Int J Mater Form (2016) 9:71–84
DOI 10.1007/s12289-014-1208-5

ORIGINAL RESEARCH

An accurate time integration scheme for arbitrary rotation motion: application to metal forming simulation

Koffi Kpodzo · Lionel Fourment · Patrice Lasne · Pierre Montmitonnet

Received: 19 May 2014 / Accepted: 9 December 2014 / Published online: 30 December 2014

Abstract Within the frame of implicit velocity based formulations with solid elements, usual time integration schemes often turn out unsatisfactory when the movement has large rotations, especially in metal forming applications such as ring rolling or cross-wedge rolling. These rotations generally require using a much higher order integration scheme with inherent difficulties in implementing such schemes. For pure rotation motions, it is possible to use a low order integration scheme by rewriting the motion equations in the cylindrical frame that is supported by the rotation axis. Accordingly, a first order scheme is sufficient to accurately integrate the movement but it is restricted to specific problems. In the more general case, it is possible to derive parts of the domain where rotations are predominant along with the governing rotation axis from the velocity field gradient. The motion equations are then rewritten in the resulting local cylindrical frame. Performances of this first order scheme are first evaluated and highlighted over simple analytical problems, before being applied to the finite element simulation of the torsion test, and then to more complex metal forming problems involving large rotations. The accuracy and efficiency of this scheme is so numerically demonstrated.

Keywords Time integration scheme · Large rotations · Implicit formulation · Velocity formulation · Finite elements · Metal forming · Torsion test · Cross-wedge rolling

K. Kpodzo (✉) · L. Fourment · P. Montmitonnet
Mines ParisTech, CEMEF -Centre for Material Forming, CNRS UMR 7635, CS 10 207, 06904 Sophia Antipolis Cedex, France
e-mail: koffi.kpodzo@mines-paristech.fr

P. Lasne
Transvalor, 694 Avenue Maurice Donat Parc de Haute Technologie, 06250 Mougins, France

Introduction

Time integration of large rotation motions has been long an important scientific issue. The applications cover many fields such as navigation in aerospace, computer graphics, solid mechanics.

Within the field of solid mechanics, most of the studies encountered in the literature mainly relate to rigid body dynamics problems [1–5] or to structural analysis problems [6–12]. For such problems, the rotation is generally regarded as a separate degree of freedom, deriving from an angular velocity through a differential equation. The main challenge lies in developing an accurate but computationally economic scheme to integrate the strongly nonlinear differential equation under consideration. The key point of the algorithms proposed in the literature often relies on a combination of two techniques: suitable parameterization of the rotation and use of an accurate time discretization scheme. The suitable parameterization of the rotation plays a major role since it allows a reduction of the required parameters for representing the rotation. Several parameterizations can be distinguished: The Euler-Rodrigues parameterization [4, 8] the rotation vector [3, 13], Cayley's parameterization [11], and quaternions [14]. In [13], Argyris discusses the choice of different parameterizations while the combined influences of parameterizations and integration schemes are discussed by Zupan and Saje [14].

For bulk forming simulation problems, especially when an implicit velocity based formulation, with solid finite elements (tetrahedra) is employed, the problem is slightly different. The rotation is not described as a separate degree of freedom; it is included within the velocity field. All the kinematics of the domain is thus dictated by the velocity field through the evolution equation. A robust and accurate integration scheme is therefore required to describe precisely the trajectory of each material point of the domain. Another major difficulty comes from the implicit formulation employed, since it allows relatively large time steps unlike explicit formulation where time steps are usually smaller. For many processes such as cogging, drawing or strip rolling, the use of classical first order schemes such as Euler one step schemes provides good accuracy. However, for other processes where the domain undergoes large rotations, such low order schemes require very small time step sizes to ensure an acceptable accuracy, which results in excessive computational time. Therefore, higher order schemes like Runge–Kutta [15–17] or sub-stepping [16] have been introduced. In [15], Traore uses a second-order Runge–Kutta scheme to handle the large rotations of the ring rolling process, while a fourth-order Runge–Kutta is regarded as necessary by Dialami [16] to properly trace Lagrangian particles in Friction Stir Welding (FSW). In this particular case, a backward Euler scheme with sub-stepping and a back and forth error compensation and correction is not sufficient to provide the required accuracy. The main shortcoming of high order or multistep schemes regards the calculation of intermediate configurations, which can be either complex or computationally expensive. In [18] for the application of ring rolling process where, the domain undergoes constant rotation movement around a fixed and predetermined axis, the authors show that the use of a cylindrical integration scheme provides better volume conservation and a good accuracy of the results. The main idea of their method consists in writing the motion equation into the cylindrical frame that is supported by the rotation axis, such a frame is quite appropriate to describe the rotational kinematics. An Euler one step scheme is then used to integrate the equation.

In this paper, the idea of [18] is resumed and extended to any kind of motion and to any kind of problem. The first important point of the proposed scheme is the automatic identification of effective rotation zones by means of criteria concerning the velocity gradient. These criteria enable switching from a standard time integration scheme (one-step Euler and RK2) to the proposed one. A second key point of the proposed scheme is the automatic detection of the local rotation axes necessary for the construction of the local cylindrical frames.

After a brief description of the finite elements model used for the studied metal forming problems, the cylindrical scheme is first presented for a constant rotation around the (Oz) axis before being extended to any arbitrary rotation motion, which requires both identifying large rotation zones and the governing rotation axis. Resulting modifications on contact equations are also discussed. In the last section, the cylindrical scheme is evaluated over analytical problems and metal forming processes.

Numerical model for metal forming simulation

This section focuses on a brief description of the mechanical equations and their finite element discretization for metal forming simulations. Equations are written within the context of a simple rigid-viscoplastic material (neither elasticity nor thermal dilation) as described in [19], although more complex constitutive models taking into account the thermo-elastic part of the deformation can also be considered [20]. A velocity (and pressure) based formulation has been employed, a suitable formulation that is widely used for modelling such material flows. For a more complete description of scientific approaches of the finite elements formulations used for metal forming problems, the reader may refer to [21].

Constitutive equation and integral formulation

The mass conservation of the domain undergoing incompressible deformation is expressed as a function of velocity **v** by:

$$div(\mathbf{v}) = 0 \tag{1}$$

The rate form of the mechanical equations and boundary conditions at any time t are expressed by the virtual power principle, as a function of the velocity **v** and p pressure fields satisfying the contact equations (which are presented later on), for any virtual velocity **v*** and pressure p* fields:

$$\begin{cases} \int\limits_{\Omega} \mathbf{s} : \dot{\boldsymbol{\varepsilon}}^* d\Omega - \int\limits_{\Omega} p\,div\left(\mathbf{v}^*\right) d\Omega + \int\limits_{\partial\Omega_c} \boldsymbol{\tau}.\mathbf{v}^* dS = 0 \\ \int\limits_{\Omega} p^* div(\mathbf{v}) d\Omega = 0 \\ +\ contact \end{cases} \tag{2}$$

Where Ω and $\partial\Omega_c$ respectively denote the domain occupied by the body and its contact surface at time t. $\dot{\boldsymbol{\varepsilon}}$ is the strain rate tensor. The deviatoric part **s** of the stress tensor is given by the Norton-Hoff constitutive law:

$$\mathbf{s} = 2K\left(\sqrt{3}\overline{\varepsilon}^{\cdot}\right)^{m-1}\dot{\boldsymbol{\varepsilon}} \tag{3}$$

Where K is the material consistency $\overline{\varepsilon} = \sqrt{{}^{2}/{}_{3}\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}}$ is the equivalent strain rate, m is the strain rate sensitivity coefficient. $\boldsymbol{\tau}$ is the tangential shear stress resulting from the friction. For example, with the Norton friction model, $\boldsymbol{\tau}$ is expresed by:

$$\boldsymbol{\tau} = -\alpha_f K \left\| \Delta \mathbf{v}_g \right\|^{q-1} \Delta \mathbf{v}_g \tag{4}$$

Where α_f is a friction coefficient, q is the sensitivity to the sliding velocity and $\Delta\mathbf{v}_g$ is the relative sliding velocity between the tool and the part.

Finite elements discretization

The problem is discretized using the P1+/P1 compatible mixed finite elements [22, 23]. The pressure p is linearly interpolated on each tetrahedron, while the velocity $\mathbf{v}$ is interpolated using linear shape functions and further enriched with a bubble function:

$$\mathbf{v}_h(\mathbf{x}) = \sum_{k=1}^{nbnoe} \mathbf{v}_k N_k(\mathbf{x}) + \sum_{e=1}^{nbelt} \mathbf{v}_e b_e(\mathbf{x}) \;\;, \;\; \forall \mathbf{x} \in \Omega p_h(\mathbf{x})$$
$$= \sum_{k=1}^{nbnoe} p_k N_k(\mathbf{x}) \;, \;\; \forall \mathbf{x} \in \Omega \qquad (5)$$

Where $\mathbf{v}_k$ and p_k are the nodal velocity and pressure values. *nbnoe* and *nbelt* respectively represent the number of nodes and elements. N_k is the linear shape function at a node k and $\overline{\mathbf{v}}_e$ is the enrichment of the velocity field interpolated by the bubble function b_e and computed at the center of each tetrahedron.

Time evolution of the domain

Within the updated lagrangian framework, with our velocity based formulation, the evolution of the domain over the time is obtained by integrating evolution Eq. (6) at each time t of the simulation; unlike displacement based formulations where the configuration updating is straightforward and given by the displacement fields provided by the finite elements calculations.

$$\frac{d\mathbf{x}^t}{dt} = \mathbf{v}^t(\mathbf{x}^t, t) \equiv \mathbf{v}^t \qquad (6)$$

Where $\mathbf{x}^t$ and $\mathbf{v}^t$ respectively denote the position and velocity at time t. Notice that the above equation is the fundamental equation around which revolves the study.

A widely used integration scheme is the simple explicit or implicit Euler one step scheme (Fig. 1a). For this scheme, the updated $\mathbf{x}^{t+\Delta t}$ configuration is given by:

$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \mathbf{v}^t \Delta t \;\; or \;\; \mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \mathbf{v}^{t+\Delta t} \Delta t \qquad (7)$$

When the rotational component of the movement is moderate, it is often observed that second order Runge–Kutta scheme (Fig. 1b) provides considerable improvements over the Euler scheme [15]. ($\mathbf{x}^t$, $\mathbf{v}^t$) being the current configuration, this scheme can be summarized by:

- from ($\mathbf{x}^t$, $\mathbf{v}^t$), compute the middle configuration: $\mathbf{x}^{t+\Delta t/2} = \mathbf{x}^t + \mathbf{v}^t \frac{\Delta t}{2}$
- on the middle configuration ($\mathbf{x}^{t+\Delta t/2}$), compute a new FEM solution: $\mathbf{v}^{t+\Delta t/2}$
- from ($\mathbf{x}^t$, $\mathbf{v}^{t+\Delta t/2}$), compute the updated configuration: $\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \mathbf{v}^{t+\Delta t/2} \Delta t$

For a pure rotation motion, Fig. 1 illustrates the time integration error inherent in the Euler scheme, along with the improvement provided by the Runge–Kutta scheme. In the Euler scheme, the velocity field (which is normal to the rotation axis) is assumed to remain constant during the time step. The integrated point at time $t+\Delta t$ consequently moves in the tangent direction and leaves the exact curve (see also Fig. 2b). The resulting error is proportional to Δt. The second order Runge–Kutta is significantly more accurate. The error is second order, then proportional to $(\Delta t)^2$. However, it doesn't allow perfect integration of a pure rotation motion either. A fourth order Runge–Kutta scheme would then be required, as utilized in [16] where it is shown to be necessary to accurately integrate FSW particles motions when the time steps cannot be sufficiently reduced.

Contact treatment

Forming tools are considered to be rigid. Their surfaces are discretized by triangular facets. The signed distance between any point $\mathbf{M}^t$ of the part and the surface of the tool is defined by:

$$\delta^t = (\mathbf{M}^t \mathbf{P}^t).\mathbf{n}^t \qquad (8)$$

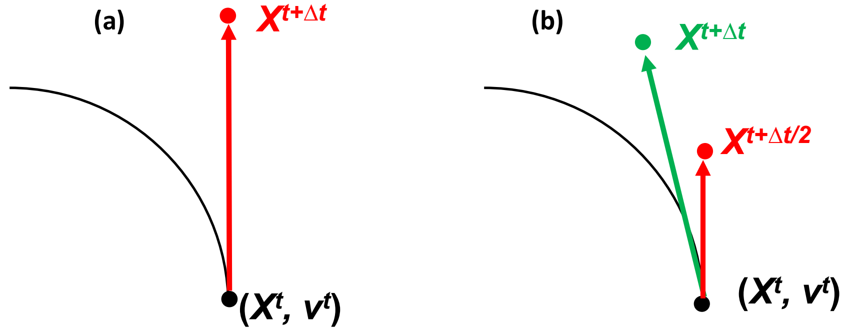


Fig. 1 Illustration of (**a**) first-order Euler and (**b**) second-order Runge–Kutta time integration schemes

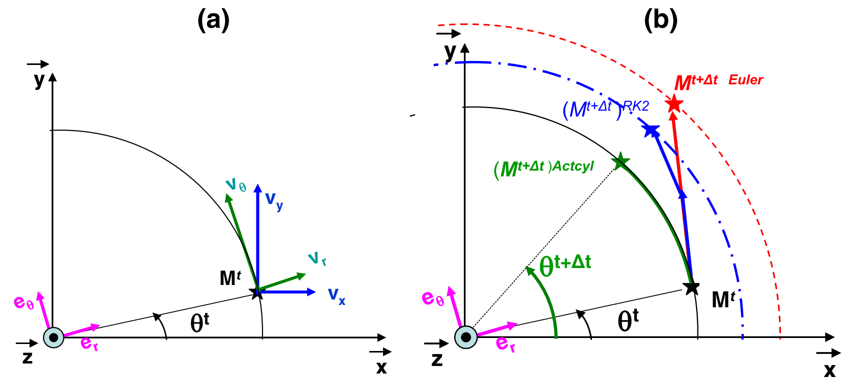


Fig. 2 **a** velocity decomposition in a cylindrical frame **b** material point trajectory according to the different time integration schemes

Where $\mathbf{n}^t$ is the surface normal vector at point $\mathbf{P}^t$ of the tool surface; it is the closest point of the tool from the point $\mathbf{M}^t$. $\delta^t>0$ if the point is outside of the tool, $\delta^t=0$ if it is on the tool surface, $\delta^t<0$ if it has penetrated the tool.

A nodal contact (node to facet) formulation [24] is employed. For any node n, the contact condition is prescribed at the end of the time increment:

$$\delta^{t+\Delta t} = \left(\mathbf{M}^{t+\Delta t}\mathbf{P}^{t+\Delta t}\right).\mathbf{n}^{t+\Delta t}\geq 0 \tag{9}$$

The Eq. (9) is linearized by assuming that the tool can be locally approximated by its tangential plane during the time step. So, $\mathbf{n}^{t+\Delta t}\approx\mathbf{n}^t$, which results into:

$$-\frac{\delta^{t+\Delta t}}{\Delta t}\approx\left(\frac{\mathbf{M}^t\mathbf{M}^{t+\Delta t}}{\Delta t}-\mathbf{v}^t_{tool}\right).\mathbf{n}^t-\frac{\delta^t}{\Delta t} = h(\mathbf{v}_n)\leq 0 \tag{10}$$

Where $\mathbf{v}^t_{tool}$ is the tool velocity, $h(\mathbf{v}_n)$ is a contact function. The discrete contact condition Eq. (10) is enforced by a penalty formulation, which leads to the following contributions for the gradient and Hessian of the discretized equations:

$$\begin{aligned}\mathbf{R}_n^{contact}(\mathbf{v}_n) &= \rho[h(\mathbf{v}_n)]^+ S_n\frac{\partial h(\mathbf{v}_n)}{\partial \mathbf{v}_n},\ \forall n\in\partial\Omega_c\\ \mathbf{H}_n^{contact}(\mathbf{v}_n) &= \rho[h(\mathbf{v}_n)]^+ S_n\frac{\partial h(\mathbf{v}_n)}{\partial \mathbf{v}_n}\otimes S_n\frac{\partial h(\mathbf{v}_n)}{\partial \mathbf{v}_n^T},\ \forall n\in\partial\Omega_c\end{aligned} \tag{11}$$

Where ρ is the penalty coefficient and S_n is a surface associated to the node n, $[\bullet]^+$ denotes the positive part of $(\bullet)$.

The set of non-linear discretized equations is solved by a Newton–Raphson algorithm, within which underlying linear systems of equations are iteratively solved by preconditioned conjugate residual that is fully compatible with parallel computations [23]. For large deformation problems, the updated Lagrangian formulation results into large mesh distortions that require frequent and regular remeshings [22].

Cylindrical time integration scheme

Case of rotation around (Oz) axis

For a material point $\mathbf{M}^t$ of a body that is spinning around the (Oz) axis, the curvilinear trajectory is essentially driven by the tangential component of the velocity (v^t_θ) which is given by the rotation velocity (angular velocity). The decomposition of the velocity in the corresponding cylindrical frame provides the v^t_θ component and enables a more accurate integration of the rotational motion (see Fig. 2b). The *cylindrical scheme* consists in three steps:

- Transformation into the cylindrical frame
- Time integration of motion equations in the cylindrical frame
- Inverse transformation into the original Cartesian frame

The coordinates $\mathbf{x}^t$ and the velocity $\mathbf{v}^t$ at a time t are mapped into the cylindrical basis $(\mathbf{e}_r,\mathbf{e}_\theta,\mathbf{e}_z)$ built around the (Oz) axis. For any node $\mathbf{M}^t$, its (r^t,θ^t,z^t) coordinates are expressed as follow:

$$(\mathbf{M}^t)_{Cyl}\left\{\begin{aligned} r^t &= \left[\left(x^t-x^t_{axis}\right)^2+\left(y^t-y^t_{axis}\right)^2\right]^{1/2}\\ \theta^t &= \arctan\left(\frac{y^t-y^t_{axis}}{x^t-x^t_{axis}}\right)\\ z^t &\end{aligned}\right. \tag{12}$$

Where $\mathbf{x}^t_{axis}=(x^t_{axis},y^t_{axis},z^t_{axis})$ is a point belonging to the rotation axis. The three components of the velocity v^t_r, v^t_θ and v^t_z are given by the following relations:

$$(\mathbf{v}^t)_{Cyl}\begin{cases} v_r^t = v_x^t\cos(\theta^t) + v_y^t\sin(\theta^t) \\ v_\theta^t = \dfrac{v_y^t\cos(\theta^t) - v_x^t\sin(\theta^t)}{r^t} \\ v_z^t \end{cases} \tag{13}$$

Using the first-order Euler scheme in the cylindrical frame, the updated cylindrical coordinates $(r^{t+\Delta t}, \theta^{t+\Delta t}, z^{t+\Delta t})$ (Fig. 2b) are written as:

$$\left(\mathbf{M}^{t+\Delta t}\right)_{Cyl}\begin{cases} r^{t+\Delta t} = r^t + v_r^t\Delta t \\ \theta^{t+\Delta t} = \theta^t + v_\theta^t\Delta t \\ z^{t+\Delta t} = z^t + v_z^t\Delta t \end{cases} \tag{14}$$

In other words, during the time increment the velocity field is assumed to remain constant in the cylindrical frame rather than in the Cartesian frame (7). The updated Cartesian coordinates are then obtained by performing an inverse transformation of the updated cylindrical coordinates into the original Cartesian frame:

$$\mathbf{M}^{t+\Delta t}\begin{cases} x^{t+\Delta t} = r^{t+\Delta t}cos\left(\theta^{t+\Delta t}\right) \\ y^{t+\Delta t} = r^{t+\Delta t}sin\left(\theta^{t+\Delta t}\right) \\ z^{t+\Delta t} \end{cases} \tag{15}$$

Generalization to any arbitrary rotation motion

In practice, the rotation axis can be in any direction, and for many processes such as cross wedge-rolling or screw forming, this axis is moving all along the process. Moreover, there are many problems, such as bending, within which only a part of the domain is actually rotating. Furthermore, in the most general case, there is a combination of rotation and translation so that such integration scheme is necessary only when and where rotation should be regarded as dominant. Therefore, in order to generalize the *cylindrical scheme* to any kind of motion, two important features have been introduced: an automatic detection of the rotation axis and the identification of the actually rotating zones.

- Detection of the rotation axis

The rotation axis can be derived from the rotation velocity vector $\boldsymbol{\omega}^t$ $(\omega_x^t, \omega_y^t, \omega_z^t)$ which itself derives from the skew-symmetric part of the velocity gradient $\boldsymbol{\Omega}(\mathbf{v}^t)$ as proposed in [2–5] and [7, 8, 13, 25] where a similar tensor-vector representation of the rotation vector is employed:

$$\boldsymbol{\Omega}(\mathbf{v}^t) = \frac{1}{2}\left[\nabla\mathbf{v}^t - (\nabla\mathbf{v}^t)^T\right] = \begin{bmatrix} 0 & \omega_z^t & -\omega_y^t \\ -\omega_z^t & 0 & \omega_x^t \\ \omega_y^t & -\omega_x^t & 0 \end{bmatrix} \tag{16}$$

Using a P1 finite elements interpolation for the velocity (at this level, the enrichment provided by the bubble interpolation in order to satisfy the compatibility condition between the velocity and pressure interpolations should be neglected), the gradient $\nabla\mathbf{v}^t$ and consequently the angular velocity vector $\boldsymbol{\omega}^t$ are discontinuous. They are only defined at the Gauss points of the elements. Nodal values $\boldsymbol{\omega}_k^t$ are then obtained by averaging elementary values over nodal patches P_k (the set of elements containing node k) presented in Fig. 3:

$$\boldsymbol{\omega}_k^t = \frac{1}{\mathrm{d}}\sum_{e=1}^{d}\boldsymbol{\omega}_e^t \quad , \quad \mathrm{d} = card(P_k) \tag{17}$$

- Identification of the effective rotating zones

To account for possible local rotations (only a zone of the domain is rotating), it is necessary to evaluate the intensity of the angular velocity. If it is not sufficient, then translation is dominant and the standard one-step Euler scheme should be used. The selected criterion consists in evaluating the variations of the computed rotation velocity (16) and (17) within a patch of elements. Considering a patch P_k centred on a node k (see Fig. 3), this criterion is twofold:

(a) The intensity of the averaged rotation velocity $\|\boldsymbol{\omega}_k^t\|$ at a node k (17) must be closed enough to that of its surrounding elements (16):

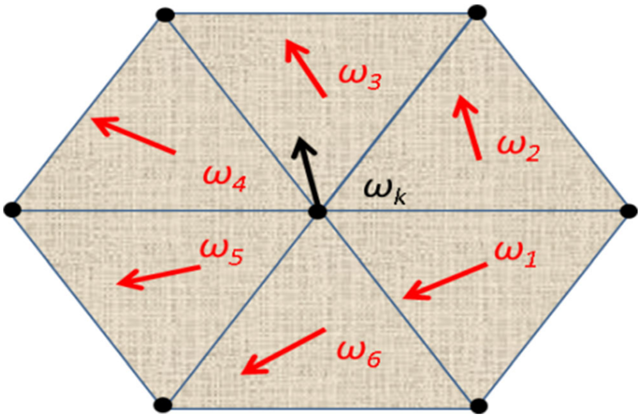


Fig. 3 Nodal patch centered on a node k

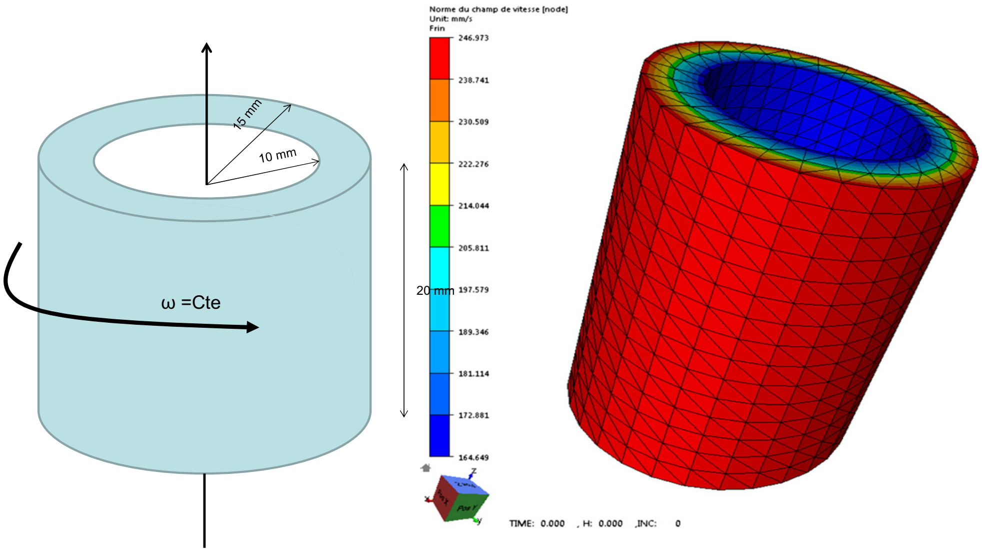


Fig. 4 Rigid body rotation: simulation set up

$$0<\lambda_1\le\min\left(\frac{\|\boldsymbol{\omega}_k^t\|}{\|\boldsymbol{\omega}_e^t\|},\frac{\|\boldsymbol{\omega}_e^t\|}{\|\boldsymbol{\omega}_k^t\|}\right)\le 1 \quad \forall e\in P_k \tag{18}$$

(b) The orientation of the average rotation velocity $\boldsymbol{\omega}_k^t$ at a node k (17) must be closed enough to that of its surrounding elements (16):

$$0<\lambda_2\le\left(\frac{\boldsymbol{\omega}_k^t.\boldsymbol{\omega}_e^t}{\|\boldsymbol{\omega}_k^t.\boldsymbol{\omega}_e^t\|}\right)\le 1 \quad \forall e\in P_k \tag{19}$$

Where λ_1 and λ_2 are parameters used to check the criteria. In practice, λ_1 and λ_2 must be close to 1.

- Updating procedure

Once the effective rotating zones and the rotation axis are known, the updating procedure presented in the case of a rotation around (Oz) axis can be utilized after performing a change of reference frame such that the rotation axis coincides with (Oz). The three basis vectors $(\mathbf{u}_0^t,\mathbf{v}_0^t,\mathbf{w}_0^t)$ of the new local frame are expressed as follow:

$$\begin{cases}\mathbf{w}_0^t=\dfrac{\boldsymbol{\omega}^t}{\|\boldsymbol{\omega}^t\|}\\ \mathbf{v}_0^t=\dfrac{\boldsymbol{\omega}^t\times\mathbf{v}^t}{\|\boldsymbol{\omega}^t\times\mathbf{v}^t\|}\\ \mathbf{u}_0^t=\mathbf{v}_0^t\times\mathbf{w}_0^t\end{cases} \tag{20}$$

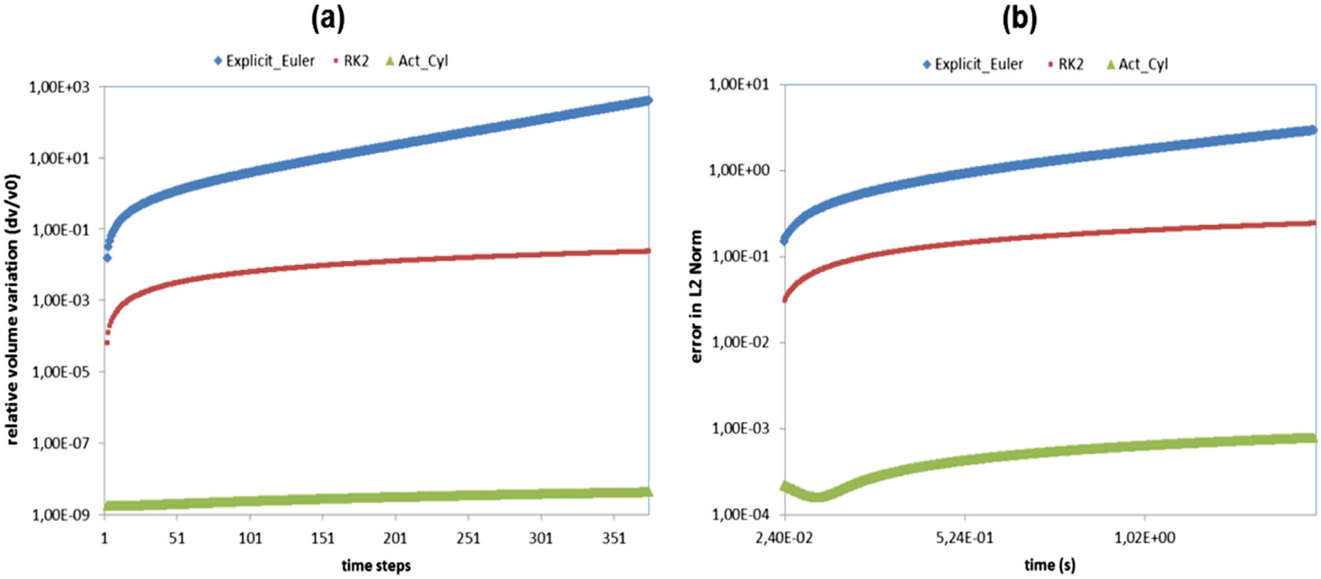


Fig. 5 **a** Relative volume variation and **b** position error for rigid body rotation for few increments

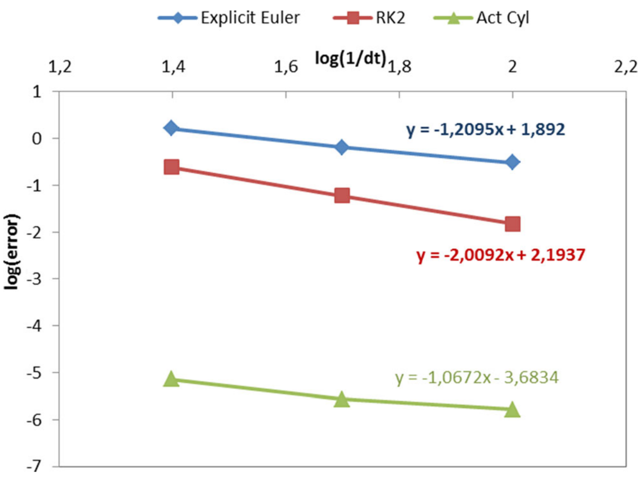


Fig. 6 Convergence curve for the different time integration schemes

The resulting transformation matrix is then given by:

$$\mathbf{P} = \begin{pmatrix} u^t_{0x} & v^t_{0x} & w^t_{0x} \\ u^t_{0y} & v^t_{0y} & w^t_{0y} \\ u^t_{0z} & v^t_{0z} & w^t_{0z} \end{pmatrix} \qquad (21)$$

To fully define the projection of a point $\mathbf{M}^t$ into the local frame, the position of a point $\mathbf{O}^t$ of the rotation axis is required. For a uniform rigid body rotation around a fixed axis, $\mathbf{O}^t\mathbf{M}^t$ satisfies the following relation: $\mathbf{v}^t = \boldsymbol{\omega}^t \times \mathbf{O}^t\mathbf{M}^t$. This vector equation has no unique solution, therefore the geometrical condition $\mathbf{O}^t\mathbf{M}^t . \boldsymbol{\omega}^t = 0$ is added ($\mathbf{O}^t$ belongs to the same plane as $\mathbf{M}^t$ and is perpendicular to the rotation axis, as for a uniform plane rotation). Then, the solution of the vector equation is given by the following relation:

$$\mathbf{O}^t\mathbf{M}^t = -\frac{\boldsymbol{\omega}^t \times \mathbf{v}^t}{\|\boldsymbol{\omega}^t\|^2} \qquad (22)$$

It should be noticed that when the local material movement is not a pure rotation then the instantaneous rotation axis is not strictly defined. This is not a real problem because in practical applications this “rotation axis” moves slowly, and because this assumption is only used for building the time integration scheme.

Using the defined mapping, a position vector $\mathbf{O}^t\mathbf{M}^t$ in the global frame is transformed into $\mathbf{o}^t\mathbf{m}^t$ in the local frame. The updated vector $\mathbf{o}^{t+\Delta t}\mathbf{m}^{t+\Delta t}$ is then computed by the *cylindrical scheme* presented in section 3.1. The updated vector $\mathbf{O}^{t+\Delta t}\mathbf{M}^{t+\Delta t}$ in the global frame is then deduced from $\mathbf{o}^{t+\Delta t}\mathbf{m}^{t+\Delta t}$ by an inverse mapping.

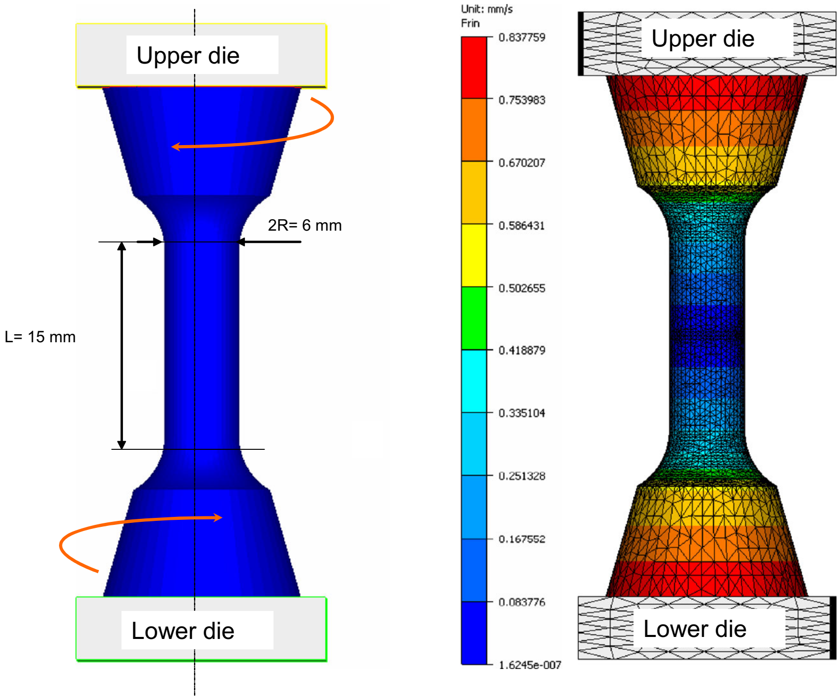
Modified contact equations

Because contact equations are enforced at $t+\Delta t$ (9–10), values of contact condition (10), gradient and hessian (11) depend on the utilized time integration scheme. With the one-step Euler scheme, $\frac{\mathbf{M}^t\mathbf{M}^{t+\Delta t}}{\Delta t} = \mathbf{v}^t$ in Eq. (10) so a simple expression is obtained for the derivative terms of Eq (11):

$$\frac{\partial h(\mathbf{v}_n)}{\partial \mathbf{v}_n} = \mathbf{n}^t \qquad (23)$$

With the *cylindrical scheme*, $\frac{\mathbf{M}^t\mathbf{M}^{t+\Delta t}}{\Delta t}$ cannot be approximated by $\mathbf{v}^t$. In fact, $\mathbf{M}^t\mathbf{M}^{t+\Delta t}$ depends on terms calculated in the cylindrical frame and involved in the updating procedure.

Fig. 7 Hot torsion simulation set up



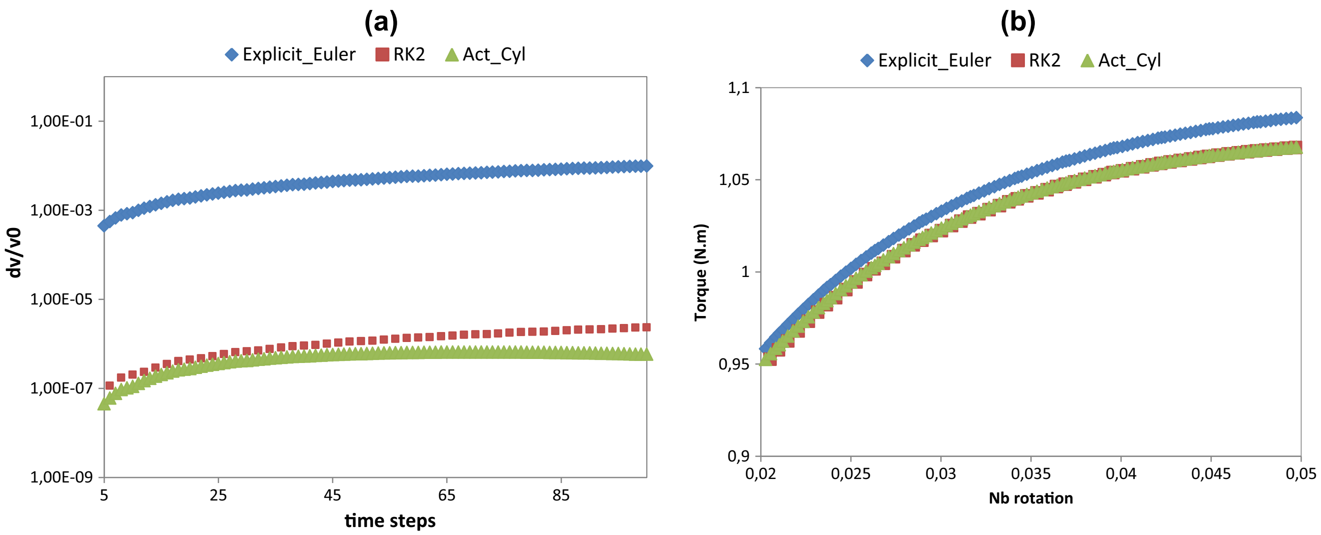


Fig. 8 **a** Relative volume variation and **b** torque evolution for the hot torsion test

So, considering the mapping of $\mathbf{M}^t\mathbf{M}^{t+\Delta t}$ into $\mathbf{m}^t\mathbf{m}^{t+\Delta t}$ and neglecting the dependency of matrix $\boldsymbol{P}$ with respect to the velocity field, the following relations are obtained:

$$\frac{\partial h(\mathbf{v}_n)}{\partial \mathbf{v}_n} = \frac{1}{\Delta t}\left(\mathbf{P}\frac{\partial \mathbf{m}^t\mathbf{m}^{t+\Delta t}}{\partial \mathbf{v}_n^*}\mathbf{P}^{-1}\right).\mathbf{n}^t \tag{24}$$

Where $\mathbf{v}_n^*$ denotes the mapping of vector $\mathbf{v}_n$ into the local basis. $\frac{\partial \mathbf{m}^t\mathbf{m}^{t+\Delta t}}{\partial \mathbf{v}_n^{\rightarrow}}$ is calculated as follows :

$$\frac{\partial \mathbf{m}^t\mathbf{m}^{t+\Delta t}}{\partial \mathbf{v}_n^*}\begin{cases} \frac{\partial\left(r^{*t+\Delta t}cos\left(\theta^{*t+\Delta t}\right)-r^{*t}cos\left(\theta^{*t}\right)\right)}{\partial \mathbf{v}_n^*} = \frac{\partial r^{*t+\Delta t}}{\partial \mathbf{v}_n^*}cos\left(\theta^{*t+\Delta t}\right) + r^{*t+\Delta t}\frac{\partial cos\left(\theta^{*t+\Delta t}\right)}{\partial \mathbf{v}_n^*} \\ \frac{\partial\left(r^{*t+\Delta t}\sin\left(\theta^{*t+\Delta t}\right)-r^{*t}\sin\left(\theta^{*t}\right)\right)}{\partial \mathbf{v}_n^*} = \frac{\partial r^{*t+\Delta t}}{\partial \mathbf{v}_n^*}\sin\left(\theta^{*t+\Delta t}\right) + r^{*t+\Delta t}\frac{\partial \sin\left(\theta^{*t+\Delta t}\right)}{\partial \mathbf{v}_n^*} \\ \frac{\partial\left(z^{*t+\Delta t}-z^{*t}\right)}{\partial \mathbf{v}_n^*} = \frac{\partial z^{*t+\Delta t}}{\partial \mathbf{v}_n^*} \end{cases} \tag{25}$$

Where $(r^{*t},\theta^{*t},z^{*t})$ and $(r^{*t+\Delta t},\theta^{*t+\Delta t},z^{*t+\Delta t})$ are respectively the cylindrical coordinates of $\mathbf{m}^t$ and $\mathbf{m}^{t+\Delta t}$. The partial derivatives of (25) are then evaluated considering the cylindrical updating in the local basis (Eq. (14)) and the decomposition of the velocity (13):

$$\frac{\partial r^{*t+\Delta t}}{\partial \mathbf{v}_n^*}\begin{cases} \frac{\partial r^{*t+\Delta t}}{\partial v_{x,n}^*} = cos\left(\theta^{*t}\right)\Delta t \\ \frac{\partial r^{*t+\Delta t}}{\partial v_{y,n}^*} = sin\left(\theta^{*t}\right)\Delta t \\ \frac{\partial r^{*t+\Delta t}}{\partial v_{z,n}^*} = 0 \end{cases}, \quad \frac{\partial \theta^{*t+\Delta t}}{\partial \mathbf{v}_n^*}\begin{cases} \frac{\partial \theta^{*t+\Delta t}}{\partial v_{x,n}^*} = -\frac{1}{r^{*t}}\sin\left(\theta^{*t}\right)\Delta t \\ \frac{\partial \theta^{*t+\Delta t}}{\partial v_{y,n}^*} = \frac{1}{r^{*t}}\cos\left(\theta^{*t}\right)\Delta t \\ \frac{\partial \theta^{*t+\Delta t}}{\partial v_{z,n}^*} = 0 \end{cases}, \quad \frac{\partial z^{*t+\Delta t}}{\partial \mathbf{v}_n^*}\begin{cases} \frac{\partial z^{*t+\Delta t}}{\partial v_{x,n}^*} = 0 \\ \frac{\partial z^{*t+\Delta t}}{\partial v_{y,n}^*} = 0 \\ \frac{\partial z^{*t+\Delta t}}{\partial v_{z,n}^*} = 1 \end{cases} \tag{26}$$

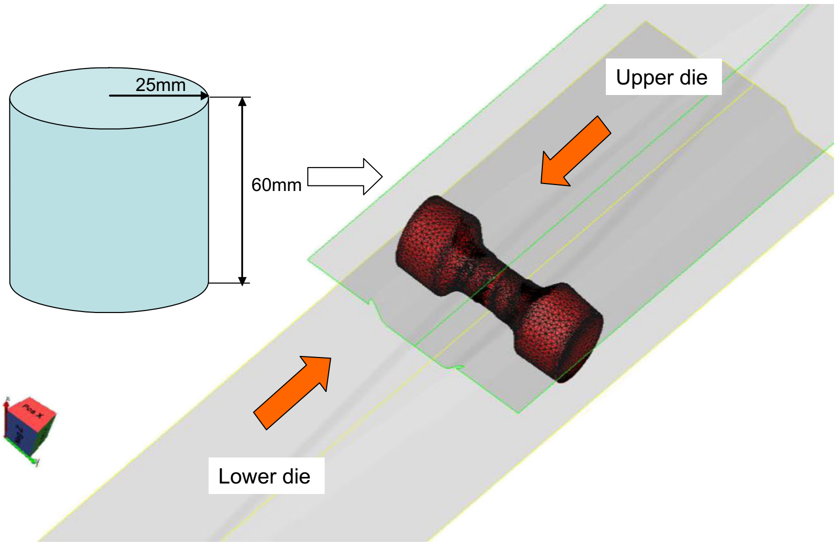


Fig. 9 Cross-wedge rolling with dies having a translation motion

By combining Eqs. (26) and (27), a simplified form is obtained for $\frac{1}{\Delta t}\frac{\partial \mathbf{m}^t \mathbf{m}^{t+\Delta t}}{\partial \mathbf{v}_{n}^{\rightarrow}}$ which can be expressed through a matrix $\mathbf{\Psi}$ as follows:

$$\mathbf{\Psi} = \begin{bmatrix} \Psi_{11} & \Psi_{12} & 0 \\ \Psi_{21} & \Psi_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \ \mathit{Where} \begin{cases} \Psi_{11} = \cos\left(\theta^{*t+\Delta t}\right)\cos\left(\theta^{*t}\right) + \frac{r^{*t+\Delta t}}{r^{*t}}\sin\left(\theta^{*t+\Delta t}\right)\sin\left(\theta^{*t}\right) \\ \Psi_{22} = \sin\left(\theta^{*t+\Delta t}\right)\sin\left(\theta^{*t}\right) + \frac{r^{*t+\Delta t}}{r^{*t}}\cos\left(\theta^{*t+\Delta t}\right)\cos\left(\theta^{*t}\right) \\ \Psi_{12} = \cos\left(\theta^{*t+\Delta t}\right)\sin\left(\theta^{*t}\right) - \frac{r^{*t+\Delta t}}{r^{*t}}\sin\left(\theta^{*t+\Delta t}\right)\cos\left(\theta^{*t}\right) \\ \Psi_{21} = \sin\left(\theta^{*t+\Delta t}\right)\cos\left(\theta^{*t}\right) - \frac{r^{*t+\Delta t}}{r^{*t}}\cos\left(\theta^{*t+\Delta t}\right)\sin\left(\theta^{*t}\right) \end{cases} \tag{27}$$

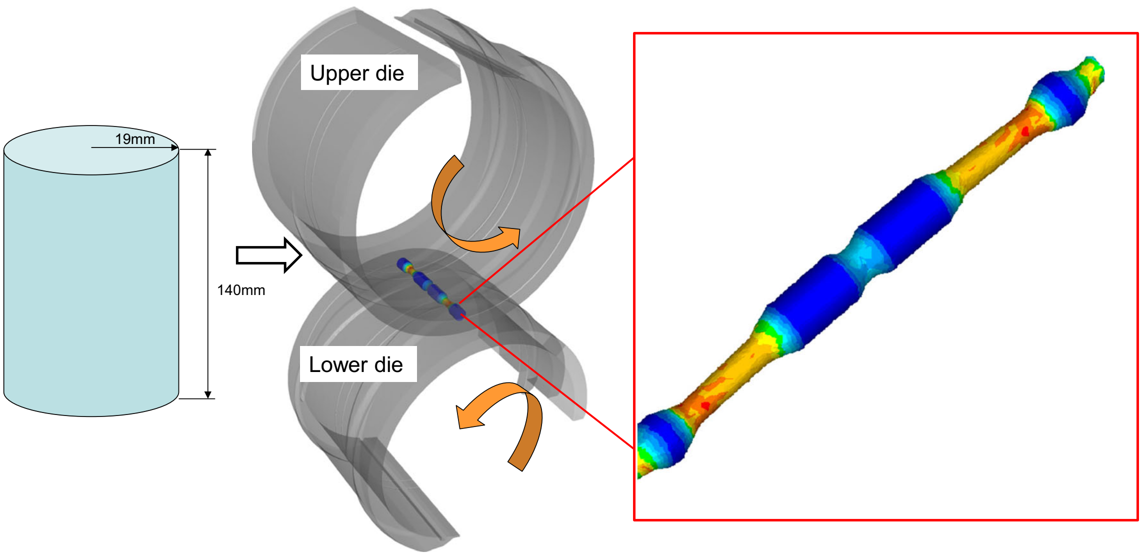


Fig. 10 Cross-wedge rolling with dies having a rotation motion

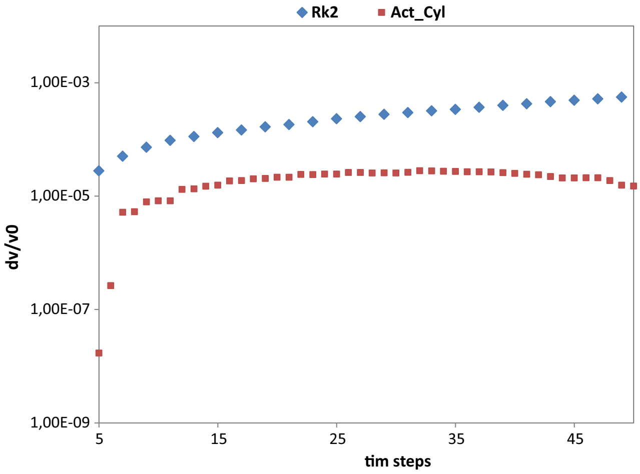


Fig. 11 Relative volume variation for cross-wedge rolling simulation with dies in translation

It is important to notice that the resulting contact Eq. (10) is now non-linear with respect to the velocity field. This would increase the non-linearities of the mechanical problem and consequently the number of Newton Raphson iterations required to solve it.

Applications

For all the applications considered, the material is assumed to follow the Norton-Hoff constitutive Eq. (3) which parameters are: K=1881 MPa and m=0.15. Also, a value of 0.8 has been used for λ_1 and λ_2 (parameters for detecting effectively rotating nodes), a best compromise provided by numerical tests.

Rigid body rotation

An academic case of rigid body rotation is first presented in order to evaluate the accuracy and the order of the proposed scheme. The body consists of a hollow cylinder (see Fig. 4) subjected to a constant rotation velocity ω=10π rad/s.

For such a problem, the analytical solution is well known ($x(t) = \sqrt{x^2+y^2}\cos(\omega t+\theta_0)$, $y(t) = \sqrt{x^2+y^2}\sin(\omega t+\theta_0)$, $z(t)$=z_0, where θ_0 and z_0 respectively denotes the coordinates at time t=0 s). The *cylindrical scheme* is compared to the first-order Euler and second-order Runge–Kutta schemes, in terms of accuracy and convergence order. For the accuracy study, a constant time step size of $\Delta t = 10^{-2} s$ has been used. Time step sizes of Δt, $2\Delta t$ and $4\Delta t$ are then considered for the convergence analysis. Figure 5a and b respectively show the relative error in the L_2 classical norm for the position vector and the relative volume variation for few time steps. The relative error for the position vector in the L_2 norm is defined as $error = \sqrt{\frac{\sum_k \left\|\mathbf{X}_k^{exact}-\mathbf{X}_k\right\|^2}{\sum_k \left\|\mathbf{X}_k^{exact}\right\|^2}}$, where $\mathbf{X}_k^{exact}$ and $\mathbf{X}_k$ respectively represent the exact position vector and the calculated one on a node k. Logarithmic scale is used for the vertical axis. As expected, with the Euler scheme a very large volume dilatation is detected; the volume increases about ten

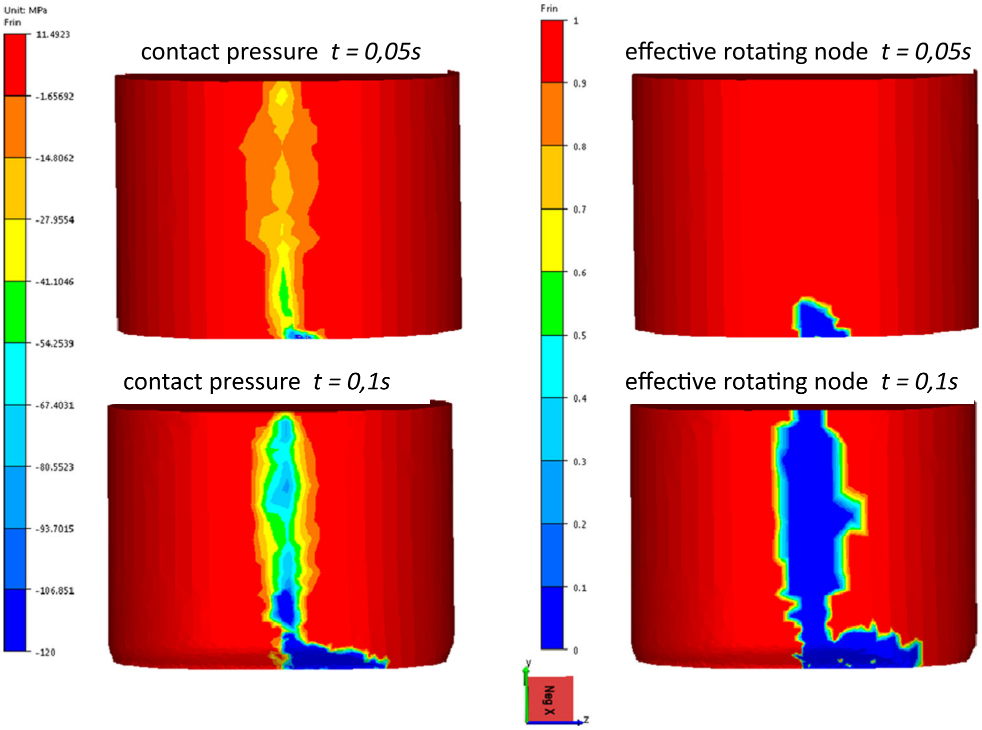


Fig. 12 For two different steps of cross-wedge rolling, *Left*: contact pressure. *Right*: flag of effective rotating node (isovalue equal to 1 if rotation dominates and equal 0 if translation prevails)

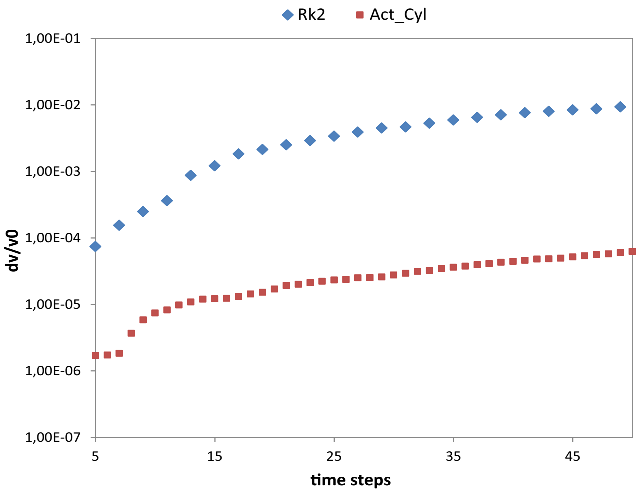


Fig. 13 Relative volume variation for cross-wedge rolling simulation with rotating dies

times. It also results into very large position errors. Second-order Runge–Kutta scheme significantly improves the results (the relative volume variation, $\Delta V/V_0 \sim 10^{-2}$) but it remains not accurate enough. The *cylindrical scheme* provides the best results. The volume variation is very negligible ($\Delta V/V_0 \sim 10^{-8}$). Positions are also very accurate. Figure 6 presents the relative error in the L_2 classical norm for the position vector as a function of the inverse of the time step size in a Log-Log scale. It can be pointed out that the *cylindrical scheme* is a first order scheme (As shown by the slope∼−1), just like the Euler scheme (but with a much better accuracy), while the second-order Runge–Kutta is actually a higher order scheme (but with a poorer accuracy for the considered time steps).

Hot torsion test

The following application aims at showing the impact of volume variation on simulation results. In this hot torsion test, the impacts on the values of calculated torques versus the number of turns are pointed out. The two jaws of the torsion machine (see Fig. 7) have opposite rotation velocities of $\omega_1=\pi/60\ rad/s$ and $\omega_2=-\pi/60\ rad/s$. A time step size of $\Delta t=10^{-1}s$ has been employed.

Figure 8a presents the relative volume variation for few time steps. Once again the lowest volume variation is provided by the *cylindrical scheme*. In spite of the low number of turns, the Euler scheme provides a large volume increase. In terms of torque (Fig. 8b), results obtained with the *cylindrical scheme* are almost identical to that of second-order Runge–Kutta which can be considered here as a reference solution. On the other hand, the Euler scheme provides significantly higher torques, as a consequence of the volume increase.

Cross-wedge rolling

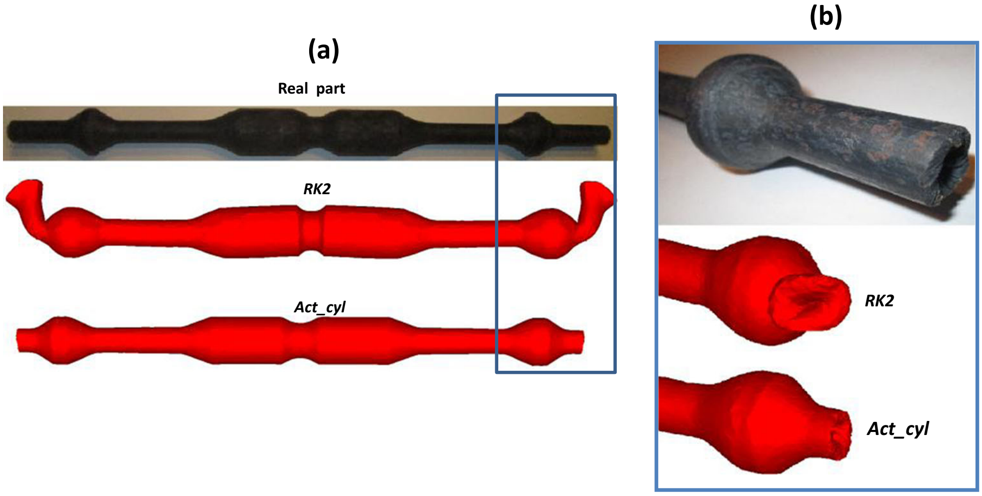
Cross-wedge rolling is employed for manufacturing axisymmetric automotive parts. Starting from a cylindrical billet, the final component is obtained through the plastic deformation imposed by two moving dies (by translation or rotation) that make the billet turn around its axis. For such a tricky process, a slight default in the kinematics modelling results into significant defects on the shape of the final part. It is therefore important to very accurately integrate the large material rotations. For this study, two different dies technologies are considered: with dies in translation (Fig. 9) and with dies in rotation (Fig. 10).

In the die translation process (see Fig. 9), the relative velocity of the dies is $v=30\ mm/s$. In the rotating process (see Fig. 10), the velocities of the counter rotating dies are: $\omega_1=1.25\ rad/s$ and $\omega_2=-1.25\ rad/s$. A time step size of $10^{-2}s$ has been used in the two cases. Since few remeshings are performed to account for the large mesh deformation, our analysis only regards the first fifty time steps (before the first remeshing).

- Translating dies cross-wedge rolling

With translating dies kinematics, the rotation axis of workpiece is continuously moving during the process, so it has to

Fig. 14 Final shapes of the billet during the cross-wedge rolling simulation with rotating dies



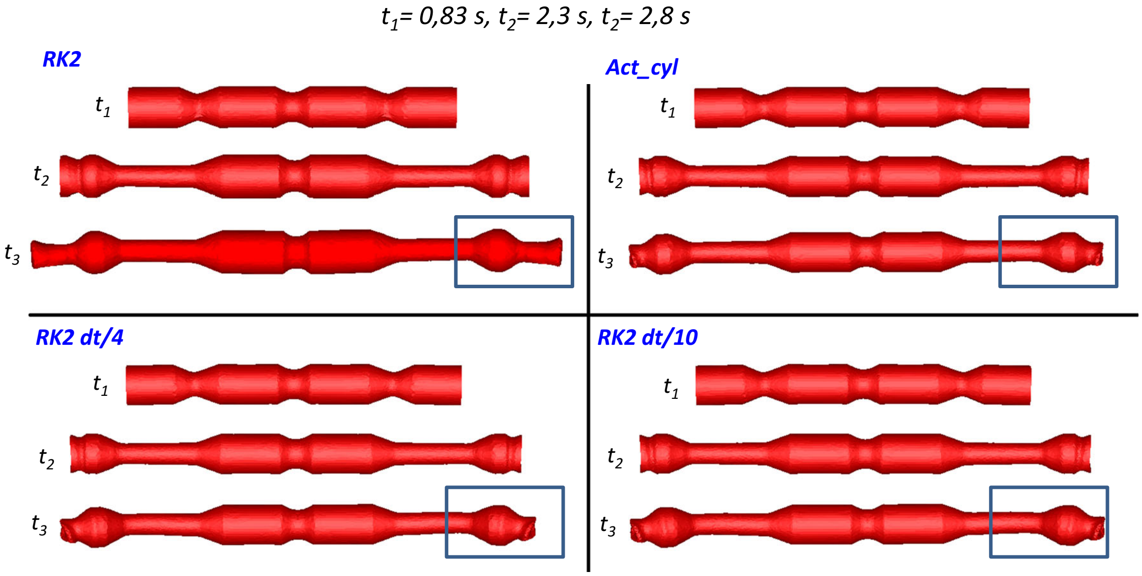


Fig. 15 Shape evolution during the cross-wedge rolling simulation with rotating dies at different stages of the process: comparison between the *cylindrical scheme* and RK2 with decreasing time steps

be continuously updated within the *cylindrical scheme*. As can be seen in Fig. 11, once again the volume conservation is much better along with the *cylindrical scheme* rather than with the second-order Runge–Kutta scheme; the error reduces of two orders of magnitude, from 10^{-3} to 10^{-5}.

Figure 12 (right) presents isovalues of effectively rotating nodes, in other words the flags that evaluate whether rotation (flag equal to 1) or translation (flag equal to 0) prevails. Although the workpiece is continuously rotating around its axis, it is found that translation is able to dominate rotation at some locations between the dies. These isovalues can be easily correlated to contact pressure isovalues of Fig. 12 (left), at different deformation stages of the process. The good agreement between these fields indicates that a translation motion dominates where the maximum compression (negative contact pressure) is observed between the dies.

- Rotating dies cross-wedge rolling

With rotating dies, the rotation axis of the workpiece is almost steady. Figure 13 shows that similar conclusions can be derived for the volume conservation: clear superiority of the

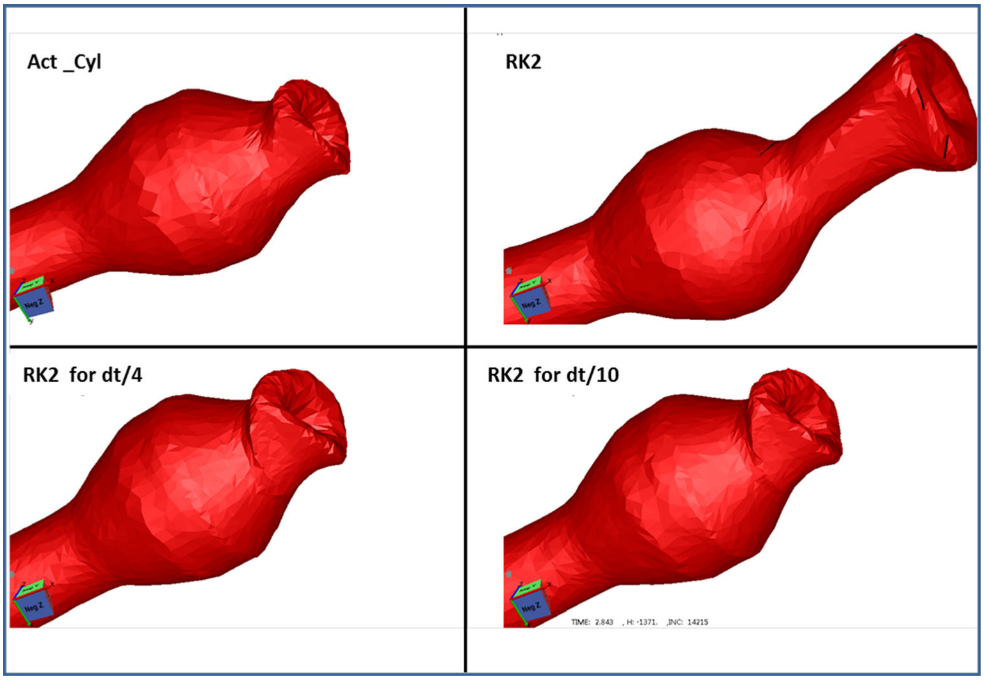


Fig. 16 Zoom on the end of final shape of the billet for the *cylindrical scheme* and RK2 with decreasing time steps

Table 1 Information about the simulations

Schemes	CPU time on 4 cores (hours)	Average number of non linear iterations per step
Act_cyl (Δ*t*)	5.4	20
RK2 (Δ*t*)	5.28	17
RK2 (Δ*t/4*)	17	15
RK2 (Δ*t/10*)	33.7	13

cylindrical scheme over the second-order Runge–Kutta scheme with also two orders of difference on the error. In this case, with the second-order Runge–Kutta scheme, the volume increase is one order of magnitude larger than with the previous kinematics.

Analyzing the shape of the billet can also highlight the accuracy and robustness of the cylindrical scheme. Figure 14 presents the final shapes of the billet that have been obtained with the two schemes, as well as the real shape of the billet resulting from the actual forming process. The *cylindrical scheme* allows predicting a very realistic final shape, while the RK2 scheme provides a noticeable forming defect (see Fig. 14b) at the end of the process simulation. This unexpected defect (with RK2) can be explained by the numerical increase of material volume observed in Fig. 13. Although low (1 %), it can be enough to modify the forming route of such a sensitive process. This hypothesis is confirmed by carrying out a convergence study with reduced time steps, *Δt/4* and *Δt/10*. Figure 15 shows the shape evolution over the time and Fig. 16 provides a zoom on one end of the final shape. They clearly show that the RK2 scheme converges toward the solution provided by *cylindrical scheme*.

Table 1 provides information about the simulations. For the time step Δ*t,* the *cylindrical scheme* is just slightly (2.3 %) more expensive than the RK2 scheme, due to additional calculations and additional non-linearities in the residual equations as the average number of iterations needed for the Newton Raphson algorithm shows it. However, with such time step Δ*t*, the RK2 scheme provides inaccurate results, with much more smaller time steps (*Δt/4* or better *Δt/10*) accurate results are reached but much higher computational times (respectively 3 and 6 times larger) is needed. The *cylindrical scheme* is then both accurate and much more efficient.

To illustrate the necessity to take into account the *cylindrical scheme* in the contact Eqs. (24–27) and to solve the resulting non-linear equations, a comparison of results obtained with and without changes in the contact equations has been carried out when the cylindrical scheme is used. Figure 17 shows the isovalues of the contact distance (a positive or zero value indicates that the concerned node is in contact) at time *t*=0.37 s. It can be clearly noticed that the contact area is not similar; the contact is detected earlier and is more pronounced when properly taking into account the *cylindrical scheme* in the contact equations.

Conclusion

The studied *cylindrical* time integration scheme consists in expressing the motion equations into a cylindrical frame, the rotation axis of which is computed from the skew-symmetric part of the velocity gradient. This scheme is only applied in domain areas where the velocity field corresponds to a rotational movement rather than to a translational movement. It provides a general scheme that can be applied to any problem. This *cylindrical scheme* is only first order but it shows more accurate (and much easier to implement) than a higher order

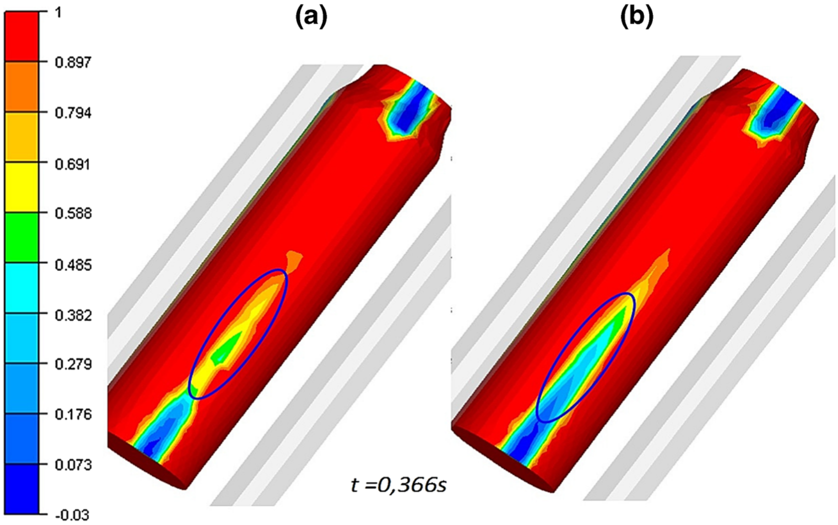


Fig. 17 Isovalue of the contact distance (**a**) without taking the cylindrical scheme into account and (**b**) taking it into account in the contact equations

scheme such as second-order Runge–Kutta, which is clearly shown by the analytical example of the rotating rigid body. With respect to the original Euler scheme, which is often used in implicit metal forming formulations, the accuracy of the *cylindrical scheme* is several orders of magnitude higher. It is particularly suited to problems where the rotation axis continuously varies over time, while making it possible to take into account that, in some particular portions of the domain subjected to large deformations (between the rolling dies), the material flow rather corresponds to a translation and should therefore be integrated with a standard scheme. For studied forming problems involving very large rotations, such as the torsion test or cross-wedge rolling, the cylindrical scheme provides significantly better results than the second-order Runge–Kutta, both for the volume preservation; the torsion torques values or else the computed geometries at the end of rolling.

This scheme can rather easily be introduced into existing code without specific difficulties. Changing an Euler time integration scheme into the *cylindrical scheme* is rather straight forward (equations of section 3.1), while detecting the rotation axis and the rotation occurrence, as well as introducing the changes to the contact equations (section 3.3) requires going further into the data structure of the code.

Acknowledgments Financial support of AREVA Cezus (Paimbœuf & Ugine) for this work is gratefully acknowledged.

References

1. Atluri SN, Cazzani A (1995) Rotations in computational solid mechanics. Arch Comput Methods Eng 2(1):49–138
2. Buss SR (2000) Accurate and efficient simulation of rigid-body rotations. J Comput Phys 164(2):377–406
3. Ghosh S, Roy D (2010) An accurate numerical integration scheme for finite rotations using rotation vector parametrization. J Frankl Inst 347(8):1550–1565
4. Rubin MB (2007) A simplified implicit Newmark integration scheme for finite rotations. Comput Math Appl 53(2):219–231
5. Ibrahimbegovic A (1997) On the choice of finite rotation parameters. Comput Methods Appl Mech Eng 149(1–4):49–71
6. Ibrahimbegovic A, Taylor RL (2002) On the role of frame-invariance in structural mechanics models at finite rotations. Comput Methods Appl Mech Eng 191(45):5159–5176
7. Betsch P, Menzel A, Stein E (1998) On the parametrization of finite rotations in computational mechanics. Comput Methods Appl Mech Eng 155(3–4):273–305
8. Pimenta PM, Campello EMB, Wriggers P (2008) An exact conserving algorithm for nonlinear dynamics with rotational DOFs and general hyperelasticity. Part 1: Rods. Comput Mech 42(5):715–732
9. Cardona A, Geradin M (1988) A beam finite element non-linear theory with finite rotations. Int J Numer Methods Eng 26(11): 2403–2438
10. Bottasso CL, Borri M (1998) Integrating finite rotations. Comput Methods Appl Mech Eng 164(3–4):307–331
11. Borri M, Trainelli L, Bottasso CL (2000) On representations and parameterizations of motion. Multibody Syst Dyn 4(2–3): 129–193
12. Brank B, Mamouri S, Ibrahimbegovic A (2005) Constrained finite rotations in dynamics of shells and Newmark implicit time-stepping schemes. Eng Comput 22(5/6):505–535
13. Argyris J (1982) An excursion into large rotations. Comput Methods Appl Mech Eng 32(1–3):85–155
14. Zupan E, Saje M (2011) Integrating rotation from angular velocity. Adv Eng Softw 42(9):723–733
15. K. Traore, R. Forestier, K. Mocellin, P. Montmitonnet, and M. Souchet, *Three dimensional finite element simulation of ring rolling*. 2001
16. Dialami N, Chiumenti M, Cervera M, de Saracibar C A, and Ponthot J P (Dec. 2013) "Material flow visualization in Friction Stir Welding via particle tracing". *Int. J. Mater. Form.*, pp. 1–15
17. Chiumenti M, Cervera M, Agelet de Saracibar C, Dialami N (2013) Numerical modeling of friction stir welding processes. Comput Methods Appl Mech Eng 254:353–369
18. Gines Losilla, Montmitonnet Pierre, and Bouziane M., "Modélisation du laminage circulaire par éléments finis. Calculs thermomécaniques et calcul microstructuraux," presented at the Matériaux 2002, Tours, France, 2002.
19. Soyris N, Fourment L, Coupez T, Cescutti JP, Brachotte G, Chenot JL (1992) Forging of a connecting rod: 3D finite element calculations. Eng Comput 9(1):63–80
20. Gay C, Montmitonnet P, Coupez T, Chenot JL (1994) Test of an element suitable for fully automatic remeshing in 3D elastoplastic simulation of cold forging. J Mater Process Technol 45(1–4):683–688
21. Wagoner RH, Chenot J-L (2001) Metal Forming Analysis, 1st edn. Cambridge University Press, Cambridge
22. Coupez T, Soyris N, Chenot J-L (1991) 3-D finite element modelling of the forging process with automatic remeshing. J Mater Process Technol 27(1–3):119–133
23. Mocellin K, Fourment L, Coupez T, Chenot JL (2001) Toward large scale F.E. computation of hot forging process using iterative solvers, parallel computation and multigrid algorithms. Int J Numer Methods Eng 52(5–6):473–488
24. Fourment L, Chenot JL, Mocellin K (1999) Numerical formulations and algorithms for solving contact problems in metal forming simulation. Int J Numer Methods Eng 46(9):1435–1462
25. Jelenić G, Saje M (1995) A kinematicsally exact space finite strain beam model — finite element formulation by generalized virtual work principle. Comput Methods Appl Mech Eng 120(1–2):131–161