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

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

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

Abstract

In order to solve highly non-linear dynamic problems, an explicit method, which is conditionally stable, is the most adapted. Such an algorithm presents the advantage of being non-iterative, and the conditional stability is not a disadvantage since time steps must be small enough for an accurate computation. But for more linear dynamics, an implicit method, which is iterative, presents the advantage of more stability and of unconditional stability. Therefore, time step size can be increased and the method becomes cheaper than an explicit one. Typical sheet metal forming processes are governed by high non-linearities during the stamping process and by quasi-linear dynamics during the spring-back process. 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 shift from a method to another, depending on the dynamics, have been developed. Those criteria are based on CPU costs and integration error evaluations. Implicit restarting conditions are also proposed that annihilate numerical oscillations resulting of explicit calculation. The combined method then allows computation of problems such as the sheet metal forming into a "S" shaped rail. In such a problem, an implicit solution is not stable or is expensive during the stamping process. On the other hand, during the springback simulation, the contact configuration does not evolve rapidly, and an explicit method with a small time step is much more expensive than an implicit method that can increase the time step. The combined implicit-explicit algorithms are then the solution that minimize the CPU cost.

Keywords: implicit, explicit, combined method, dynamics, non-linearities, metal-forming

1 Introduction

Many industrial problems of today need to be simulated with non-linear models. 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 need to be solved iteratively on 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 convergence problems, are therefore generally used for highly non-linear problems with many

*This work was supported by the Belgian National Fund for Scientific Research (F.N.R.S.).
†Corresponding author. E-mail: l.stainier@ulg.ac.be.

degrees of freedom, for which iterations are very expensive and convergence problems are frequent [14]. On the other hand, for slower dynamics problems with less non-linearities, implicit algorithms allow to work with larger time step size, resulting in more numerical stability and accuracy [5, 13, 14]. A 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 and they were all developed for sheet metal forming analysis. Jung and Yang [9] 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.* [4] and by Narkeeran and Lovell [10], 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. This method has been generalized in this work and automatic criteria that decide to shift from a family to another have been developed. They depend on an integration error [6, 11] that allows to determine 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 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, a numerical simulation of an "S" shaped rail sheet forming will be exposed to validate the methodologies.

2 Numerical integration of transient problems

2.1 *Equations of motion*

FEM (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 [2, 7, 12]:

$$R = M\ddot{x} + F^{int}(x, \dot{x}) - F^{ext}(x, \dot{x}) = 0 \tag{2.1}$$

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.

2.2 *Implicit schemes*

The most general scheme for implicit integration of equation (2.1) is a generalized trapezoidal scheme [2, 3, 7] 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

$$\dot{x}_{n+1} = \dot{x}_n + (1-\gamma)\,\Delta t\ddot{x}_n + \gamma\Delta t\ddot{x}_{n+1} \tag{2.2}$$

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{x}_n + \beta \Delta t^2 \ddot{x}_{n+1} \tag{2.3}$$

The discretized motion equation (2.1) can be rewritten under the form proposed by Chung and Hulbert [3]:

$$\begin{aligned} R_{n,n+1} &= \tfrac{1-\alpha_M}{1-\alpha_F} M \ddot{x}_{n+1} + \tfrac{\alpha_M}{1-\alpha_F} M \ddot{x}_n + \left(F_{n+1}^{int} - F_{n+1}^{ext}\right) \\ &+ \tfrac{\alpha_F}{1-\alpha_F} \left(F_n^{int} - F_n^{ext}\right) = 0 \end{aligned} \tag{2.4}$$

where $R_{n,n+1}$ is the residual vector of time step n to $n+1$.
Iterative solution of the nonlinear equation (2.4) requires the writing of the Hessian matrix of the system, i.e.

$$S = \left[\frac{1}{\beta \Delta t^2} \left(\frac{1-\alpha_M}{1-\alpha_F}\right) M + \frac{\gamma}{\beta \Delta t} C_T + K_T\right] \tag{2.5}$$

where K_T, C_T are respectively the tangent stiffness and damping matrices. Using equations (2.2) to (2.5) and a Newton-Raphson technique, the iterative solution of the problem can be written as:

$$S \Delta x = -R \tag{2.6}$$

2.3 Explicit Scheme

Chung and Hulbert have extended their implicit scheme to an explicit one, taking $\alpha_F = 1$ in equation (2.4) [8]. Its principal advantage is its numerical dissipation property. Time integration is then:

$$\ddot{x}_{n+1} = \frac{M^{-1}\left(F_n^{ext} - F_n^{int}\right) - \alpha_M \ddot{x}_n}{1 - \alpha_M} \tag{2.7}$$

$$\dot{x}_{n+1} = \dot{x}_n + \Delta t \left[(1-\gamma) \ddot{x}_n + \gamma \ddot{x}_{n+1}\right] \tag{2.8}$$

$$x_{n+1} = x_n + \Delta t \dot{x}_n + \Delta t^2 \left[\left(\frac{1}{2} - \beta\right) \ddot{x}_n + \beta \ddot{x}_{n+1}\right] \tag{2.9}$$

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

$$\Delta t = \gamma_s \Delta t_{crit} = \gamma_s \frac{\Omega_s(\rho_b)}{\omega_{max}} \tag{2.10}$$

with

$$\Omega_s(\rho_b) = \sqrt{\frac{12(1+\rho_b)^3(2-\rho_b)}{10 + 15\rho_b - \rho_b^2 + \rho_b^3 - \rho_b^4}} \tag{2.11}$$

In equation (2.10), γ_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 [6], extended to highly non-linear problems by Noels *et al.* [11]. 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 equation (2.2) and equation (2.3). 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 ε [6, 11]. 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)[11]:

$$e_{int} = \frac{\Delta t^2}{6} \frac{\sum_{i=1}^{N} \Delta \|\ddot{x}_i\|}{\varepsilon \|x_0\|} \tag{2.12}$$

Time step size is deduced from the integration error defined in equation (2.12) and from a tolerance $PRCU$ fixed by the user. The relation to be verified is:

$$e_{int} < PRCU \tag{2.13}$$

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 by Geradin [6]:

$$\left(\frac{\Delta t_{new}}{\Delta t_{cur}}\right)^{\eta} = \frac{PRCU}{2e_{int,cur}} \tag{2.14}$$

with $\eta \in [2, 3]$ a user specified parameter [6, 11]. The time step size management, based on equations (2.13) and (2.14), is the one developed by Noels *et al.* [11].

3 Shifts from an algorithm family to another

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 averaged for each step, in order to be able to shift from a method to another for non-linear simulation. Shift to explicit method occurs if:

$$\mu \Delta t_{impl} < r^* \Delta t_{expl} \tag{3.15}$$

where Δt_{expl} is evaluated with equation (2.10). The factor μ is taken greater than unity (typical values disscussed 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 algorithms 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 (2.10). 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*), nodal acceleration variations can provide us with this implicit time step size. Using equation (2.14), acceleration variation is proportional to Δt^{η}. Inverting equation (2.12) the implicit time step size is (with N the number of nodes):

$$\Delta t_{impl} = \left[6 \frac{\frac{PRCU}{2} \varepsilon \|x_0\| (\Delta t_{expl})^{\eta-2}}{\sum_{i=1}^{N} \Delta \|\ddot{x}_i\|}\right]^{\frac{1}{\eta}} \tag{3.16}$$

Therefore the explicit to implicit shift criterion is similar to equation (3.15). It yields:

$$\Delta t_{impl} > \mu r^* \Delta t_{expl} \tag{3.17}$$

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

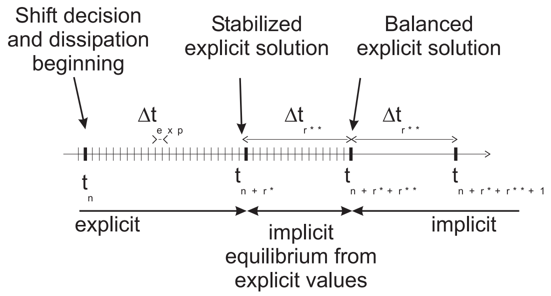

Figure 1: Transition scheme from an explicit scheme to an implicit one.

3.3 Initial conditions when shifting to an implicit scheme

Classical explicit schemes such as the central difference method [2] 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-α explicit scheme. Indeed, when equation (3.17) 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 a user parameter). Thus, numerical oscillations have been greatly reduced at time t_{n+r^*} (Figure 1).
The second step in the algorithm is to determine a balanced configuration at time $t_{n+r^*+r^{**}}$. Therefore, we act in two stage. 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 $\ddot{x}^{expl}_{n+r^*+r^{**}}$, 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 illustrated on Figure (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 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 $\Delta t = \mu r^* \Delta t_{expl}$

4 Numerical examples

The numerical example consists in sheet forming of an "S"-shaped rail [1]. A description of the bench and of the die is given on Figure (2). Properties of the material are reported on Table (1). The stamping process is simulated in a time of $5ms$. It consists of a doped stamping process with the true density but with a shorter time of stamping. The dies are removed in a total time of $1s$ to simulate springback of the process.
The simulation used the proposed combined implicit-explicit algorithm. There is 1800 elements (30 in length, 30 in width and 2 on 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. Moreover,the

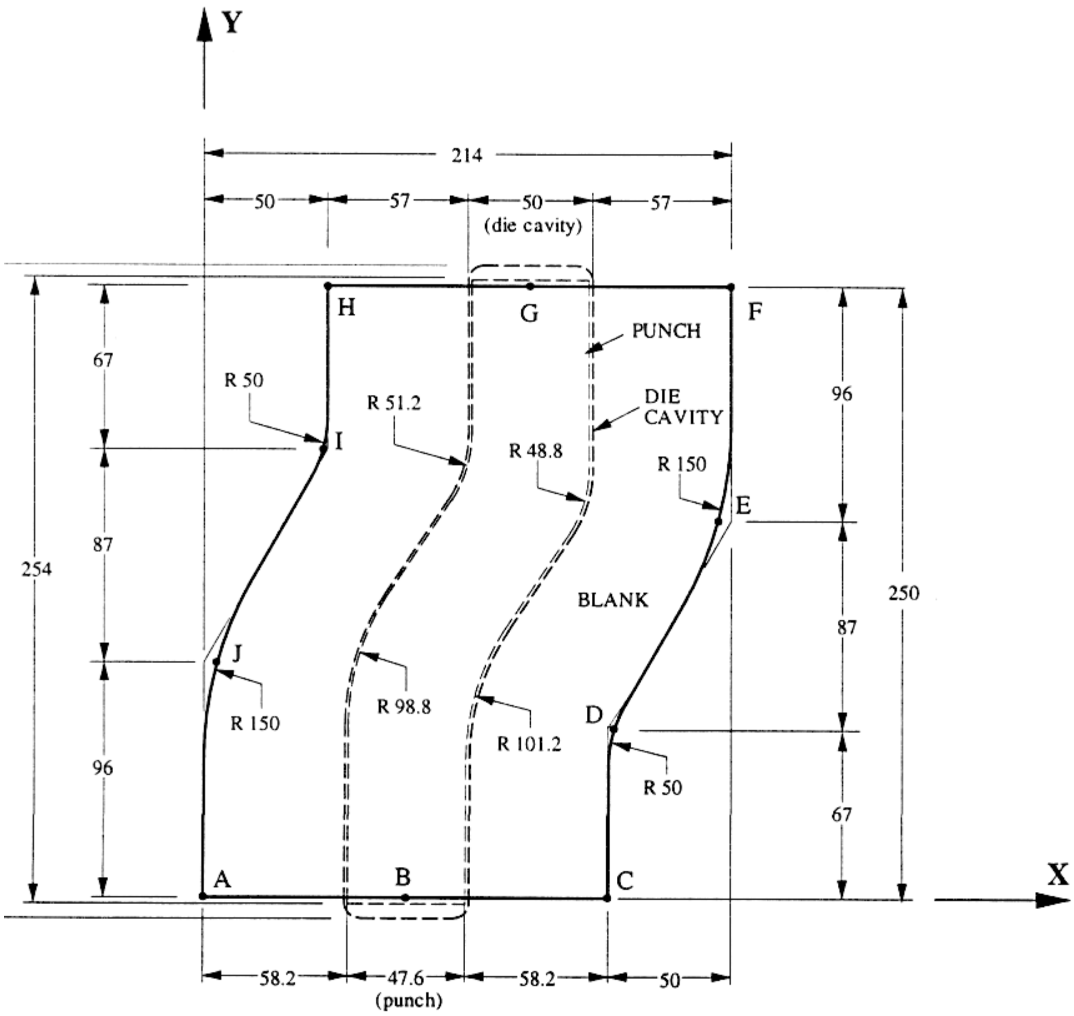

Figure 2: Description of the "S"-shaped sheet forming.

Thickness	$e = 0.92mm$
Density	$\rho = 8900kg/m^3$
Young's modulus	$E = 206000N/mm^2$
Poisson's ratio	$\nu = 0.31$
Initial yield stress	$\sigma_0 = 158N/mm^2$
Hardening parameter	$h = 1000N/mm^2$

Table 1: Properties of the sheet.

α_M	-0.97
α_F	0.01
β	0.9801
γ	1.48
δ	$10e^{-6}$
$PRCU$	$10e^{-4}$

Table 2: Numerical properties for the implicit scheme.

ρ_b	0.2
α_M	-1.6
β	5.5
γ	3.1
γ_s	0.9

Table 3: Numerical properties for the explicit scheme.

parameters η of equation (3.16) and μ of equations (3.15, 3.17) are respectively taken equal to 2.5 and 1.5. Decreasing η or μ 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 (section 2) are reported in (table 2 and 3). The frictional contact simulation uses the penalty method with a normal penalty of $10e^6$ and a tangent penalty of $10e^5$. The Coulomb 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, 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 on Figure (3). During the springback the implicit scheme is selected until the end of the springback process ($1s$). The advantage of the implicit scheme during the springback process is the accuracy obtained [14, 5, 13]. The solution obtained after springback is illustrated on Figure (4).

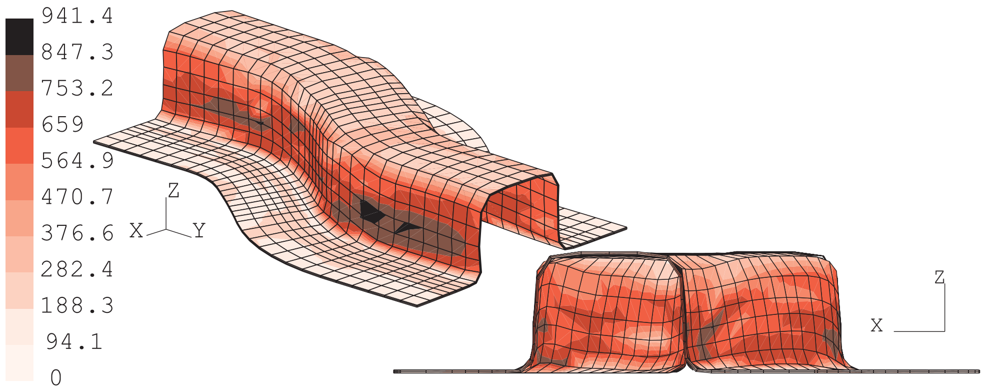

Figure 3: Deformation and von Mises stress (N/mm^2) after stamping of the "S"-shaped sheet.

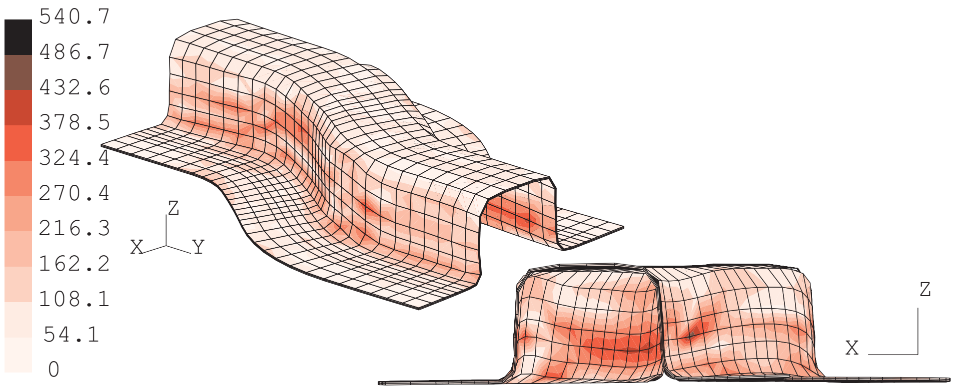

Figure 4: Deformation and von Mises stress (N/mm^2) after springback of the "S"-shaped sheet
.

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] NUMISHEET 96. 3rd international conference: numerical simulation of 3-d sheet metal forming processes - verification of simulations with experiments. Technical report, NUMISHHET 96, Dearborn, Michigan, USA, 1996.

[2] T. Belytschko and T.J.R. Hughes. *Computational methods for transient analysis.* North Holland, 1983.

[3] J. Chung and J.M. Hulbert. A time integration algorithms for structural dynamics with improved numerical dissipations: the generalized-α method. *Journal of Applied Mechanics* **60** (1993) 371–375.

[4] M.J. Finn, P.C. Galbraith, L. Wu, J.O. Hallquist, L. Lum, and 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.

[5] J.C. Gelin, L. Boulmane, and 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.

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

[7] T.J.R. Hughes. *The finite element method.* Prentice Hall, 1987.

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

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

[10] N. Narkeeran and 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.

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

[12] J.-P. Ponthot and 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*, volume 2, Baden-Baden, Germany, Sept 1994. 128–148.

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

[14] D.Y. Yang, D.W. Jung, I.S. Song, D.J. Yoo, and J.H. 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.