ABSTRACT: Weight reduction of vehicles can be achieved by using high strength steels or aluminum. The formability of aluminum can be improved by applying the forming process at elevated temperatures. A thermo-mechanically coupled material model and shell element is developed to accurately simulate the forming process at elevated temperatures. The use of high strength steels enlarges the risk of wrinkling. Wrinkling indicators are developed which are used to drive a local mesh refinement procedure to be able to properly capture wrinkling.

Besides, to intensify the use of implicit finite element codes for solving large-scale problems, a method is developed which decreases the computational time of implicit codes by factors. The method is based on introducing inertia effects into the implicit finite element code. It is concluded that the computation time is decreased by a factor 5-10 for large-scale problems.

Key words: wrinkling, aluminum warm forming, dynamic implicit FE code
125°C, the ultimate tensile strength decreases and the strain rate sensitivity increases.

A numerical model is developed to simulate the warm forming of Al-Mg sheet. Both the hardening behaviour, including temperature and strain rate effects, and the biaxial stress-strain response of the sheet are considered. Hardening is modeled with the physically-based Bergström-model. This model incorporates the influence of the temperature on the flow stress and on the hardening rate based on dynamic recovery. The biaxial stress-strain response of the material is experimentally determined by uniaxial stress, plane strain, simple shear and equibiaxial stress tests. The Vegter yield criterion is used as it is able to accurately represent the shape of the yield locus and the anisotropy.

To check the performance of the numerical model, three experiments at room temperature and at elevated temperatures were analysed numerically. In the simulated experiments, the drawing ratio was fixed at 2.09. For this ratio, cups could be drawn at all temperatures, without failure, and the influence of the flange temperature on the thickness can be compared, Fig. 2 and Fig. 3. In Fig. 3, the arc-length is given starting at the outer radius in the transverse direction, continuing to the outer radius in rolling direction and returning to the transverse direction along the outer radius. In Fig. 2, the thickness strain along the rolling direction is given. The influence of the temperature on the thickness after the 80 mm punch stroke is most pronounced in the bottom of the cup. An increase in temperature yields a decrease in thickness strain in the bottom of the cup. However, in the simulations, the thickness reduction in the bottom of the cup is overestimated. Probably, the deviation is caused by an inaccurate prediction of the friction behavior; changing the friction properties largely influences the thickness strain distribution [1].

3 WRINKLING PREDICTION

Wrinkling is becoming an important failure mode in sheet metal forming mainly because of the trend toward thinner, high strength sheet metals. In a numerical simulation, wrinkles can be detected by visual inspection of the deformed mesh, provided that this mesh is fine enough to allow a proper capture of the wrinkles. On the other hand a fine mesh means high computational costs. Therefore the objective is to drive local mesh refinement with wrinkling indicators, which allows us to spot the wrinkles properly while keeping the computational costs as low as possible.

In this work, the analysis of Hutchinson and Neale [2,3] is used. However, Hutchinson analysis is limited to contact free regions. Therefore, a different approach must be used in regions where contact applies. Consequently, a geometric indicator based on the local change of curvatures has been developed by Selman [4,5]. In this paper, only the geometric wrinkling indicator will be focused on.

As stated, Hutchinson analysis is restricted to contact free wrinkling. Therefore, a new indicator had to be developed which can be used in contact regions. This indicator is based on the change of
curvatures under compressive stresses. Taking the change of curvature instead of taking the curvature itself, filters out all changes in curvature that are not due to compressive stresses, such as those caused for example by the geometry of the tool and the die. The new wrinkling indicator in the $i^{th}$-principal direction is defined as

$$ e_i^w = \frac{1}{A} \int \left( \frac{R_i^t - R_i^{t+\Delta t}}{R_i^t R_i^{t+\Delta t}} \right) dA \quad i = 1,2 $$  (1)

with $i$ representing the principal curvature direction and $R_i^t$, $R_i^{t+\Delta t}$ are the radii of curvatures in the $i$-direction at the beginning and at the end of a given deep drawing step, respectively. The principal directions (stress and curvature) are considered one at a time, as is the case with Hutchinson approach, and the maximum change for each element is taken. Consequently, the wrinkling indicator value $e_i^w$ is defined as the maximum of $e_i^w$, i.e.

$$ e_i^w = \max(e_i^w) \quad i = 1,2 $$  (2)

The wrinkling indicator value $e_i^w$ is only calculated for elements under compressive stresses in the 1 and/or 2 principal directions. The wrinkling risk factor can now be defined as:

$$ f_{gw} = \frac{e_i^w}{e_{avr}} $$  (3)

where $e_{avr}$ is the average value of $e_i^w$ over all rotating elements.

The performance of the wrinkling indicators to drive the local mesh refinement will be demonstrated with a deep drawing simulation of a hemispherical product, applying a low blankholder force. A low blankholder force is chosen to enforce the product to wrinkle under the blankholder. The simulation is started with a mesh comprising 2050 elements. After a punch stroke of 30 mm, the mesh refinement in the wall of the product is driven by the Hutchinson wrinkling criterion. After a punch stroke of 60 mm, the elements under the blankholder are refined, driven by the geometric wrinkling indicator. At the final stage (punch stroke of 100 mm), severe wrinkling is observed under the blankholder, see Fig. 4.

The hemispherical product has been used as a benchmark test with various BHF and drawn experimentally in a series of tests at different forming depths to spot the onset, location and mode of wrinkling. In comparing the numerical results with those obtained through experimental testing, a good agreement has been found in terms of location, onset and modes of wrinkling [6]. However, the numerical analyses tend to overestimate the experimentally observed amplitudes of the wrinkles slightly as well as it predicts an onset of wrinkling somewhat earlier than experimentally observed. This makes the numerical analysis conservative compared to experimental testing.

**Fig. 4.** Mesh refinement, driven by the Hutchinson and the geometric wrinkling indicator

4 DYNAMIC IMPLICIT INTEGRATION

At present time the competition between implicit and explicit finite element codes is still in full swing, where explicit codes are favored for solving large problems although implicit codes yield more accurate results. To intensify the use of implicit codes for solving large problems, the computation time of these codes has to be decreased drastically. A method is developed which decreases the computational time of implicit codes by factors, which makes the implicit finite element code competitive with explicit finite element codes for large scale problems. This method is based on introducing dynamics contributions into the implicit finite element code. Using a lumped mass matrix approach, this method yields an improved condition of the system matrix, which makes an effective use of iterative solvers possible.

The dynamics contributions are implemented for shell elements which have 3 displacement degrees of freedom (d.o.f.) and 3 rotational d.o.f. per node. The condition number of the system matrix will improve considerably when mass contributions are taken into account for both the displacement and rotational d.o.f. A mass moment of inertia approach is used to account for the mass contributions for the rotational d.o.f. For details about the implementation, the reader is referred to [7,8,9].

First the influence of the dynamics contributions on the deep drawing of a rectangular product is investigated, using different iterative solvers.
Several simulations are performed in which the punch velocity is varied between 0 m/s (no dynamics contributions) and 50 m/s. The system matrix at step 40 is stored whereafter the system is solved, using the different iterative solvers. The unbalance criterion is set to $10^{-5}$, the maximum number of iterations is set to 500. The performance of these iterative solvers is given in Fig. 5 (labeled as *rotdis).

It can be concluded that the convergence rate drastically increases with an increase of the deep drawing velocity. Since, the results between a simulation performed with dynamics contributions until a deep drawing speed of 25 m/s and one without dynamics contributions do not differ significantly [8], it is possible to decrease the computation time by factors without affecting the results. Besides a set of simulations is performed without dynamics contributions for the rotational d.o.f. (labeled as *_dis) This figure shows that it is important to take into account the dynamics contributions for the rotational d.o.f, since they have a large influence on the convergence behavior. Next a simulation of large-scale problem is performed, viz. the Numisheet 2002 Fender [10]. The deep drawing velocity is set to 10 m/s. One simulation is performed using the Bi-CGSTAB iterative solver with SSOR preconditioning and one simulation is performed with a direct solver (Cholesky decomposition). The simulation starts with 25000 d.o.f and ends up with 170000 d.o.f. Fig. 7. The simulation took 16 hours on an ALPHA DS20E workstation in case of using the iterative solver. In case of using the direct solver, the simulation stopped due to memory capacity problems. Therefore, the computation time for one iteration is considered to make a comparison between the computation times of both solvers, Fig. 6. It is concluded that using an iterative solver drastically decreases the computation time.

REFERENCES