2nd International Conference on Computational Plasticity Barcelona, 18-22 sept 1989

LARGE STRAIN PLASTIC COLLAPSE: A COMPARISON OF EXPLICIT AND IMPLICIT SOLUTIONS

F.Beltrán and J.Goicolea

Principia, S.A.
Velázquez, 94
28006 Madrid, SPAIN

SUMMARY

Two shock-absorber devices, based on the turning inside-out of thin walled ductile metal tubes, are modelled numerically using four different computer codes: NIKE2D, ABAQUS, PR2D and DYNA2D. The formulations implemented in these programs are briefly introduced: NIKE2D and ABAQUS are implicit codes, while PR2D and DYNA2D are explicit.

The results obtained with each code and experimental data are compared, and conclusions about the methods employed by the programs and the general guidelines in modelling this kind of problems are drawn.

1. INTRODUCTION

Plastic deformation of short metallic tubes has been successfully used as an energy absorbing mechanism during impacts [1,2].

The design of such mechanisms involves the collapse of thin-walled ductile metal tubes through a folding process, developing very large plastic strains. In this manner the kinetic energy of the impacting body is dissipated in a controlled fashion at an acceptable rate. The yield limit of the material keeps the transmitted forces below an acceptable upper bound and, hence, the deceleration is slower and less harmful.

A number of energy absorbers based upon metallic tubes have been conceived during the past [2,3]. In this paper we shall focus on two so-called "invertube" devices (figs.1 and 2), in which large plastic deformations are produced by the turning inside-out of a thin-walled tube. In the first example (fig.1) the tube has been pre-formed and its flared end is clamped in order to force the inversion proccess as the tube is displaced axially. In the second case (fig.2), a plain tube is confronted with a hard die to produce the inversion. Numerical modelling of the collapse of such energy dissipating structures requires not only to take into account the plastic behaviour of the tube material, but also to consider the very large strains, displacements and rotations that develop and, in the second case, an

accurate calculation of contact forces. Thus, a great number of advanced code capabilities are tested by running this kind of problems.

In the present work the load-shortening curves of two particular "invertubes", corresponding to the categories described above, have been obtained using four different non-linear computer codes: two of them implicit, NIKE2D [4] and ABAQUS [5], and the other two explicit, PR2D [6] and DYNA2D [7]. The comparison of these curves between each other and with available experimental results allows to draw some conclusions about the formulations implemented in each code and the general guidelines in modelling this kind of situations.

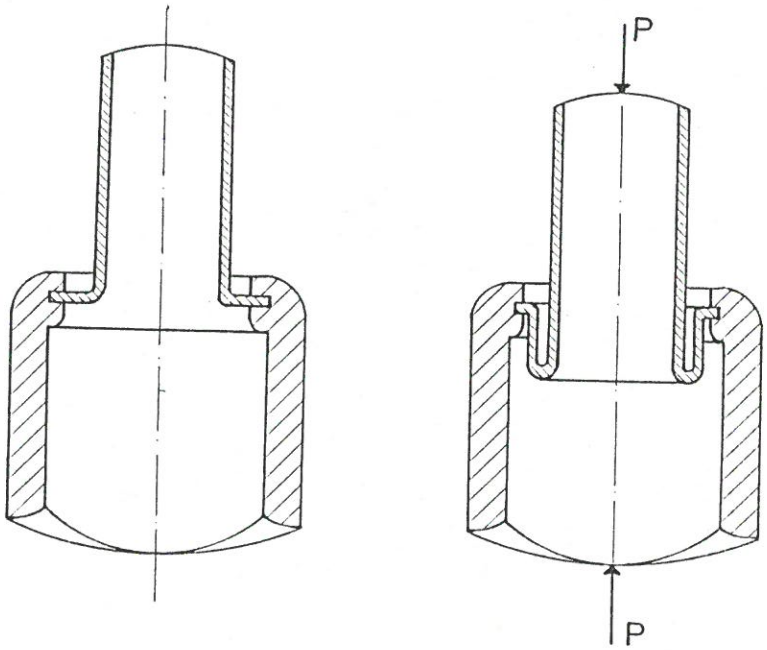

Figure 1. Energy absorber device with pre-formed tube.

2. COMPUTER CODES

In the present section a brief description of each code is given. Further information may be found in [4,5,6,7].

2.1. NIKE2D

NIKE2D is an implicit, finite deformation, finite element code designed to perform static or dynamic stress analysis of two-dimensional inelastic solids.

Spatial discretization is done by using three or four node elements that are integrated with a 1 or 2 point Gauss quadrature rule. An hourglass formulation is used to stabilize the elements with one point quadrature, and a constant pressure approach is used with the 2 point quadrature rule to avoid the locking problems that may arise when modelling plastic flow.

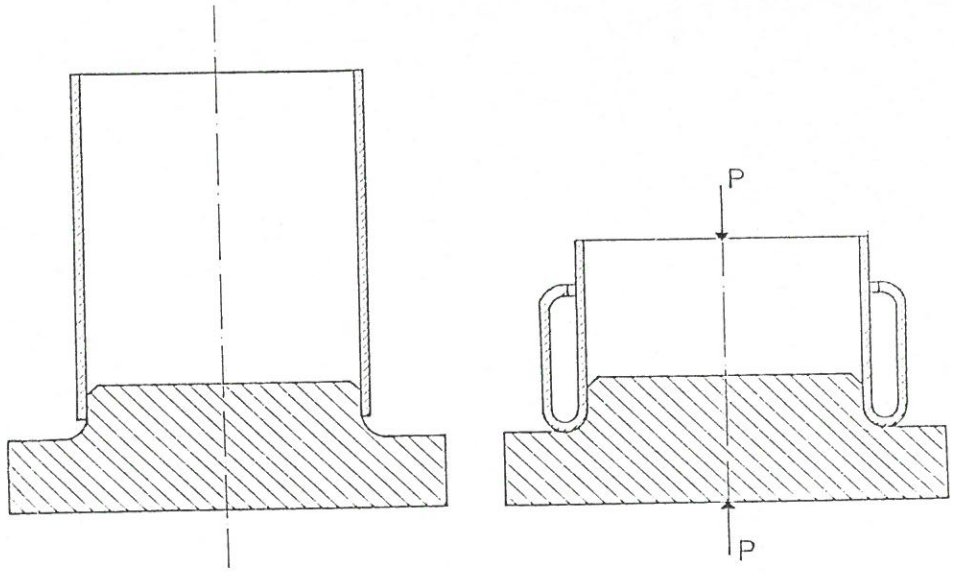

Figure 2. Energy absorber device with plain tube and die.

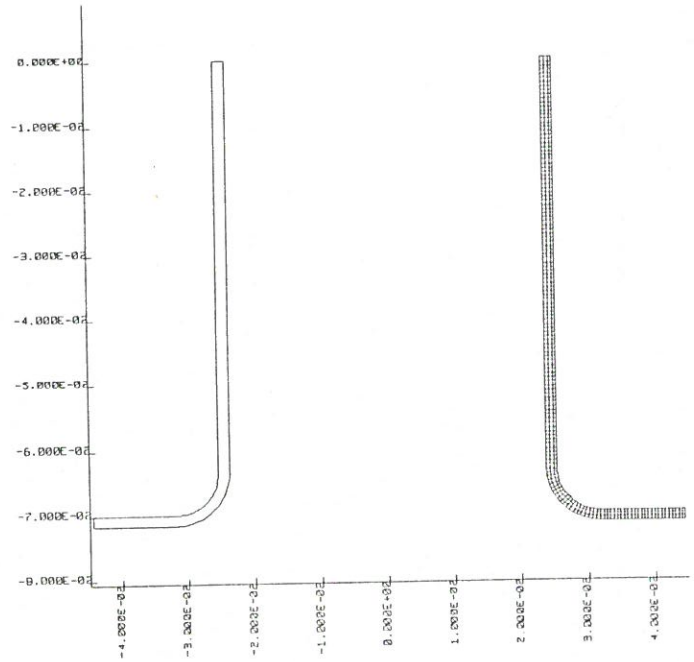

Figure 3. Case 1 - Undeformed mesh.

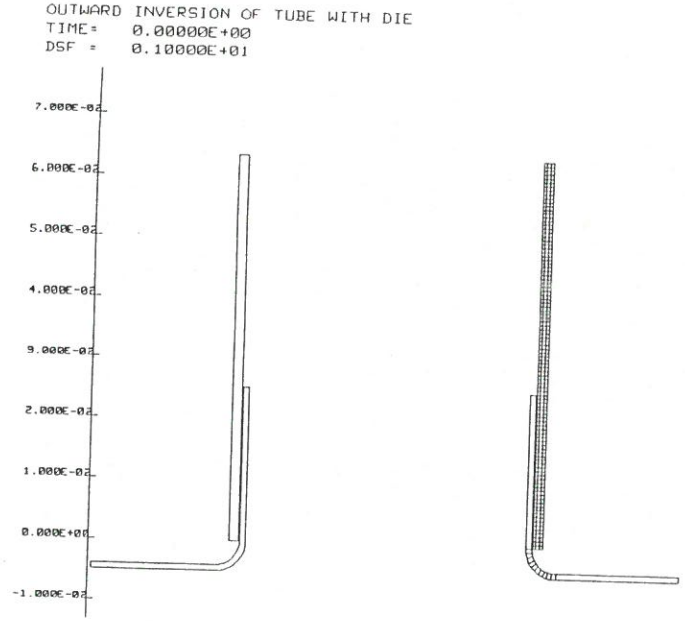

Figure 4. Case 2 - Undeformed mesh

An incremental-iterating numerical algorithm based on the Green-Naghdi stress rate formulation is implemented in the code. The method can handle non-linear problems that involve finite strains and arbitrarily large rotations. The load increment or time step size is only limited by accuracy considerations. Twelve quasi-Newton, modified-Newton and modified constant arc length iterating schemes are available. A line search is used with each of these schemes along with automatic stiffness reformations as needed to avoid nonconvergence.

The program has a general slideline capability based on a penalty function formulation. In the region of contact, interface springs are used to preclude penetration.

2.2. ABAQUS

ABAQUS is a general purpose finite element program designed specifically to serve advanced structural analysis needs, specially in nonlinear applications.

Despite having an explicit integration module, it is mainly and for the present an implicit code that allows to carry out a wide range of analyses (static, dynamic, buckling, heat transfer ...) of 2D or 3D problems. It has a large library of elements but, for our purposes (axisymmetric geometry and plastic flow), possible choices would be: 1) four node elements with reduced integration and hourglass control, 2) four node elements with independent interpolation for pressure (hybrid formulation), 3) four node elements with constant pressure, 4) eight node elements with reduced integration and 5) eight node hybrid elements.

ABAQUS uses Lagrangian and updated Lagrangian formulations for finite strain elastic-plastic problems. The load increment or time step size is again only limited by accuracy considerations. For nonlinear equations ABAQUS uses the full Newton method, and, in a static stress/displacement analysis, an arc-length algorithm (modified RIKS), may be used.

Contact problems are modelled by means of interface elements.

2.3. PR2D

PR2D is an explicit finite difference non linear code for the dynamic stress analysis of two-dimensional solids.

Geometry discretization is accomplished by using "mixed discretization" quadrilateral cells. This concept means that deviatoric components of stress and strain are interpolated independently in each of the two triangles into which a quadrilateral is divided. The volumetric components of those fields are given an average value for the complete quadrilateral. In this way hourglass modes are resisted "naturally" by the deviatoric forces without introducing artificial anti-hourglass viscosities.

Resembling a finite element discretization, the geometry of quadrilaterals does not need to be neither geometrically nor topologically regular.

To handle finite strains and rotations, a Jaumann stress rate formulation is used. The equations of motion are integrated by a central difference scheme and as in all explicit codes, the step size is limited by the Courant condition for stability reasons. Contacts between bodies are treated using a penalty formulation in which forces that are proportional to the penetration appear on the interfaces.

2.4. DYNA2D

DYNA2D is an explicit, axisymmetric and plane strain finite element code for analyzing the large deformation dynamic and hydrodynamic response of inelastic solids.

Spatial discretization is achieved by the use of four node solid elements with single point integration and a hourglass viscosity to control the zero energy modes.

For elastic-plastic materials, a Jaumann stress rate formulation is used. The equations of motion are also integrated by the central difference method and the time step is then limited by Courant condition.

Several "slideline" algorithms are available to represent contact problems, including a penalty formulation and a distributed parameter formulation, in which bodies are not allowed to interpenetrate.

3. MODELLING APPROACH AND NUMERICAL RESULTS

3.1. Case 1: pre-formed tube

In this case a hypothetical shock-absorber is studied. It consists of an aluminium pre-formed tube (fig.3) with one end flared and clamped. The outside

diameter of the tube is 50.8mm and its wall thickness is 1.63mm. The total length is 71.50mm and the inner radius of the pre-forming is 6.35mm. The total diameter of the flared end is 88.9mm.

The material is supposed to have Young's modulus of 67 GPa, a Poisson's ratio of 0.3 and a yield stress of 300 MPa. Isotropic hardening is assumed, with a hardening modulus of 44.7 MPa.

The purpose of the simulation is to predict the load-shortening curve for the device under quasi-static loading.

The mesh employed in the analysis has 3x145=435 quadrilateral elements (or cells in the case of PR2D), and 4x146=584 nodes. The outermost nodes at the flared end are clamped, and at the other end of the tube the nodes have a prescribed displacement of 75mm downwards.

With NIKE2D a quasi-static simulation was carried out using 4-node constant pressure quadrilaterals and a quasi-Newton (BFGS) implicit solution algorithm.

The simulation with ABAQUS was also quasi-static with elements of the same type as in NIKE2D, but a full Newton algorithm was used.

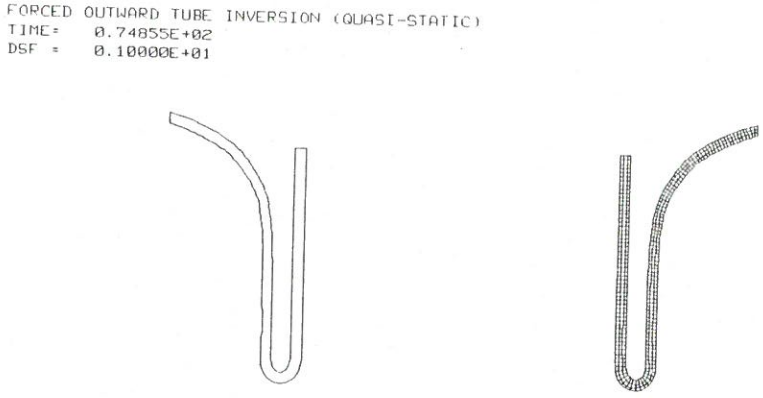

Figure 5. Case 1 - Mesh at final state.

The explicit nature of PR2D and DYNA2D, with a limited time-step, does not allow for a quasi-static or "very slow" simulation since very large computer times would be necessary. Thus a dynamic simulation was carried out in which the free end of the tube starts with cero velocity, it reaches 40m/sec in 0.2msec and then this value is kept constant. The analysis ends when an axial displacement of 75mm is reached.

All four codes were successful in modelling this problem correctly, without appreciable difficulty. Very similar deformation patterns were obtained in all cases. As an example, PR2D final mesh is given in figure 5; results from the other programs are undistinguishable.

In figure 6 the predicted force at the fixed end vs free end displacement curve is represented for each program.

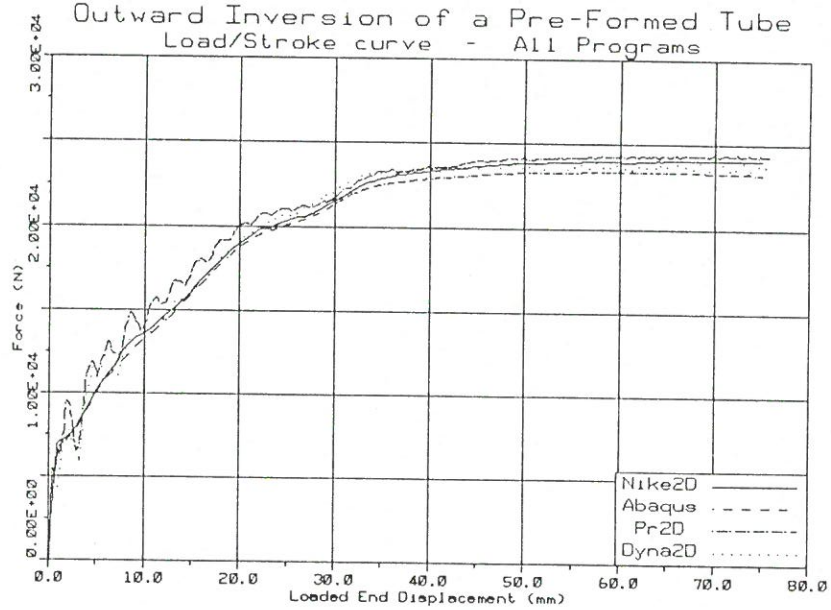

Figure 6. Case 1 - Computed load-displacement curves.

There is also good general agreement between the load-displacement curves obtained from the various programs. There are slight differences between the computed steady-state forces but they are not higher than 5% of the average value. During the initial transient the curves obtained with NIKE2D and ABAQUS are almost the same, whereas the corresponding to DYNA2D and PR2D show oscillations due to the presence of inertial forces. These oscillations are much stronger if we look at the force computed by the explicit codes at the end of the tube with the prescribed displacement (figure 7). This is so because the zone of plastic flow "filters" the high frequency stress waves, giving a smoother response for the force at the clamped end. Apart from considerations of this kind, it is remarkable how the explicit codes, whose traditional field of application is the modelling of fast transient dynamics , can give engineering accurate results for an essentially quasi-static problem.

Table 1 gathers the total number of computing cycles and iterations done by each program, together with the CPU time used in a CONVEX C120 vector computer.

As it might be expected, the number of cycles is considerably higher for the explicit codes, but each cycle takes, in these programs, much less CPU time. It can be seen also how the full Newton method (ABAQUS) needs less iterations per

cycle than a quasi-Newton method (NIKE2D) in order to achieve convergence. However the CPU time per cycle is higher in ABAQUS.

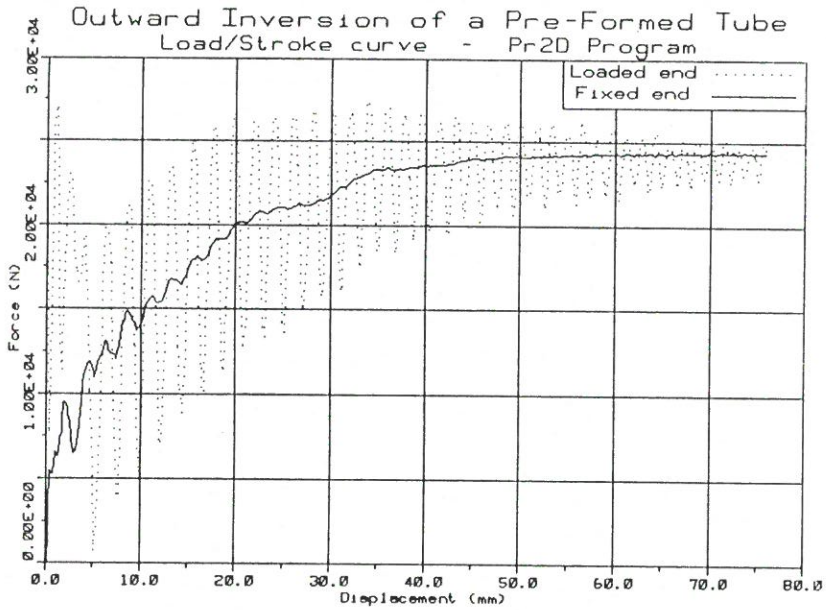

Figure 7. Case 1 - Load/displacement curves computed with PR2D.

PR2D is slightly slower that DYNA2D because of the mixed discretization, which in fact forces to consider two integration points per quadrilateral (one for each triangular cell) whereas DYNA2D has only one integration point per quadrilateral.

3.2. Case 2: plain tube and die

In this second case an experiment of Al-Hassani et al [2] is modelled. An aluminium tube of 50.8mm outside diameter x 63.5mm length x 1.63mm wall thickness is driven against a lubricated mild steel 3.97mm radius die (fig.4), at a 0.05 in/min speed.

The material of the tube has the same properties as in case 1 except the yield stress, which is estimated from the data supplied by Al-Hassani et al [2], to be 150MPa.

The tube has been modelled using 300 quadrilaterals and 404 nodes, and the die has been represented with 20 four node elements and 42 nodes. The material of the die is assumed elastic with a Young's modulus of 210GPa and a Poisson's ratio of 0.3.

The contact between the die and the tube is supposed to be frictionless, and

a 50mm axial displacement is prescribed for the nodes at the upper end of the tube.

The simulation has been conducted only using NIKE2D and PR2D, with the same choices of element, procedure, etc, as in case 1. Now the problem is more difficult because contacts between the tube and the die are involved.

The deformed meshes at different states obtained with NIKE2D are shown in figure 8. In figure 9 the predicted force-displacement curves are presented together with the experimental results. The value of the force corresponds to the end of the tube with prescribed displacements, and the curve from PR2D has been smoothed using a Centred Moving Average (CMA) algorithm to soften the spurious oscillations due to inertial forces and contact springs.

Both codes predict a similar steady state force, slightly greater than the experimental value. This slight difference is due to the fact that no accurate data for material properties were given in [2]. The yield stress used for the material is only an estimate. The main difference between the curves predicted by each code is at the initial peak, corresponding to the situation represented in figure 8a. The peak has a greater value in PR2D due to the initial inertia of the tube before it comes into contact with the die. On the other hand, NIKE2D underestimates the peak and gives a curve a slightlydisplaced to the left in this area. The modelling of the contacts is the cause of the oscillations that appear in both curves within the theoretical steady state region.

Table 2 gives the number of cycles and CPU time consumed by both simulations. It appears that the presence of contacts increases the cpu time per cycle in both codes but, being implicit, NIKE2D is more effected. Now PR2D, the explicit code, is a bit faster than NIKE2D.

Program	Total cycles	Total iterations	CPU time (sec)	CPU time (per cycle)
NIKE2D	263	2872	1167	4.44
ABAQUS	184	516	9014	49.00
PR2D	42070	--	2289	0.054
DYNA2D	37916	--	1854	0.049

Table 1. Number of cycles and CPU time for case 1 simulation.

Program	Total cycles	Total iterations	CPU time (sec)	CPU time (per cycle)
NIKE2D	100	2533	2264	22.64
PR2D	24012	--	2240	0.093

Table 2. Number of cycles and CPU time for case 2 simulation.

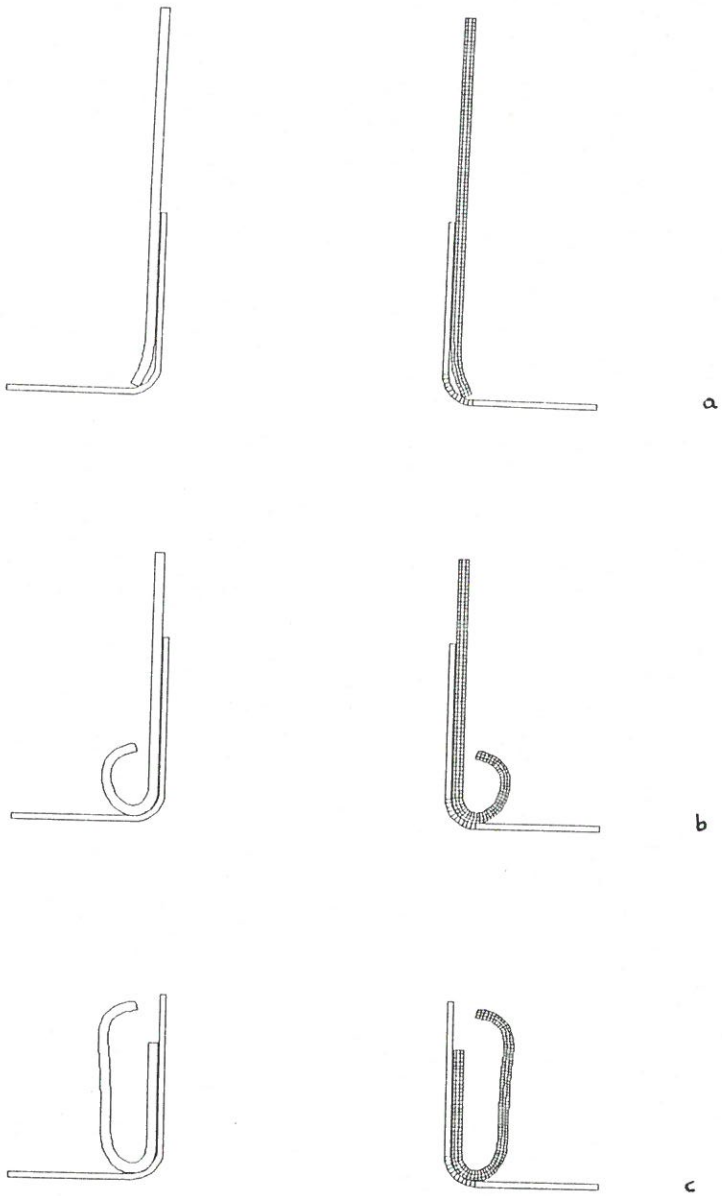

Figure 8. Case 2 - Deformed meshes.

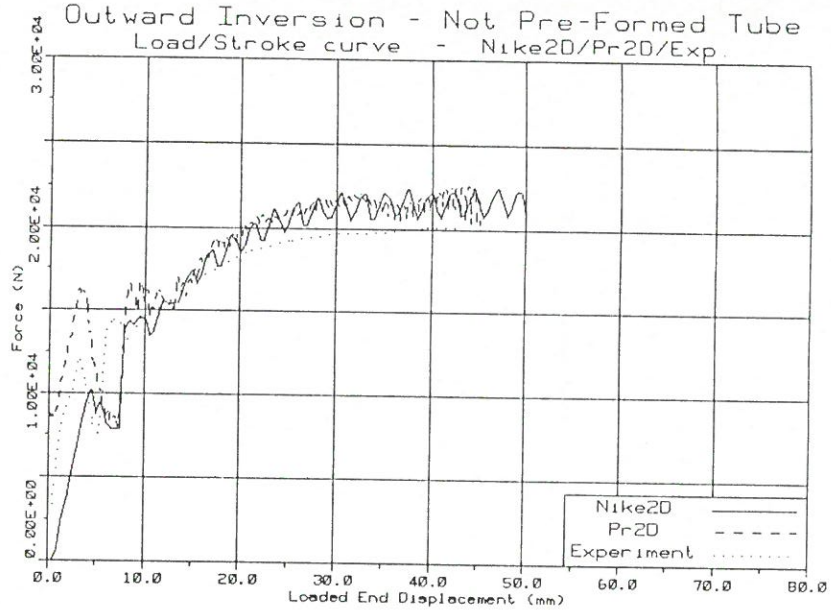

Figure 9. Case 2 - Computed load-displacement curves.

4. CONCLUSIONS

The four programs studied in this work have been able to model successfully the highly non linear tube inversion problems considered, with a reasonable use of computer resources. Results did not differ appreciably between the codes, and in particular between explicit and implicit approaches.

As explicit codes cannot model directly quasi-static processes, it would appear that implicit codes are the obvious approach for these problems. However, explicit programs using velocity scaling can provide also engineering accurate solutions at a reasonable cost.

In particular, when highly non-linear situations are encountered (e.g.presence of contacts), explicit programs are a good alternative because the explicit formulation of non linearities is simpler and more robust.

For the implicit solution of non-linear softening systems studied in this paper a quasi-Newton iteration method with line search seems to be computationally more efficient than a full Newton method. It needs more iterations to achieve convergence, but the total CPU time is smaller.

5. REFERENCES

1. JOHNSON, W. - Impact strength of Materials. Edward Arnold Publishers, London, 1972.

2. AL-HASSANI, S.T.S., JOHNSON, W. and LOWE, W.T.- Characteristics of inversion tubes under axial loading. Journal of Mechanical Engineering Science, 14, p.370, 1972.

3. GOICOLEA, J. - Numerical modelling in large strain plasticity with application to tube collapse analysis. Ph.D.Tesis. University of London, 1985.

4. HALLQUIST, J. - NIKE2D. A vectorized implicit, finiter deformation, finite element code for analyze the static and dynamic response of 2D solids with interactive rezoning and graphics. Report UCID-19677 rev.1. Lawrence Livermore National Lab. Livermore (California), 1987.

5. HIBBIT, KARLSSON & SORENSEN INC.-ABAQUS. User's Manual ver.4.7. Providence (Rhode Island), 1988.

6. PRINCIPIA, S.A. - PR2D. User's Guide. Report 11/87. Madrid, 1987.

7. HALLQUIST, J. - User's Manual for DYNA2D. An explicit two-dimensional hydrodynamic finite element code with interactive rezoning and graphic display. Report UCID-18756 rev.3. Lawrence Livermore National Lab. Livermore (California), 1987.

Application of Component Crash Simulation

Heng-yee Chen

Ford Motor Company

SUMMARY

The load-path, the fineness of the finite element mesh, the choice of element types, the location and the simulation of the welds, and the formulation of the contact/sliding surface, each contribute in determining the collapse mode of the analyzed structure.

In crash simulation, welds should be simulated as hinges, contact surfaces and fine meshes must be used; the usage of beams, triangular elements and spring elements should be avoided if possible. Notwithstanding the fact that the load-displacement curve of the computer simulation deviates from the test curve, crash simulation can be used for comparative study.

INTRODUCTION

In general, the crash simulation of a vehicle system has been an extremely difficult task. It was not until the "contact element" from DYNA3D of Lawerence Livermore National Laboratory became available to the non-linear finite element community, that stress analysts were then able to perform a crash simulation more realistically.

The DYNA3D Non-linear Computer Program has proven to be an important computational tool for performing dynamic crash analyses of vehicle systems. The CDC/DYNA3D, MSC/DYNA, PAMCRASH, and RADIOSS computer codes are each derivatives of DYNA3D, some being used now as design tools by various car manufacturers around the world.

Vehicle crash simulation is, in essence, a detailed stress analysis. The finer the finite element mesh is, the better the results correlate.

Due to the complexity of the vehicle structure, as well as the intensity of the calculation involved in vehicle crash simulation, it is impractical to determine the critical mesh size of the vehicle model. However, disregarding the fineness of the finite element model, crash simulation can undoubtedly determine the crash mode in the global sense. For a given mesh size and constraints, comparative analyses are reliable in finite element analysis.