Combined implicit-explicit algorithms for non-linear structural dynamics

Ludovic Noels[*] — **Laurent Stainier**[*] — **Jean-Philippe Ponthot**[*]
Jérôme Bonini[**]

[*] *University of Liège*
LTAS-Thermomechanics - Bât. B52/3
1 Chemin des Chevreuils
B-4000 Liège 1, Belgium

{l.noels,l.stainier,jp.ponthot}@ulg.ac.be

[**] *SNECMA-Moteurs*
Engineering Division - centre de Villaroche
F-77550 Moissy-Cramayel, France

ABSTRACT. *To solve fast dynamic problems, an explicit method is the most adapted. But for slower dynamics, an implicit method is more stable. The industrial problems are governed by high frequency (impact, ...) during short time intervals and slower dynamics (spring-back, ...) during other time intervals. The optimal solution is then to have both implicit algorithm and explicit methods readily available in the same code and to be able to switch automatically from one to another. Criteria that decide when to shift from a method to another have been developed here. Implicit balanced restarting conditions that annihilate numerical oscillations resulting for an explicit calculation are also proposed.*

RÉSUMÉ. *Pour résoudre des problèmes de dynamique rapide une méthode explicite est la mieux adaptée. Par contre, pour des dynamiques plus lentes, une méthode implicite est plus stable. Les problèmes industriels sont gouvernés par des hautes fréquences (impact, ...) pendant de courts intervalles temporels et par une dynamique plus lente (retour élastique, ...) pendant les autres intervalles. La solution optimale est donc de posséder les deux algorithmes, disponibles dans le même code et de pouvoir basculer automatiquement de l'un à l'autre. Des critères qui décident automatiquement quand passer d'une méthode à l'autre, sont développés dans ce travail. Des conditions initiales stables et équilibrées pour le passage de l'explicite vers l'implicite sont aussi proposées.*

KEYWORDS: *Contact, impact, implicit-explicit, time integration, dynamics, non-linearities.*

MOTS-CLÉS : *Contact, impact, intégration temporelle, implicite-explicite, dynamique, non-linéaire.*

1. Introduction

Most industrial problems of today need to be simulated with non-linear models. This is especially true in aeronautical domains where the constructor must assure that components stand up to impact. At the present time, those structures are developed with a lot of experimental studies. Since they require manufacturing (and subsequent loss) of samples, those experimental studies are extremely onerous. Adequate non-linear numerical finite elements simulations can then avoid those expensive experiments and only a final specimen is necessary in order to be validated and qualified for flight.

Engine manufacturers thus need to be able to simulate the response of a full non-linear structure (e.g. jet engine, ...) under external dynamics loading (e.g. load unbalance, ...). Model and simulation of those non-linear dynamics phenomena can only be solved with a time-incremental approach. The choice of a time integration algorithm, adapted to the studied problem 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. Within time integration algorithms, there are two families: explicit algorithms [BEL 83, HUG 87, GER 94, PON 95, HUL 96, BEN 98], and implicit algorithms [BEL 83, HUG 87, CAS 91, CHU 93, GER 94, PON 95, GER 97, BEN 98] among other references. Implicit ones need to be solved iteratively on each time increment (time step), contrarily to explicit ones. Indeed, explicit algorithms compute solution at time t_{n+1} only from solution at time t_n. On the opposite, implicit algorithms compute solution at time t_{n+1} from solution at time t_n, but also from solution at time t_{n+1}. But, for stability reasons, explicit methods use smaller time steps than implicit ones. Explicit methods, avoiding iterations and convergence problems, are therefore generally used for fast dynamics simulation for which small time step size is always necessary to capture high frequency solutions. It is especially true for problems with many degrees of freedom for which iterations are very expensive and convergence problems are frequent [YAN 95]. On the other hand, for slower dynamics problems, implicit algorithms allow to work with greater time step size, resulting in more numerical stability and accuracy [YAN 95, GEL 95, SUN 00]. Nevertheless, it appears that, for some clearly fast dynamics problems, implicit solutions are cheaper and more accurate than explicit solutions [PON 94, HOG 96, GRA 99]. Therefore, choosing an integration algorithm is a difficult task on which depend the cost and the accuracy of the numerical solution. More, most of industrial problems can take advantage from a solution method that combines both families of integration algorithms.

Three methods allow this combination. A first combination integrates in time some forces with an implicit method and other forces with an explicit one. In [KAN 99] the contact accelerations are treated with an implicit algorithm and the internal accelerations are treated with an explicit algorithm. In [PLE 85] the linear part and the non-linear part of the constitutive relation are respectively integrated by implicit and explicit method. A second combination separates the mesh into sub-domains. Each sub-domain can be treated with a different integration algorithm and a different time

step size [BEL 78, OWE 80, LIU 84, BEL 92, DAN 98, WU 00, COM 01, GRA 01]. A third combination integrates 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 and they were all developed for sheet metal forming analysis. In [JUN 98], a stamping simulation 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 simulates stamping (as a fast dynamics problem) with an explicit scheme and springback phase (slow dynamics) is subsequently analyzed with an implicit one [FIN 95, NAR 99]. 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.

This third way is the method that has been generalized in this work because nonlinear problems under consideration have significant changes of properties in time. For example, in sheet metal forming, stamping (with a more adapted explicit scheme) has radically different properties than springback (with a more adapted implicit scheme). For simulation of a jet engine under load unbalance, when rotational velocity increases (or when impact with a bird occurs) contacts and large plastic deformations appear, leaving explicit algorithm to be the most adapted. On the other hand, when rotational velocity decreases, accuracy allows to work with larger time step size and an implicit method becomes cheaper and more stable than an explicit one. Nevertheless, a criterion that decides to shift from a family to another must be automatic and not fixed by the user. There are two reasons to have an automatic criterion. Firstly: the choice of a family could not be made only from physical criteria, but must also consider numerical conditions. For example, some problems of fast dynamics can be simulated with more accuracy and less computational time with an implicit method rather than with an explicit one [PON 94, HOG 96, GRA 99]. Secondly: for complex simulations, shift from a method to another can be followed by a comeback to the original one. For example when jet engine rotation velocity decreases, simulation can use an implicit scheme, after having simulated the impact phase (short duration event) with an explicit scheme. But if a critical mode is excited, the algorithm must be able to shift to an explicit scheme again. Another example can be found in hemming problems where a lot of stamping and springbacks are followed by one another [NOE 02]. Automatic criteria must then be developed. They depend on integration error [CAS 91, GER 94, HUL 95, DUT 98, NOE 00, NOE 01, NOE 02] that allows to determinate implicit time step size and they also depend on a ratio between the computational time (or CPU) needed to solve an implicit time step and the CPU needed to solve an explicit time step. Initial conditions, when shifting from explicit scheme to implicit scheme occurs, are also defined to avoid lack of stability and convergence.

This paper will be organized into three sections. First, time integration algorithms will be briefly explained. Second, the mentioned criteria and initial restarting conditions will be detailed. Third, numerical simulations will be exposed to validate the methodologies.

2. Numerical integration of transient problems

In this section, the equations of motion and both implicit and explicit schemes of integration are rapidly explained. Next, integration parameters management, subsequently needed to develop shifting criteria, are explained.

2.1. *Equations of motion*

FEM (space Finite Element Model) semi-discretization of the equations of motion of a nonlinear structure leads to the following coupled set of second order nonlinear differential equations [BEL 83, HUG 87, GER 94, PON 94, PON 95, HOG 96]:

$$R = M\ddot{x} + F^{int}\left(x, \dot{x}\right) - F^{ext}\left(x, \dot{x}\right) = 0 \qquad [1]$$

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^{int} the vector of internal forces resulting from body's deformation and F^{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 or geometrical non-linearities. The set of equations [1] is completed by two sets of given initial conditions at time zero:

$$x_0 = x\left(t = 0\right) \quad \dot{x}_0 = \dot{x}\left(t = 0\right) \qquad [2]$$

Semi-discretized internal and external forces can be written:

$$F^{int}\left(x, \dot{x}\right) = \int_{V(t)} [B]^T \{\sigma\}\, dV \qquad [3]$$

$$F^{ext}\left(x, \dot{x}\right) = \int_{V(t)} [N]^T \{b\}\, dV + \int_{S(t)} [N]^T \{f\}\, dS \qquad [4]$$

where b is the body force per unit mass, σ is the Cauchy stress, B is the matrix of the derivatives of the FEM shape function, f is the surface tractions, N the matrix of the FEM shape function, $S\left(t\right)$ the current surface of the body and $V\left(t\right)$ is the current volume of the body. Note that expression [4] collects all types of loading (applied through local or distributed actions, in a follow-up way or not, reactions to imposed displacements and contact situations) and that the consistent mass matrix reads:

$$M = \int_{V(t)} \rho\, [N]^T\, [N]\, dV = \int_{V_0} \rho_0\, [N]^T\, [N]\, dV_0 \qquad [5]$$

where ρ is the current mass density of the material, ρ_0 the initial mass density of the material and S_0 is the initial surface of the body.

2.2. *Implicit schemes*

Implicit schemes are classically designed for vibrations and low speed dynamics of structures. For an implicit scheme, the elements of the solution at time t_{n+1}, implicitly depends on other elements of the solution at time t_{n+1} itself. The problem must then be solved in an iterative fashion. Stability (i.e. positive damping of initial perturbations) imposes different restrictions on this family of algorithms and a proper choice of parameters. The time step size does not need to be lower than a limit to have a stable time step since the scheme is unconditionally stable.

Implicit schemes: The generalized-α trapezoidal scheme

The most general scheme for implicit integration of [1] is a generalized trapezoidal scheme [BEL 83, HUG 87, CHU 93] 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:

$$\dot{x}_{n+1} = \dot{x}_n + (1 - \gamma)\,\Delta t \ddot{x}_n + \gamma \Delta t \ddot{x}_{n+1} \tag{6}$$

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \left(\frac{1}{2} - \beta\right)\Delta t^2 \ddot{x}_n + \beta \Delta t^2 \ddot{x}_{n+1} \tag{7}$$

or equivalently:

$$\ddot{x}_{n+1} = \frac{1}{\beta \Delta t^2}\left[x_{n+1} - x_n - \Delta t \dot{x}_n - \left(\frac{1}{2} - \beta\right)\Delta t^2 \ddot{x}_n\right] \tag{8}$$

$$\dot{x}_{n+1} = \frac{\gamma}{\beta \Delta t}\left[x_{n+1} - x_n + \left(\frac{\beta}{\gamma} - 1\right)\Delta t \dot{x}_n + \left(\frac{\beta}{\gamma} - \frac{1}{2}\right)\Delta t^2 \ddot{x}_n\right] \tag{9}$$

The discretized equations of motion [1] can be rewritten under the form proposed by Chung and Hulbert [CHU 93]:

$$\begin{aligned} R_{n,n+1} \quad = \quad & \frac{1 - \alpha_M}{1 - \alpha_F} M \ddot{x}_{n+1} + \frac{\alpha_M}{1 - \alpha_F} M \ddot{x}_n + \left(F^{int}_{n+1} - F^{ext}_{n+1}\right) + \\ & \frac{\alpha_F}{1 - \alpha_F}\left(F^{int}_n - F^{ext}_n\right) = 0 \end{aligned} \tag{10}$$

where $R_{n,n+1}$ is the residual vector of time step n to $n + 1$.

Particular choices of parameters lead to well-known [BEL 83, HUG 87, CHU 93] schemes such as:

$- \alpha_M = \alpha_F = 0$ for Newmark scheme

$- \alpha_M = 0$ for Hilber-Hughes-Taylor scheme

$- \alpha_F = 0$ for Wood-Bossak-Zienkiewicz scheme

Unconditional stability and second order accuracy of the scheme, for linear problems [CHU 93], require that:

$$\begin{align}
\gamma &\geq \frac{1}{2} - \alpha_M + \alpha_F \\
\alpha_M &\leq \frac{1}{2} \\
\beta &\geq \frac{1}{4}\left(1 + \alpha_F - \alpha_M\right)^2
\end{align}$$

[11]

Associated rules for Newmark scheme are the classical:

$$\begin{align}
\gamma &\geq \frac{1}{2} \\
\beta &\geq \frac{1}{4}\left(\gamma + \frac{1}{2}\right)^2
\end{align}$$

[12]

Maximum of accuracy is provided in frequency and amplitude for [CHU 93]:

$$\begin{align}
\alpha_F &\in \left[0, \frac{1}{2}\right] \\
\alpha_M &= 3\alpha_F - 1 \\
\gamma &= \frac{1}{2} - \alpha_M + \alpha_F \\
\beta &= \frac{1}{4}\left(1 + \alpha_F - \alpha_M\right)^2
\end{align}$$

[13]

It's worth pointing out that though classical schemes require $0 \leq \alpha_F \leq 1/2$ (i.e. sampling the force in the second half of $[t_n, t_{n+1}]$), here, no such rule is followed for α_M, the sampling parameter for the inertia terms: it might be negative for instance, thus leading to an extrapolation at t_{n+1} instead of an interpolation.

Iterative solution of the nonlinear system [10] first requires the elimination of acceleration and velocity at time t_{n+1} with the help of [8] and [9] and, secondly, the writing of the Hessian matrix of the system, i.e.:

$$S = \left[\frac{1}{\beta\Delta t^2}\left(\frac{1 - \alpha_M}{1 - \alpha_F}\right)M + \frac{\gamma}{\beta\Delta t}C_T + K_T\right]$$

[14]

where K_T, C_T are respectively the tangent stiffness and damping matrices defined by:

$$K_T = \frac{\partial}{\partial x}\left(F^{int} - F^{ext}\right)$$

[15]

$$C_T = \frac{\partial}{\partial \dot{x}}\left(F^{int} - F^{ext}\right)$$

[16]

The residual after iteration number i is defined by:

$$
\begin{aligned}
R \quad = \quad & \frac{1 - \alpha_M}{1 - \alpha_F} M \ddot{x}_{n+1}^i + \frac{\alpha_M}{1 - \alpha_F} M \ddot{x}_n + \\
& \left[F_{n+1}^{int} \left(x_{n+1}^i, \dot{x}_{n+1}^i \right) - F_{n+1}^{ext} \left(x_{n+1}^i, \dot{x}_{n+1}^i \right) \right] + \\
& \frac{\alpha_F}{1 - \alpha_F} \left[F_n^{int} - F_n^{ext} \right]
\end{aligned}
\qquad [17]
$$

Using equations [14] to [17] and a Newton-Raphson technique, the iterative solution of system [8], [9] and [10] can be written as:

$$
S \Delta x = -R \qquad [18]
$$

Iterations stop when the non-dimensional residual r becomes lower than the accuracy tolerance δ that is defined by the user. Therefore, the following relation is verified:

$$
r = \frac{\|R\|}{\|F^{int}\| + \|F^{ext}\| + \|M\ddot{x}\|} \leq \delta \qquad [19]
$$

Other types of implicit schemes exist. For example, we can cite the generalized-θ mid-point scheme [PON 95, PON 94, HOG 96], the conservative schemes such as "Energy Momentum Conserving Algorithms" [SIM 92, GON 96, GON 99, GER 00] or the conservative scheme with relative dissipation such as "Energy Dissipative Momentum Conserving Algorithms" [ARM 99, ROM 00, ARM 01]. Nevertheless, in this paper only the α-generalized scheme is used. Proposed methods can be extended to other implicit schemes.

2.3. *Explicit schemes*

This is the most advocated scheme [BEL 83, HUG 87] for integrating [1] in case of wave propagation and impact problems, i.e. high speed dynamics. For an explicit algorithm, the elements of solution at time t_{n+1} depend only on the solution at time t_n. Therefore, the resolution does not need to be iterative. Stability (i.e. positive damping of initial perturbations) imposes the time step size to be lower than a limit. The scheme is conditionally stable.

Explicit scheme: The generalized-α explicit scheme

Chung and Hulbert have extended their implicit scheme to an explicit one, taking $\alpha_F = 1$ in [10] [HUL 96]. Its most principal interest is its numerical dissipation property. Time integration is then:

$$
\ddot{x}_{n+1} = \frac{M^{-1} \left(F_n^{ext} - F_n^{int} \right) - \alpha_M \ddot{x}_n}{1 - \alpha_M} \qquad [20]
$$

$$
\dot{x}_{n+1} = \dot{x}_n + \Delta t \left[(1 - \gamma) \ddot{x}_n + \gamma \ddot{x}_{n+1} \right] \qquad [21]
$$

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{x}_n + \beta \ddot{x}_{n+1} \right] \qquad [22]$$

Optimal numerical dissipation is reached for the above parameters depending on the spectral radius ρ_b that conditions the numerical damping of the high frequency [HUL 96]:

$$
\begin{aligned}
\alpha_M &= \frac{2\rho_b - 1}{1 + \rho_b} \\
\gamma &= \frac{3}{2} - \alpha_M \\
\beta &= \frac{5 - 3\rho_b}{(1 + \rho_b)^2 (2 - \rho_b)}
\end{aligned}
\qquad [23]
$$

This scheme is conditionally stable and time step size is limited, depending on maximal model frequency ω_{max} but also depending on spectral radius:

$$\Delta t = \gamma_s \Delta t_{crit} = \gamma_s \frac{\Omega_s (\rho_b)}{\omega_{max}} \qquad [24]$$

with:

$$\Omega_s (\rho_b) = \sqrt{\frac{12 (1 + \rho_b)^3 (2 - \rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}} \qquad [25]$$

In relation [24], γ_s is a safety factor (< 1) that accounts for the destabilizing effects of non-linearities. Evolution of Ω_s (figure 1) indicates that for a low spectral radius (i.e. high numerical dissipation), time step size is 22.5% lower than for a high spectral radius (i.e. low numerical dissipation). Nevertheless, numerical dissipation allows to work with higher security.

Other explicit schemes exist. The most popular one is the central difference. Its principal drawback is its absence of numerical dissipation, leading to important numerical oscillations or even to instability (rapid amplification of the numerical oscillations resulting from a physical or a numerical perturbation) [BEL 83, HUG 87]. As with implicit scheme, the proposed methodology can be extended to this scheme (excepting the setting of the spectral radius in section (3.4) that is an user parameter only in the generalized-α explicit scheme).

2.4. *Parameters control for numerical integration*

First the time step size control for implicit schemes is studied. Next an algorithm deciding when the Hessian matrix needs to be updated is explained. Finally the evaluation of explicit time step size is exposed.
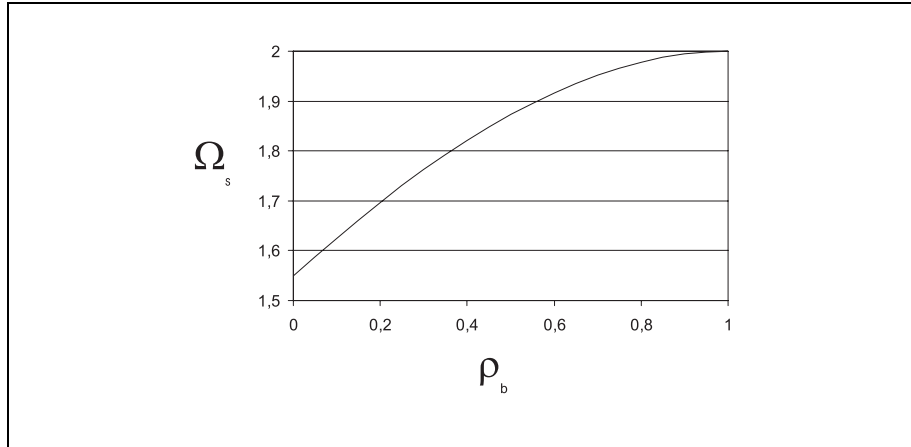
Figure 1. *Stability limit evolution depending on spectral radius ρ_b*

2.4.1. *Implicit time step size control*

The implicit time step size control is the one proposed by Géradin [GER 97], extended here to highly non-linear problems [NOE 00, NOE 02]. This scheme continuously adapts time step size to physical modes evolution and keeps time step size constant during long time intervals. To estimate time step size an integration error is computed.

The integration error is deduced from truncated terms of relations [6] and [7]. This error is to the third order: $O\left(\frac{1}{6}\Delta t^3 \dddot{x}\right) \simeq O\left(\frac{1}{6}\Delta t^2 \Delta \ddot{x}\right)$. Then it leads to:

$$e = \frac{\Delta t^2}{6}\|\Delta \ddot{x}\| \tag{26}$$

To have a problem independent error, it is made non dimensional (x_0 is the initial position vector):

$$e_{nd} = \frac{\Delta t^2}{6\|x_0\|}\|\Delta \ddot{x}\| \tag{27}$$

This error is then divided by a reference error ε. This reference is the average error of a linear oscillator of pulse ω and of non-dimensional pulse $\Omega = \Delta t \omega$ [GER 94, NOE 00, NOE 02]:

$$\varepsilon\left(\Omega\right) = \frac{\left(1 - \alpha_F\right)\Omega^3 \sqrt{1 + \frac{\Omega^2}{4}}}{3\pi\left[1 - \alpha_M + \left(1 - \alpha_F\right)\Omega^2 \beta\right]} \tag{28}$$

Ω must be known to evaluate [28]. Ten time steps give a good accuracy for integrating a linear oscillator over one period. Therefore, since $\omega = 2\pi\nu$ (ν is the frequency),

taking $\Delta t = (10\nu)^{-1}$ leads to a reference non-dimensional pulse $\Omega_k \simeq 0.6$. Then [27] is rewritten:

$$e_{nd,ref} = \frac{\Delta t^2}{6} \frac{\|\Delta \ddot{x}\|}{\varepsilon (\Omega_k) \|x_0\|} \qquad [29]$$

Nevertheless, for an uniform rotation, the components of the nodal vector of accelerations $\ddot{x}_i$ are not a constant. Indeed, if its modulus is constant, direction changes. Therefore, even if the motion is perfectly integrated, error [29] would not be equal to zero. The integration error is then rewritten by taking the variation of the nodal acceleration modulus (Nn is the number of nodes):

$$e_{int} = \frac{\Delta t^2}{6} \frac{\sum_{i=1}^{Nn} \Delta \|\ddot{x}_i\|}{\varepsilon (\Omega_k) \|x_0\|} \qquad [30]$$

which also remains consistent for translation motions.

Time step size is deduced from the integration error [30] and from a tolerance $PRCU$ fixed by the user. The relation to be verified is:

$$e_{int} < PRCU \qquad [31]$$

The new time step size Δt_{new} to reach a reference integration error (half of the tolerance $PRCU$) is deduced from the current time step size (Δt_{cur}) and from the current integration error ($e_{int,cur}$), using the following relation developed in [GER 94]:

$$\left(\frac{\Delta t_{new}}{\Delta t_{cur}}\right)^{\eta} = \frac{PRCU}{2e_{int,cur}} \qquad [32]$$

with $\eta \in [2, 3]$ a user specified parameter. This range is obtained by evaluating [28] for Ω asymptotically equal to 0 and to ∞ [GER 94, NOE 00, NOE 02]. The time step size management, based on relations [31] and [32], used is the one developed in [NOE 00, NOE 02] that is able to integrate with accuracy and without modifying the time step size too often.

2.4.2. *Hessian matrix updating*

For non-linear problems, if the Hessian matrix is not recomputed and inverted, the iteration is much less expensive, but the convergence of the resulting modified Newton-Raphson iterations is slower than if the Hessian matrix were recomputed and inverted at each iteration. For some step, divergence could even occur. Therefore, the criterion must consider two facts [NOE 01, NOE 02]:

– Convergence of the iterations must be ensured.

– Not updating the Hessian matrix must reduce the total computation cost.

The evolution of the non-dimensional residual r [19] from one iteration to another could indicate if the problem is converging or not. While r decreases, iterations converge even if the Hessian matrix is not recalculated and not inverted. An indication of how it could be interesting not to recalculate the Hessian matrix is the ratio *VALRF* between the CPU time needed for an iteration with re-calculation and the CPU time needed for an iteration without re-calculation [NOE 01, NOE 02]. This ratio is generally much larger than one. The proposed algorithm is the following:

   – The Hessian matrix is recalculated at the first iteration only if the time step size has changed. Indeed, S depends on Δt, see [14]. Therefore, a modification of the time step size requires the Hessian matrix to be updated in order to avoid divergence.

   – After the first iteration, if the residual r is divided by a ratio chosen equal to $RAPRES = \frac{VALRF}{10} \in [0.2, 0.85]$ (thus ensuring a good rate of convergence), the Hessian matrix is kept constant until the iteration number become larger than *VALRF*. Doing so, we hope to reach convergence before that. And if it is the case, the cost of the time steps was actually cheaper.

   – If the non-dimensional residual has not been divided by *RAPRES*, the next iteration then needs updating of the Hessian matrix. But, if the residual has increased, this iteration does not take as initial values (x, $\dot{x}$, $\ddot{x}$) the values at the end of the previous iteration, but the values at the end of the last iteration which has converged (i.e. the penultimate iteration). Some divergences of the iterations are then avoided.

   – When the number of iteration becomes larger than *RAPRES*, the Hessian matrix is recomputed at each subsequent iteration.

This algorithm avoids some needless re-calculations and inversions of the Hessian matrix. For problems with a lot of degrees of freedom, this algorithm reduces substantially the computation time [NOE 01, NOE 02].

2.4.3. *Explicit time step size evaluation*

Explicit schemes are conditionally stable. Time step size must be lower than a limit i.e. the critical time step [24]. In this relation, the maximal structure pulse ω_{max} is computed by the power iteration method proposed in [BEN 98]. ω_{max} is then correctly evaluated. The critical time step ensures stability but does not ensure accuracy. Therefore, the safety factor γ_s [24] can be deduced from the integration error as it was the case for the implicit time step size [NOE 02].

3. Shifts from an algorithm family to another

Shifting from an implicit algorithm to an explicit one, in one way or another, leads to three major practical problems. The first problem is to determine a criterion that is able to decide to shift from an implicit algorithm to an explicit one. A second one is to

determinate a criterion that is able to decide of the opposite shift. And a third problem is to have stable (i.e. without too much numerical oscillations leading to divergence of the implicit iteration) initial conditions for the implicit algorithm when coming from an explicit one. In this paper only α-generalized implicit (section 2.2) and explicit (section 2.3) algorithms are actually used.

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. Such a ratio was used in [RAM 89] to determinate if the whole computation of a linear system must be made with an explicit or an implicit method. In this paper, this ratio is averaged for each step to be able to shift from a method to another for non-linear simulation. For example, if the current algorithm is implicit, the actual cost of an implicit step is evaluated. The average cost of an implicit step is then obtained by adding 0.9 time the old one and 0.1 time the actual. The average cost of an explicit step is obtained with the same technique. The average cost of an implicit step divided by the explicit one gives r^*. Initialization of r^* is obtained by beginning the analyze with an explicit step. The CPU needed to compute this explicit step is the initial value of the averaged cost of an explicit step. The result of this step is ignored and the analyze begins again with an implicit method. The CPU needed for the first implicit step gives the initial value for the averaged cost of an implicit step. With these two values, the initial value of r^* can be computed. The ratio r^* is re-evaluated only if the implicit time step size does not change, avoiding to take into account the computation cost inherent from an updating of the Hessian matrix coming from a time step size change (section 2.4.2). Shift to explicit method occurs if:

$$\mu \Delta t_{impl} < r^* \Delta t_{expl} \tag{33}$$

where Δt_{expl} is evaluated with [24]. Factor μ is taken greater than unity (typical value is disscussed in section (4)) to avoid shifting from a method to another too frequently.

This methodology allows to consider the number of degrees of freedom, the algorithms efficiency, the updating of the Hessian matrix (section 2.4.2), the residual tolerance required and the non-linearities evolution. During implicit computations, time step size depends on integration error (section 2.4.1). Therefore time step size is the largest one that integrates the conservation equations with the tolerance required. Then the criterion [33] also depends implicitly on this tolerance.

Once the implicit algorithm has shifted to an explicit one, we have to be able to return to an implicit method when the dynamics conditions allow us. This is the topic of the following section.

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

While the method used is an implicit one, the explicit time step size could always be easily computed from [24]. When the current method is explicit, the implicit time step size, which correctly integrates the problem, does not remain directly accessible. Using developments of section (2.4.1), nodal acceleration variations can provide us with this implicit time step size. Using relation [32], acceleration variation is proportional to Δt^{η}. Inverting relation [30] the implicit time step size is (with Nn the number of nodes):

$$\Delta t_{impl} = \left[6 \frac{\frac{PRCU}{2} \varepsilon \left(\Omega_k \right) \|x_0\| \left(\Delta t_{expl} \right)^{\eta - 2}}{\sum_{i=1}^{Nn} \Delta \|\ddot{x}_i\|} \right]^{\frac{1}{\eta}} \tag{34}$$

Therefore the explicit to implicit shift criterion is similar to [33]. It leads:

$$\Delta t_{impl} > \mu r^* \Delta t_{expl} \tag{35}$$

with Δt_{expl} the current explicit time step size.

3.3. *Other considerations of the construct of the implicit/explicit algorithm*

A problem results from the evaluation of r^*. Since the implicit algorithm generally shifts to an explicit one when dynamics is fast and convergence problems are frequent, the number of iteration with updating of the Hessian matrix is very large and the subsequent implicit step computation time (CPU) used to evaluate r^* is also large. When the algorithm is able to shift to an implicit one, the dynamics is slower and the convergence problems will be less frequent. Then, the cost of an implicit step, computed before shifting to an explicit method is overvalued. Therefore, to have a valid ratio r^*, the cost of an implicit step must be adapted with the dynamics evolution of the problem when the algorithm in used is the explicit one. Then, if predicted implicit time step size Δt_{impl} increases, the cost of an implicit step is decreased, to take into account the current smoothness of the dynamics. For example, if Δt_{impl} increases of 10%, then the average cost of an implicit step is decreased by $d\%$ (d will be defined on section 4).

The implicit convergence problems are now examined. Consider an implicit step for which integration error satisfies relation [31] but for which the next time step diverges. Relation [33] could be verified and then the algorithm shifts to explicit. But then relation [35] is also verified and algorithm shifts back to an implicit algorithm that has divergence problems. Therefore, the treatment of divergence problems differs from the one proposed in [NOE 02] (where the parameters $PRCU$ -see relation [31]- has been reduced) to take into account the shift from an algorithm to another. When divergence problems occurs in a implicit step, time step size is reduced and implicit step is recomputed. Then the non-dimensional accelerations and the time step size are

kept in memory. Non-dimensional accelerations are defined considering relations [30 and 32]:

$$\Delta \ddot{x}_{nd} = \frac{\sum_{i=1}^{Nn} \Delta \|\ddot{x}_i\|}{\|x_0\| (\Delta t)^{\eta-2}} \qquad [36]$$

and the stored ones are noted as $\Delta \ddot{x}_{nd,s}$. Stored time step size (Δt_s) is the average between non-converged time step size and actual time step size (it is taken greater than actual time step size to allow working with this time step size). Therefore when an implicit scheme is used, the time step size is limited to Δt_s and when an explicit scheme is used, predicted implicit time step size is limited to Δt_s. For both schemes, this maximum time step size is increased if non-dimensional accelerations computed by [36] are lower than $\Delta \ddot{x}_{nd,s}$. When occurring, new values are:

$$\Delta t_s^{new} = \Delta t_s^{old} \left(\frac{\Delta \ddot{x}_{nd,s}^{old}}{\Delta \ddot{x}_{nd}} \right)^{\frac{1}{\eta}}$$

$$\Delta \ddot{x}_{nd,s}^{new} = \Delta \ddot{x}_{nd} \qquad [37]$$

Shifting from a balanced implicit step with a large time step size to an explicit step with a smaller time step size does not lead to numerical problems. On the other hand, shifting to a non-balanced explicit step with a small time step size to an implicit step with a larger time step size generally leads to non-convergence behaviours.

3.4. *Initial conditions when shifting from an explicit scheme to an implicit scheme*

Classical explicit scheme such as the central difference method [BEL 83] is well known to generate oscillatory (though stable) solutions. Especially the velocities, the accelerations and the values at the Gauss points are concerned. Two solutions are here provided to stabilize and balance the Gauss points values and the nodal values.

First, numerical oscillations of the Gauss points values and of the nodal values are annihilated thanks to the numerical dissipation property of the used α-generalized explicit scheme. Indeed, when relation [35] 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 ρ_b (section 2.3) set equal to zero (ρ_b is an user parameter). Thus, numerical oscillations have been greatly reduced at time t_{n+r^*} (figure 2).

The second step in the algorithm will be to determine a balanced configuration at time $t_{n+r^*+r^{**}}$. Therefore, we act twofold. First an explicit solution using r^{**} (r^{**} will be defined on next paragraph) explicit steps is computed. This solution results in $x_{n+r^*+r^{**}}^{expl}$, $\dot{x}_{n+r^*+r^{**}}^{expl}$ and in $\ddot{x}_{n+r^*+r^{**}}^{expl}$, which in turn is used as a predictor value 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 unbalanced solution. Therefore, a balanced step of size equal to the implicit time step size is reached. The methodology is thus (figure 2):

– At time t_{n+r^*} forces (internal, external and inertial) are stored.

– From time t_{n+r^*} to time $t_{n+r^*+r^{**}}$, r^{**} explicit steps are computed.

– Using forces at time t_{n+r^*}, forces at time $t_{n+r^*+r^{**}}$ are balanced to satisfy a discretized equilibrium relation such as [10].

This third point is reached considering an implicit time step size equal to $\Delta t_{r^{**}} = r^{**}\Delta t_{expl}$. Prediction values for this equilibrium are the values computed by the last explicit step:

$$
\begin{aligned}
x^0_{n+r^*+r^{**}} &= x^{expl}_{n+r^*+r^{**}} \\
\dot{x}^0_{n+r^*+r^{**}} &= \dot{x}^{expl}_{n+r^*+r^{**}} \\
\ddot{x}^0_{n+r^*+r^{**}} &= \ddot{x}^{expl}_{n+r^*+r^{**}}
\end{aligned}
\tag{38}
$$

Equilibrium relation [10] is then rewritten (indices $n + r^* + r^{**}$ are omitted for the sake of clarity):

$$
\begin{aligned}
R_{n+r^*,n+r^*+r^{**}} &= \frac{1-\alpha_M}{1-\alpha_F}M\ddot{x} + \frac{\alpha_M}{1-\alpha_F}M\ddot{x}_{n+r^*} + \left(F^{int} - F^{ext}\right) \\
&+ \frac{\alpha_F}{1-\alpha_F}\left(F^{int}_{n+r^*} - F^{ext}_{n+r^*}\right) = 0
\end{aligned}
\tag{39}
$$

with the new Hessian matrix [18]:

$$
S = \left[\frac{1}{\beta\Delta t^2_{r^{**}}}\left(\frac{1-\alpha_M}{1-\alpha_F}\right)M + \frac{\gamma}{\beta\Delta t_{r^{**}}}C_T + K_T\right]
\tag{40}
$$

with K_T, C_T respectively defined by [15] and [16]. Corrections for iteration i are:

$$
\begin{aligned}
x^{i+1}_{n+r^*+r^{**}} &= x^i + \Delta x \\
\dot{x}^{i+1}_{n+r^*+r^{**}} &= \dot{x}^i + \frac{\gamma\Delta x}{\beta\Delta t_{r^{**}}} \\
\ddot{x}^{i+1}_{n+r^*+r^{**}} &= \ddot{x}^i + \frac{\Delta x}{\beta\Delta t^2_{r^{**}}}
\end{aligned}
\tag{41}
$$

with Δx computed from [18]. In general, the iterative process necessary to reach this equilibrium quickly converges and this allows to begin the implicit method with a balanced solution at time $t_{n+r^*+r^{**}}$. Anyway r^{**} must be defined. It is always lower or equal to μr^*. It is lower if r^* is too large to lead to convergence of the first truly implicit step after time $t_{n+r^*+r^{**}}$. In this work r^{**} is limited to 100. But if r^{**} is lower than μr^*, time step size is increased (multiplied by 2) each two steps to reach $\Delta t = \mu r^*\Delta t_{expl}$.
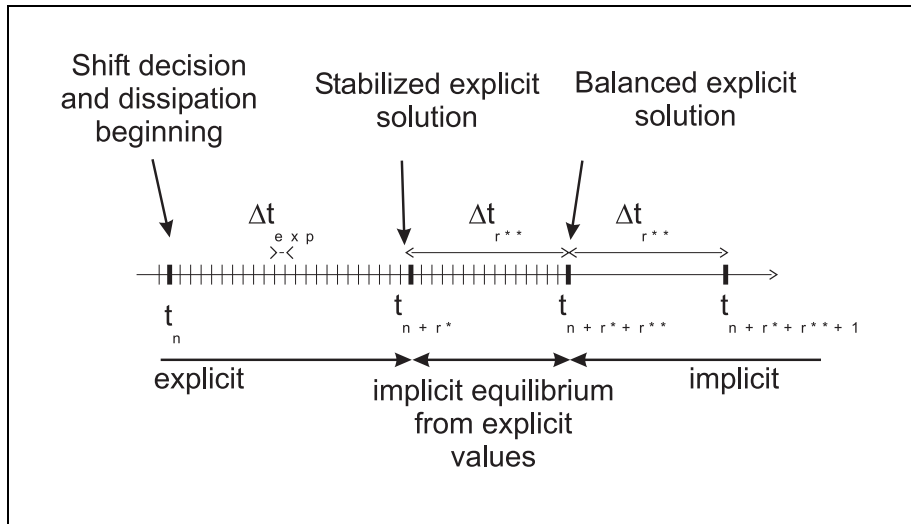
Figure 2. *Transition scheme from an explicit scheme to an implicit one*
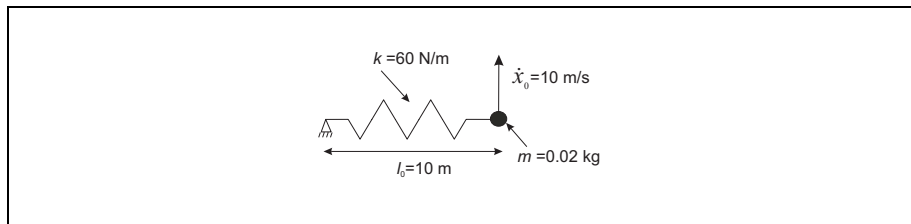

Figure 3. *Squetch of the linear spring model*

The numerical stability of such initial conditions is evaluated on a dynamic linear example. This example is inspired from the one proposed in [ARM 01] (figure 3). It consists in a mass of $0.02kg$ fixed on a spring with a stiffness of $60N/m$ and an initial length of $10m$. The other extremity of the spring is fixed and the mass has an initial velocity of $10m/s$. The numerical computation is reached with an implicit scheme ($\alpha_M = -0.97$, $\alpha_F = 0.01$, $\beta = 0.9801$ and $\gamma = 1.48$, $\Delta t = 0.147s$), with an explicit scheme ($\alpha_M = -1.6$, $\beta = 5.5$, $\gamma = 3.1$, $\Delta t = 0.028s$) and finally with a combined implicit/explicit scheme. For this last computation, the analysis begins with 15 implicit steps ($\Delta t = 0.147s$). Then the scheme is manually constrained to compute 55 explicit steps ($\Delta t = 0.028s$). Next, 5 explicit steps are reached (corresponding to the time interval t_n to t_{n+r*} of figure 2) with a spectral radius ρ_b (section 2.3) set equal to zero. The next 5 explicit steps (corresponding to the time interval t_{n+r*} to $t_{n+r*+r**}$ of figure 2) provide the predicted value to compute the balanced implicit step between t_{n+r*} and $t_{n+r*+r**}$. The rest of the computation occurs with the implicit scheme. The total energy (kinetic energy plus potential energy) of the system if illustrated for
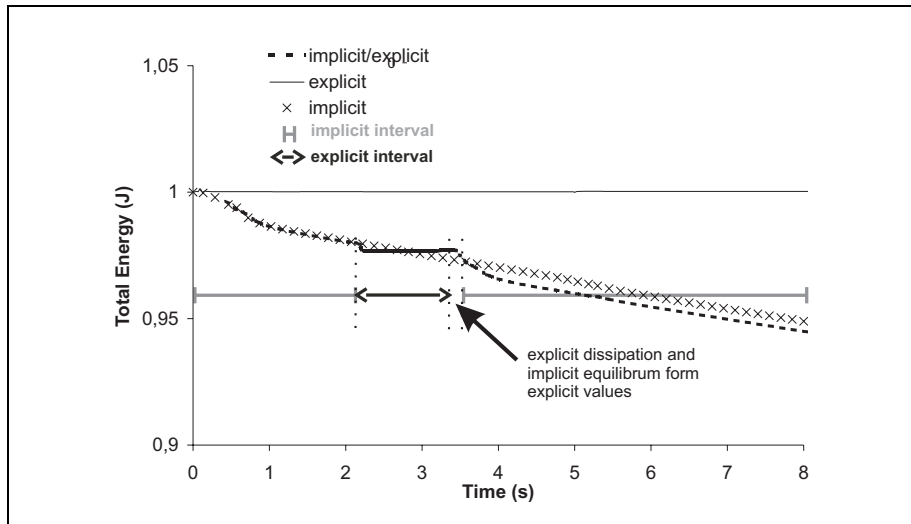
Figure 4. *Total energy evolution of the linear spring model*

each computation (figure 4). If the total energy is preserved for the explicit computation, the implicit scheme dissipates it. Assuming that a complete revolution of the mass is reached after about $6.2s$, the implicit scheme dissipates about 5% of the energy during a revolution. The first implicit interval of the combined scheme dissipates at the same rate as the all-implicit scheme. Shifting to the explicit scheme corresponds to a loss of about 0.25% in energy. The explicit interval conserves the total energy as the all explicit computation and returning to the implicit scheme corresponds to a loss of about 0.7% in the total energy. The rest of the implicit dissipation gives a curve of energy parallel to the all-implicit computation. In conclusion, we can assume that shifting from a method to the other does not produce numerical instability (increase of the total energy) and that the numerical dissipation (loss of accuracy) resulting from the shift is less important $(< 1\%)$ than the numerical dissipation resulting from the implicit computation (5% per revolution).

4. Numerical examples

Numerical examples will be computed with three methods. The first method uses a purely implicit algorithm, the second method a purely explicit algorithm, and the third method can shift from an algorithm family to another. The accuracy of the numerical results and the computation costs are compared. For all those examples, the finite elements use selective reduced integration, to avoid volumetric locking resulting from the incompressibility condition of plastic deformations. There are 8 deviatoric Gauss points and 1 volumetric Gauss point. Moreover, the parameters d (section 3.2), η [34, 36] and μ [33, 35] are respectively taken equal to 2.5, 2.5 and 1.5. Increasing

External diameter	$d_e = 6.4mm$
Length	$l = 32.4mm$
Density	$\rho = 8930kg/m^3$
Young's modulus	$E = 117E9N/m^2$
Poisson's ratio	$\nu = 0.35$
Yield stress	$\sigma_0 = 400N/mm^2$
Hardening parameter	$h = 100N/mm^2$
Initial velocity	$\dot{x}_0 = 227m/s$
Initial distance from plane	$x_0 = 1mm$

Table 1. *Geometrical and material properties of Taylor bar impact*

d or decreasing η or μ will result in more shifts from a method to another and thus will revert efficiency of the algorithm. In fact, if d is increased, the ratio r^* will decrease more rapidly (section 3.3), the relation [35] will be more rapidly verified and the algorithm will therefore shift to an implicit one. If η is decreased, the predicted implicit step will be increased [34] and the algorithm will shift sooner from an explicit scheme to an implicit scheme. As mentioned in section (2.4.1), the range of variation of η is $[2, 3]$. If μ is decreased, the relations [33, 35] will be less severe and the algorithm will shift more often from a method to another. Since a return to an implicit scheme leads to some iterations (section 3.4), computation costs can increase. On the other hand, decreasing d or increasing η or μ will result in less shifts from a method to another. Therefore, the used method can be the less appropriate for some time interval, increasing the computation cost. The proposed values result from intensive numerical experiments.

4.1. *Numerical example 1: Taylor bar impact*

A cylindrical bar (table 1), discretized with 1080 elements (27 in each section times 40 along the axis), is at distance x_0 from a rigid plane. It has an initial velocity $\dot{x}_0$ directed towards the plane. After the wave has propagated across it, the bar looses contact with the wall (figure 5).

This problem is well suited to evaluate the performances of the algorithm combination method. Indeed, the integration time could be decomposed in three intervals of different dynamics properties:

– Rigid translation motion before impact has slow dynamics properties.

– Impact with large plastic straining (about 300%) has fast dynamics properties.

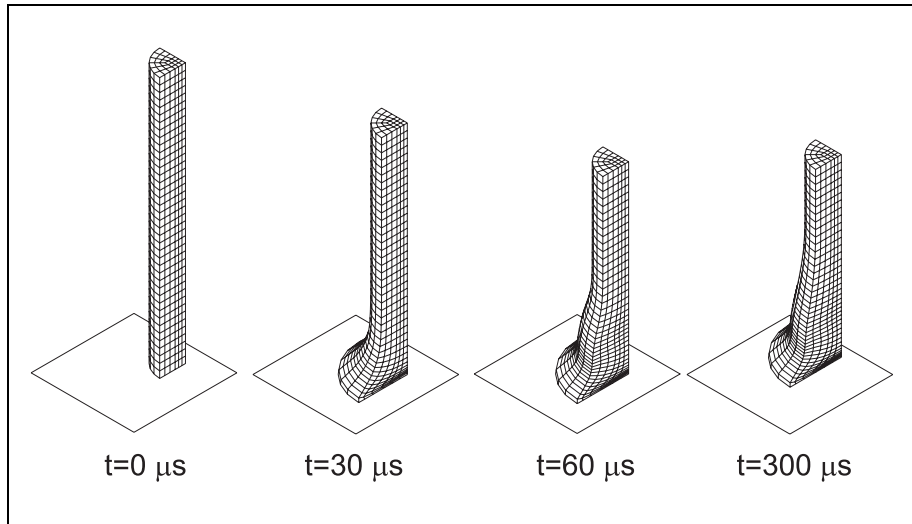– Rebound after the shock wave propagation has slow dynamics properties.

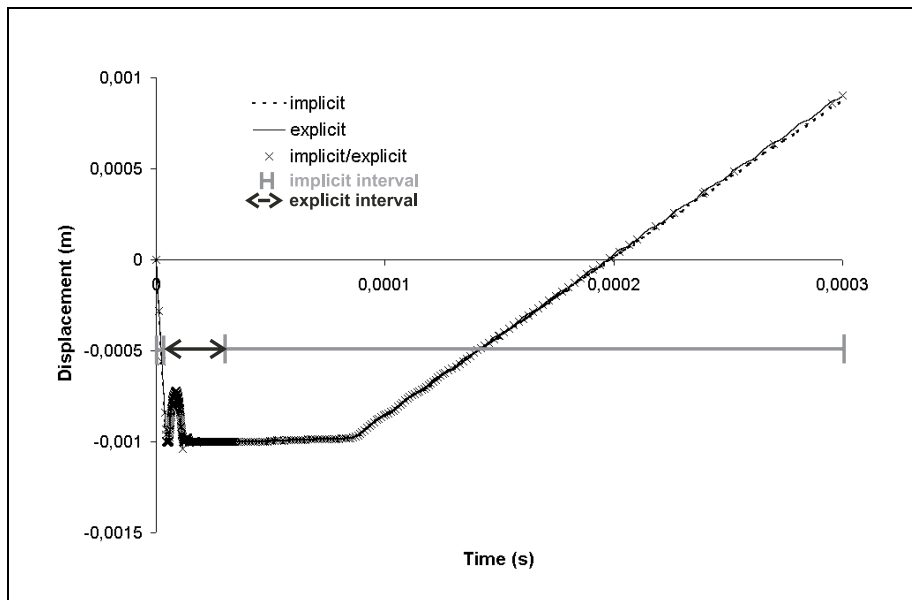**Figure 5.** *Deformation of Taylor bar impact*

Figure 6. *Central point displacement comparison for Taylor bar impact*

α_M	-0.97
α_F	0.01
β	0.9801
γ	1.48
δ	$1e-6$
$PRCU$	$1e-3$

Table 2. *Numerical properties for the implicit scheme*

ρ_b	0.2
α_M	-1.6
β	5.5
γ	3.1
γ_s	0.8

Table 3. *Numerical properties for the explicit scheme*

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. Numerical parameters used for the time integration scheme (section 2) are reported in (table 2 and 3).

Central point displacements obtained with the three methods (full implicit, full explicit and combined implicit/explicit) are compared in figure 6. Zones treated with implicit or explicit scheme when the combined implicit/explicit algorithm is used are also indicated. 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. When algorithm shifts to explicit method, r^* is equal to 6.2 and it is equal to 3 when it comes back to an implicit scheme. Comparing computed displacement, it appears that all the solution are quite similar. Computational costs (CPU) (table 4) confirm the marked advantage of the combined algorithm. All the computations were done on a Digital Compaq XP1000 workstation ($667Mhz$). It is about 30% cheaper than the implicit solution and about 700% cheaper than the explicit solution. When shift to implicit scheme occurs, using the defined initial conditions (section 3.4), with an implicit time step size $\Delta t_{impl} = \mu r^* \Delta t_{expl} = 6\Delta t_{expl}$, does not lead to convergence problem.

Implicit	11.45
Explicit	66.2
Implicit/explicit	8.3

Table 4. *CPU comparison for Taylor bar impact (min.)*

Casing diameter	$c_d = 1.305m$
External blade diameter	$d_e = 1.3m$
Internal blade diameter	$d_i = 0.1m$
Length of section	$L = 0.11m$
Width of section	$l = 0.032m$
Twisting angle	$\alpha_t = 45°$
Density	$\rho = 4450kg/m^3$
Young's modulus	$E = 110E9N/m^2$
Poisson's ratio	$\nu = 0.31$
Initial yield stress	$\sigma_0 = 1000N/mm^2$
Saturated yield stress	$\sigma_\infty = 1300N/mm^2$
Exponential hardening parameter	$h_e = 100$
Hardening parameter	$h = 300N/mm^2$
Rotational velocity	$\Omega_r = 3333.3rpm$

Table 5. *Properties of contact between a blade and a casing*

4.2. *Numerical example 2: contact between a blade and a casing*

A blade, with a twisted rectangular cross section, discretized with 204 elements, is in an uniform rotation in a rigid casing. The rotational velocity is $\Omega_r = 3333.3rmp$ (table 5 and figure 7). The plastic hardening law (σ_v depending on effective plastic strain $\bar{\varepsilon}^{pl}$) is $\sigma_v = \sigma_0 + (\sigma_\infty - \sigma_0)\left(1 - e^{-h_e\bar{\varepsilon}^{pl}}\right) + h\bar{\varepsilon}^{pl}$.

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 and the subsequent deformations of the blade are illustrated on figure (8). The initial position of the center of rotation is coincident with the center of the casing and there is no contact (figure 8: Deformation 1). During the first half revolution, the rotation center is moved $6cm$ in the direction opposite to the initial blade position (figure 8) so that contact interactions between the blade and the casing is generated. These frictional contact interactions lead to bending of the blade with irreversible plastic deformation (figure 8: Deformation 2). During the second half of the first revolution, the center of rotation is brought back to its initial position and the blade, after elastic unloading, remains deformed due to the plastic deformation (figure 8: Deformation 3). Scheme parameters (section 2) are given in (table 2 and 3).

Potential energies (i.e. work done by internal forces) obtained with the three methods (full implicit, full explicit and combined implicit/explicit) are compared in figure 9. Time intervals treated with implicit or explicit scheme when the combined implicit/explicit algorithm is used are indicated. This combined scheme behaves as expected. It computes rotation with no large plastic strain (before first half revolution and after first revolution) with an implicit scheme and interval with important plastic
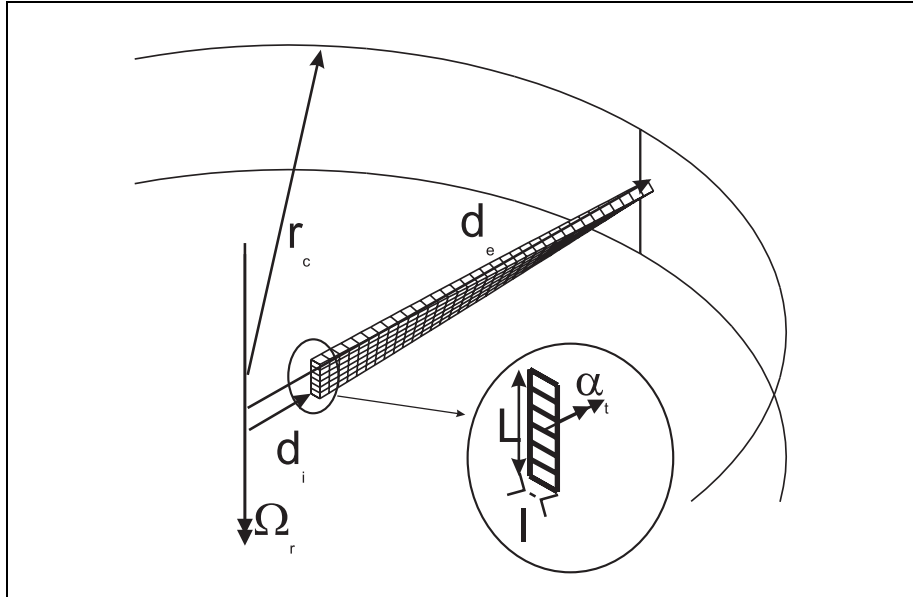
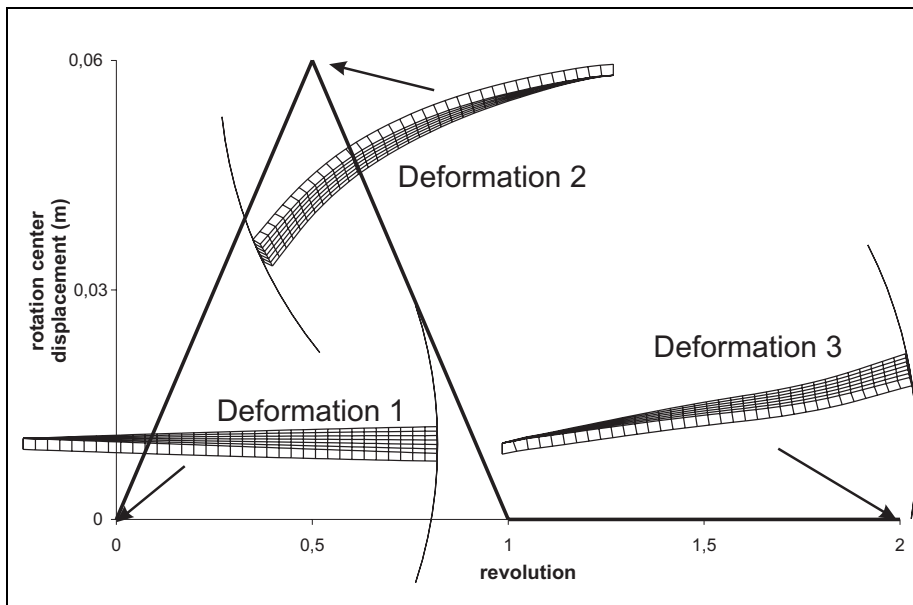**Figure 7.** *Squetch of the contact between a blade and a rigid casing*

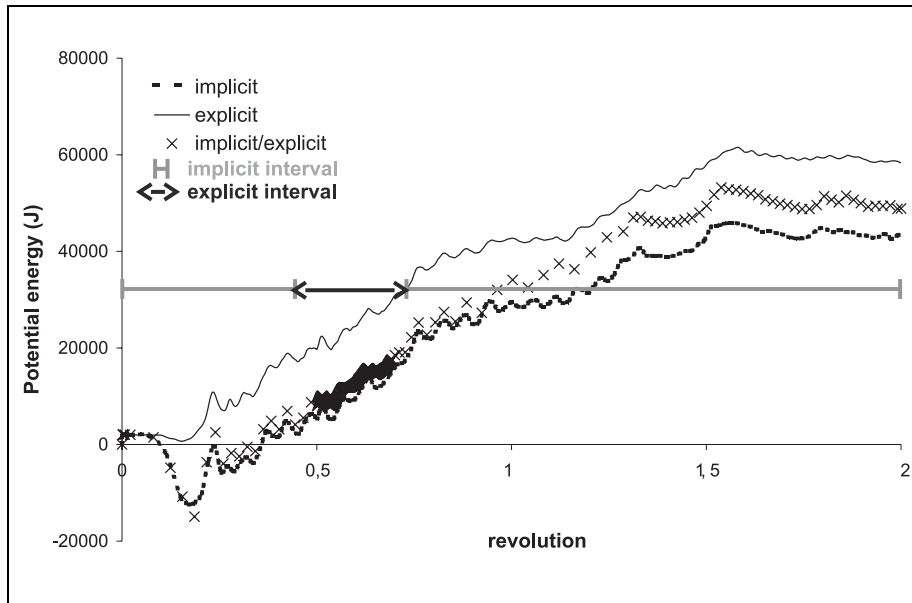**Figure 8.** *Deformation and center of rotation displacement for contact between a blade and a rigid casing*

Figure 9. *Potential energies comparison for contact between a blade and a rigid casing*

Implicit	5.4
Explicit	11
Implicit/explicit	5.1

Table 6. *CPU comparison for contact between a blade and a casing (min.)*

deformation (between first and second half revolution) with an explicit scheme. When algorithm shifts to explicit method, r^* is equal to 8.4 and it is equal to 7.4 when it comes back to an implicit scheme. Comparing full implicit and full explicit computed energies, it appears that both solution are qualitatively identical but are about 25 % different quantitatively. Nevertheless the implicit/explicit solution is bracketed by the two other solutions. It proves that the combined method does not lead to numerical instability. Computational costs (CPU) (table 6) confirm the advantage of the combined algorithm. It is about 8% cheaper than the implicit solution and about 50% cheaper than the explicit solution. Even if r^* is greater than for numerical case 1, the gain is lower (8% for 30%). In fact, implicit method has more divergence problem for the first numerical example than for numerical case 2, which advantages the shift to explicit method. When shift to implicit scheme occurs, using the defined initial conditions (section 3.4), with an implicit time step size $\Delta t_{impl} = \mu r^* \Delta t_{expl} = 14.8 \Delta t_{expl}$, it does not lead to any convergence problem.

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. Those criteria, based on integration error and ratio between CPU of an implicit step and CPU of an explicit step, aim at minimizing the CPU time of the whole simulation. More, they take into account tolerance (on the residual and on the integration error) imposed by the user, the non-linearities (both geometrical, material and contact) evolution and the numerical conditions (number of elements, implicit and explicit algorithm relative performances, ...). Next, stable balanced initial conditions have also been proposed when shifting from an explicit algorithm to an implicit algorithm. These initial conditions avoid divergence problems even if implicit step size is much larger than explicit time step size and does not introduce expensive computation operations. Finally, numerical examples were proposed that confirm the interest of the combined algorithm.

6. References

[ARM 99] ARMERO F., ROMERO I., "Dissipative integration algorithms for nonlinear elasto-dynamics", *ECCM99, Proceedings of the European Conference on Computational Mechanics*, Munich, Germany, 1999, CD-ROM.

[ARM 01] ARMERO F., ROMERO I., "On the formulation of high-frequency dissipative time-stepping algorithms for non)linear dynamics. Part I: low-order methods for two model problems and nonlinear elastodynamics", *Computer Methods in Applied Mechanics and Engineering*, vol. 190, 2001, p. 2603-2649.

[BEL 78] BELYTSCHKO T., MULLEN R., "Stability of explicit-implicit mesh partitions in time integration", *International Journal of Numerical Methods in Engineering*, vol. 12, 1978, p. 1575-1586.

[BEL 83] BELYTSCHKO T., HUGHES T., *Computational methods for transient analysis*, North Holland, 1983.

[BEL 92] BELYTSCHKO T., LU Y., "Stability analysis of elemental explicit-implicit partitions by Fourier methods", *Computer Methods in Applied Mechanics and Engineering*, vol. 95, 1992, p. 87-96.

[BEN 98] BENSON D. J., "Stable time step estimation for multi-material Eulerian hydrocodes", *Computer Methods in Applied Mechanics and Engineering*, vol. 167, 1998, p. 191-205.

[CAS 91] CASSANO A., CARDONA A., "A comparaison between three variable-step algorithms for the integration of the equations of motion in structural dynamics", *Latin American Research*, vol. 21, 1991, p. 187-197.

[CHU 93] CHUNG J., HULBERT J., "A time integration algorithms for structural dynamics with improved numerical dissipations: the generalized-α method", *Journal of Applied Mechanics*, vol. 60, 1993, p. 371-375.

[COM 01] COMBESCURE A., GRAVOUIL A., "A time-space multi-scaled algorithm for transient structural nonlinear problems", *Mécanique & Industrie*, vol. 2, 2001, p. 43-55.

[DAN 98] DANIELS W., "Subcycling first- and second-order generalizations of the trapezoidal rule", *International Journal of Numerical Methods in Engineering*, vol. 42, 1998, p. 1091-1119.

[DUT 98] DUTTA A., RAMAKRISHNAN C., "Accurate computation of design sensitives for structures under transient dynamic loads using time marching scheme", *International Journal of Numerical Methods in Engineering*, vol. 41, 1998, p. 977-999.

[FIN 95] 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", *Journal of Material Processing Technology*, vol. 50, 1995, p. 395-409.

[GEL 95] GELIN J., BOULMANE L., BOISSE P., "Quasi-static implicit and transient explicit analyses of sheet-metal forming using a C° three-nodes shell element", *Journal of Material Processing Technology*, vol. 50, 1995, p. 54-69.

[GER 94] GERADIN M., RIXEN D., *Mechanical vibrations (Theory and applications to structural dynamics)*, John Wiley and Sons Inc, Paris, 1994.

[GER 97] GERADIN M., *Analyse, simulation et conception de systèmes polyarticulés et déployables*, Cours IPSI, Paris, mars 11-13 1997.

[GER 00] GERADIN M., *Flexible multibody dynamics (A finite element approach)*, John Wiley and Sons Inc, 2000.

[GON 96] GONZALEZ O., SIMO J., "On the stability of sympletic and energy-momentum algorithms for non-linear Hamiltonian systems with symmetry", *Computer Methods in Applied Mechanics and Engineering*, vol. 134, 1996, p. 197-222.

[GON 99] GONZALEZ O., "Mechanical systems subject to holonomic constraints: Differential-algebraic formulations and conservative integration", *Physica D*, vol. 132, 1999, p. 165-174.

[GRA 99] GRAILLET D., PONTHOT J.-P., "Efficient implicit schemes for the treatment of the contact between deformable bodies: Applications to shock-absorber devices", *International Journal of Crashworthiness*, vol. 4, num. 3, 1999, p. 173-286.

[GRA 01] GRAVOUIL A., COMBESCURE A., "Multi-time-step explicit-implicit method for non-linear structural dynamics", *International Journal of Numerical Methods in Engineering*, vol. 50, 2001, p. 199-225.

[HOG 96] HOGGE M., PONTHOT J.-P., "Efficient implicit schemes for transient problems in metal forming simulation", *NUPHYMAT'96, Numerical and physical study of material forming processes*, CEMEF-Ecole nationale supèrieure des mines de Paris Sophia-Antipolis, France, June 5-7 1996.

[HUG 87] HUGHES T., *The finite element method*, Prentice Hall, 1987.

[HUL 95] HULBERT G., JANG I., "Automatic time step control algorithms for structural dynamics", *Computer Methods in Applied Mechanics and Engineering*, vol. 126, 1995, p. 155-178.

[HUL 96] HULBERT G., CHUNG J., "Explicit time integration algorithms for structural dynamics with optimal numerical dissipation", *Computer Methods in Applied Mechanics and Engineering*, vol. 137, 1996, p. 175-188.

[JUN 98] JUNG D., YANG D., "Step-wise combined implicit-explicit finite-element simulation of autobody stamping process", *Journal of Material Processing Technology*, vol. 83, 1998, p. 245-260.

[KAN 99] KANE C., REPETTO E., ORTIZ M., MARDSEN J. E., "Finite element analysis of nonsmooth contact", *Computer Methods in Applied Mechanics and Engineering*, vol. 180, 1999, p. 1-26.

[LIU 84] LIU K., BELYTSCHKO T., ZHANG Y., "Implementation and accuracy of mixed-time implicit-explicit methods for structural dynamics", *Computers & Structures*, vol. 19, num. 4, 1984, p. 521-530.

[NAR 99] NARKEERAN N., LOVELL M., "Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure", *Finite Elements in Analysis and Design*, vol. 33, 1999, p. 29-42.

[NOE 00] NOELS L., "Détermination automatique de la taille du pas de temps pour les schémas implicites en dynamique non-linéaire", Travail de fin d'étude, june 2000, Université de Liège, Liège, in French.

[NOE 01] NOELS L., STAINIER L., PONTHOT J., BONINI J., "Automatic time stepping algorithms for implicit numerical simulations of blade/casing interactions", *International Journal of Crashworthiness*, vol. 6, num. 3, 2001, p. 351-361.

[NOE 02] NOELS L., STAINIER L., PONTHOT J., "Self-adapting time integration management in crash-worthiness and sheet metal forming computations", *International Journal of Vehicle Design*, vol. 30, num. 2, 2002, p. 1-48.

[OWE 80] OWEN D., HINTON E., *Finite elements in plasticity*, Pineridge Press, 1980.

[PLE 85] PLESHA M., BELYTSCHKO T., "A constitutive operator splitting method for nonlinear transient analysis", *Computers & Structures*, vol. 20, num. 4, 1985, p. 767-777.

[PON 94] PONTHOT J.-P., HOGGE M., "On relative merits of implicit schemes for transient problems in metal forming simulation", *International Conference on Numerical Methods for Metal Forming in Industry*, vol. 2, Baden-Baden, Germany, Sept 1994, p. 128-148.

[PON 95] PONTHOT J.-P., "Traîtement unifié de la mécanique des milieux continus solides en grandes transformations par la méthode des éléments finis", PhD thesis, Université de Liège, 1995, in French.

[RAM 89] RAMIREZ M., BELYTSCHKO T., "An expert system for setting time steps in dynamics finite elemnt programs", *Engineering with Computers*, vol. 5, 1989, p. 205-219.

[ROM 00] ROMERO I., ARMERO F., "High-frequency dissipative time-stepping algorithms for the dynamics of nonlinear shells", *ECCOMAS 2000, Proceedings of the Congress on Computational Methods in Applied Sciences and Engineering*, Barcelona, Spain, 2000.

[SIM 92] SIMO J., TARNOW N., WONG K., "Exact energy-momentum conserving algorithms and sympletic schemes for nonlinear dynamics", *Computer Methods in Applied Mechanics and Engineering*, vol. 100, 1992, p. 63-116.

[SUN 00] Sun J., Lee K., Lee K., "Comparaison of implicit and explicit finite element methods for dynamic problem", *Journal of Material Processing Technology*, vol. 105, 2000, p. 110-118.

[WU 00] Wu Y., Smolinski P., "A multi-time step integration algorithm for structural dynamics based on the modified trapezoidal rule", *Computer Methods in Applied Mechanics and Engineering*, vol. 187, 2000, p. 641-660.

[YAN 95] 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", *Journal of Material Processing Technology*, vol. 50, 1995, p. 39-53.