



**HAL**  
open science

## Combined implicit/explicit algorithms for crashworthiness analysis

Ludovic Noels, Laurent Stainier, Jean-Philippe Ponthot

► **To cite this version:**

Ludovic Noels, Laurent Stainier, Jean-Philippe Ponthot. Combined implicit/explicit algorithms for crashworthiness analysis. *International Journal of Impact Engineering*, 2004, 30 (8-9), pp.1161-1177. 10.1016/j.ijimpeng.2004.03.004 . hal-01005861

**HAL Id: hal-01005861**

**<https://hal.science/hal-01005861>**

Submitted on 10 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Combined implicit/explicit algorithms for crashworthiness analysis

L. Noels<sup>1</sup>, L. Stainier<sup>2</sup>, J.-P. Ponthot\*

University of Liège, LTAS-Milieux Continus & Thermomecanique, Chemin des Chevreuils 1, B-4000 Liège, Belgium

In order to simulate an industrial process, an explicit method, which is conditionally stable, is the most adapted while the non-linearities evolve rapidly (impact phase, stamping process, etc.). But when the dynamics becomes quasi-linear (post-impact analysis, springback simulation, etc.), an implicit method, which is iterative, presents the advantage of unconditional stability. The optimal solution is then to have both implicit and explicit methods readily available in the same code and to be able to switch automatically from one to the other. Criteria that decide to switch from one method to another, depending on the current dynamics, have been developed. Implicit restarting conditions are also proposed that annihilate numerical oscillations resulting from an explicit calculation.

*Keywords:* Implicit; Explicit; Dynamics; Finite element; Large strains

## 1. Introduction

The choice of a time integration algorithm is an essential criterion to ensure efficiency and robustness of the numerical simulations. Difficulty in this choice resides in being able to combine robustness, accuracy and stability of the algorithm. Implicit algorithms require iterative solutions for each time increment (time step), contrarily to explicit ones. But, for stability reasons, explicit methods use smaller time steps than implicit ones. Explicit methods, avoiding iterations and

---

\*Corresponding author. Tel.: +32-4-366-9310; fax: +32-4-366-9141.

*E-mail address:* jp.ponthot@ulg.ac.be (J.-P. Ponthot).

<sup>1</sup> Research Fellow at the Belgian National Fund for Scientific Research (FNRS).

<sup>2</sup> Research Associate at the Belgian National Fund for Scientific Research (FNRS).

## Nomenclature

|                  |   |
|------------------|---|
| $e_{\text{int}}$ | integration error   |
| $E$              | Young modulus   |
| $F^{\text{ext}}$ | external force vector   |
| $F^{\text{int}}$ | internal force vector   |
| $h$              | hardening parameter   |
| $M$              | consistent mass matrix  |
| $N$              | number of nodes   |
| $PRCU$           | tolerance on the integration error  |
| $r^*$            | ratio between the CPU time of an implicit step and the CPU time of an explicit step |
| $r^{**}$         | number of explicit steps balanced   |
| $R$              | residual vector   |
| $t$              | time  |
| $x$              | nodal positions vector  |

### *Greek symbols*

|                                 |   |
|---------------------------------|---|
| $\Delta t$                      | time step size  |
| $\Omega_s$                      | explicit non-dimensional pulsation stability limit          |
| $\alpha_M$                      | extrapolation parameter of the inertial forces              |
| $\alpha_F$                      | extrapolation parameter of the internal and external forces |
| $\beta$                         | first Newmark parameter                                     |
| $\gamma$                        | second Newmark parameter                                    |
| $\gamma_s$                      | security on the explicit time step                          |
| $\varepsilon$                   | reference integration error                                 |
| $\bar{\varepsilon}^{\text{pl}}$ | equivalent plastic strain                                   |
| $\eta$                          | parameter for the time step size evaluation                 |
| $\mu$                           | security of the automatic shift criterion                   |
| $\nu$                           | Poisson's coefficient                                       |
| $\rho$                          | density   |
| $\rho_b$                        | spectral radius at the bifurcation limit                    |
| $\sigma_0$                      | initial yield stress  |
| $\sigma_v$                      | von Mises stress  |
| $\sigma_\infty$                 | saturated stress  |
| $\omega_{\text{max}}$           | maximal pulsation of the structure                          |

### *Subscript*

|            |   |
|------------|---|
| 0          | initial value   |
| $n$        | value evaluated at configuration $n$                        |
| $n, n + 1$ | value that refers to the two configurations $n$ and $n + 1$ |
| new        | new computed value  |
| cur        | current value   |
| expl       | value that refers to the explicit method                    |
| impl       | value that refers to the implicit method                    |

convergence problems, are therefore generally used for highly non-linear problems with many degrees of freedom, for which iterations are very expensive and convergence problems are frequent [1]. On the other hand, for slower dynamics problems with less non-linearities, implicit algorithms allow use of a larger time step size, resulting in more numerical stability and accuracy [1–3]. However, in general, an actual impact simulation has some time intervals governed by highly non-linear dynamics (during the impact itself) and others governed by slower non-linear dynamics (before and after the impact). Thus, one can take advantage of this situation by integrating over some time intervals with an implicit method and other time intervals with an explicit one.

Few works have been developed with this latter combination. Jung and Yang [4] have simulated a stamping simulation that begins with an implicit scheme and shifts to an explicit one when a problem of convergence appears. No return to implicit scheme is actually planned. Another method, developed by Finn et al. [5] and by Narkeeran and Lovell [6], simulates stamping (as a fast dynamics problem) with an explicit scheme, and the springback phase (slow dynamics) is subsequently analyzed with an implicit one. The time of transition is fixed by the user and initial conditions for the implicit phase, such as velocities and accelerations, are set to zero. Automatic criteria that decide to shift from one family to another have been developed by the authors in Ref. [7] for impact problems and were applied to sheet-metal forming problems in Ref. [8]. They depend on an integration error that allows determination of the implicit time step size. They also depend on a ratio between the computational time (or CPU time) needed to solve an implicit time step and the CPU time needed to solve an explicit time step. Initial conditions, when shifting from explicit to implicit scheme occurs, are also defined to avoid lack of stability and convergence.

In the present paper, we focus on the practical application of the combined method. We will present in Section 2 the principles of the implicit and explicit algorithms. In Section 3, after having recalled the theory of shifting, developed in Ref. [7], we will present a practical way to implement this combined method in a Finite-Element code. Finally, in Section 4, we will analyze in details the impact dynamics of Taylor bar and of a blade casing interaction. This last simulation requires to model interactions between deformable bodies. We will prove that our combined method is able to simulate with accuracy these complex dynamic phenomena. A quantitative discussion on the parameters in use will also be provided to improve the practical use of such a method.

## 2. Numerical integration of transient problems

Finite element method (FEM) semi-discretization of the equations of motion of a non-linear structure leads to the following coupled set of second-order non-linear differential equations (see Refs. [9–11]):

$$R = M\ddot{x} + F^{\text{int}}(x, \dot{x}) - F^{\text{ext}}(x, \dot{x}) = 0, \quad (1)$$

where  $R$  is the residual vector,  $x$  the vector of the nodal positions at current time,  $\dot{x}$  the vector of nodal velocities,  $\ddot{x}$  the vector of nodal accelerations.  $M$  is the mass matrix,  $F^{\text{int}}$  the vector of internal forces resulting from body deformations and  $F^{\text{ext}}$  the vector of external forces. Both vectors are non-linear in  $x$  and in  $\dot{x}$  due to the coupled phenomena of contact, plastic deformations and geometrical non-linearities.

## 2.1. Implicit schemes

The most general scheme for implicit integration of Eq. (1) is a generalized trapezoidal scheme [9,10,12] where updating of positions and velocities is based on ‘‘averaged’’ accelerations stemming from associated values between  $t_n$  and  $t_{n+1}$ . It reads for instance

$$\dot{x}_{n+1} = \dot{x}_n + (1 - \gamma)\Delta t\ddot{x}_n + \gamma\Delta t\ddot{x}_{n+1}, \quad (2)$$

$$x_{n+1} = x_n + \Delta t\dot{x}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2\ddot{x}_n + \beta\Delta t^2\ddot{x}_{n+1}. \quad (3)$$

The discretized motion Eq. (1) can be rewritten under the form proposed by Chung and Hulbert [12]:

$$\begin{aligned} R_{n,n+1} = & \frac{1 - \alpha_M}{1 - \alpha_F} M\ddot{x}_{n+1} + \frac{\alpha_M}{1 - \alpha_F} M\ddot{x}_n + (F_{n+1}^{\text{int}} - F_{n+1}^{\text{ext}}) \\ & + \frac{\alpha_F}{1 - \alpha_F} (F_n^{\text{int}} - F_n^{\text{ext}}) = 0, \end{aligned} \quad (4)$$

where  $R_{n,n+1}$  is the residual vector of time step  $n$  to  $n + 1$ . Using Eqs. (2)–(4) and a Newton–Raphson technique, the iterative solution of the problem can be computed.

The implicit time step size control is the one proposed by G eradin [13], extended to highly non-linear problems by Noels et al. [14,15]. This scheme continuously adapts time step size to evolution of the physical modes and keeps time step size constant during long time intervals. To estimate the current time step size, an integration error is computed. The integration error  $e_{\text{int}}$  (see Refs. [13–15] for details) is deduced from truncated terms of Eqs. (2) and (3). This error is of third order:  $O(\frac{1}{6}\Delta t^3\dot{\ddot{x}}) \simeq O(\frac{1}{6}\Delta t^2\Delta\ddot{x})$ . To have a problem independent error, it is made non-dimensional, using  $x_0$  (the initial position vector) and a reference error  $\varepsilon$  [13–15]. To take into account the rotation, the integration error is then rewritten by taking the variation of the nodal acceleration modulus ( $N$  is the number of nodes) [7]. Finally, it leads to

$$e_{\text{int}} = \frac{\Delta t^2}{6} \frac{\sum_{i=1}^N \Delta \|\ddot{x}_i\|}{\varepsilon \|x_0\|}. \quad (5)$$

Time step size is deduced from the integration error defined in Eq. (5) and from a tolerance PRCU ( $10^{-3}$  is a typical value) fixed by the user. The relation to be verified is

$$e_{\text{int}} < \text{PRCU}. \quad (6)$$

The new time step size  $\Delta t_{\text{new}}$  to reach a reference integration error (half of the tolerance PRCU) is deduced from the current time step size ( $\Delta t_{\text{cur}}$ ) and from the current integration error ( $e_{\text{int,cur}}$ ), using the following relation developed by G eradin [13]:

$$\left(\frac{\Delta t_{\text{new}}}{\Delta t_{\text{cur}}}\right)^\eta = \frac{\text{PRCU}}{2e_{\text{int,cur}}}. \quad (7)$$

Let us determine the parameter  $\eta$ . If we analyze the reference error [13–15]  $\varepsilon$ , it leads to  $\lim_{\Omega \rightarrow 0} \varepsilon = \mathcal{O}(\Omega^3)$  and  $\lim_{\Omega \rightarrow \infty} \varepsilon = \mathcal{O}(\Omega^2)$ . Therefore, one gets  $\eta \in [2,3]$ . Practically, we take  $\eta = 2.5$ . Now, let us explain briefly the methodology to determine the time step size in the non-linear dynamics (more details can be found in Refs. [14,15]). Let us assume we compute the time step from time  $t_n$  to time  $t_{n+1}$ . From accelerations at these two configurations, the integration error  $e_{\text{int,cur}}$  is

computed from Eq. (5). This error is compared to the tolerance PRCU. If it is greater than 1.5 PRCU, the step is rejected and the time step is reduced using Eq. (7). If it is between 1.5 PRCU and PRCU, the step is accepted but following time step will be computed with a smaller step size deduced from Eq. (7). If the error is between PRCU and PRCU/2, the time step must be reduced, but since the dynamics is non-linear, to avoid to change too often the time step size, it is decreased only if this happened three successive times. On the other hand, if the error is lower than PRCU/16, five successive times, the time step size is increased.

## 2.2. Explicit scheme

Chung and Hulbert [16] have extended their implicit scheme to an explicit one, taking  $\alpha_F = 1$  in Eq. (4). Its principal advantage is its numerical dissipation property. Time integration is then

$$\ddot{x}_{n+1} = \frac{M^{-1}(F_n^{\text{ext}} - F_n^{\text{int}}) - \alpha_M \ddot{x}_n}{1 - \alpha_M}, \quad (8)$$

$$\dot{x}_{n+1} = \dot{x}_n + \Delta t[(1 - \gamma)\ddot{x}_n + \gamma\ddot{x}_{n+1}], \quad (9)$$

$$x_{n+1} = x_n + \Delta t\dot{x}_n + \Delta t^2 \left[ \left(\frac{1}{2} - \beta\right)\ddot{x}_n + \beta\ddot{x}_{n+1} \right]. \quad (10)$$

This scheme is conditionally stable and time step size is limited, depending on maximal model frequency  $\omega_{\max}$ , but also depending on spectral radius ( $\rho_b$ ):

$$\Delta t = \gamma_s \Delta t_{\text{crit}} = \gamma_s \frac{\Omega_s(\rho_b)}{\omega_{\max}} \quad (11)$$

with [16]:

$$\Omega_s(\rho_b) = \sqrt{\frac{12(1 + \rho_b)^3(2 - \rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}}. \quad (12)$$

In Eq. (11),  $\gamma_s$  is a safety factor ( $<1$ ) that accounts for the destabilizing effects of non-linearities.

## 3. Shifts from an algorithm family to another

This section exposes the methodology of the shifting methods. More details can be found in Ref. [7].

### 3.1. Shift from an implicit algorithm to an explicit algorithm

First the ratio  $r^*$  between the CPU needed for an implicit time step computation and the CPU needed for an explicit time step computation is evaluated. In this paper, this ratio is updated for each step, in order to be able to shift from a method to another in non-linear simulations. Shift to explicit method occurs if:

$$\mu \Delta t_{\text{impl}} < r^* \Delta t_{\text{expl}}, \quad (13)$$

where  $\Delta t_{\text{expl}}$  is evaluated from Eq. (11). This relation has the following meaning: since  $r^*$  represents the number of explicit steps that can advantageously be replaced by an implicit step, if the size of the implicit step becomes smaller than  $r^*$  multiplied by the size of an explicit step, an explicit method is then cheaper than an implicit one. Then the implicit method shifts to an explicit one. The factor  $\mu$  is taken larger than unity (typical values discussed in Section 4) to avoid shifting from a method to another too frequently. This methodology takes into account the number of degrees of freedom, the algorithm current efficiency, the residual tolerance required and the evolution of non-linearities.

### 3.2. Shift from an explicit algorithm to an implicit algorithm

When the method in use is an implicit one, the explicit time step size can always be easily computed from Eq. (11). When the current method is explicit, the implicit time step size, which correctly integrates the problem, is not directly accessible. Using developments of Section 2.1, nodal acceleration variations can provide us with this implicit time step size. Using Eq. (7), acceleration variation is proportional to  $\Delta t^\eta$ . Inverting Eq. (5) the implicit time step size becomes (with  $N$  the number of nodes):

$$\Delta t_{\text{impl}} = \left[ 6 \frac{\frac{\text{PRCU}}{2} \varepsilon \|x_0\| (\Delta t_{\text{expl}})^{\eta-2}}{\sum_{i=1}^N \Delta \|\ddot{x}_i\|} \right]^{1/\eta}. \quad (14)$$

This expression is the relation between the accelerations and the time step size. Since  $\sum_{i=1}^N \Delta \|\ddot{x}_i\|$  is the accelerations difference that corresponds to a time step equal to  $\Delta t_{\text{expl}}$ , using Eqs. (5) and (7), one can deduce the equivalent implicit step  $\Delta t_{\text{impl}}$  that can lead to an error equal to  $\text{PRCU}/2$ .

Therefore, the explicit to implicit shift criterion is similar to Eq. (13). It yields

$$\Delta t_{\text{impl}} > \mu r^* \Delta t_{\text{expl}} \quad (15)$$

with  $\Delta t_{\text{expl}}$  the current explicit time step size. This relation has same meaning that Eq. (13): since  $r^*$  represents the number of explicit steps that can be advantageously replaced by an implicit step, if the size of the implicit step becomes larger than  $r^*$  multiplied by the size of an explicit step; therefore, an implicit method is cheaper than an explicit one.

### 3.3. Initial conditions when shifting to an implicit scheme

First, numerical oscillations resulting from the explicit scheme are annihilated thanks to the numerical dissipation property of the generalized- $\alpha$  explicit scheme. Indeed, when Eq. (15) is satisfied, thus resulting in the choice to switch to implicit, at step number  $n$  (at time  $t_n$ ),  $r^*$  explicit steps occur with a spectral radius  $\rho_b$  (Section 2.2) set equal to zero ( $\rho_b$  is a user defined parameter). Thus, numerical oscillations have been greatly reduced at time  $t_{n+r^*}$  (Fig. 1).

The second step in the algorithm is to determine a balanced configuration at time  $t_{n+r^*+r^{**}}$ . Therefore, we act in two stages. First an explicit solution using  $r^{**}$  ( $r^{**}$  will be defined in next paragraph) explicit steps is computed. This solution results in  $x_{n+r^*+r^{**}}^{\text{expl}}$ ,  $\dot{x}_{n+r^*+r^{**}}^{\text{expl}}$  and in  $\ddot{x}_{n+r^*+r^{**}}^{\text{expl}}$ , which in turn are used as predictor values for an implicit solution in one time step between time  $t_{n+r^*}$  (where numerical oscillations have been reduced) and time  $t_{n+r^*+r^{**}}$ . This procedure proved to be very effective in order to restart an implicit solution based on explicit

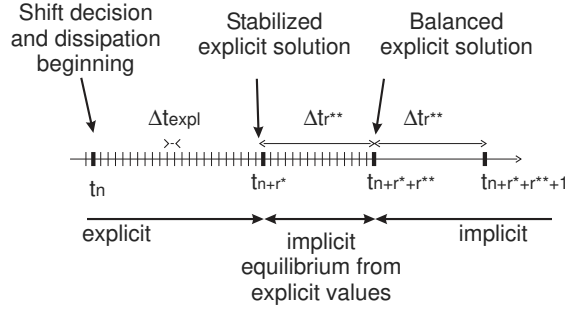


Fig. 1. Transition scheme from an explicit scheme to an implicit one.

unbalanced solution. Therefore, a balanced step of size equal to the implicit time step size is reached leading to solutions at time  $t_{n+r^*+r^{**}}$  from solutions (nodal and Gauss points values) at time  $t_{n+r^*}$ . The methodology is illustrated in Fig. 1. This balanced solution is reached considering an implicit time step size equal to  $\Delta t_{r^{**}} = r^{**} \Delta t_{\text{expl}}$ . In general, the iterative process necessary to reach this equilibrium quickly converges and this allows the implicit method to begin with a balanced solution at time  $t_{n+r^*+r^{**}}$ . Anyway  $r^{**}$  must be defined. It is always lower or equal to  $\mu r^*$ . It is equal to  $\mu r^*$  or limited to an user defined parameter (e.g. 100).

### 3.4. Numerical implementation in a Finite-Element code

The practical implementation of the combined method is based on the full compatibility of the explicit formulation with the implicit formulation in the same Finite-Element code. This compatibility results from the same element (volume, contact, etc.) formulation, and from the same material law (elasto-plastic hypoelastic, frictional contact law, etc.) models for both methods. This avoids some inaccurate procedure of transition between an explicit and an implicit finite element code.

Once this compatibility is guaranteed, we can integrate the two resolution methods in a time integration algorithm. We will explain this integration that is represented in Fig. 2. Let us assume that all the values (time, nodal values and values at the Gauss points) are known at time  $t_n$ .

Let us first study the case where the integration method is chosen to be explicit. Therefore, a direct resolution of Eqs. (8)–(10) determines the complete solution at time  $t_{n+1}$ . Using the CPU needed for this step, the ratio  $r^*$  can be updated. Moreover, using accelerations at time  $t_n$  and  $t_{n+1}$ , the new explicit time step can be computed from Eq. (11) and the equivalent implicit time step is evaluated from Eq. (14). Then if the balance criterion defined in Section 3.2 is not verified and if we are not in an explicit steps balancing situation (i.e. *shift* = *false*), the method remains explicit. Then the new integration time is evaluated and the step number is incremented.

If the balance criterion defined in Section 3.2 is verified and if we are not in an explicit steps balancing situation (i.e. *shift* = *false*), the process to determinate the initial conditions described in Section 3.3 begins. Firstly, the shift indicator is set to *true*, a step counter  $i$  is initialized to zero, and the spectral radius is set to zero. The explicit integration continues during  $r^*$  steps, incrementing  $i$  at each step. When  $i = r^*$ , the solution obtained (time, nodal values and values at Gauss points) is saved (on disk), before the explicit integration continues during  $r^{**}$  steps. When  $i = r^* + r^{**}$  the balance step can be finalized. First, the values previously saved on disk are



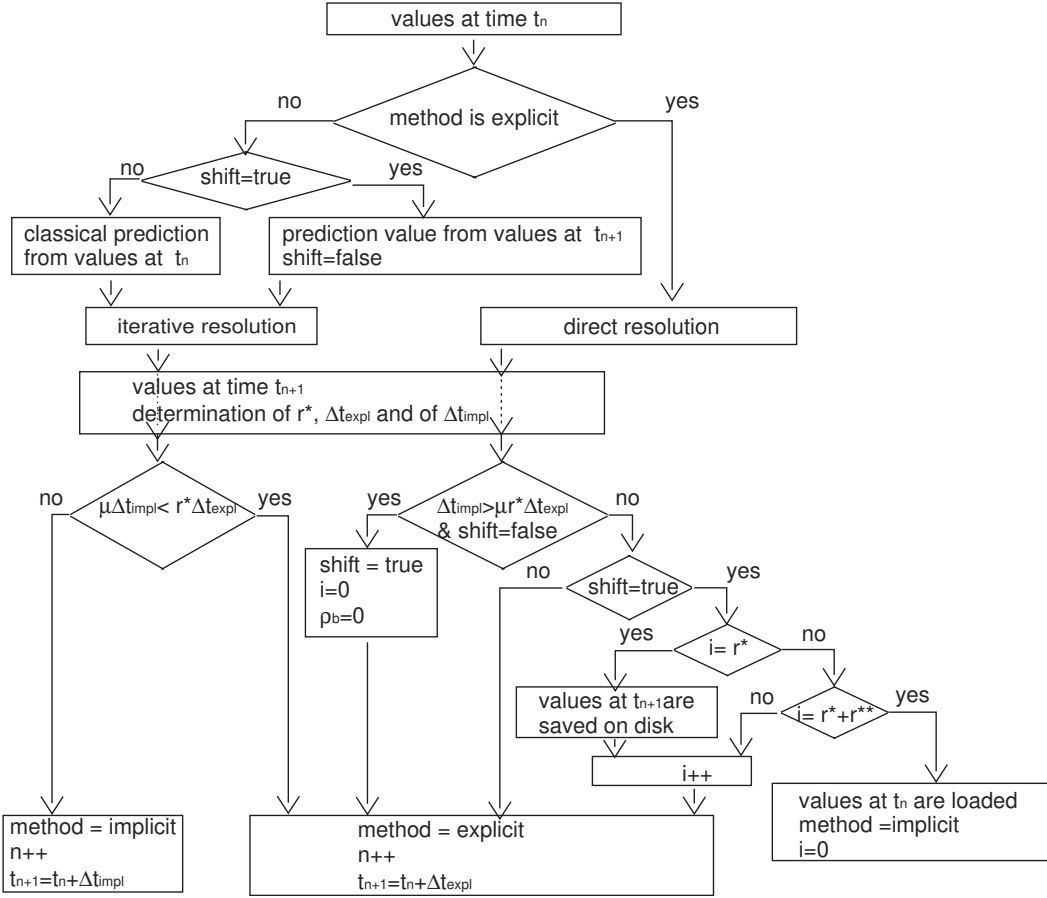


Fig. 2. Implementation of the combined method.

reloaded and stored as the solution at time  $t_n$ . The method is set to *implicit*, but the time step number  $n$  is not incremented. Then, during the step resolution, the method used is the implicit one, but the prediction values are the previous explicit solution reached at time  $t_{n+1}$  ( $x_{n+1}^{\text{expl}}$ ,  $\dot{x}_{n+1}^{\text{expl}}$  and  $\ddot{x}_{n+1}^{\text{expl}}$ ). A classical iteration procedure corrects these values, leading to a balanced solution with a time step equal to  $\Delta t = r^{**} \Delta t_{\text{expl}}$ .

Then the method remains implicit and the step number is incremented. The next implicit steps use the classical prediction formula, using the solution at time  $t_n$  leading to the solution at time  $t_{n+1}$ . The new implicit time step can then be computed by Eq. (7) and the critical explicit time step by Eq. (11). Finally, if Eq. (13) is verified, the method shifts to an explicit time integration algorithm.

#### 4. Numerical examples

In the numerical examples, the finite elements use selective reduced integration, to avoid volumetric locking resulting from the incompressibility condition on plastic deformations. There

Table 1  
Geometrical and material values of the Taylor bar

| Property                       | Value                              |
|--------------------------------|------------------------------------|
| External diameter              | $d_e = 6.4$ mm                     |
| Length                         | $l = 32.4$ mm                      |
| Density                        | $\rho = 8930$ kg/m <sup>3</sup>    |
| Young's modulus                | $E = 117E9$ N/m <sup>2</sup>       |
| Poisson's ratio                | $\nu = 0.35$                       |
| Yield stress                   | $\sigma_0 = 400$ N/mm <sup>2</sup> |
| Hardening parameter            | $h = 100$ N/mm <sup>2</sup>        |
| Initial distance from the wall | $x_0 = 1$ mm                       |
| Initial velocity               | $\dot{x}_0 = 227$ m/s              |

are eight deviatoric Gauss points and one volumic Gauss point. When studying the first problem, we will discuss on the influence of user parameters. The second problem will show how to take into account the oscillations successive to the impact.

#### 4.1. Taylor's bar impact

A cylindrical bar (properties in Table 1), discretized with 1080 elements (27 in each section times 40 along the axis), is at an initial distance  $x_0$  from a rigid plane. It has an initial velocity  $\dot{x}_0$  directed towards the plane. Numerical parameters used for the time integration scheme are for the implicit scheme (Section 2.1):  $\alpha_M = -0.97$ ,  $\alpha_F = 0.01$ ,  $\beta = 0.9801$ ,  $\gamma = 1.48$ . The tolerance on the Newton–Raphson residual is  $10^{-6}$ . For the explicit scheme, the parameters are (Section 2.2):  $\rho_b = 0.2$  and  $\gamma_s = 0.9$ .

Let us first compare a full implicit solution with an integration error tolerance  $\text{PRCU} = 10^{-3}$  to a full explicit solution and to a combined solution. For the combined solution, we first choose a security  $\mu = 1.5$  (Eqs. (13) and (15)). Discussion over this value will be achieved later. This problem is well suited to evaluate the performances of the algorithm combination method. Indeed, the simulation could be decomposed into three intervals of different dynamics properties: first a rigid translation motion before impact has slow dynamics properties, second an impact with large plastic straining (about 300%) has fast dynamics properties, and third a rebound after the shock wave propagation has slow dynamics properties. The method is expected to shift from implicit to explicit when impact occurs and to come back to an implicit scheme after the plastic deformations have occurred.

Central point displacements, of the respectively bottom section and top section, obtained with the three methods (full implicit, full explicit and combined implicit/explicit) are compared in Fig. 3(a) and (b). Zones treated with the explicit scheme when the combined implicit/explicit algorithm is used are also indicated. Impact occurs at time  $t = 4.4$   $\mu\text{s}$ . At this time, the central node at the bottom face enters into contact, and the external force change the sign of the velocity (Fig. 4(a)). This change of sign leads the node to be separated from the wall. But, due to the inertial forces, the velocity changes of sign again and the node reenters into contact. Moreover,

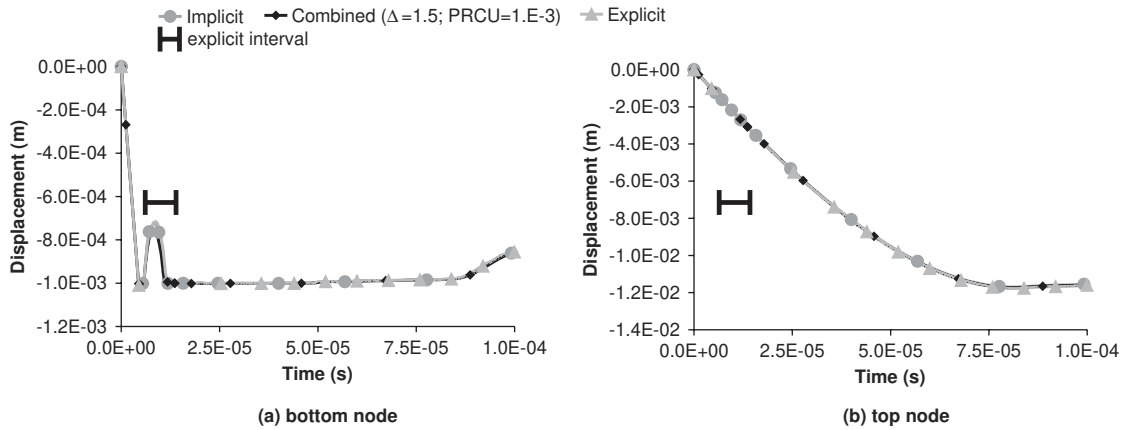


Fig. 3. Central point displacement for Taylor bar impact: (a) bottom node, (b) top node.

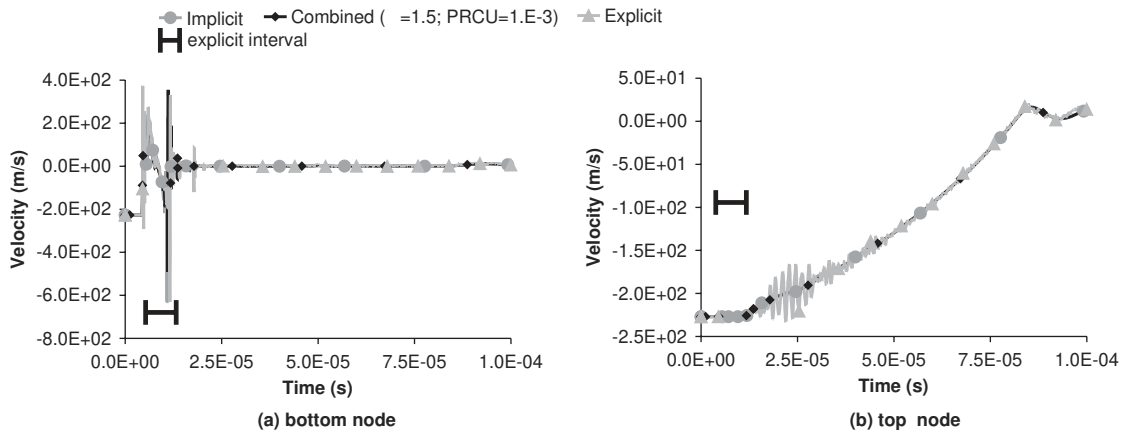


Fig. 4. Central point velocity for Taylor bar impact: (a) bottom node, (b) top node.

the impact creates a wave of velocity<sup>3</sup> equal to 2857 m/s, which reaches the top side of the cylinder at time<sup>4</sup>  $t \approx 4.4 \mu\text{s} + (0.0324/2857)\text{s} \approx 15.7 \mu\text{s}$ . At this time, the velocity of the top central node becomes to decrease to zero (Fig. 4(b)). It appear from this result that the bottom side is subjected to impact, leading to overshoot in the velocity, but that the top side is subject to wave reflection and its values evolve smoothly. After about 80  $\mu\text{s}$ , the bar rebounds from the rigid surface. Fig. 5 illustrates the deformations and the equivalent plastic strains for each simulation. All the solutions are identical to about 6%.

The combined method reacts as expected. It computes the rigid translation motion and rebound with an implicit scheme and the plastic impact with an explicit scheme. Fig. 6(a) illustrates the

<sup>3</sup>The sound speed (longitudinal wave) in a solid is equal to  $\sqrt{E(1+\nu)(1-2\nu)/\rho(1-\nu)}$ .

<sup>4</sup>Since the length of the beam change, this is an overestimate.

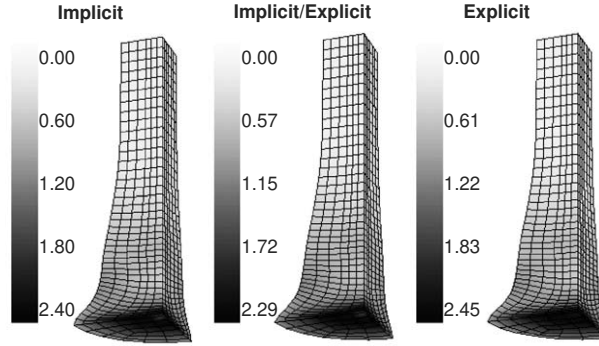


Fig. 5. Equivalent plastic strain for Taylor bar impact.

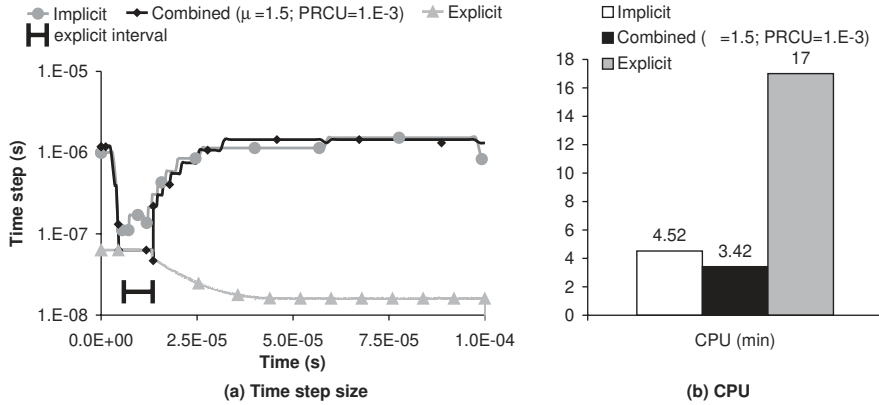


Fig. 6. Method comparison for Taylor bar impact: (a) time step size evolution, (b) CPU.

evolution of the time step size. For the full explicit solution, the time step is reduced when the cylinder impacts and then it is increased while the dynamics becomes smoother. For the explicit simulation, since the elements becomes smaller, the critical time step size follows, and the step is reduced after the impact. The combined solution take advantage of the explicit method when the implicit time step is decreased, and take advantage of the implicit method when the dynamics becomes smoother. This results into a reduction of the CPU time (Fig. 6(b)) of about 30% from the implicit solution and of about 75% from the explicit solution.

Let us now discuss the influence of the security parameter  $\mu$ . If the security  $\mu$  is decreased, both Eqs. (13) and (15) will be verified sooner when the combined method is used. Let us assume that  $\mu = 1$  and that the algorithm shifts from an implicit method to an the explicit one. At this point Eq. (13) is verified. But if  $\mu = 1$ , Eq. (15) will be verified simultaneously, and the method will shift to an implicit one. We will observe this shift until the implicit scheme is able to converge without verifying Eq. (13). We compare in Fig. 7 the influence of  $\mu$  on the combined method. It appears (Fig. 7(a)) that, as expected, for  $\mu = 1$ , the combined method shift from a method to another eight times before the implicit method could be continuously used. This result in a increase of the CPU time when compared to the value  $\mu = 1.5$  (Fig. 7(b)). If we use  $\mu = 2$ , the combined method shift a

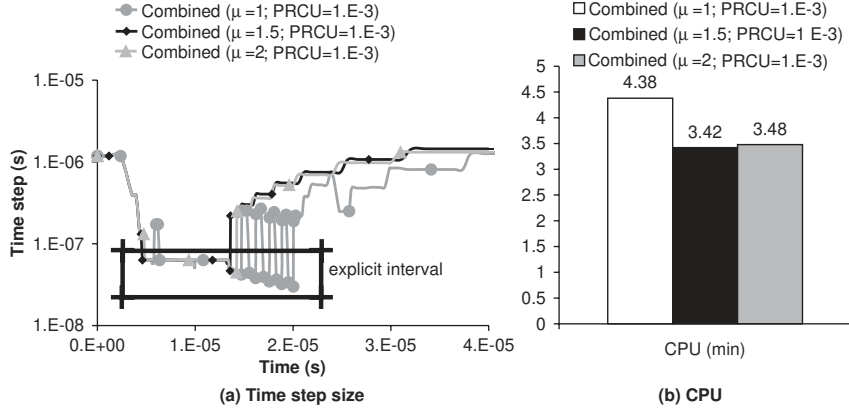


Fig. 7. Security influence for Taylor bar impact: (a) time step size evolution, (b) CPU.

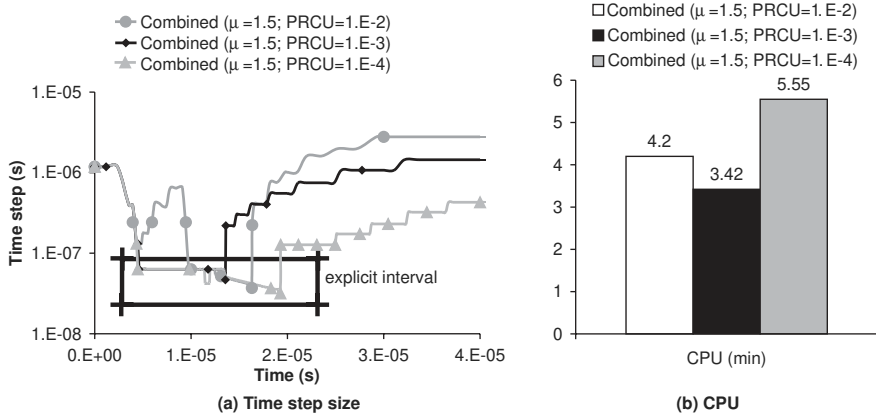


Fig. 8. Integration error tolerance influence for Taylor bar impact: (a) time step size evolution, (b) CPU.

little later to the implicit algorithm (Fig. 7(a)) than when  $\mu = 1$ , leading to a small increase in CPU time (Fig. 7(b)).

Now, let us study the influence of the integration tolerance PRCU on the combined method. Fig. 8 illustrates the results obtained for three values ( $10^{-2}$ ,  $10^{-3}$  and  $10^{-4}$ ) of this tolerance. When the tolerance is increased from  $10^{-3}$  to  $10^{-2}$ , the implicit step used is larger and the combined method shifts to an explicit scheme later (Fig. 8(a)). Surprisingly, even if the predicted implicit step computed by Eq. (14) is larger since PRCU is larger, the combined method return to an implicit method later. It results from the fact that the ratio  $r^*$  is also larger since the larger the PRCU, the larger the CPU time needed to compute an implicit step. Eq. (15) is verified later. Therefore, the total CPU time is larger for a less accurate solution as we can see in Fig. 8(b). If the PRCU parameter is taken smaller (i.e.  $10^{-4}$ ), the predicted implicit step obtained by Eq. (15) is much smaller than for  $\text{PRCU} = 10^{-3}$  and the combined method shifts later to an implicit step (Fig. 8(a)). This results to a more expensive solution (Fig. 8(b)), but since the integration error is always smaller, this solution is also more accurate.

#### 4.2. Blade-casing interaction

We consider the rotor blade illustrated in Fig. 9(a). It has a twisted rectangular cross section and is placed in a casing (properties reported in Table 2). The material properties of the blade are identical to the material properties of the casing. The plastic isotropic hardening law

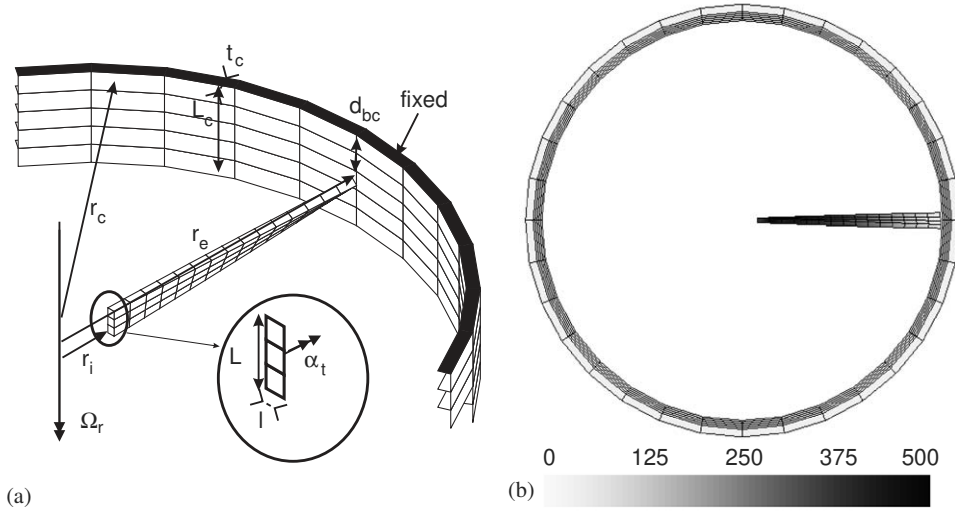


Fig. 9. Description of the blade-casing interaction: (a) geometry, (b) initial von Mises stress (MPa).

Table 2  
Geometrical and material values of the blade-casing interaction

| Property   | Value                                      |
|--|--|
| Casing radius  | $r_c = 1.305$ m                            |
| Casing thickness                                     | $t_c = 5$ mm                               |
| Casing length  | $L_c = 0.4$ m                              |
| External blade radius                                | $r_e = 1.3$ m                              |
| Internal blade radius                                | $r_i = 0.1$ m                              |
| Length of section                                    | $L = 0.11$ m                               |
| Width of section                                     | $l = 0.032$ m                              |
| Twisting angle                                       | $\alpha_t = 45^\circ$                      |
| Distance between the blade and the top of the casing | $d_{bc} = 0.145$ m                         |
| Density  | $\rho = 4450$ kg/m <sup>3</sup>            |
| Young's modulus                                      | $E = 110e^9$ N/m <sup>2</sup>              |
| Poisson's ratio                                      | $\nu = 0.31$                               |
| Initial yield stress                                 | $\sigma_0 = 1e^9$ N/m <sup>2</sup>         |
| Saturated yield stress                               | $\sigma_\infty = 1.3e^9$ N/mm <sup>2</sup> |
| Exponential hardening parameter                      | $h_e = 100$                                |
| Linear hardening parameter                           | $h = 300$ N/mm <sup>2</sup>                |
| Rotational velocity                                  | $\Omega_r = 3333.3$ rpm.                   |

( $\sigma_v$  depending on effective plastic strain  $\bar{\epsilon}^{pl}$ ) is  $\sigma_v = \sigma_0 + (\sigma_\infty - \sigma_0)(1 - e^{-h_c \bar{\epsilon}^{pl}}) + h \bar{\epsilon}^{pl}$ . The blade is discretized with 45 elements and the casing is discretized with 160 elements. The rotational velocity of the blade is constant ( $\Omega_r$ ). At time  $t = 0$  s, the initial configuration of the rotor is computed for a rotation velocity of 4775 rpm. This initial configuration is computed with a Newton–Raphson scheme where the external forces are the analytical inertial forces computed from the nodal positions and from the imposed rotation velocity. The von Mises stresses resulting from this uniform rotation velocity are illustrated at Fig. 9(b). To simulate a load unbalance phenomenon, the center of rotation of the blade is moved during the first rotation so that the blade comes into contact with the casing and is subsequently deformed. The displacement of the center of rotation is the following. Its initial position is coincident with the center of the casing and there is no contact. During the first half revolution, the rotation center is moved 7.5 cm to the opposite direction of the initial blade position, so that contact interactions between the blade and the casing are generated. These interactions lead to irreversible plastic deformation of the blade. During the second half of the first revolution, the center of rotation is brought back to its initial position. The frictional contact simulation uses the penalty method with a normal penalty of  $10^9$  and a tangent penalty of  $10^8$ . The Coulomb friction coefficient is equal to 0.2. Numerical parameters used for the time integration scheme are for the implicit scheme (Section 2.1):  $\alpha_M = -0.97$ ,  $\alpha_F = 0.01$ ,  $\beta = 0.9801$ ,  $\gamma = 1.48$ . The tolerance on the Newton–Raphson residual is  $10^{-9}$  and the tolerance on the integration error (PRCU) is  $10^{-3}$ . For the explicit scheme, the parameter are (Section 2.2):  $\rho_b = 0.2$  and  $\gamma_s = 0.9$ . Moreover, for the combined implicit/explicit method, parameter  $\mu$  of Eqs. (13) and (15) is taken equal to 1.5.

The resulting force on the extremity of the blade, obtained with the three methods (full implicit, full explicit and combined implicit/explicit), is illustrated in Fig. 10(a). Zones treated with explicit scheme when the combined implicit/explicit algorithm is used are also indicated. We see that the impact occurs after 0.4 revolution. At this time the combined method shifts to an explicit algorithm. The impact results in the apparition of plastic deformations. Fig. 10(b) illustrates the time evolution of the energy plastically dissipated. It appears that half of this energy is dissipated between 0.4 revolution and the first half revolution. This time interval corresponds to the explicit resolution of the combined method. Fig. 11 illustrates the equivalent plastic strain obtained after a

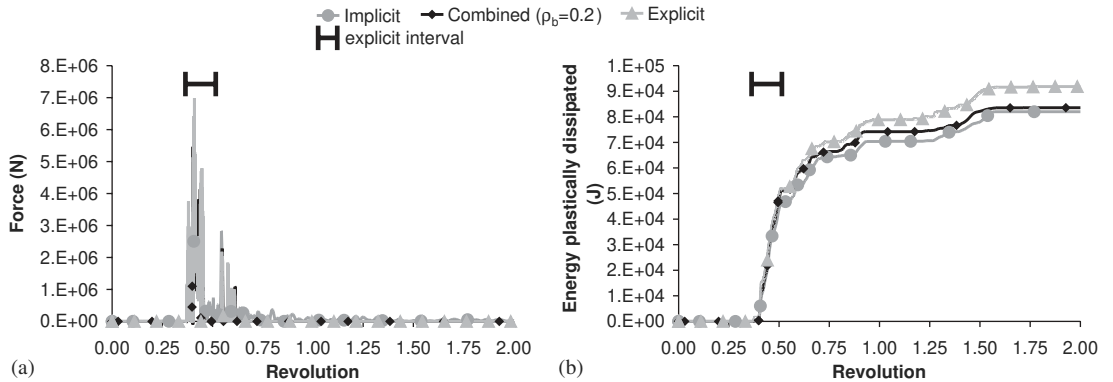


Fig. 10. Time evolution of the results for the blade-casing interaction: (a) resulting force on the blade, (b) energy plastically dissipated.

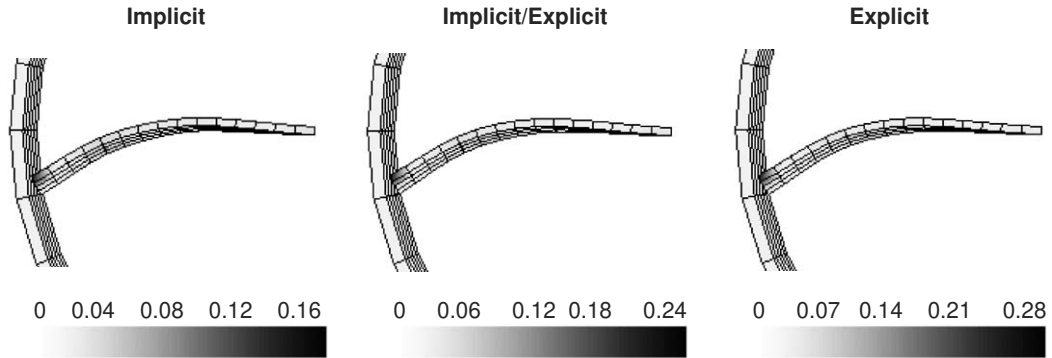


Fig. 11. Equivalent plastic strain of the blade-casing interaction after a half revolution.

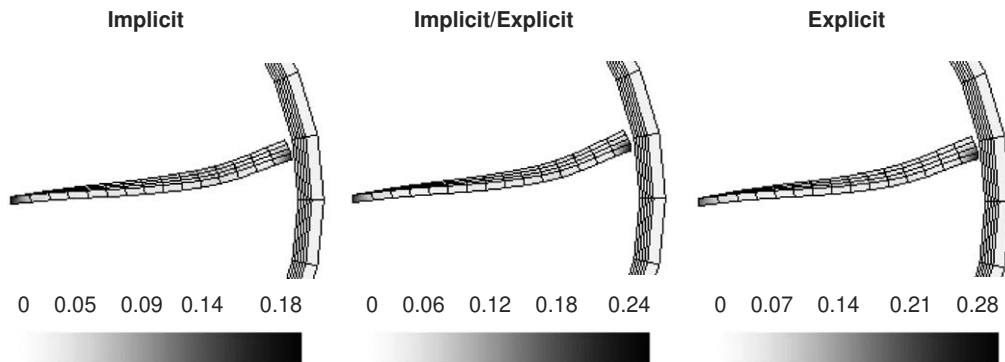


Fig. 12. Equivalent plastic strain of the blade-casing interaction after two revolutions.

half revolution. We see that the higher plastic deformations are localized at the extremity of the blade. The explicit method overestimates this maximal value, when comparing to the implicit solution. The combined method gives a solution between these two values. After the impact, the blade oscillates until the end of the simulation. During these oscillations, contact with the casing can occur (Fig. 10(a)) and the plastically dissipated energy increases. During these oscillations, new plastic strains appear at the root of the blade as we can see in Fig. 12. When analyzing the time step size in Fig. 13(a), we see that the implicit method must decrease the step size to simulate the impact. Then the combined method shifts to an explicit method. The end of the simulation can be simulated with an implicit scheme, since the time steps size that integrates the motion with accuracy, is much more important than the explicit one. When analyzing the CPU time needed for each simulation (Fig. 13(b)), we see that, since the number of elements is small and since the implicit solution has not led to important convergence problems, the gain resulting from the combined solution, compared to the implicit one, is small (2%). Nevertheless, the combined method has reacted as expected.

Let us now show the importance of the numerical dissipation when simulating a problem subject to oscillations. We compare the solution of the combined method with two different



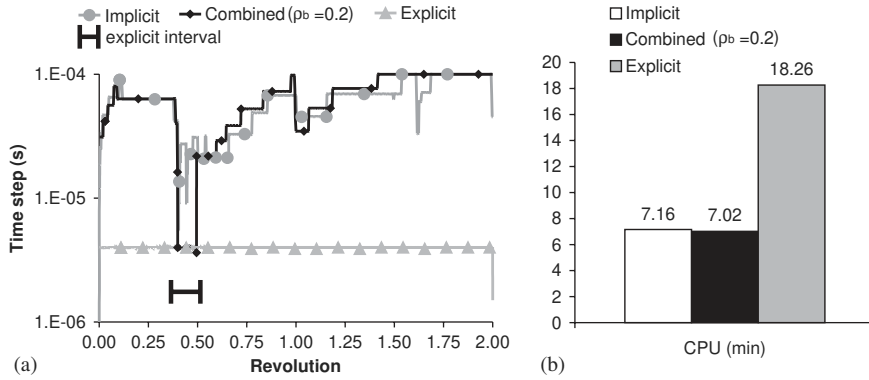


Fig. 13. Method comparison for the blade-casing interaction: (a) time step size evolution, (b) CPU.

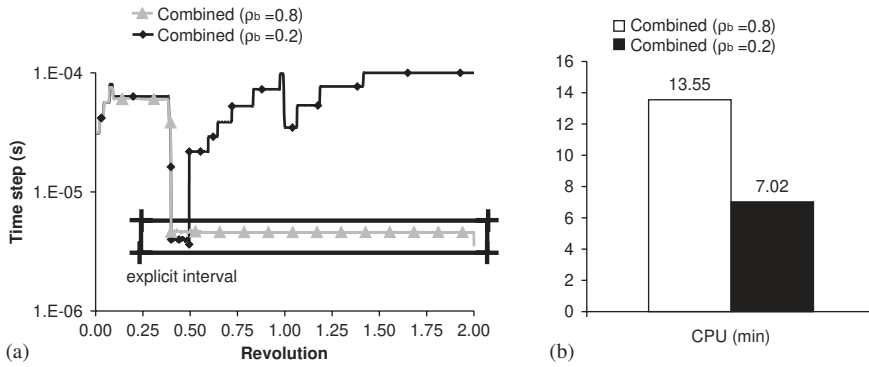


Fig. 14. Numerical dissipation influence: (a) time step size evolution, (b) CPU.

spectral radii:  $\rho_b = 0.2$  that leads to a high numerical dissipation and  $\rho_b = 0.8$  that leads to a low numerical dissipation. If we compare the evolution of the time step size in Fig. 14(a), we see that, during the explicit simulation, if the high-frequency modes are not numerically dissipated, the predicted implicit step is never large enough to shift to an implicit algorithm. This results in an increase of the CPU time needed to solve problem (Fig. 14(b)).

## 5. Conclusions

An integration scheme that combines implicit and explicit schemes was presented. This scheme integrates some time intervals with an implicit scheme, and others with an explicit scheme. First, automatic criteria that decide to shift from an algorithm family to another were developed. Next, stable balanced initial conditions have also been proposed when shifting from an explicit algorithm to an implicit algorithm. Then, we explain how to implement the combined scheme in a finite element code. Finally, numerical examples of impact were proposed. The combined method was shown to be as accurate as a full implicit or explicit method. But the computation time was

reduced thanks to this combination. Next, we see that the choice of the user parameters can modify the computational time of the combined method. For example, if the security factor of the shift criteria is chosen to low or if the integration error tolerance is chosen to high, the combined method leads to some uninteresting change of methods. Finally, we show that to be able to shift automatically to an implicit method when the impact was computed with an explicit algorithm, the explicit method needs to possess numerical dissipation. Without this dissipation, oscillations resulting from the impact do not allow the integration error to verify the shift criterion.

## References

- [1] Yang D, Jung D, Song I, Yoo D, Lee J. Comparative investigation into implicit, explicit, and iterative implicit/explicit schemes for the simulation of sheet-metal forming processes. *J Mater Process Technol* 1995;50:39–53.
- [2] Gelin J, Boulmane L, Boisse P. Quasi-static implicit and transient explicit analyses of sheet-metal forming using a  $C^\circ$  three-nodes shell element. *J Mater Process Technol* 1995;50:54–69.
- [3] Sun J, Lee K, Lee K. Comparison of implicit and explicit finite element methods for dynamic problem. *J Mater Process Technol* 2000;105:110–8.
- [4] Jung D, Yang D. Step-wise combined implicit–explicit finite-element simulation of autobody stamping process. *J Mater Process Technol* 1998;83:245–60.
- [5] Finn M, Galbraith P, Wu L, Hallquist J, Lum L, Lin T-L. Use of a coupled explicit–implicit solver for calculating spring-back in automotive body panels. *J Mater Process Technol* 1995;50:395–409.
- [6] Narkeeran N, Lovell M. Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure. *Finite Elements Anal Des* 1999;33:29–42.
- [7] Noels L, Stainier L, Ponthot J-P, Bonini J. Combined implicit–explicit algorithms for non-linear structural dynamics. *Rev Eur Elements-Finis, Dyn Rapide* 2002;11:565–91.
- [8] Noels L, Stainier L, Ponthot J-P. Combined implicit/explicit time integration algorithms for the numerical simulation of sheet metal forming. *J Comput Appl Math*, in press.
- [9] Belytschko T, Hughes T. *Computational methods for transient analysis*. Amsterdam: North-Holland; 1983.
- [10] Hughes T. *The finite element method*. Englewood Cliffs, NJ: Prentice-Hall; 1987.
- [11] Ponthot J-P, Hogge M. On relative merits of implicit schemes for transient problems in metal forming simulation. In: *International Conference on Numerical Methods for Metal Forming in Industry*, vol. 2, Baden-Baden, Germany, 1994. p. 128–48.
- [12] Chung J, Hulbert J. A time integration algorithms for structural dynamics with improved numerical dissipations: the generalized- $\alpha$  method. *J Appl Mech* 1993;60:371–5.
- [13] Gérardin M, Cardona A. *Flexible multibody dynamics (A finite element approach)*. New York: Wiley; 2000.
- [14] Noels L, Stainier L, Ponthot J. Self-adapting time integration management in crash-worthiness and sheet metal forming computations. *Int J Vehicle Des* 2002;30(2):67–114.
- [15] Noels L, Stainier L, Ponthot J-P, Bonini J. Automatic time stepping algorithms for implicit numerical simulations of non-linear dynamics. *Adv Eng Software* 2002;33(10):581–95.
- [16] Hulbert G, Chung J. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Comput Methods Appl Mech Eng* 1996;137:175–88.