# Combined implicit/explicit time integration algorithms for the numerical simulation of sheet metal forming

L. Noels [a,1,*], L. Stainier [a,1], J.-P. Ponthot [a]

[a] *University of Liège, LTAS-Milieux Continus & Thermomécaniques, Chemin des Chevreuils 1- Bât. B52/3, B-4000 Liège, Belgium*

**Abstract**

In order to simulate stamping processes, an explicit method, which is conditionally stable, is generally thought to be the most adapted. Such an algorithm presents the advantage of being non-iterative while, the contact configuration evolves rapidly, and the conditional stability is not a disadvantage since time steps must be small enough anyway for an accurate computation. But during the springback simulation, 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 a method to another, depending on the current dynamic, have been developed. Implicit restarting conditions are also proposed that annihilate numerical oscillations resulting from an explicit calculation.

*Key words:* Implicit, explicit, combined method, dynamics, non-linearities, metal-forming

## 1. Introduction

The choice of a time integration algorithm is an essential criterion to ensure efficiency and robustness of numerical simulations. The 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,

[*] Corresponding author: E-mail l.noels@ulg.ac.be

[1] The authors thank the Belgian National Fund for Scientific Research (F.N.R.S.) for its financial support.

explicit methods use smaller time steps than implicit ones. Explicit methods, avoiding iterations and 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 fewer non-linearities, implicit algorithms allow to work with larger time step size, resulting in more numerical stability and accuracy [1–3]. However, in general, an actual sheet metal forming process has some time intervals governed by high non-linear dynamics (stamping) and others governed by slower non-linear dynamics (springback). Then, one can take advantage from a solution method that combines both families of integration algorithms.

A solution is to integrate 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 process that begins with an implicit scheme and shifts to an explicit one when a problem of convergence appears. No return to an 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 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 a family to another have been developed by the authors in Ref. [7] for impact problems. They depend on an integration error (see Refs. [8,9]) that allows to determine the 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 to implicit, are also defined to avoid loss of stability and convergence. In the present paper, the formulation is enhanced and extended to sheet-metal forming problems.

## 2. Numerical integration of transient problems

### *2.1. Equations of motion*

FEM (Finite Element Method) semi-discretization of the equations of motion of a nonlinear structure leads to the following coupled set of second order nonlinear differential equations (see Refs. [10–12]):

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

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 the 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 presence of contact, plastic deformations and geometrical non-linearities.

### *2.2. Implicit schemes*

The most general scheme for implicit integration of equation (1) is a generalized trapezoidal scheme [10,11,13] 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:

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

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

The discretized equation of motion (1) can be rewritten under the form proposed by Chung and Hulbert [13]:

$$\begin{aligned} R_{n,n+1} &= \tfrac{1-\alpha_M}{1-\alpha_F} M \ddot{x}_{n+1} + \tfrac{\alpha_M}{1-\alpha_F} M \ddot{x}_n + \left(F^{int}_{n+1} - F^{ext}_{n+1}\right) \\ &\quad + \tfrac{\alpha_F}{1-\alpha_F}\left(F^{int}_n - F^{ext}_n\right) = 0 \end{aligned} \tag{4}$$

where $R_{n,n+1}$ is the residual vector of time step $n$ to $n{+}1$. Non-linear equations (2) to (4) can be solved by a Newton-Raphson technique.

### *2.3. Explicit Scheme*

Chung and Hulbert [14] have extended their implicit scheme to an explicit one, taking $\alpha_F = 1$ in equation (4). Its principal advantage is its numerical dissipation property. Time integration is then given by:

$$\ddot{x}_{n+1} = \frac{M^{-1}\left(F^{ext}_n - F^{int}_n\right) - \alpha_M \ddot{x}_n}{1-\alpha_M} \tag{5}$$

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

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

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

$$\Delta t = \gamma_s \Delta t_{crit} = \gamma_s \frac{\Omega_s(\rho_b)}{\omega_{max}} \tag{8}$$

with [14]:

$$\Omega_s(\rho_b) = \sqrt{\frac{12\,(1+\rho_b)^3\,(2-\rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}} \tag{9}$$

In equation (8), $\gamma_s$ is a safety factor ($< 1$) that accounts for the destabilizing effects of non-linearities.

### *2.4. Implicit time step size control*

The implicit time step size control is the one proposed by Géradin [8], extended to highly non-linear problems by Noels *et al.* [9]. This scheme continuously adapts time step size to the evolution of physical modes and keeps time step size constant during long time intervals. The current time step size is estimated on the basis of an integration error.

The integration error $e_{int}$ is deduced from truncated terms of equation (2) and equation (3). This error is of the third order: $O\left(\frac{1}{6}\Delta t^3 \dddot{x}\right) \simeq O\left(\frac{1}{6}\Delta t^2 \Delta \ddot{x}\right)$. To have a problem independent error, it is made non dimensional, using $x_0$ (the initial position vector) and a reference error $\varepsilon$ (see Refs. [8,9] for details). 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:

$$e_{int} = \frac{\Delta t^2}{6} \frac{\sum_{i=1}^{N} \Delta \left\| \ddot{x}_i \right\|}{\varepsilon \left\| x_0 \right\|} \tag{10}$$

Time step size is deduced from the integration error defined in equation (10) and from a tolerance $PRCU$ fixed by the user ($10^{-4}$ is a typical value). The relation to be verified is:

$$e_{int} < PRCU \tag{11}$$

The new time step size $\Delta t_{new}$ to reach a reference integration error (half of the tolerance $PRCU$) is deduced from the current time step size ($\Delta t_{cur}$) and from the current integration error ($e_{int,cur}$), using the following relation developed by Géradin [8]:

$$\left(\frac{\Delta t_{new}}{\Delta t_{cur}}\right)^{\eta} = \frac{PRCU}{2e_{int,cur}} \tag{12}$$

with $\eta \in [2, 3]$ a user specified parameter [8,9].

## 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 actualized for each step, in order to be able to shift from a method to another for non-linear simulation. One shifts to an explicit method if:

$$\mu \Delta t_{impl} < r^{*} \Delta t_{expl} \tag{13}$$

where $\Delta t_{expl}$ is evaluated from equation (8). The factor $\mu$ is taken greater than unity (typical values are discussed in section 4.) to avoid shifting from a method to another too frequently. This methodology allows to take into account the number of degrees of freedom, the algorithm's current efficiency, the residual tolerance required and the non-linearities evolution.

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

While the method used is an implicit one, the explicit time step size can always be easily computed from equation (8). When the current method is explicit, the implicit time step size, which correctly integrates the problem, is not directly accessible. Using developments of section (2.4.), nodal acceleration variations can provide us with this implicit time step size. Using equation (12), one sees that acceleration variations are proportional to $\Delta t^{\eta}$. Inverting equation (10), the implicit time step size becomes (with $N$ the number of nodes):

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

Therefore the explicit to implicit shift criterion is similar to equation (13). It yields:

$$\Delta t_{impl} > \mu r^{*} \Delta t_{expl} \tag{15}$$

with $\Delta t_{expl}$ the current explicit time step size.

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

Classical explicit schemes such as the central difference method [10] are well known to generate oscillatory (though stable) solutions. Two solutions are provided here 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 generalized-$\alpha$ explicit scheme. Indeed, when equation (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 2.3.) 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 on next paragraph) explicit steps is computed. This solution results in $x^{expl}_{n+r^{*}+r^{**}}$, $\dot{x}^{expl}_{n+r^{*}+r^{**}}$ and in $\ddot{x}^{expl}_{n+r^{*}+r^{**}}$. From the Gauss point

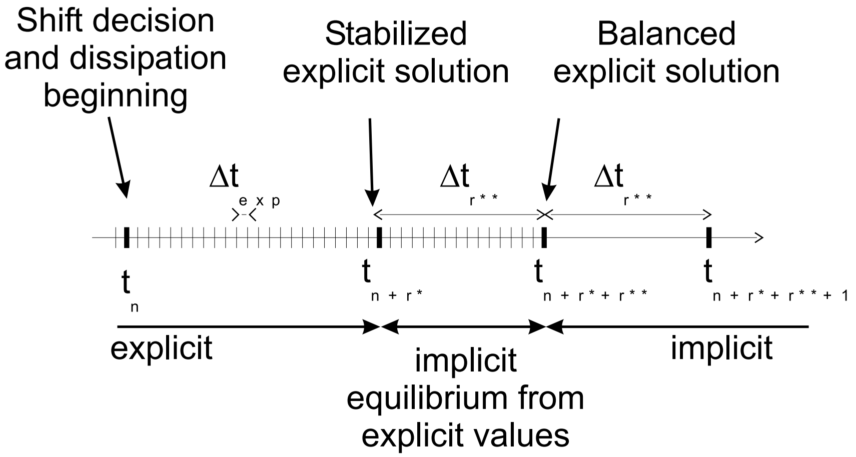
Shift decision and dissipation beginning
Stabilized explicit solution
Balanced explicit solution
Δt exp
Δt r**
Δt r**
t n
t n+r*
t n+r*+r**
t n+r*+r**+1
explicit
implicit equilibrium from explicit values
implicit

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

values and the nodal values obtained at time $t_{n+r^*}$ (where numerical oscillations have been reduced), we compute an implicit step of size $t_{n+r^*+r^{**}}$-$t_{n+r^*}$ that uses $x^{expl}_{n+r^*+r^{**}}$, $\dot{x}^{expl}_{n+r^*+r^{**}}$ and $\ddot{x}^{expl}_{n+r^*+r^{**}}$ as predictor values. This procedure proved to be very effective in order to restart an implicit solution based on an explicit unbalanced solution. The methodology is illustrated on Fig. (1). This balanced solution is reached considering an implicit time step size equal to $\Delta t_{r^{**}} = r^{**}\Delta t_{expl}$. 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 equal to $\mu r^*$ or limited to a user defined parameter (e.g. 100).

## 4. Numerical example

The numerical example consists in a sheet metal forming process of an "S"-shaped rail (thickness $e = 0.92mm$) (see Ref. [15] for more detail on the benchmark). A description of the sheet and of the die is given in Fig. (2). Properties of the material are: density $\rho = 8900kg/m^3$, Young's modulus $E = 206000N/mm^2$, Poisson's ratio $\nu = 0.31$, initial yield stress $\sigma_0 = 158N/mm^2$ and hardening parameter $h = 1000N/mm^2$. There are 1800 elements (30 in length, 30 in width and 2 through the thickness). 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 per element.

The simulation compares the solutions obtained with the proposed combined implicit-explicit algorithm with full implicit and full explicit algorithm. For the full implicit and the combined implicit-explicit schemes, the stamping process is simulated in a time of $5ms$ (a very short time that does favor the explicit scheme). It consists of a doped stamping process with the true density of the material but with a shorter time of stamping. The dies are removed in a total time of $1s$ to simulate springback of the sheet. However, for the

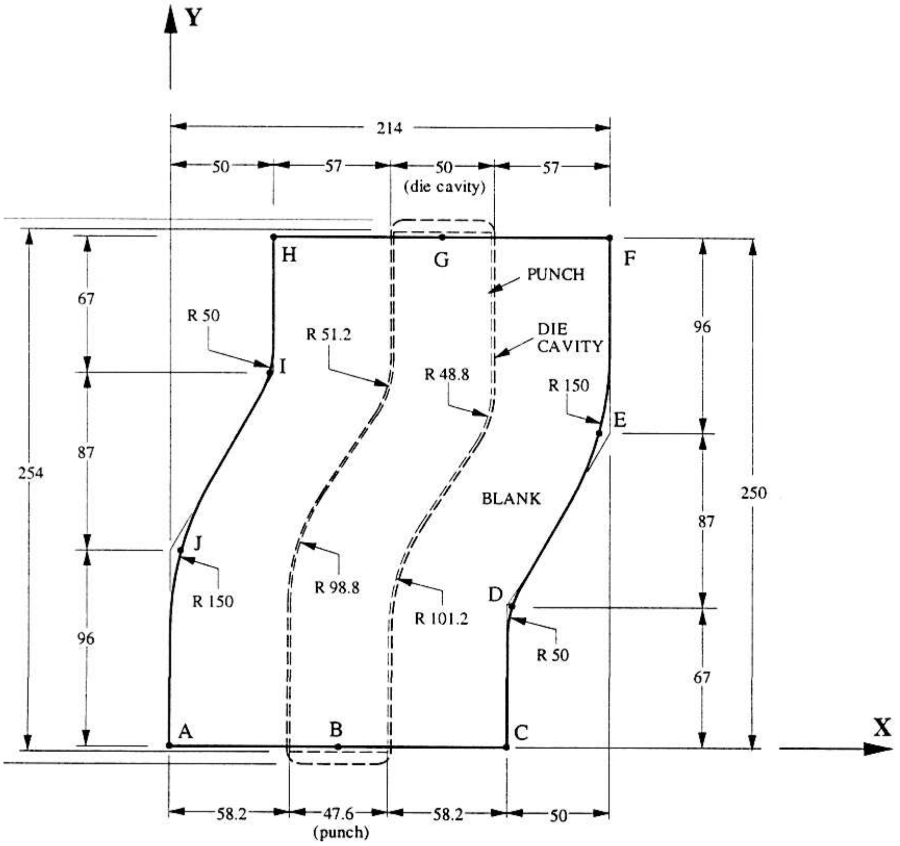

Fig. 2. Description of the "S"-shaped sheet forming.

full explicit process, these times lead to a very expensive computational time (several weeks). Therefore, the stamping process and the springback process are doped (stamping process is simulated in $0.5ms$ and springback process is simulated in $1.5ms$). The parameters $\eta$ of equation (14) and $\mu$ of equations (13, 15) are respectively taken equal to 2.5 and 1.5. Decreasing $\eta$ or $\mu$ will result in more shifts from a method to another and thus will degrade the efficiency of the algorithm. Since a return to an implicit scheme leads to some iterations (section 3.3.), computation costs can increase. Numerical parameters used for the time integration scheme are for the implicit scheme (section 2.2.): $\alpha_M = -0.97$, $\alpha_F = 0.01$, $\beta = 0.9801$, $\gamma = 1.48$, tolerance on the Newton-Raphson residual$= 10^{-7}$ and $PRCU = 10^{-4}$. For the explicit simulation of the combined implicit-explicit scheme, the parameters are (section 2.3.): $\rho_b = 0.2$, $\alpha_M = -1.6$, $\beta = 5.5$, $\gamma = 3.1$ and $\gamma_s = 0.9$. For the full explicit scheme, $\rho_b$ is taken equal to zero ($\alpha_M = -1$, $\beta = 2.5$, $\gamma = 2.5$) to dissipate the whole kinetic energy during the springback process. The frictional contact simulation uses the penalty method with a normal penalty of $10^6$ and a tangent penalty of $10^5$. The Coulomb friction coefficient is equal to 0.2.

During the stamping process (from time $= 0s$ to time $= 5ms$), the combined scheme shifts 5 times from an implicit scheme to an explicit scheme,

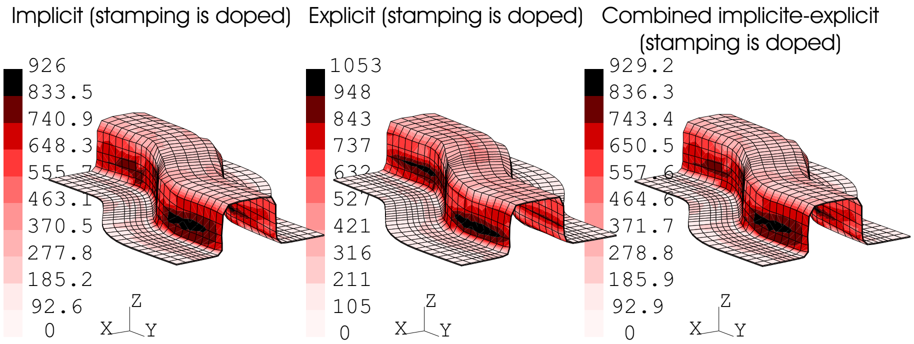

Fig. 3. Deformation and von Mises stress ($N/mm^2$) after stamping of the "S"-shaped sheet.

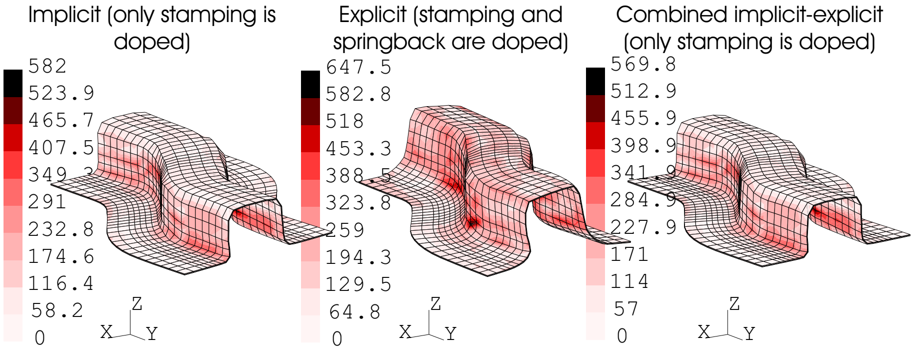

Fig. 4. Deformation and von Mises stress ($N/mm^2$) after springback of the "S"-shaped sheet.

when problems of convergence appear, before returning to the implicit scheme. During the $5ms$ of the stamping process, there are about $3ms$ computed with an implicit scheme and $2ms$ with an explicit scheme. The solution obtained at the end of the stamping is illustrated in Fig. (3). During the springback the implicit scheme is selected until the end of the springback process ($1s$). The solution obtained after springback is illustrated in Fig. (4). It appears that a simulation of the springback with a (doped) explicit method leads to a totally different solution (although the kinetic energy is dissipated thanks to the dissipative explicit algorithm) and that the combined implicit/explicit method gives the same solution as the full implicit method. The CPU needed for the combined implicit-explicit simulation is the lower one (16.7 hours), while the full implicit computation requires 23.3 hours and the full (doped) explicit computation requires 29.4 hours.

## 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. Finally, a numerical example of sheet metal forming was proposed that confirms the interest of the combined algorithm. In this example, the stamping was processed with an explicit scheme when divergence problems appear. On the other hand, the springback process was performed with an implicit scheme that has a dynamic balanced solution.

## References

[1] D. Yang, D. Jung, I. Song, D. Yoo, J. Lee, Comparative investigation into implicit, explicit, and iterative implicit/explicit schemes for the simulation of sheet-metal forming processes, *Journal of Material Processing Technology* **50** (1995) 39–53.

[2] J. Gelin, L. Boulmane, P. Boisse, Quasi-static implicit and transient explicit analyses of sheet-metal forming using a C° three-nodes shell element, *Journal of Material Processing Technology* **50** (1995) 54–69.

[3] J. Sun, K. Lee, K. Lee, Comparaison of implicit and explicit finite element methods for dynamic problem, *Journal of Material Processing Technology* **105** (2000) 110–118.

[4] D. Jung, D. Yang, Step-wise combined implicit-explicit finite-element simulation of autobody stamping process, *Journal of Material Processing Technology* **83** (1998) 245–260.

[5] M. Finn, P. Galbraith, L. Wu, J. Hallquist, L. Lum, T.-L. Lin, Use of a coupled explicit-implicit solver for calculating spring-back in automotive body panels, *Journal of Material Processing Technology* **50** (1995) 395–409.

[6] N. Narkeeran, M. Lovell, Predicting springback in sheet metal forming: an explicit to implicit sequential solution procedure, *Finite Elements in Analysis and Design* **33** (1999) 29–42.

[7] L. Noels, L. Stainier, J.-P. Ponthot, J. Bonini, Combined implicit-explicit algorithms for non-linear structural dynamics, *Revue Européenne des Elements-Finis, Dynamique Rapide* **11** (2002) 565–591.

[8] M. Géradin, A. Cardona, *Flexible multibody dynamics (A finite element approach)*, John Wiley and Sons Inc, 2000.

[9] L. Noels, L. Stainier, J. Ponthot, Self-adapting time integration management in crash-worthiness and sheet metal forming computations, *International Journal of Vehicle Design* **30** (2) (2002) 1–48.

[10] T. Belytschko, T. Hughes, *Computational methods for transient analysis*, North Holland, 1983.

[11] T. Hughes, *The finite element method*, Prentice Hall, 1987.

[12] J.-P. Ponthot, M. Hogge, 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, pp. 128–148.

[13] J. Chung, J. Hulbert, A time integration algorithms for structural dynamics with improved numerical dissipations: the generalized-$\alpha$ method, *Journal of Applied Mechanics* **60** (1993) 371–375.

[14] G. Hulbert, J. Chung, Explicit time integration algorithms for structural dynamics with optimal numerical dissipation, *Computer Methods in Applied Mechanics and Engineering* **137** (1996) 175–188.

[15] NUMISHEET96, 3rd international conference: Numerical simulation of 3D sheet metal forming processes - verification of simulations with experiments, Dearborn, Michigan, USA, 1996.