et al (1989), PAM-CRASH and ANSYS, among others. Belytschko elements are formulated under a co-rotational strain rate approach providing an straightforward way to handle the complex nonlinear kinematics involved in such problems, whereas Cauchy stress rate is used as the conjugated stress measure. Through-thickness numerical integration of internal forces accounts for the nonlinear material behavior. The transverse shear stresses are computed

considering a linear elastic law, based on Reissner-Mindlin assumption. A perturbed hourglass control is used to overcome the rank deficiency of the resulting discrete operators due to the reduced integration scheme used over the element area. This under-integration technique used in the computation of internal forces for each element reduces bookkeeping requirements and decreases the number of computations involved, thus saving memory and CPU time. Internal forces computation is the most critical and computationally intensive phase of a nonlinear finite element analysis (NL-FEA) and therefore a subject of particular interest for the development of high performance implementations for NL-FEA. The explicit central difference operator is used for time integration of the governing momentum conservation equations.

The BLT element is constructed by combining a flat, 4-node membrane element with a plane quadrilateral 4-node bending element, so warping is not considered. Due to the co-rotational system construction, this system is not exactly normal to the mid-surface at the element centroid. On the other hand, the BWC element is based on the continuum-based approach, and so projection operators deal with transverse shear and bending for in-plane distorted as well as for out-of-plane warped element geometry. The co-rotational system is built to be exactly normal to the mid-surface at the element centroid.

The improved elements based on BLT and BWC elements consider a special projection operator, **P**, allowing shell elements derived on 5 degrees-of-freedom per node to be used in general 6 degrees-of-freedom codes. This projection enforces invariance under rigid body motion and drill rotation, according to Belytschko *et al*. (1994) and Zhu *et al*. (1996). Indeed, this projection depends on the warping and mesh refinement. It is sub-divided into two cases: total projection operator (for high warping) and partial operator (for slight warping).

For a comprehensive review of shell elements, co-rotational theories and the intricacies of the complete formulation, which are out of the scope of this work, the reader can refer to the work of Stolarski et al. (1995); and specifically for element formulations see Belytschko et al., (1989, 1992) and Belytschko et al., (1984), Belytschko *et al*. (1994) and Zhu *et al*. (1996). Overall data structure and code techniques used in this work are addressed in Alves (1991) and Silva (1997, 2003).

Considering all these developments, several currently available commercial codes, endowed with capabilities for dynamic nonlinear analysis of shell structures, implement at least one element of this family. Most often these codes provide the standard BLT element, and in some cases they may provide as alternative the BWC element. There is some controversy about the relative computational cost for each one of these formulations, and eventually the user is left with the task of selecting the element of his convenience and warned about the additional cost involved that might arise from this selection.

In this work we address some of the key aspects of such elements implementation under a special coding strategy, namely the *lamina-by-lamina* approach, which render a uniform framework for these special formulations at little additional cost compared to the standard BLT element, even for the more involved members of the family. This particular approach also renders good vectorization and parallelism which were both exploited in the implementation reported herein. Two computational platforms for the implementation were used for the performance assessment of the resulting codes: the CRAY SV1, a cache-memory/parallel vector processor system with 8 CPUs installed

at the NACAD/COPPE, and the CRAY T90, a shared-memory and parallel vector processor system with 2 CPU's.

The remaining of this work is organized as follows. Sections 2, 3 and 4 are devoted to briefly describe the basics of the element formulations and problem equations, trying to convey a flavor of the computational complexity involved in the problem. The BLT and BWC element co-rotational system are described at section 2; the laminar system, stresses and nodal forces are defined at section 3; the governing equations and the basic formulation concepts common to all elements are presented at section 4. The implemented code structure is treated in Section 5; performance analysis and conclusions are respectively addressed in sections 6 and 7.

2 Co-rotational Reference Systems for Members of the Belytschko Element Family

The co-rotational system is defined based on the nodal coordinates at the mid-surface of the quadrilateral shell element, following element deformation along the analysis. The rotation of the material is accounted according to Belytschko *et al*. (1984) and all computations are carried out at the element centroid.

2.1 Belytschko-Lin-Tsay Element – BLT

The co-rotational unit normal vector, $\hat{\mathbf{e}}_3$, is constructed by the cross product of the main diagonals of the element. The unit vector, $\hat{\mathbf{e}}_1$, is obtained to a vector $\mathbf{r}_1$ which is *nearly* parallel (in warped configurations) to the vector $\mathbf{r}_{21}$, as defined in Fig. 1.

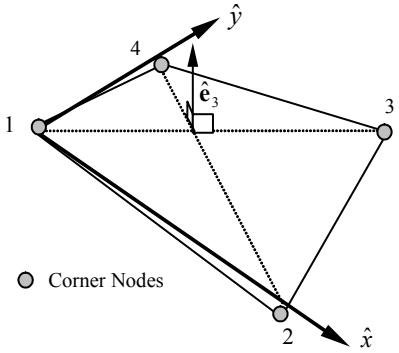

Fig. 1. Reference surface and BLT co-rotational system

The whole set of co-rotational unit vectors are given by the following expressions.

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{r}_{31} \times \mathbf{r}_{42}}{\left\| \mathbf{r}_{31} \times \mathbf{r}_{42} \right\|} \tag{1}$$

where $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{x}_j$ is the vector of nodal coordinates (x,y,z) for node i.

$$\mathbf{r}_1 = \mathbf{r}_{21} - \left(\mathbf{r}_{21}^T \hat{\mathbf{e}}_3\right)\hat{\mathbf{e}}_3 \tag{2}$$

and, finaly

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{r}_1}{\|\mathbf{r}_1\|} \tag{3}$$

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1 \tag{4}$$

2.2 Belytschko-Wong-Chiang Element – BWC

The BWC co-rotational system $(\hat{x}, \hat{y}, \hat{z})$, for the bilinear shell element defined at its mid-surface is shown in Fig 2. The vectors $\mathbf{r}_{ac}$ and $\mathbf{r}_{bd}$ are respectively vectors in the directions of the lines (ac and bd) connecting the mid-side points of the opposite element sides.

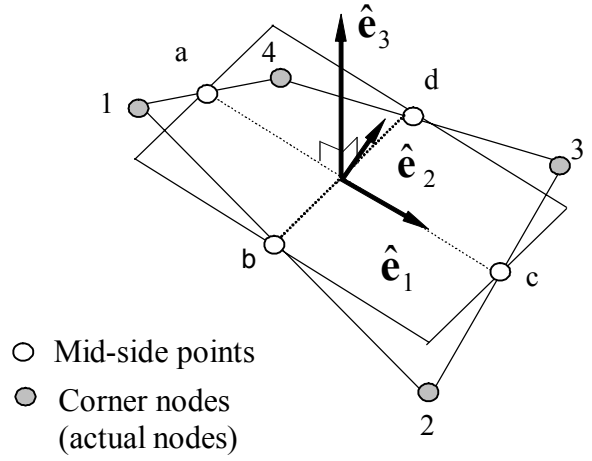

Fig. 2. Reference surface and BWC co-rotational system

The BWC co-rotational unit triad $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_3$ are obtained by:

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{r}_{ac}}{\|\mathbf{r}_{ac}\|} \tag{5}$$

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{r}_{ac} \times \mathbf{r}_{bd}}{\|\mathbf{r}_{ac} \times \mathbf{r}_{bd}\|} \tag{6}$$

The unit vector $\hat{\mathbf{e}}_2$ is obtained from the vector cross product given by the Eq. (4)

3 Laminar System for Stresses and Nodal Forces

To account nonlinear physical material behavior, a laminar system (ξ, η, ζ) is constructed, in a way that the axes: $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$ are tangent to the laminae, and $\hat{\mathbf{e}}_3$ is orthogonal to it. Each surface of constant ζ is called laminae. Figure 3 defines a laminar system at the mid-surface, ($\zeta = 0$). Fibers are lines along ζ axis that connect a point from the bottom to the top surfaces. The fiber is assumed to remain straight.

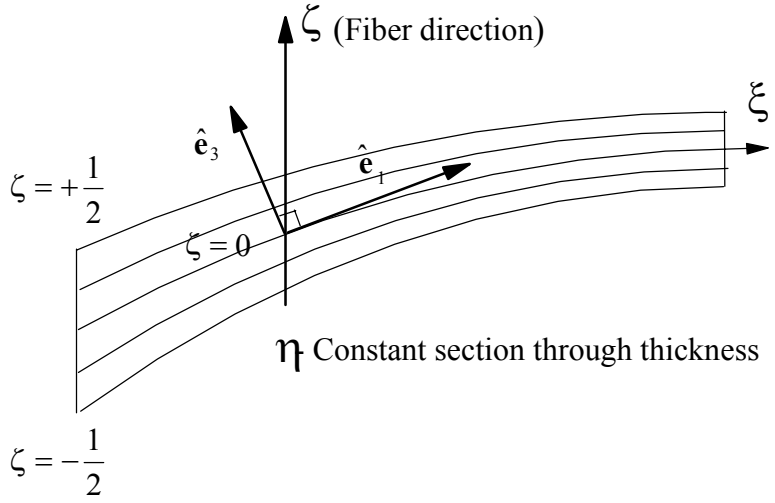

Fig. 3. Definition of laminar system at mid-surface

Reissner-Mindlin assumption is used for transverse shear, in which a projection scheme is employed in order to avoid locking, Belytschko *et al*. (1989). Plane stress assumption is used at each lamina. The strain rates are evaluated at the mid-surface, $\zeta = 0$, at the quadrature point $(\xi = \eta = 0)$, using the quadrilateral bilinear shape functions, Belytschko *et al*. (1989).

For each element, stresses are stored at a pre-set number of sampling points through the thickness; three or five points are sufficient in most cases. These points are located for each lamina at the centroid of the element $(\xi = \eta = 0)$. The stresses are updated at each time step accounting material nonlinear behavior.

The stresses are computed in two blocks: plane stress assumption at each lamina, $\hat{\boldsymbol{\sigma}}'^{\mathrm{T}}$, and an elastic law for the transverse shear, $\hat{\boldsymbol{\sigma}}''^{\mathrm{T}}$, which is assumed constant through the thickness.

$$\hat{\boldsymbol{\sigma}}'^{T} = \left\{\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_{xy}\right\} \tag{7}$$

$$\hat{\boldsymbol{\sigma}}''^{T} = \left\{\boldsymbol{\sigma}_{xz}, \boldsymbol{\sigma}_{yz}\right\} \tag{8}$$

$\hat{\boldsymbol{\sigma}}_z = 0$, as usual in thin shell theory. (9)

Cauchy stresses are evaluated and updated at each sampling point ($\xi = \eta = 0$) for each lamina.

The forces per unit length developed at the shell cross section are computed integrating the stress field across the thickness. This is carried out only along the fiber ($\xi = \eta = 0$) at the centroid of element area, evaluating moment and membrane forces, as shown in Fig. 4 and given in Eqs.10, 11 and 12, respectively for membrane forces, bending forces and transverse shear forces per unit length.

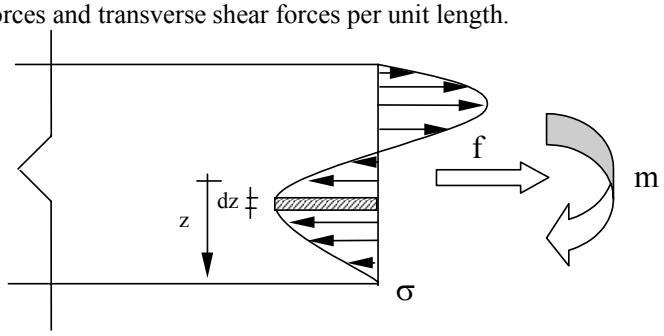

Fig. 4. Evaluation of membrane and bending forces through thickness.

$$\hat{\mathbf{f}}_{ij} = \int \hat{\boldsymbol{\sigma}}_{ij} \mathrm{dz} \tag{10}$$

$$\widehat{\mathbf{m}}_{ij} = -\int \hat{z}\, \hat{\boldsymbol{\sigma}}_{ij}\, \mathrm{dz} \tag{11}$$

$$\widehat{\mathbf{q}}_i = \int \hat{\boldsymbol{\sigma}}_{iz} \mathrm{dz} \tag{12}$$

Since forces per unit length are only computed at the centroid, a proper and low cost stabilization procedure is included in order to control the undesirable hourglass modes, which may arise as a side effect of this under-integrated strategy, over the element area. The stabilization forces, derived from the perturbed hourglass control, are represented by the superscript “h” and are added to the internal resisting forces given by:

$$\hat{\mathbf{f}}_{iI} = \hat{\mathbf{f}}_{iI} + \hat{\mathbf{f}}_{iI}^{h} \tag{13}$$

$$\hat{\mathbf{q}}_{iI} = \hat{\mathbf{q}}_{iI} + \hat{\mathbf{q}}_{iI}^{h} \tag{14}$$

$$\hat{\mathbf{m}}_{iI} = \widehat{\mathbf{m}}_{iI} + \hat{\mathbf{m}}_{iI}^{h} \tag{15}$$

The perturbed hourglass control evaluation is detailed at Belytschko *et al*., (1984). Therefore, the element internal forces vector is represented by:

$$\hat{\mathbf{f}}_{\mathbf{e}}^{int^T} = \left[\hat{\mathbf{f}}_1^{\text{int}} \quad \hat{\mathbf{f}}_2^{\text{int}} \quad \hat{\mathbf{f}}_3^{\text{int}} \quad \hat{\mathbf{f}}_4^{\text{int}}\right] \tag{16}$$

Finally, the projection operator, **P**, (Belytschko *et al*, 1994; Rankin *et al* (1988), Zhu *et al*, 1996) is applied:

$$\hat{\bar{\mathbf{f}}}_{\text{e}}^{\text{int}} = \mathbf{P}^T \hat{\mathbf{f}}_{\text{e}}^{int} \tag{17}$$

The main idea of the operator **P** is to extract a pure deformation field from an arbitrary displacement field, ensuring invariance to rigid body motion. The operator **P** represents both the total and the partial operators.

4 Governing Equations

The discrete form of the equations of motion for a discrete structural system is given by the set of second order differential equations is:

$$\mathbf{M}\,\mathbf{a} = \mathbf{F}^{\text{ext}} - \mathbf{F}^{\text{int}} \tag{18}$$

where **M** is the lumped (diagonal) mass matrix; **a** is the acceleration vector; $\mathbf{F}^{\text{ext}}$ is the external forces vector and $\mathbf{F}^{\text{int}}$ the internal forces vector given by:

$$\mathbf{F}^{\text{int}} = \mathop{\mathsf{A}}_{\text{e}}^{\text{nel}} \mathbf{f}_e^{\text{int}} \tag{19}$$

where A is the assembling operator; nel is the number of elements and $\mathbf{f}_{\text{e}}$ is the vector of element contribution to the internal forces in the global reference systems, defined as:

$$\mathbf{f}_e^{\text{int}} = \mathbf{R}^T \ \hat{\mathbf{f}}_e^{\text{int}} \tag{20}$$

The **R** matrix is the transformation (rotation) operator between the global-local systems and will be defined later. The element internal forces in the co-rotational reference system, $\hat{\mathbf{f}}_e^{\text{int}}$, are evaluated by:

$$\hat{\mathbf{f}}_{\mathbf{e}}^{\text{int}} = \int_{\text{A}_\text{e}} \mathbf{b}^{t} \begin{Bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{m}} \\ \hat{\mathbf{q}} \end{Bmatrix} d\text{A}_e \tag{21}$$

where $\hat{\mathbf{f}}$, $\hat{\mathbf{m}}$ *and* $\hat{\mathbf{q}}$ are respectively membrane, bending and shear contributions in the co-rotational system and **b** the discrete gradient operator. For short transient problems the equations of motion Eq.18 are integrated using the explicit algorithm based on the central difference operator in time. Also, it is worthy to note that the diagonal form of the mass matrix in (Eq.18) uncouples the equations. This fact added to the simple structure of the central differences approximation allows a straightforward implementation comprised of simple and fully vector operations for computing the accelerations, velocities and displacements at each time step.

5 Code Structure for the Shell Element Implementation

In the implementation reported herein, the internal forces are evaluated based on three routines. The first routine computes the co-rotational system and the strains per laminae, both according to element type, performing a loop over the elements present in the discretization. The second routine is basically a loop over every laminae for stress updating considering elastic-plastic behavior under von Mises criterion and plane stress assumption. Sub-incremental correction with Euler-forward technique is employed to achieve admissible stress state for plastic laminae. This routine is entirely independent of the Belytschko element choice. The third routine is responsible for the internal force computations on a single loop over the elements. Updated stresses are recovered and integrated using Simpson rule through thickness at each element centroid and finally integrated over its area.

If the Belytschko element is used into the six-degree-of freedom explicit code, the operator, **P**, must be employed to extract the spurious modes from the strain field at the first sub-routine and to remove these modes from the stress field at the third sub-routine.

Routine #1 – Strain rate computation

The strain rate routine loops over every shell element present in the computational model defining the reference system transformations and computing the strain rates at each lamina applying the CB kinematic relations, according to the following steps:

(a)	Transform coordinates to local-Global system	$\hat{\mathbf{x}} = \mathbf{R}\mathbf{x}$
(b)	Compute local velocities	$v_L = \mathbf{R}v$
(c)	Transform velocities to co-rotational system	$\hat{\mathbf{v}} = \hat{\mathbf{R}}\mathbf{v}$
(d)	Apply the projection operator **P**	$\overline{\hat{\mathbf{v}}} = \mathbf{P}\hat{\mathbf{v}}$
(e)	Compute and store membrane strain rates at each laminae	$\hat{\mathbf{d}} = \hat{\mathbf{d}}^{mid-surf} + z\mathbf{k}$

Routine #2 - Elastic-plastic stress updating over every laminae (lamina-by-lamina)

Integrate stress rates within elements laminae, according to an elastic-plastic constitutive law. This step performs a loop over every lamina, for the whole discrete model, irrespective to the parent element. Computations are based solely on local information.

(a)	Retrieve strain rates for the laminae
(b)	Update stresses at the lamina using a return mapping (sub-incremental technique Euler-forward) accounting the elastic-plastic behavior and plane stress assumption.

Routine #3 - Internal forces evaluation
Compute generalized stresses and element contributions to global internal forces vector. Loop over the elements present in the discrete model, gathering stresses computed at each child lamina, performing the through thickness and over the area numerical integration of internal forces, perform final transformations and make the results available to the next phase, the global assembling of internal forces (Eq.19).

(a)	Integrate stresses through thickness (Simpson rule)	**f**, **m** and **q**
(b)	Compute shear stresses (linear law)	
(c)	Integrate over element area (internal forces) reduced integration	$\hat{\mathbf{f}}_e^{\text{int}} = \int \mathbf{b}^T \hat{\boldsymbol{\sigma}} dv$
(d)	Add hourglass stabilization	
(e)	Apply the projection operator **P**	$\hat{\bar{\mathbf{f}}}_e^{\text{int}} = \mathbf{P}\hat{\mathbf{f}}^{\text{int}}$
(f)	Transform local internal forces to global system	$\mathbf{f}_e^{\text{int}} = \mathbf{R}^T \ \hat{\mathbf{f}}_e^{\text{int}}$

6 Performance Analysis – Cylindrical Panel

This study assesses the computer implementation presented, comparing the performance obtained among the Belytschko element family and improved elements based on it. The code exploits the characteristics of PVP machines in a particular coding technique, namely *lamina-by-lamina*. Performance analyses were carried out for the following element formulations: BLT element; BWC element and a MIX element, ie, the combined element is formulated according to BLT assumptions, but using the BWC co-rotational system. A further option is implemented taking the latter elements and including a total or partial operator **P**, respectively refereed to by "tot" and "part" suffixes.

Target machines for the implementation are shared memory PVP´s and the experiments reported herein used the Cray SV1, installed at the NACAD/COPPE/UFRJ and the Cray T94 installed at the CESUP/UFRGS. It is worthy to note that the Cray SV1 has memory hierarchy architecture, i.e., cache memory, which for the current implementation may impact the performance, as we shall observe. Table 1 displays the principal characteristics of the considered computational platforms.

Table 1. Computational characteristic between the Cray SV1 and the Cray T94.

Cray	**CPUs Numbers**	**GFlops/ CPU**	**Memory** *(Gbytes)*	**Disc Storage** *(Gbytes)*
SV1	12	1.2	16	250
T94	2	1.8	2	53

In order to assess the performance of the implemented code a cylindrical panel impulsively loaded was selected (Belytschko *et al*., 1984, 1994 and Zhu *et al*., 1996). The initial velocity of 5650 in./sec. simulates an explosion at the region RE, as shown at the Fig. 5. For this test, according to Zhu *et al*. (1996), finite rotations around 60^0 are expected. Figure 5 and Table 2 show the problem data. The displacement time history of a point A (0.00, 6.28, 2.94) at the crown is shown in the Fig. **6**. Stress evaluation used five sampling points through thickness. Due to symmetry only one half of structure was modeled using a mesh of 80x160.

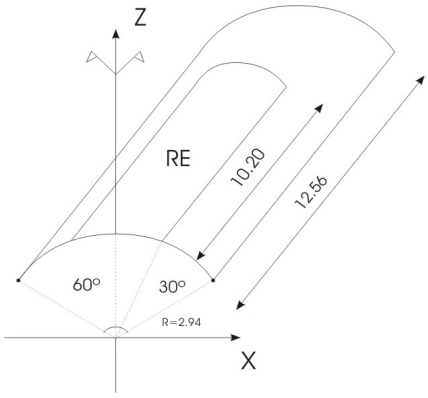

Fig.5. Cylindrical panel description

Table 2. . Problem data

Radius	R= 2.9375
Angle	$\alpha=60^o$
Thickness	t=0.125
Length	L=12.56
Elasticity modulus	$E=10.5x10^6$
Poisson coeff.	$\nu=0.33$
Yield stress	$\sigma=4.4x10^4$
Plastic modulus	$E_p=0.0$
Density	$\rho=2.5x10^{-4}$
Initial velocity	v=-5650.0
Time step	$dt=5.0x10^{-7}$

In an explicit code, the computation efforts are concentrated at the internal forces routine, precisely evaluating the stresses over the laminae. Time integration was performed for $1.0e10^{-3}$s and the incremental time step used was $5.0e10^{-8}$s. The whole analysis was carried out within 20000 steps. Table 3 presents mesh data summary for both analyses.

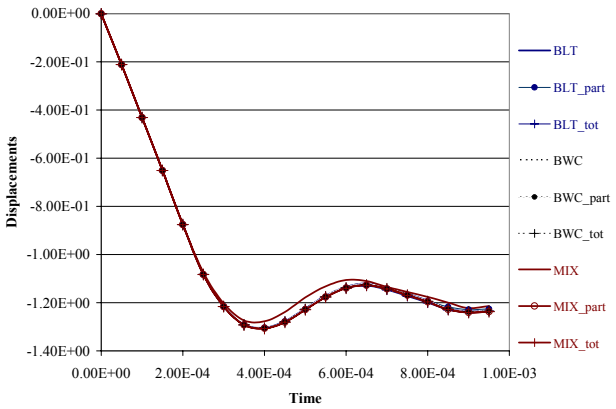

Fig. 6. The displacement time history of a point A among the implemented codes

The implemented codes presented good results for the cylindrical panel. Similar results are observed in Belytschko *et al*. (1984) and Zhu *et al*. (1996). Only the MIX

version shows a slightly stiffer behavior in contrast to the other codes. It is due to the co-rotational system proposed by Belytschko et al. (1989) inserted into the BLT formulation. Although this system captures better the geometric nonlinearities, the BLT element is a flat element and it is not adequate to warped geometries.

Table 3. Mesh data summary for the cylindrical panel problem

General DATA	***Mesh (80x160)***
Number of elements	12800
Number of nodes	13041
Number of laminae	64000
Number of equations	76319
Time step	5.0×10^{-8}s

It was observed an increase in CPU time and *Mflop/s* for the whole job due to additional cost for the improved elements, when comparing with the original Belytschko element. Table 4 and Table 5 present respectively the whole code performance in the Cray SV1 and in the Cray T94, for each considered member of the Belytschko family, including overall CPU time and overall code performance in *Mflop/s* as measured with *hpm* (Cray's hardware performance monitor). It is also presented the normalized performance of each implementation, i.e., ratio overall code performance to processor peak (theoretical) performance. The last column presents the overall computational intensity (ratio of flops to memory references) as measured by Cray's *perfview* tool.

Table 4. Overall code performance for a single CPU run in the Cray SV1 at NACAD/COPPE/UFRJ (peak performance per CPU 1200 Mflop/s)

Belytschko Element/ Code Versions		CPU Time	Performance (*Mflop/s*)	Relative Performance (%)	Computational Intensity
BLT	Original	29 min. 27s	245.6	0.20	0.88
	Partial	36 min. 44s	301.9	0.25	0.95
	Total	46 min. 08s	329.0	0.27	0.90
BWC	Original	32 min. 54s	253.0	0.21	0.85
	Partial	41 min. 43s	266.9	0.22	0.93
	Total	56 min. 35s	283.5	0.24	0.86
MIX	Original	30 min. 21s	228.8	0.19	0.86
	Partial	36 min. 17s	278.5	0.23	0.94
	Total	48 min. 22s	309.4	0.26	0.90

Table 5. Overall code performance for a single CPU run in the Cray T94 at CESUP/UFRGS (peak performance per CPU 1800 Mflop/s)

Belytschko Element/ Code Versions		CPU Time	Performance (*Mflop/s*)	Relative Performance (%)	Computational Intensity
BLT	Original	9 min. 56s	745.6	41	0.76
	Partial	13 min. 43s	837.20	47	0.89
	Total	17 min. 27s	897.5	50	0.83
BWC	Original	11 min. 24s	769.4	43	0.75
	Partial	14 min. 30s	805.0	45	0.80
	Total	18 min. 37s	893.9	50	0.79
MIX	Original	9 min. 46s	734.7	41	0.75
	Partial	12 min. 44s	805.1	45	0.81
	Total	13 min. 04s	767.3	43	0.81

As can be observed in Table 4 and Table 5 overall computational intensity is above 0.75 for any element implementation, indicating reasonable efficiency in any case. A more detailed code performance analysis concerns the three routines that perform the internal force evaluation for each element. The results for these analyses are summarized in Table 6 and Table 7. As can be seen in these tables the computational intensity of the strain rate and internal forces sub-routines, for both machines, are ranging from 0.75 to 1.0, irrespective to the code. These figures indicate an algorithm of middle efficiency. Considerable computational intensity of 1.91 and 1.72 to 1.74, are respectively obtained in the Cray SV1 and in the Cray T94, for the evaluation of lamina stresses sub-routine. These figures indicate an algorithm with high efficiency. The other implemented codes behave similarly as this one, according to the computational platform considered, ie., Cray SV1 or Cray T94. The performance for the 5 most dominant subroutines is presented in Table 6 and Table 7. These sub-routines are ranked by the crescent percent CPU time for the overall code. T strain rate, stress update and internal force subroutines were previously described in section 2. The routines responsible for explicit time integration and coordinates updating for the co-rotational formulation are also included in the subroutine list.

Table 6. Performance analysis for the top 5 sub-routines for the Cray SV1 for a single CPU. (peak performance per CPU 1200 Mflop/s)

BWC_tot Code (Sub-routines)	CPU (%)	Performance (Mflop/s)	Relative Performance (%)	Computational Intensity
Strain Rates	47.80	319.90	26.65	0.86
Internal forces	26.20	370.60	30.88	0.90
Stress Update	7.10	327.60	27.30	1.91
Time integration	2.8	208.2	17.35	0.93
Coord. updating	1.1	83.4	6.95	-
Others	15	-	-	-

Table 7. Performance analysis for the top 5 sub-routines for the Cray T94, for a single CPU (peak performance per CPU 1800 *Mflop/s*)

BWC_tot Code (Sub-routines)	***CPU (%)***	***Performance (Mflop/s)***	***Relative Performance (%)***	***Computational Intensity***
Strain Rates	50.20	959.10	53.28	0.78
Internal forces	30.50	1011.20	56.17	0.83
Stress Update	11.60	631.10	35.05	1.74
Time Integration	1.90	963.40	53.52	-
Coord. Update	0.4	690.5	38.36	-
Others	5.4	-	-	-

As expected, comparing Table 6 and Table 7, the three sub-routines related to the internal forces evaluation are the top ranked for both PVP machines. The strain rate, internal force and stress update sub-routines consume a great part of CPU time, in both cases. Notwithstanding the good code performance at the internal forces evaluation in both cases, there is a large discrepancy between their performances in the addressed computational platforms. The relative performance obtained in the Cray T94 is approximately twice the observed in the Cray SV1. This can be addressed to the differences in the memory architectures. Although being both shared-memory PVP´s, the Cray SV1 has a cache. In contrast the Cray T94 has no cache and processor-memory transfers are faster operations and not much sensitive to data locality, a severe issue in unstructured data access such as in finite element data structures.

It is worthy to note that the *lamina-by-lamina* approach used in the stress updating computations in yielded the least affected routine while considering performance in both machines. This may be accounted to the data locality resulting from the lamina-by-lamina approach, where most computations are done with stride one over the vector operands.

Next, the parallel performance is addressed for the improved BWC element code with total operator approach, motivated by the good modeling properties of this element at highly warped configurations. For the multitasked analysis, other two sub-routines show up in the list. The *asmble* sub-routine is used to assemble the nodal contributions and it appears only in the Cray SV1 run. The *disasm* sub-routine is employed to the gather-scatter operations required in the process of unstructured global-local data interchange, appearing in both cases. Table 8 and Table 9 summarizes the *atexpert* report for the Cray T94 and for the Cray SV1, according to the 5 most dominant routine loops, displaying the best parallel regions on the code considering the percent work loops of the overall code. Atexpert is a statistic tool developed to measure the auto-tasking of a job run, predicting speedups on a dedicated system by using data collected from one run on a non-dedicated system.

Table 8. Atexpert report summary of 5 dominant loops for the test run on the Cray T94.

Sub-routines	*Parallel regions(loops)*	*Workload (%)*
Internal forces	tsk_BWC02_tot_000	41.56
Strain rates	tsk_BWC01_tot_000	25.53
Stress update	tsk_Plasty_003	15.09
Plasty	tsk_Plasty_000	6.90
Disasm	tsk_Disasm_tot_000	4.07

In the Cray T94, for the 5 dominant loops, 4 are related to internal forces evaluation, as expected. In fact two dominant loops belong to the lamina stress updating. The parallel region tsk_Plasty_000 is related to a loop over all laminae defining the trial elastic stresses and identifying the yielded laminae. The other region tsk_Plasty_003 considers a loop over the yielded laminae for the sub-incremental Euler-forward technique.

Table 9. Atexpert report summary of 5 dominant loops for the test run on the Cray SV1.

Subroutines	*Parallel regions (loops)*	*workload (%)*
Internal forces	tsk_BWC02_tot_000	47.23
Strain rates	tsk_BWC01_tot_000	25.30
Disasm	tsk_Disasm_000	6.39
Stress update	tsk_Plasty_003	5.92
Asmble	tsk_Asmble_tot_000	4.76

As can be observed in the Cray SV1, the lamina stress lamina updating is less important in the workload. The parallel region tsk_Plasty_003 occupies the fourth place and the region tsk_Plasty_000 is no longer in the list of the top five dominant loops. Undesirably, the parallel regions tsk_Disasm_tot_000 and tsk_Asmble_tot_000 gain importance. It is due to the SV1 cache. The subroutines implement the process of unstructured global-local data interchange, generate bottle-necks. Table 10 and Table 11 summarize respectively the *atexpert* report according to parallel regions in the Cray T94 and in the Cray SV1 for the mesh (80x160). The strategy developed to structure the code seems to be suitable for shared-memory PVP machines, such as the Cray T94. Good levels of vectorization and parallelization are attained. For cache-memory hierarchy PVP machines, the mesh data must be reorganized in order to improve the data memory access.

Table 10. ATEXpert´s summary report for the most parallel 5 subroutine at the Cray T94.

Sub-routine	*Parallel %*	*Amdahl law Speedup*	*Dedicated System Speedup*
Internal forces	99.9	4.00	4.00
Strain rates	99.9	4.00	4.00
Stress update	99.1	3.90	3.70
Disasm	96.6	3.60	3.60
Time integration	98.1	3.80	3.10

Table 11. ATEXpert´s summary report for the most parallel 5 subroutine at the Cray SV1.

Sub-routine	*Parallel %*	*Amdahl law Speedup*	*Dedicated System Speedup*
Internal forces	99.9	10.90	10.30
Strain rates	99.9	10.80	10.20
Disasm	97.2	8.60	0.30
Stress update	97.4	8.70	6.70
Asmble	94.6	7.10	6.20

Figure 7 shows the parallel performance code according to *atexpert* report at the Cray T94 and at the Cray SV1. Notice that, the curves that are related to the ideal parallel system and the dedicated system are close, indicating a good level of parallelization. On a dedicated system, the code shows reasonable speedup of 3.51 for 4 CPUs. The Amdahl's law indicates a speedup of 3.90 for 4 CPUs. Figure 8 presents respectively the multitasked performance analysis for the whole code and for the three subroutines relates to internal forces evaluation in dedicated system, according to ATEXpert´s report in the Cray T94 and in the Cray SV1.

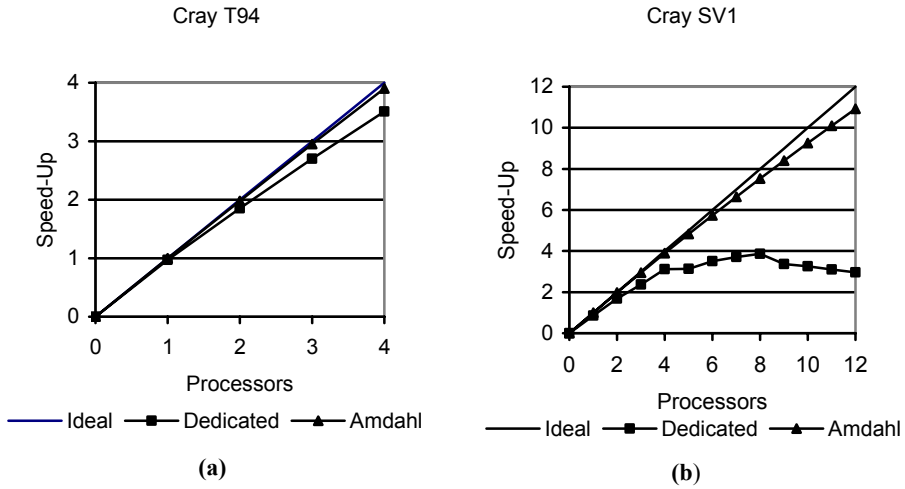

Fig. 7. Parallel performance (computational speed-up) for the whole job: (a) in the Cray T94, (b) in the Cray SV1.

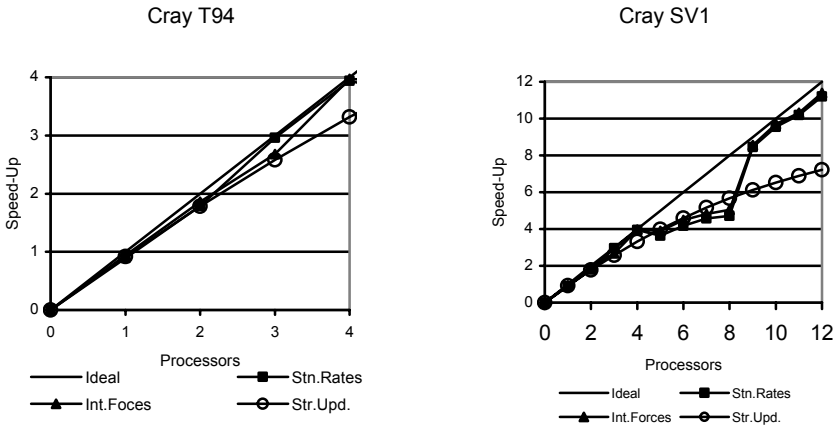

Fig. 8. Parallel performance (computational speed-up) for the element routines: (a) in the Cray T94, (b) in the Cray SV1.

Remark 1. The differences between results obtained in both platforms are due to the memory architecture issues. Since element routines for internal forces mostly handle private data with very few serial requirements to access public data, speed-up are expected to be close to the ideal. This is confirmed by the close agreement of the *ideal* and *Amdahl* curves in both systems. In the Cray T94 this is also observed in the profiling data obtained with *atexpert* tool: speed-up for the whole code and individual routines are very close to Amdahl law predictions. The Cray SV1 is a cache memory system, and therefore data locality can dramatically affect code performance. No effort was made to improve data locality in the reported implementation or for the specific test case. And the consequences can be observed in the dramatic deviation of speed-up measures from Amdahl law predictions as the number of processors is increased.

In the Cray SV1, the code appears to be 99.1 percent parallel and 0.9 percent serial. Amdahl's Law predicts that with no overhead, this program could expect to achieve a 10.9 times speedup on 12 CPU's. However, a 3.0 speedup is predicted with 12 CPU's on a dedicated system. Reasonable results for the multitasked analyses are obtained for the three sub-routines that are responsible for the evaluation of the internal forces, in spite of the whole code performance at the dedicated system.

Figure 9 depicts a sequence of snapshots showing the deformation pattern obtained in the simulation, showing the structural response to the explosive loading.

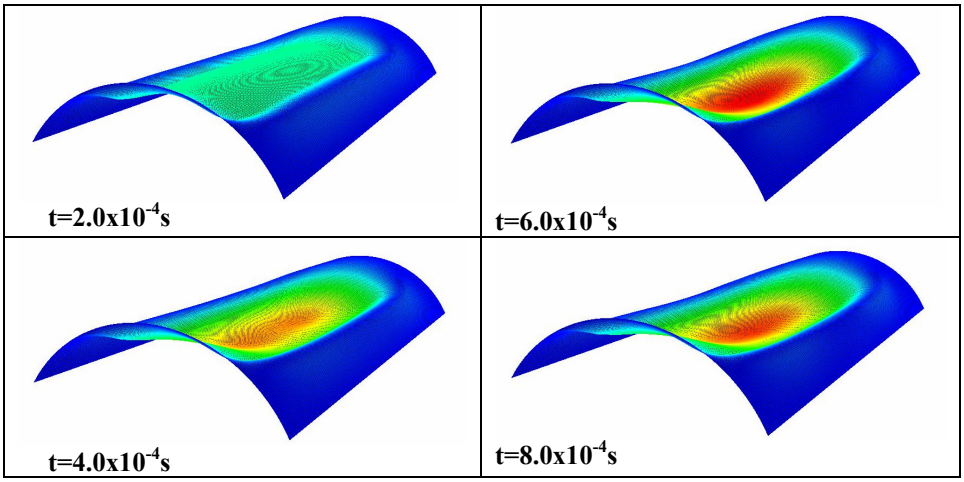

Fig. 9. Snapshot sequence for the cylindrical panel problem

Conclusion

As seen in the experiments performed in this work, members of the Belytschko shell element family do not present significant additional computational cost when implemented under the particular *lamina-by-lamina* approach. Indeed, it was observed that even the improved elements showed good modeling properties at marginal additional cost.

In the Cray SV1, it is essential to reorganize the mesh data to improve data locality alleviating possible overhead to memory access. The overall parallel code performance in system architectures with memory hierarchy, i.e., cache memory, is greatly affected. However, good results were observed for the three routines responsible for internal force evaluation (strain rate computation, elastic-plastic stress updating and internal forces evaluation). This is mainly due to data locality provided by the reduced quadrature rule combined with the *lamina-by-lamina* approach used in the implementation.

Further research is currently being carried out in order to provide additional improvement in data locality as dictated by modern processors architectures and required by high performance Nonlinear Finite Elements Analysis.

Acknowledgements

This work was partially supported by MCT/CNPq grants 522692/1995-8 and 68.0069/2003-8. Computer time on Cray SV1 and T94 was respectively provided by CESUP/UFRGS and by the Center for Parallel Computation NACAD at COPPE/UFRJ. Other computational supports were provided by the Laboratory of

Computational Methods in Engineering of the Program of Civil Engineering LAMCE/COPPE/UFRJ.

References

Alves, J. L. D., 1991, Análise transiente de grandes deformações em computadores de arquitetura Vetorial e paralela, D.Sc. thesis, Universidade Federal do Rio de Janeiro, RJ/Brasil

ANSYS User's Manual: Elements, vol. III, Elements, Swanson Analysis Systems, Inc.

Belytschko, T., Lin, J. I. & Tsay, C. S., 1984, Explicit algorithms for the nonlinear dynamics of shells. Computer Methods in Applied Mechanics and Engineering, vol. 42, pp. 225-251.

Belytschko, T., Wong, B. L. & Chiang, H.Y., 1989, Improvement in low-order shell element for explicit transient analysis. Computer Methods in Applied Mechanics and Engineering, vol. 42, pp. 383-398.

Belytschko, T., Wong, B. L. & Chiang, H.Y., 1992, Advances in one-point quadrature shell elements. Computer Methods in Applied Mechanics and Engineering, vol. 96, pp. 93-107.

Belytschko, T., Lee, S. H. & Stolarski, H., 1995, A review of shell finite elements and co-rotational theories. Computational Mechanics Advances, vol. 2, pp. 125-212.

Belytschko, T. & Leviathan I., 1994, Projection schemes for one-point quadrature shell elements. Computer Methods in Applied Mechanics and Engineering, vol. 115, pp. 277-286.

Silva, E. F., 1997, Implementação computacional de elemento de casca com formulação co-rotacional para análise transiente, M.Sc. thesis, Universidade Federal do Rio de Janeiro, RJ/ Brasil.

Silva, E. F., 2003, Implementação computacional do elemento quadrilátero de casca bilinear com com formulação co-rotacional para análise transiente de estruturas laminares, D.Sc. thesis, Universidade Federal do Rio de Janeiro, RJ/ Brasil.

Zhu, Y. & Zacharias, T., 1996, A new one-point quadrature, quadrilateral shell element with drilling degrees of freedom. Computer Methods in Applied Mechanics and Engineering, vol. 136, pp. 165-203.