PERGAMON

Computers and Structures 72 (1999) 31–47

Computers & Structures

Advances in crush analysis

Klaus-Jürgen Bathe[a,*], Jan Walczak[b], Olivier Guillermin[b], Pavel A. Bouzinov[b], Heng-Yee Chen[c]

[a]*Massachusetts Institute of Technology, Mechanical Engineering Dept., Cambridge, MA 02139, USA*
[b]*ADINA R&D, Inc., 71 Elton Avenue, Watertown, MA 02472, USA*
[c]*Ford Motor Company, 20000 Rotunda Drive, Dearborn, MI 48121, USA*

Abstract

Advances in capabilities for the crush analysis of structures are presented. Such analyses are difficult to perform and require state-of-the-art analysis procedures: efficient and reliable shell elements, an effective and general contact algorithm, efficient procedures to calculate the element stresses in elasto-plasticity, the use of consistent tangent matrices, effective nonlinear incremental solution strategies and the efficient solution of the algebraic finite element equations. Moreover, the effectiveness of the complete analysis process can only be achieved by ensuring that each of the above solution procedures is in an effective manner integrated into a complete solution scheme. This paper focuses on the advances developed in ADINA and presents various analysis cases solved with the program.

1. Introduction

Nonlinear finite element analysis of structures, fluids and fluid-structure interactions is conducted to an increasing extent. While in certain areas nonlinear analysis can now be performed in almost a routine manner, in many other areas, the state-of-the-art is continuously being advanced. One such area in which recently significant advances have taken place is the crush analysis of structures.

An important application area of such analysis capabilities is the crush analysis of motor car models.

While frequently the same analysis procedures are used for a *crush* analysis as for a *crash* analysis, it is important to recognize the fundamental differences in the crash and crush phenomena. Crash analyses simulate fast phenomena requiring a transient analysis. For example, in the crash analysis of a motor car, the crashing of the car at about 30 mph into a rigid wall is simulated. The main event takes place during a few milliseconds. These analyses are generally conducted using explicit analysis procedures with shell elements and assumptions specifically developed for modeling the transient solution. On the other hand, crush analyses simulate slow phenomena that are appropriately analysed using static analysis procedures or implicit dynamic techniques. For example, in the crush analysis of a motor car, the purpose is to establish the ultimate strength of the car body in a static situation. The ultimate strength affects the behavior of the car in various operating conditions, including when the car overturns in an accident. The laboratory experiment to identify the crush behavior is performed by crushing the car slowly, at about 0.02 mph and for about 10–30 s using a device to push a thick steel plate onto the car roof and measuring the load-deformation relation.

* Corresponding author. Tel.: +1-617-253-6645; fax: +1-617-253-2275.

0045-7949/99/$ - see front matter
PII: S0045-7949(99)00041-3

In general, a crush analysis is rather difficult to achieve. The reasons for the difficulties lie in that a slow speed almost static analysis requires particular robustness and efficiency of the solution algorithms. The shell elements must be of high predictive capability, be robust and computationally efficient for large deformation *static* analysis. The stress conditions must be evaluated efficiently including the plasticity and rupture conditions. The contact algorithm must allow three-dimensional multiple body and self-body contact on the outer and inner surfaces of the car shell, be robust and give fast convergence in the iterations for *static* equilibrium at the different deformation states. Consistent tangent matrices must be formed for the large deformation elasto-plastic and contact conditions, and the nonlinear incremental solution strategy with the solution of the algebraic finite element equations must be effective [1].

Frequently, a crush analysis is performed using an explicit crash analysis code, because the problem with the strong nonlinearities cannot be solved with implicit analysis options. However, such an analysis approach may not be appropriate as was already discussed in Ref. [2].

The objective of this paper is to summarize some thoughts regarding crush analysis and report on some analysis procedures that can be employed effectively for crush simulations. Since crash analysis codes are frequently used for crush analysis, outlined in the next section are the difficulties that can arise in a crush simulation when a crash analysis code is used. Section 3 then focuses on effective analysis procedures for crush solutions with particular emphasis on the specific requirements for such analyses. This discussion is followed in Section 4 by the presentation of some analysis cases. These analyses represent in many respects the current state-of-the-art of the nonlinear analysis of structures. Finally, in Section 5, conclusions regarding the current state of crush analysis and comments on further developments in the field are presented.

2. Crush analysis with crash analysis programs

As was pointed out above, the basic physical difference between crash and crush conditions is that a crash event (taking a few milliseconds) requires a dynamic solution, whereas a crush event (taking seconds) requires a static (or implicit dynamic) solution. However, the nonlinearities in a crush event are very large and the static solution algorithms used must be particularly powerful. For this reason, frequently, an explicit dynamic solution is performed to assess the crush behavior of the structure under consideration. The governing equations considered for a typical time step are then, using the central difference method [1],

$$\mathbf{M}{}^{t}\ddot{\mathbf{U}} = {}^{t}\mathbf{R} - {}^{t}\mathbf{F} \tag{1}$$

$${}^{t}\ddot{\mathbf{U}} = \frac{1}{(\Delta t)^2}\left({}^{t+\Delta t}\mathbf{U} - 2\,{}^{t}\mathbf{U} + {}^{t-\Delta t}\mathbf{U}\right) \tag{2}$$

$$\Delta t \le \Delta t_{\mathrm{cr}} = \frac{2}{\omega_{\mathrm{n}}} \tag{3}$$

where ${}^{t}\mathbf{R}$ is the vector of externally applied nodal loads, ${}^{t}\mathbf{F}$ is the vector of nodal point forces corresponding to the element stresses, $\mathbf{M}$ denotes the lumped mass matrix, and ${}^{t}\mathbf{U}$, ${}^{t}\ddot{\mathbf{U}}$ are the vectors of nodal point displacements and accelerations. The superscripts $(t-\Delta t,\ t,\ t+\Delta t)$ denote the times considered. With the conditions known at times $t-\Delta t$ and t, Eqs. (1) and (2) can directly be solved for the displacements at time $t+\Delta t$, and since $\mathbf{M}$ is a diagonal matrix, the solution is very effective for each time step. Special equations are used involving the initial conditions to start the integration process. In Eqs. (1) and (2), velocity-dependent damping forces are neglected and a constant time step size is assumed. These restrictions can of course easily be removed [1].

Eq. (3) expresses the fact that for a stable integration, the time step Δt must be smaller than the critical time step Δt_{cr} given by the largest frequency in the mesh, ω_{n}.

As seen in Eqs. (1) and (2), in the explicit time integration solution, a stiffness matrix is not formed, iterations for equilibrium at the time steps considered are not carried out, and instead a simple time-marching-forward solution is produced. This analysis approach is attractive because difficulties with respect to convergence in equilibrium iterations do not arise.

This explicit analysis approach can be used, but requires many time-marching steps, because the explicit solution scheme is only conditionally stable: that is, the time step used must be smaller than the critical step size Δt_{cr}. Usually, in shell analyses, the critical time step size is of the order of a microsecond. Therefore, millions of time steps must be carried out to obtain a physically correct solution. The computational time for such a solution is very high, and ways to reduce the required number of time steps have been sought. In one approach, the speed of the event is increased, resulting in a smaller number of time steps required for solution. In another, somewhat equivalent approach, the density of the material is artificially increased, resulting in a larger critical time step size, and hence also into a smaller number of required solution steps. Some success with these approaches has been reported in metal-forming analysis, largely because the structures analysed are highly constrained by displacement boundary conditions. However, in a crush analysis, the crushing device is typically applied to a structure largely free from displacement constraints, and then these approaches result in artificial inertia forces that can have a significant effect on the structural response and the crushing force. Of course, the magnitudes of the artificial inertia forces can be measured, which will give an indication of the error in the solution. However, for this error to be acceptably small, in essence the actual physical conditions must be used. For an example analysis in which these solution differences have been studied in detail, see Ref. [2].

Of course, if the explicit solution is carried out at the physical conditions of the problem (that is, without, for example, artificially increasing the speed of crushing or the material density), it is also necessary to use solution algorithms that are stable and accurate during the time integration [1]. Since the conditions then correspond to a static analysis, the shell elements, contact algorithms and other procedures used in the explicit dynamic solution must be applicable to these analysis conditions (which is not necessarily the case when a crash analysis program is employed, see Section 4.2).

Considering an implicit dynamic crush analysis, which reduces to an incremental static analysis if the time steps are large, the following equations are solved in a Newton–Raphson iteration for a typical time $t+\Delta t$ considered [1]

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F} \tag{4}$$

where the matrices and vectors are as defined below Eq. (3), but a consistent mass matrix is usually used because more accurate results can then be expected. The Newton–Raphson iteration to solve Eq. (4) is, for $i=1, 2, \ldots$, until convergence measures are satisfied,

$$\mathbf{M}\,{}^{t+\Delta t}\ddot{\mathbf{U}}^{(i)} + {}^{t+\Delta t}\mathbf{K}^{(i-1)}\Delta\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(i-1)} \tag{5}$$

where

$${}^{t+\Delta t}\mathbf{U}^{(i)} = {}^{t+\Delta t}\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)} \tag{6}$$

and the 'initial conditions' correspond to the state calculated for time t

$${}^{t+\Delta t}\mathbf{U}^{(0)} = {}^{t}\mathbf{U}, \quad {}^{t+\Delta t}\mathbf{F}^{(0)} = {}^{t}\mathbf{F}, \quad {}^{t+\Delta t}\mathbf{K}^{(0)} = {}^{t}\mathbf{K} \tag{7}$$

In Eq. (5), $\mathbf{K}$ is the tangent stiffness matrix and the iteration counter is given by the superscript (i). Eqs. (4) and (5) are used with a time integration scheme (typically the trapezoidal rule is used), and the iteration is continued until convergence to a specified tolerance is reached. Note that in Eqs. (4) and (5), the contact conditions are not explicitly indicated. If a penalty-type procedure is used, the stiffness matrix $\mathbf{K}$ and force vector $\mathbf{F}$ are appropriately modified, whereas if a Lagrange multiplier technique is employed, additional (tangent matrix) equations are assembled and solved [1].

Considering the implicit solution approach, the major computational differences to the explicit dynamic solution lie in that tangent stiffness matrices are established, equilibrium iterations are performed and, of course, much larger incremental load (or time) steps are used. Typically, for a crush analysis, about 100–200 steps are only required. In addition, it is important to note that there is also significantly more physical insight that can be gained in the analysis. Namely, if an incremental static analysis is carried out, the solution indicates bifurcation points that in an explicit dynamic solution are not detected, because no stiffness matrix is calculated and the artificial inertia effects mask these physical instabilities. It can be important that such bifurcation points are detected because they indicate that additional equilibrium paths, beyond those calculated are actually possible. In general, one of these additional equilibrium paths, and not the path calculated, may actually be observed in a physical experiment or in the field.

3. On finite element procedures for crush analysis

As pointed out in the previous section, the finite element procedures must be particularly robust and effective for a crush analysis. The objective in this section is to give some thoughts to the techniques that are considered suitable for crush solutions.

3.1. Shell elements

An important ingredient of a finite element model for crush analysis is the shell element employed. Any elements used must fulfill the reliability and effectiveness criteria enumerated by the authors for some time [1–4]: the elements must not contain any spurious zero energy modes, must not be based on artificial numerical factors, must not lock in membrane and shear actions, should be applicable to general shell geometries, and ideally should display, independent of the shell geometry, the optimal convergence rate consistent with the interpolations used. The MITC shell elements seem to represent the state-of-the-art according to these criteria as evidenced in references [5] and [6]. In particular, the element formulations are not based on any numerical factors (as are used in stabilized finite element formulations) and show good convergence behavior for membrane and bending dominated problems.

The general applicability, reliability and effectiveness referred to above are difficult to achieve, but much progress has been made through physical intuition, mathematical analysis and well-chosen numerical experiments. The mathematical model underlying the general shell elements used in practice (which are based on three-dimensional continuum mechanics theory degenerated to shell behavior), and the considerations for stability, convergence and optimality of shell elements for general applications are discussed in Refs. [4–7].

A crucial part of the evaluation of shell elements is to solve judiciously chosen problems in a study of the convergence behavior. Such problem solutions are presented in Ref. [5], where it is shown that the MITC shell elements perform well for the spectrum of types of practical analyses encountered.

3.2. Large deformations and plasticity

The classical total and updated Lagrangian formulations are effective for the solution of very large deformations of solids and structures [1]. These formulations have the advantage that they are *total* and not rate-type formulations, which means that no integration process is required in the kinematic updating process. The formulations are also efficiently employed with a fully implicit scheme to calculate the plasticity variables, and the effective-stress-function (ESF) algorithm to solve for the stresses is used [1,8–11]. The Lagrangian formulations with the elasto-plasticity ESF solution technique admit relatively large incremental steps and consistent tangent matrices can be evaluated [1].

The details of the MITC shell element formulation for large deformation analysis are given in Ref. [1]. A key ingredient in the formulation is that nodal point director vectors are used whose positions and directions are updated in each incremental solution step. This procedure allows for very large displacements and rotations, in elastic and elasto-plastic conditions.

Of particular significance is that elements with spurious zero energy modes (for example, due to reduced integration) should not be used in practice. An example solution, in which due to reduced integration, *element zero energy modes* result in *nonzero ghost frequencies* of the complete structure is presented in Ref. [1], p. 473.

In elasto-plastic analysis, there is a tendency to use simple reduced integration to relieve the locking phenomenon due to the incompressibility constraint (and the locking phenomena due to artificial membrane and shear stresses). Such reduced-integrated elements should be employed with great care because of the unphysical solutions that can be generated.

3.3. Contact algorithm

In crush analysis, contact conditions are an important phenomenon to be solved for. Let g be the gap between two contacting media (for example, two thin shells) and λ be the magnitude of the normal component of the contact traction (compression being positive). Then the conditions for normal contact are:

$$g \geq 0; \quad \lambda \geq 0; \quad g\lambda = 0 \tag{8}$$

where the last equation expresses the fact that the gap must be zero when λ is greater than zero, and vice versa.

Let t be the magnitude of the tangential component of the contact traction and $\dot{u}$ be the magnitude of the relative tangential velocity between the contacting media at the point considered; then, using Coulomb's law of friction, with μ the friction coefficient, the tangential conditions during contact are (with $\lambda > 0$)

$$\left|\frac{t}{\mu\lambda}\right| \leq 1; \quad \left|\frac{t}{\mu\lambda}\right| < 1 \quad \text{implies } \dot{u} = 0$$

$$\left|\frac{t}{\mu\lambda}\right| = 1 \quad \text{implies } \dot{u} \text{ can be} \neq 0 \tag{9}$$

The solution of solids in contact conditions therefore entails the usual equations of analysis of the media plus the satisfaction of Eqs. (8) and (9).

Developed, in ADINA, is the constraint function algorithm [1,12,13], which can be used to solve very complex contact conditions, including two-sided contact, contact of a surface onto itself, and contact with thickness offset as employed in thin shell analyses. The solution algorithm has been developed for static analysis and dynamic solutions. A key ingredient is that *continuous and differentiable* (constraint) functions for any values of g, λ, $\dot{u}$ and $t/\mu\lambda$ are used to enforce the conditions of Eqs. (8) and (9). In the implementation, a Lagrange multiplier method to impose the constraint functions is employed but, of course, a penalty method could be used as well [1]. Because of the continuity of the functions, derivatives can be taken to obtain a Jacobian that is incorporated into the tangent stiffness matrix. However, full quadratic convergence is frequently not obtained, because the models of the contact surfaces are not sufficiently smooth.

3.4. Solution of equations

The solution of a crush problem using implicit dynamic or static analysis procedures is achieved by solving Eq. (4) at the selected load levels (of course, in the case of a static analysis, neglecting the inertia forces in Eq. (4)). However, if the incremental load level is too large, difficulties in the convergence of Eq. (5) are encountered. In this case, a solution strategy must be employed that appropriately reduces and then again increases the incremental load level [1,8]. In the ADINA program, a load-displacement-constraint (LDC) procedure and an automatic-time-stepping (ATS) scheme can be used. These procedures have been employed abundantly for large deformation analyses.

In each case, using the LDC method or the ATS procedure, algebraic equations of the form given in Eq. (5) must be solved. Until a few years ago, these equations were most efficiently solved using skyline or wave front techniques. However, during the recent years, sparse solution techniques have been developed [1,14] that are much more efficient—frequently, they are an order of magnitude faster than the earlier employed skyline solution methods. It is clearly imperative that a sparse solver be used in a crush analysis. The ADINA sparse solver can be employed as an in-core or out-of-core solver, and in parallel-processing. The solution of the algebraic equations is achieved in the following steps: the reordering of the equations using graph theory, the symbolic factorization, the numerical factorization, and the vector forward reduction and back substitution. Table 1 gives some solution times obtained with the ADINA sparse solver.

4. Illustrative solutions

While all the developments described above have importance in their own right, an effective solution to an actual crush problem can only be achieved, if these techniques are implemented together in an *overall* effective solution program. Our objective in this section is to illustrate the use of the solution schemes in the ADINA program. Firstly, the solutions to two rather simple illustrative crush problems are presented, merely to show some of the features that need be available in general applications. Secondly, the analyses of two motor car models are presented, for which the crush analysis procedures mentioned above are used to obtain effective solutions.

4.1. Crush analyses of cylinder and box

To demonstrate the ADINA solution capabilities in the analysis of some relatively simple problems, the cylinder and the box problems shown in Figs. 1 and 2 are considered. The analyses of these structures correspond to crush solutions, since the time scale of applying the displacements (δ) is long—1 second. and 1.45 second. in the case of the cylinder and the box, respectively.

Figs. 3 and 4 show the calculated responses of the structures. For the cylinder, an axisymmetric model was solved of 92 9-noded displacement/pressure (9/3) elements. No initial imperfection was imposed, and a total of 200 implicit dynamic analysis steps were used to calculate the response. The shown deformations developed 'naturally' in the step-by-step solution process. The force-deflection diagram is interesting in that the force drops every time a new wrinkle develops.

For the closed square box, a three-dimensional model representing one quarter of the box was solved. The 27-noded displacement/pressure (27/4) element was employed. No initial imperfections were imposed and a total of 145 static analysis steps were used for the solution. Fig. 4 shows the mesh and calculated deformations, and also the calculated force-deflection

Table 1
Sparse solver solution time records

Number of equations: 571,966 In-core solution MITC4 shell elements and self-contact used Computer: SGI O2000		
1 processor	2 processors	4 processors
338 s	228 s	165 s
Number of equations: 745,497 Out-of-core solution 10-node tetrahedral elements used Computer: HP V-Class		
1 processor	2 processors	4 processors
2336 s	1769 s	1472 s

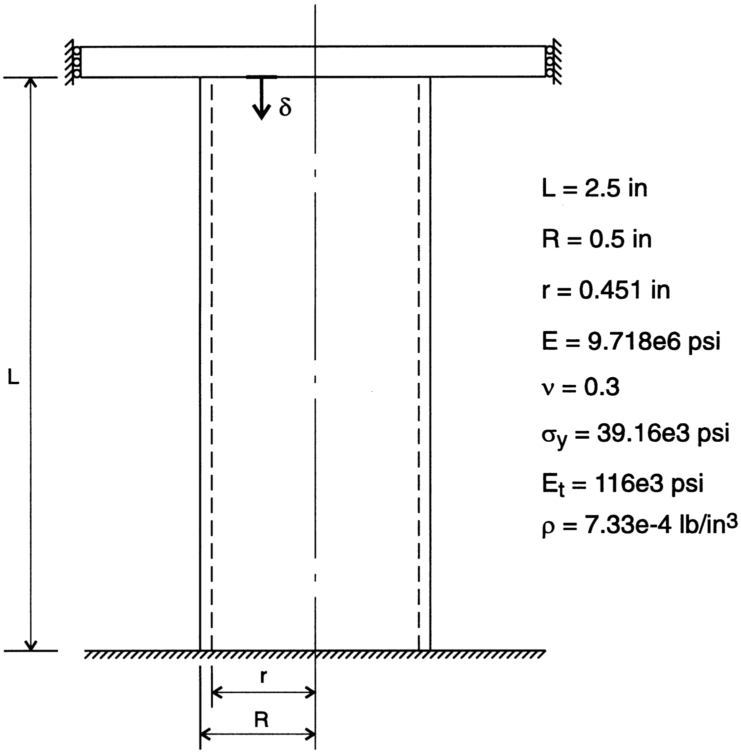


Fig. 1. Cylinder considered (not drawn to scale). Crush and crash axisymmetric analyses. δ is applied linearly as a function of time.

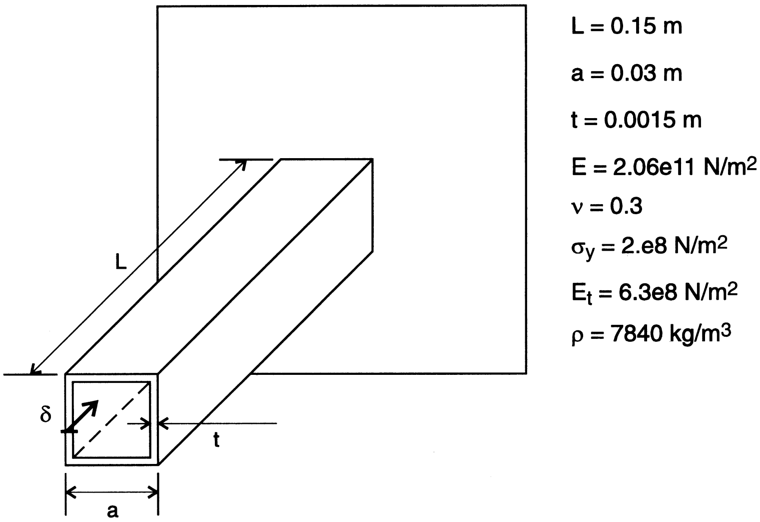


Fig. 2. Box considered (not drawn to scale). Crush and crash analyses. δ is applied linearly as a function of time.

(a)

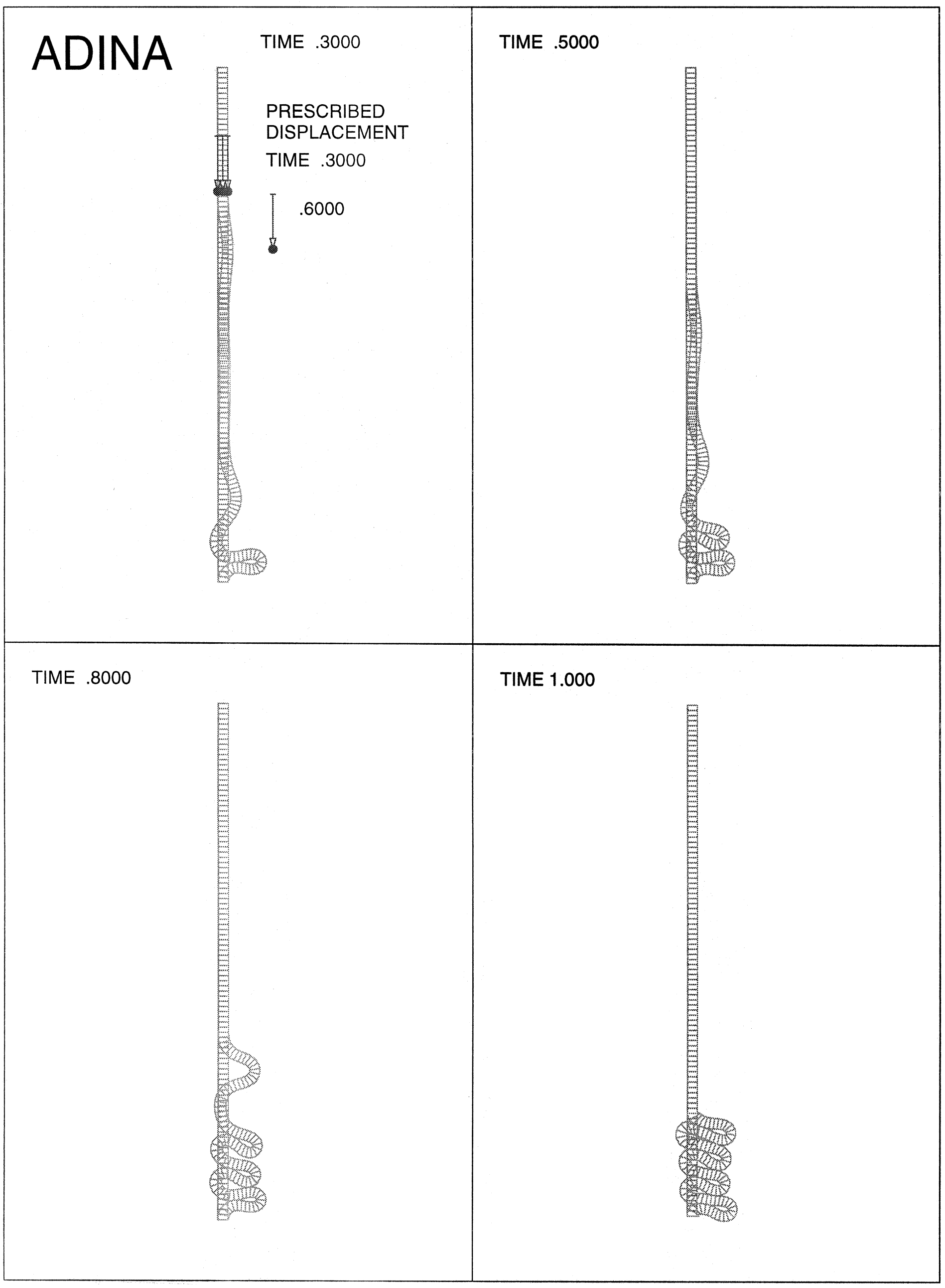
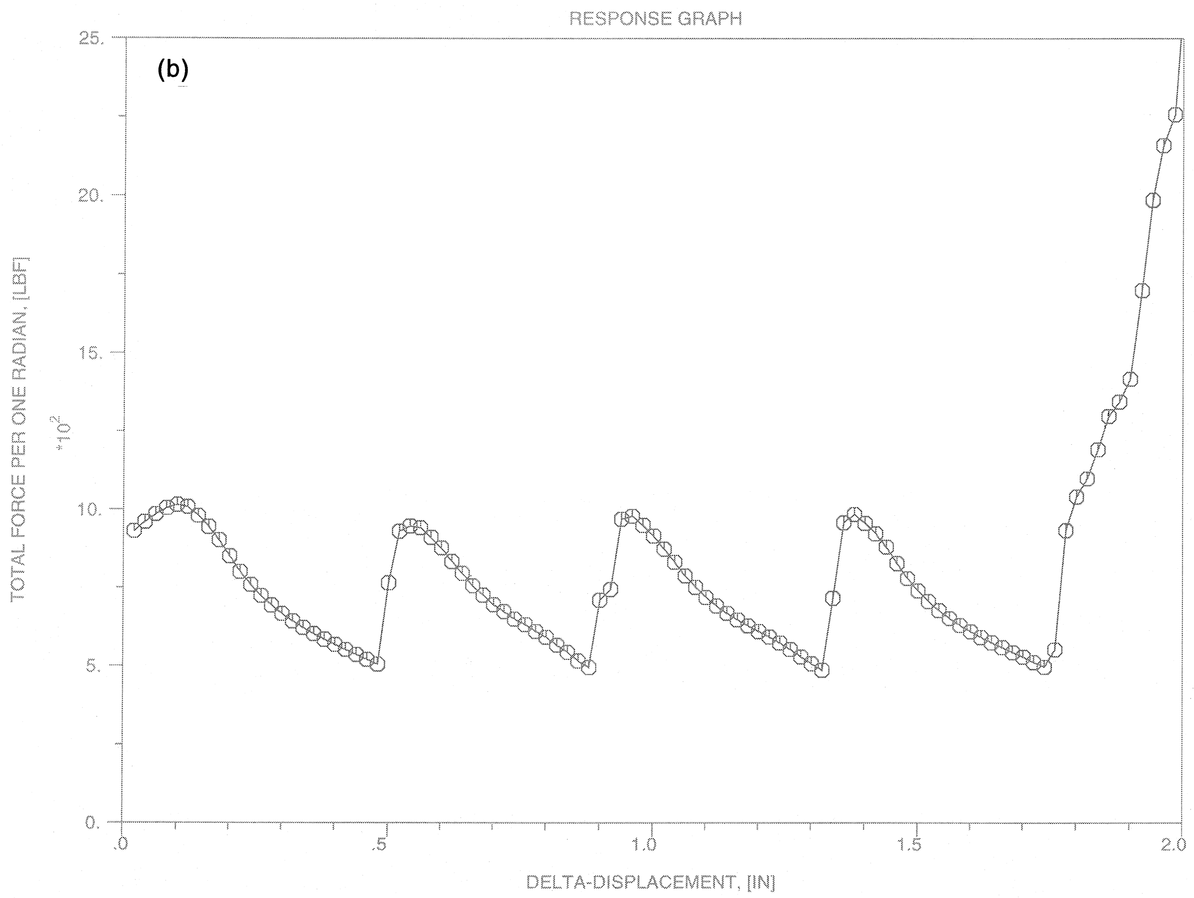


Fig. 3. Crush response of cylinder. (a) Deformations (showing development of wrinkles) (b) Crushing force-displacement relationship. (c) Effective plastic strain in crushed position at $\delta = 1.9$ in.



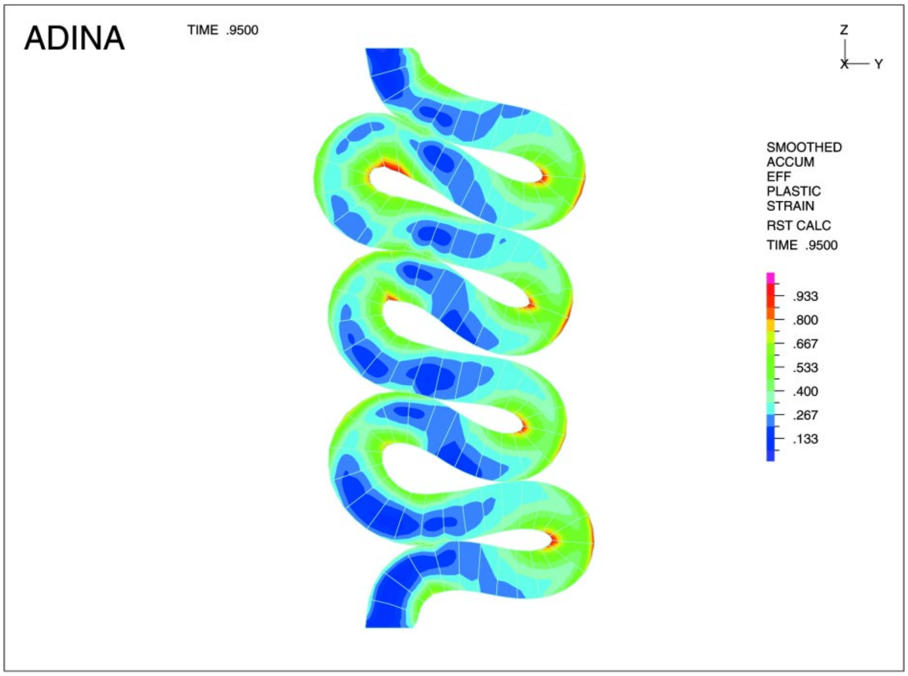


Fig. 3 (*continued*)

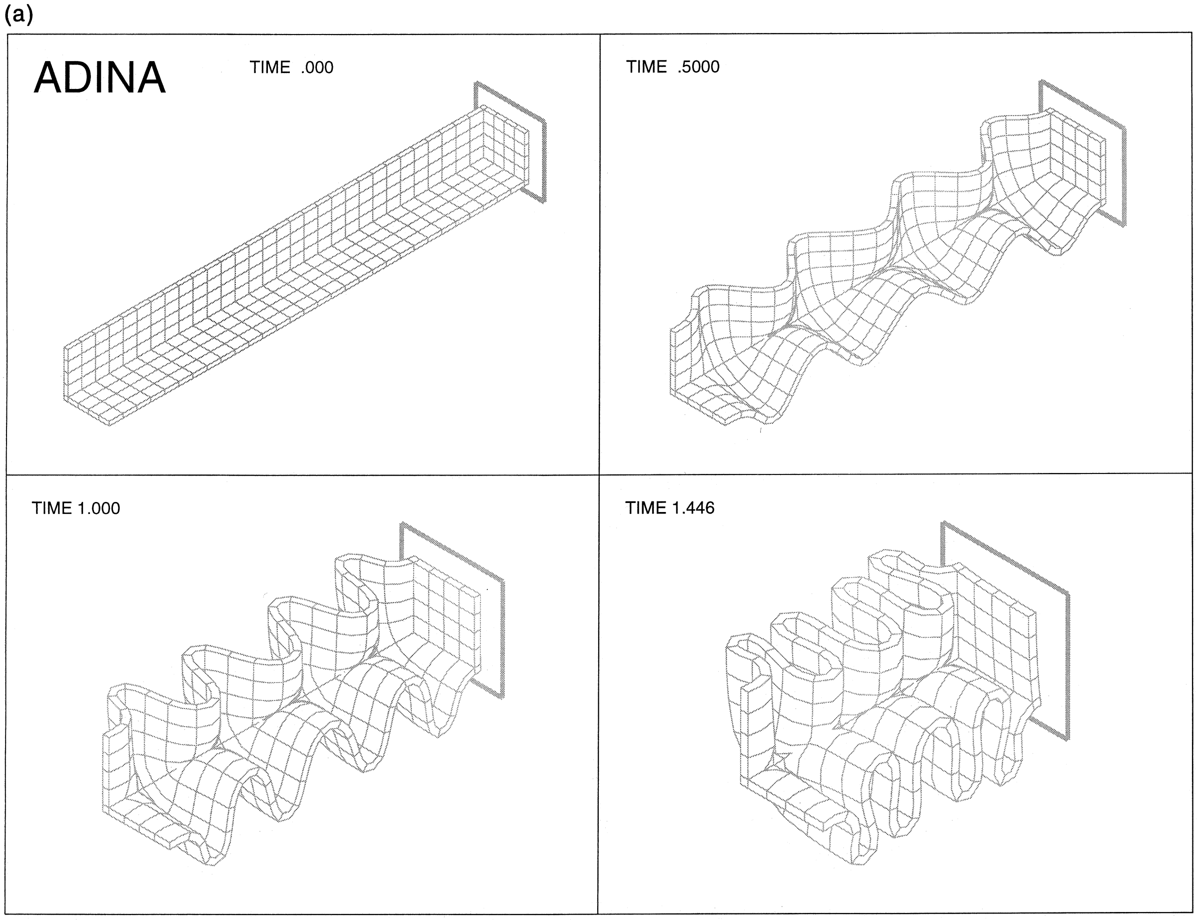


Fig. 4. Crush response of box. (a) Original mesh and deformations. (b) Crushing force-time relationship.

graph. In this case, the wrinkles do not get into contact, except in the corner of the box model, and hence the force is almost constant after the initial maximum value.

In each case, for the cylinder and the box, large strains are measured in the response solution, see Fig. 3(c). The contact conditions were modeled by specifying contact surfaces at the outer and inner surfaces of the structures and specifying that self-contact is allowed.

The two structures were also next analysed for their crash behaviors by imposing the end displacements (δ) to take place within 1 and 5 ms, respectively. The same finite element models were used and implicit time integration (the trapezoidal rule) was employed for the transient solutions. Figs. 5 and 6 show the calculated results. It is seen that in these dynamic analyses, the wrinkling takes place differently due to the significant inertia forces, and of course, the calculated time variations of end-forces are different when compared with the results given in Figs. 3 and 4. However, it is interesting to note that the maximum forces reached are almost the same in the static and dynamic analyses.

Considering all the analyses reported upon above, it was noted that while the overall response calculated in an analysis was not very sensitive to the mesh employed, the actual response path (for example, the manner in which the wrinkles developed) was somewhat sensitive to the finite element discretization used.

4.2. Crush analysis of small motor car model

The motor car finite element model shown in Fig. 7

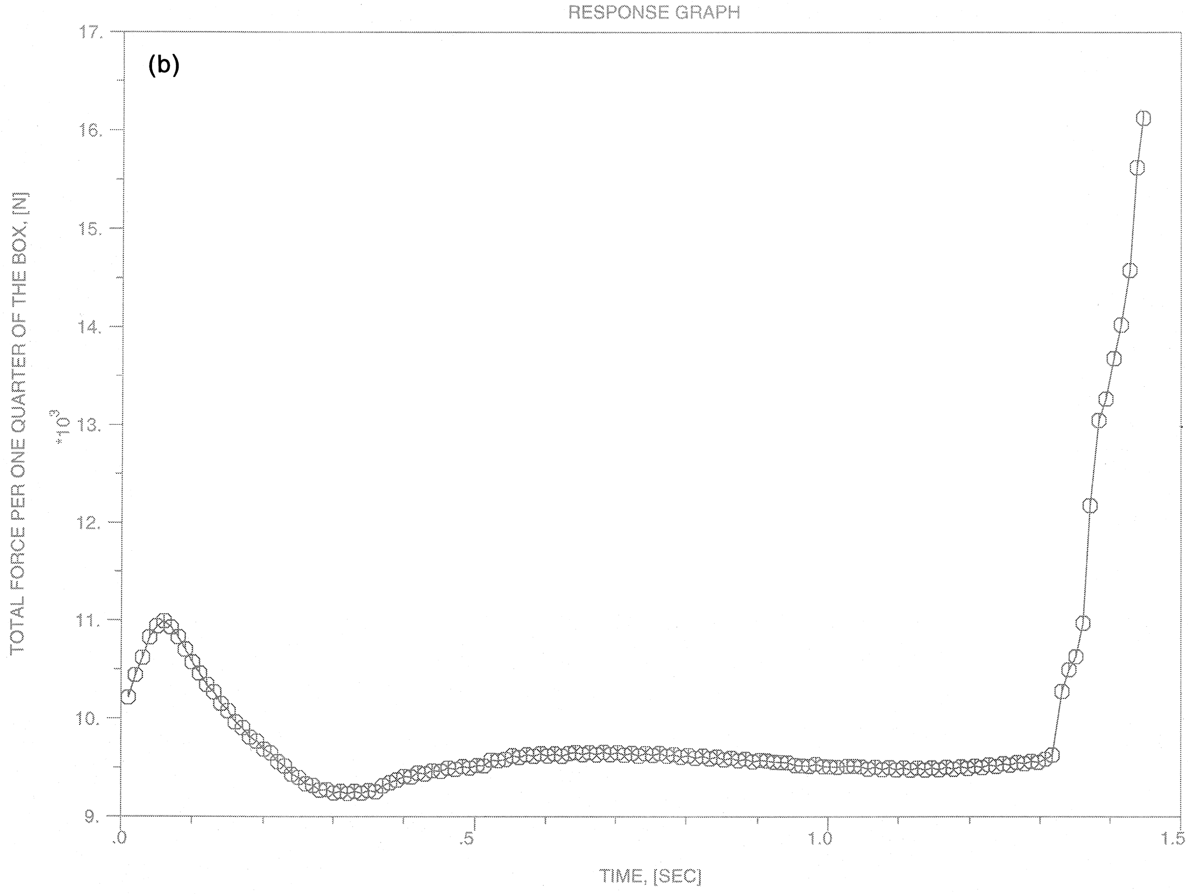


Fig. 4 (*continued*)

was analysed for its crush response. In the analysis, a heavy steel plate, assumed to be rigid, is applied to the front left corner of the car frame. The speed of crushing is 10 mm/s (which is about 0.022 mph). These conditions clearly correspond to a static analysis.

The ADINA finite element model was created from a Nastran input deck from which 4-node MITC4 shell elements were created to model the car structure and the windshield. Spring elements are employed to model connections between the doors and the windshield and frame, respectively.

Fig. 7 shows the calculated crushing load using ADINA as a function of the plate displacement. The same model was analysed as well using the LS-DYNA code (an explicit finite element program) and the calculated response is also shown in the figure. Considering the LS-DYNA solution results, as expected, the calculated crushing load depends on the speed of load application. Clearly, for a speed of 10 mph the crushing load is much too high. This would be indicated by the kinetic energy being too large in comparison to the strain energy of the finite element system. For a speed of 0.05 mph, which is deemed to be sufficiently close to the actual physical speed, however, the calculated response appears to be numerically unstable. Reasonable results are obtained using a speed of 0.5 mph, which is about 23 times larger than the actual physical speed of load application. Hence, with sufficient experience in the use of the program capabilities of an explicit code, acceptable results can probably be obtained. However, to reach these results some numerical experimentation may be necessary, which can be costly. Also, for the dynamic effects to be negligible, still many time steps need to be used (of the order of 100,000 instead of the order of one million for the actual physical speed of load application).

(a)

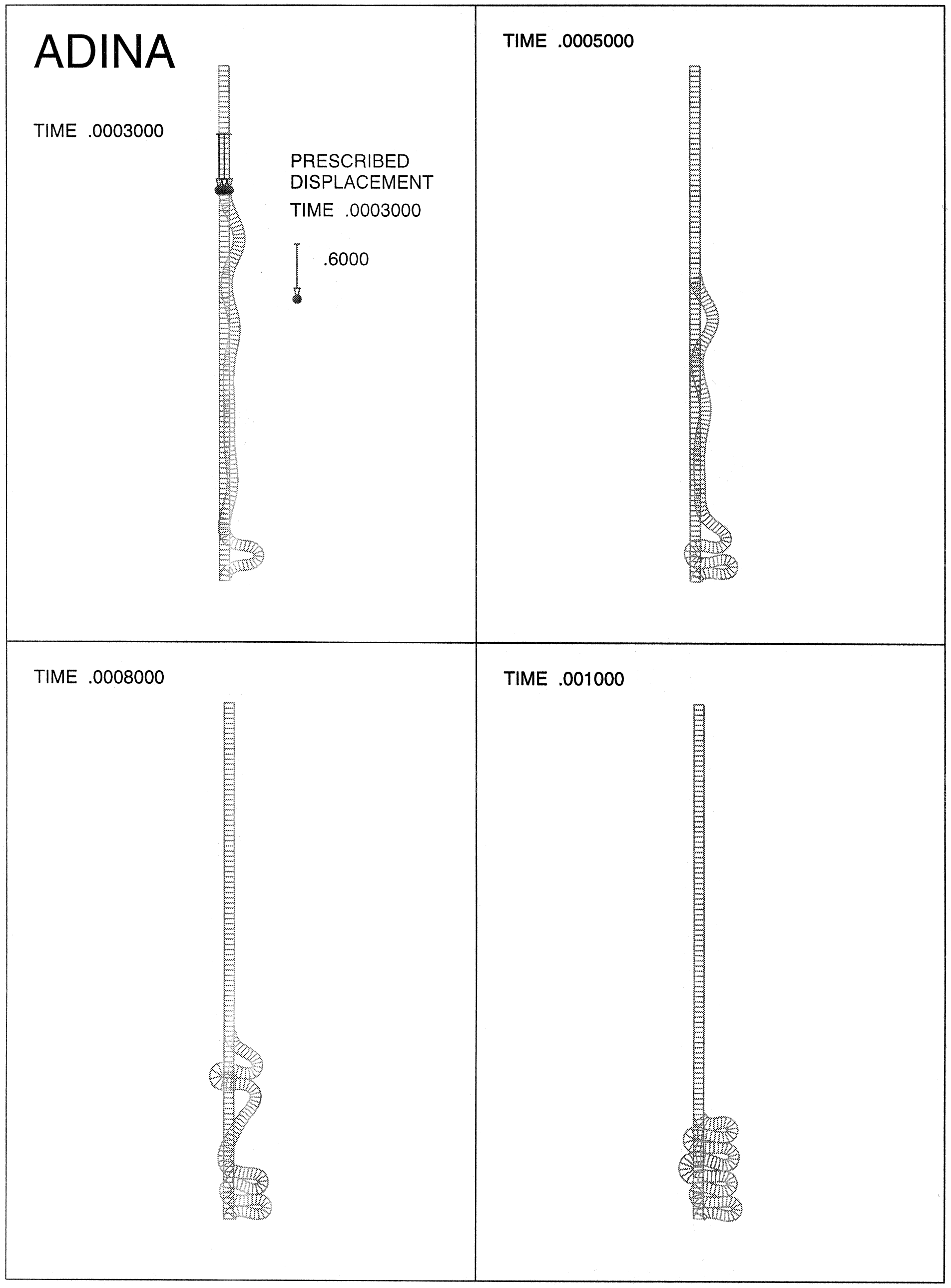
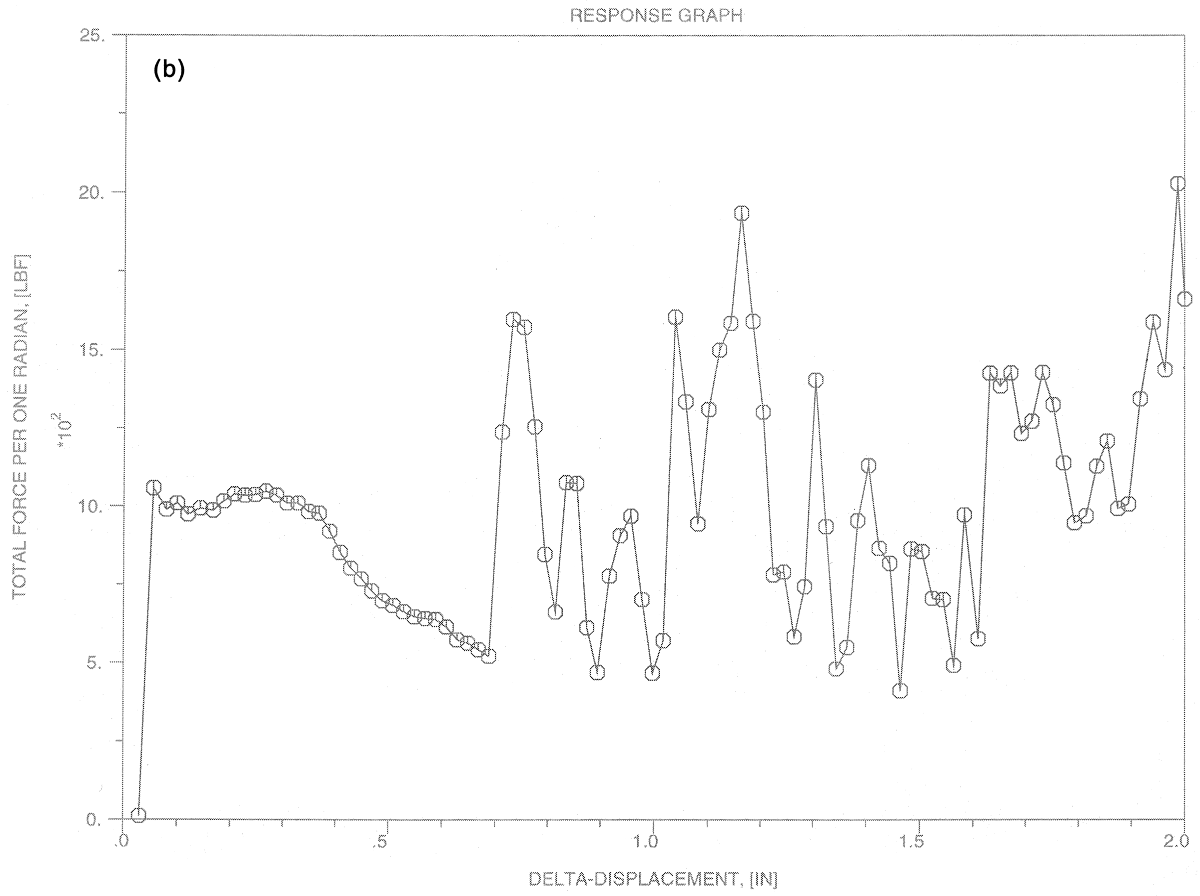


Fig. 5. Crash response of cylinder. (a) Deformations. (b) Crashing force-displacement relationship. (c) Effective plastic strain in crashed position at $\delta = 1.9$ in.



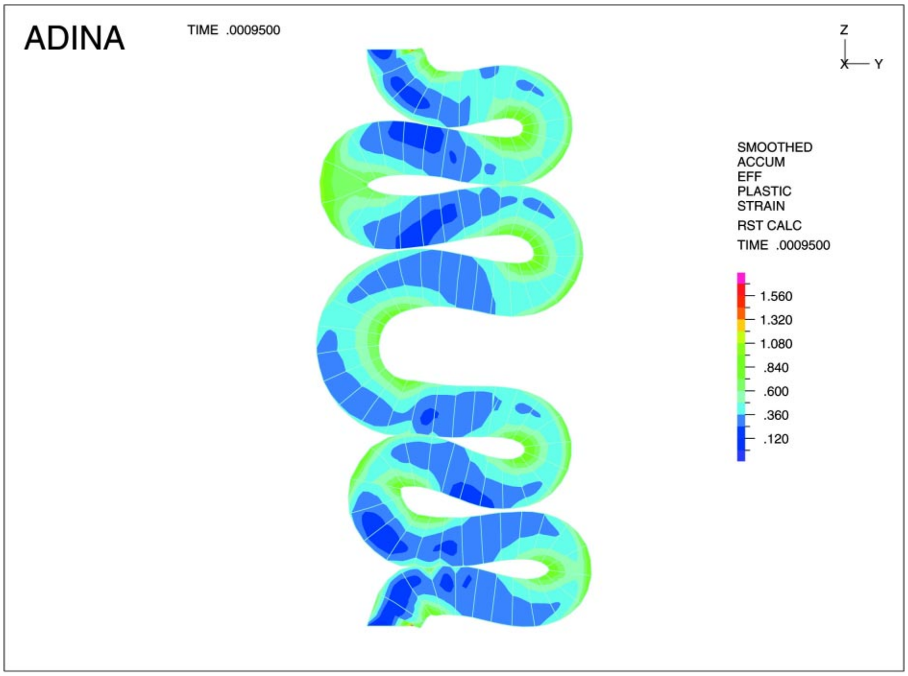


Fig. 5 (*continued*)

(a)

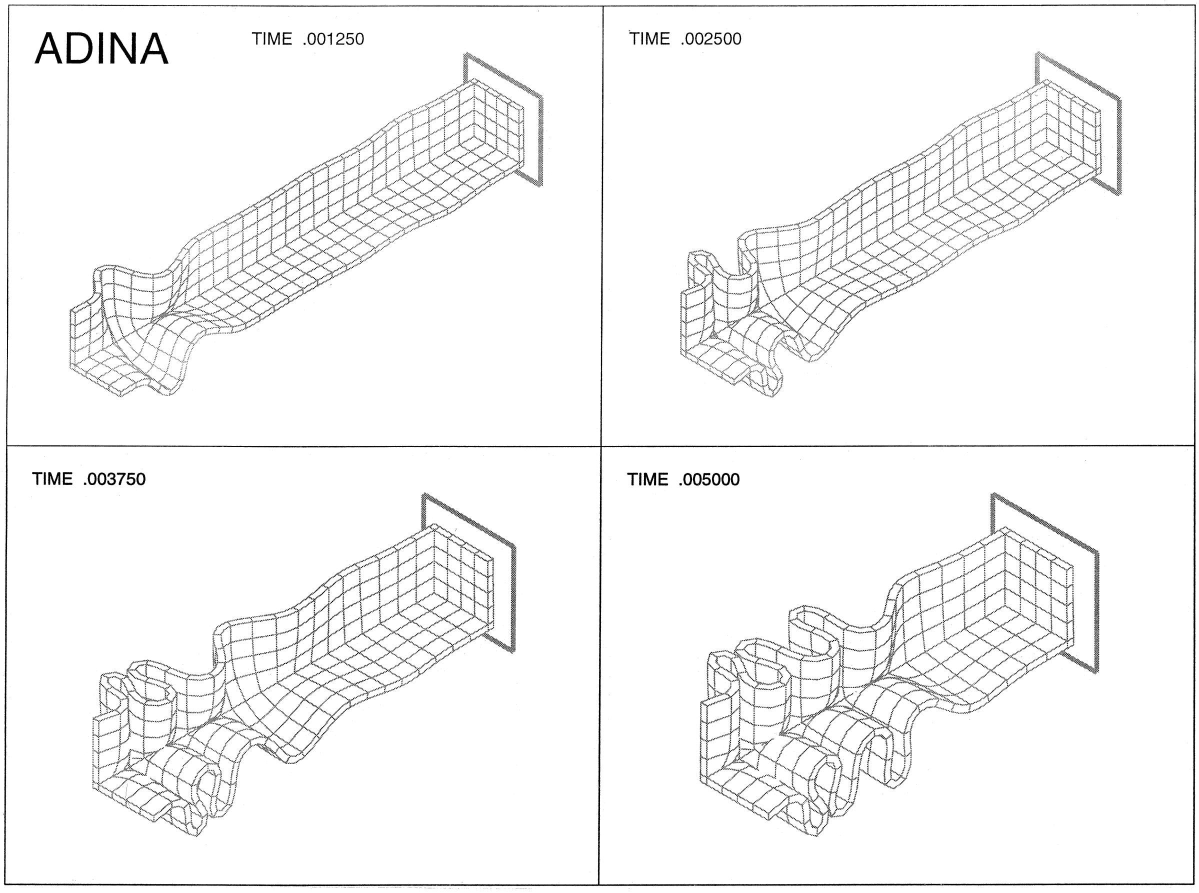


Fig. 6. Crash response of box. (a) Deformations. (b) Crashing force-time relationship.

For the ADINA solution 160 load steps were used and the complete analysis took about 8 h on a Silicon Graphics O2000 machine.

Unfortunately, for this analysis no physical test data are available for comparison.

4.3. Crush analysis of Ford Taurus model

The Ford Taurus shell model shown in Fig. 8 was also analysed for its crush response. The same physical conditions as in the above car crush analysis were imposed.

The model consists of 98,494 MITC4 shell elements, and is governed by 580,183 finite element equations. The ADINA solution shown in Fig. 8 was obtained using a total of 120 static incremental steps with a total run-time of about 48 h on a Hewlett Packard V-Class machine.

As is seen, the calculated response is quite close to the laboratory test data. It should be noted that only one model was created, *without any tuning*, and that model was run to obtain the results reported in Fig. 8.

5. Concluding remarks

The objective was to present some advances for the crush analysis of structures. Crushing a structure is a low-speed, practically static event. However, because of the solution difficulties with static analysis programs, so far, largely transient analysis procedures based on explicit time integration have been used for

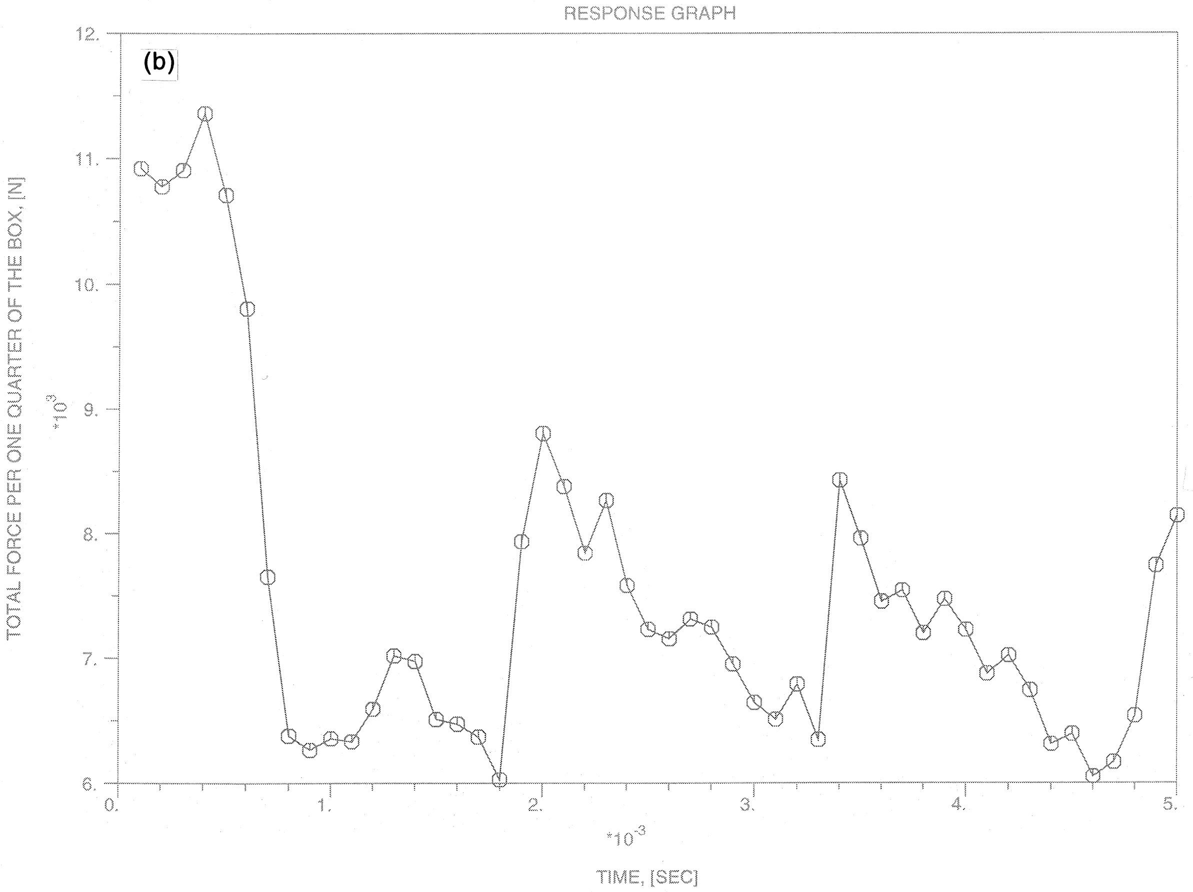


Fig. 6 (*continued*)

crush simulations. These analyses are computationally very expensive, unless severe artificial modeling assumptions regarding the speed of crushing, mass density of the materials, or other analysis ingredients are made. In addition, the transient analysis procedures may become numerically unstable at the correct physical conditions. Hence, artificial conditions are frequently used with numerical experimentation to 'assure' a proper response prediction.

It is shown in this paper that, at the current state-of-the-art, solution capabilities are available with which the correct physical conditions can be modeled and the static or low-speed dynamic solutions can be obtained in an effective manner. Indeed, the solution times used in the static analyses appear to be less than in the explicit dynamic response calculations. Hence, there is good reason to perform crush analyses in the manner demonstrated in this paper. Of course, continued efforts will be made to strive for further improvements in the static and implicit dynamic analysis capabilities, such as provided by increasing the efficiency of the contact algorithm, using possibly higher-order MITC shell elements, and through more effective parallel processing. These improvements with the continuous advances in computer hardware will render the solution of crush problems in the way demonstrated in the paper even more attractive.

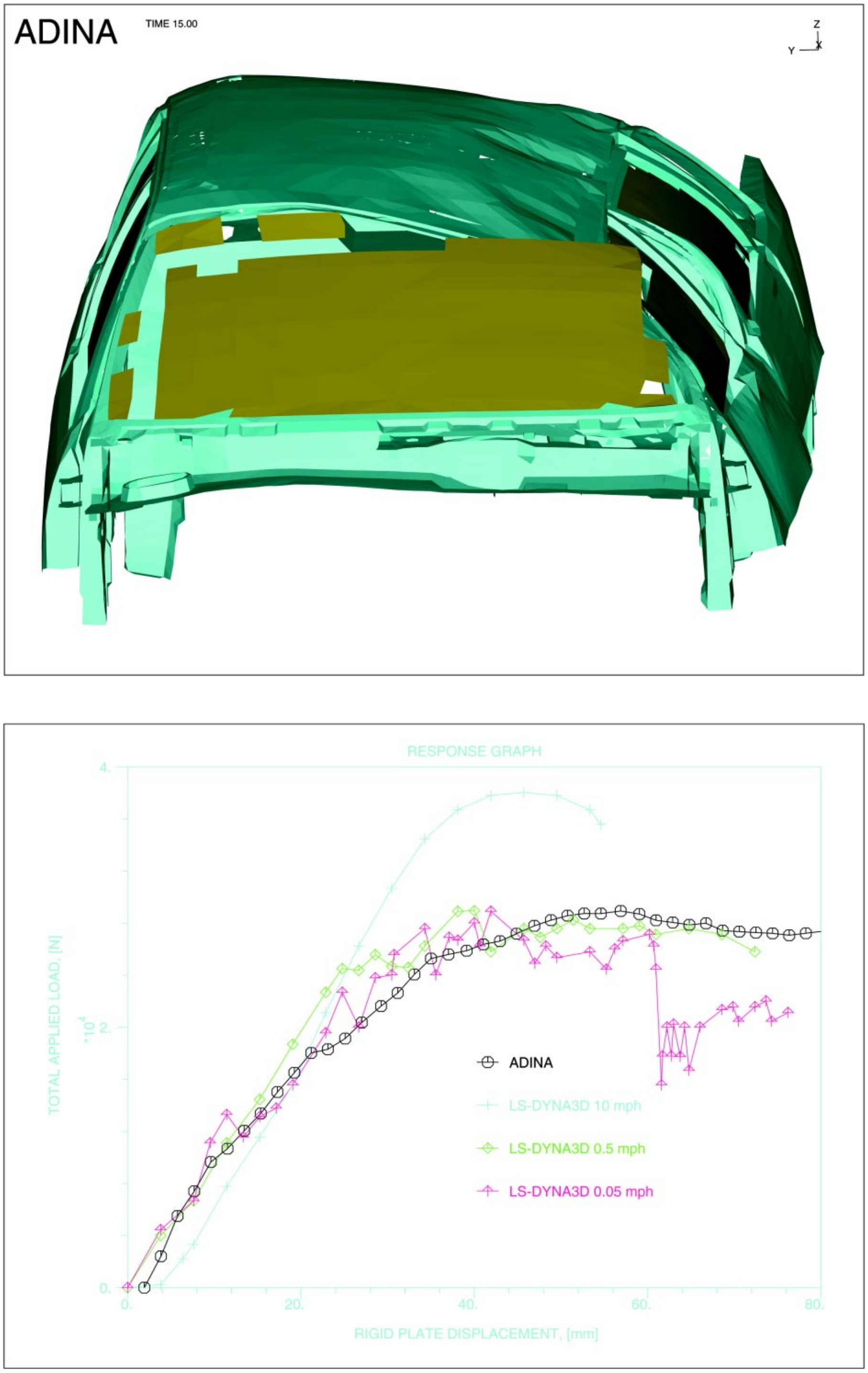
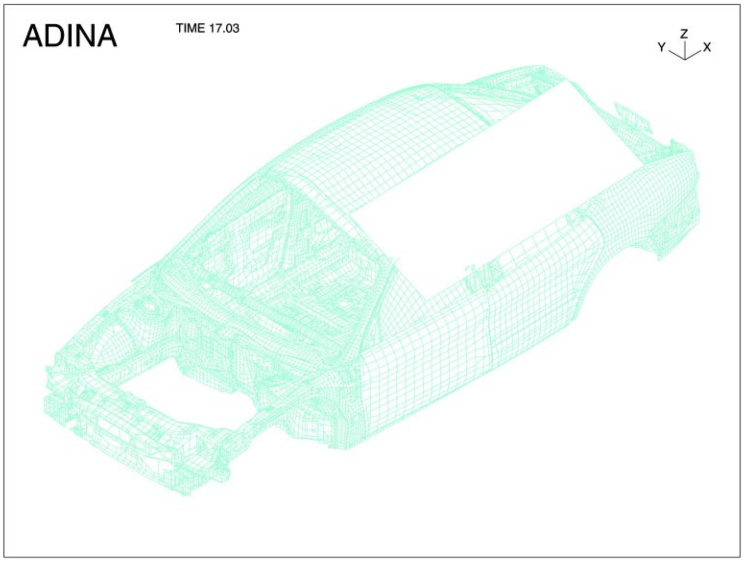


Fig. 7. Crush response of small motor car model. (Top) Deformations. (Bottom) Crushing force-displacement relationships.



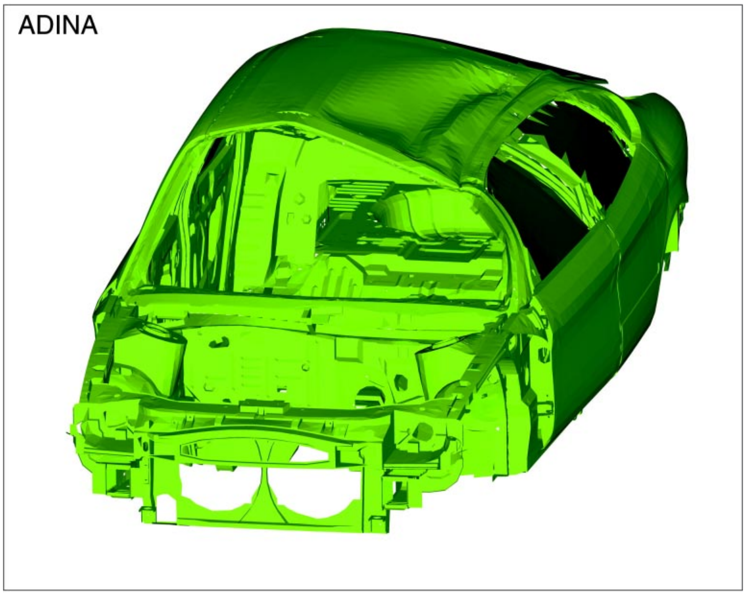


Fig. 8. Crush response of Ford Taurus car model. (Top) Mesh used (also rigid steel plate is shown). (Bottom) Deformations. (Overleaf) Crushing force-displacement relationship: calculated values and test data.

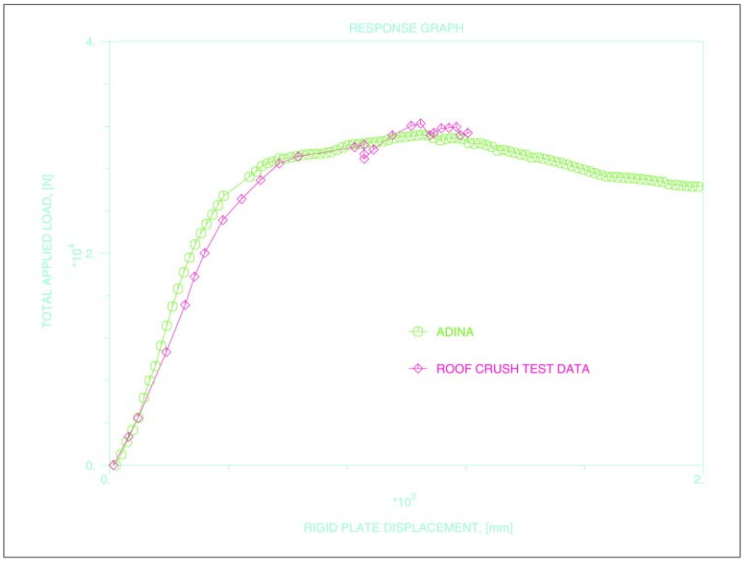


Fig. 8 (*continued*)

References

[1] Bathe KJ. Finite element procedures. Prentice Hall, 1996.

[2] Bathe KJ, Guillermin O, Walczak J, Chen H. Advances in nonlinear finite element analysis of automobiles. Computers & Structures 1997;64(5/6):881–91.

[3] Bathe KJ, Dvorkin EN. A formulation of general shell elements—the use of mixed interpolation of tensorial components. Int J Numerical Methods in Engineering 1986;22:697–722.

[4] Chapelle D, Bathe KJ. Fundamental considerations for the finite element analysis of shell structures. Computers & Structures 1998;66(1):19–36.

[5] Bathe KJ, Iosilevich A, Chapelle D. An evaluation of the MITC shell elements, Computers & Structures, submitted.

[6] Bathe KJ, Iosilevich A, Chapelle D. An inf-sup test for shell finite elements, Computers & Structures, submitted.

[7] Chapelle D, Bathe KJ. The mathematical shell model underlying general shell elements, Int J for Numerical Methods in Engineering, submitted.

[8] Bathe KJ, Chaudhary AB, Dvorkin EN, Kojić M. On the solution of nonlinear finite element equations. In: Damjanić F, et al., editors. Proceedings, International Conference on Computer-Aided Analysis and Design of Concrete Structures I. Swansea, UK: Pineridge Press, 1984. p. 289–99.

[9] Kojić M, Bathe KJ. The 'Effective-Stress-Function' algorithm for thermo-elasto-plasticity and creep. International J for Numerical Methods in Engineering 1987;24:1509–32.

[10] Eterovic AL, Bathe KJ. A hyperelastic-based large strain elasto-plastic constitutive formulation with combined isotropic-kinematic hardening using the logarithmic stress and strain measures. International J for Numerical Methods in Engineering 1990;30:1099–114.

[11] Gabriel G, Bathe KJ. Some computational issues in large strain elasto-plastic analysis. Computers & Structures 1995;56(2/3):249–67.

[12] Eterovic AL, Bathe KJ. On the treatment of inequality constraints arising from contact conditions in finite element analysis. Computers & Structures 1991;40:203–9.

[13] Bathe KJ, Bouzinov PA. On the constraint function method for contact problems. Computers & Structures 1997;64(5/6):1069–85.

[14] Graph theory and sparse matrix computation. In: George A, Gilbert JR, Liu JWH, editors. Institute for mathematics and its applications Vol. 56. Springer-Verlag, 1993.