FOI
SWEDISH DEFENCE
RESEARCH AGENCY

FOI-R--0310--SE
June 2002
ISSN 1650-1942

Scientific report

Lars Westerling

AN IMPLICIT METHOD FOR A STRAIN RATE DEPENDENT CONSTITUTIVE MODEL IMPLEMENTED IN AUTODYN AS A USER SUBROUTINE

Weapons and Protection
SE-147 25 Tumba

SWEDISH DEFENCE RESEARCH AGENCY

Weapons and Protection
SE-147 25 Tumba

FOI-R--0310--SE

June 2002

ISSN 1650-1942

Scientific report

Lars Westerling

AN IMPLICIT METHOD FOR A STRAIN RATE DEPENDENT CONSTITUTIVE MODEL IMPLEMENTED IN AUTODYN AS A USER SUBROUTINE

Issuing organization
FOI – Swedish Defence Research Agency
Weapons and Protection
SE-147 25 Tumba

Report number, ISRN
FOI-R--0310--SE

Report type
Scientific report

Research area code
5. Combat

Month year
June 2002

Project no.
E2022

Customers code
5. Commissioned Research

Sub area code
51 Weapons and Protection

Author/s (editor/s)
Lars Westerling

Project manager
Ewa Lidén

Approved by

Sponsoring agency

Scientifically and technically responsible

Report title
AN IMPLICIT METHOD FOR A STRAIN RATE DEPENDENT CONSTITUTIVE MODEL IMPLEMENTED IN AUTODYN AS A USER SUBROUTINE

Abstract (not more than 200 words)
An implicit numerical method for the strain rate dependent constitutive model by Johnson and Cook is described. It is implemented in AUTODYN-2D as a user subroutine. The reason for doing this is that the standard method sometimes leads to high frequency oscillations and underestimation of the influence of strain rate on the yield stress. The standard method, our implicit method as well as a new method (where also the mentioned problems are solved) included in the latest version of AUTODYN (v. 4.2, released autumn 2001) are compared.

Keywords
Hydrocode; Strain rate dependence; Johnson and Cook; Implicit; Autodyn;

Further bibliographic information

Language English

ISSN 1650-1942

Pages 42 p.

Price acc. to pricelist

Utgivare
Totalförsvarets Forskningsinstitut - FOI
Vapen och skydd
147 25 Tumba

Rapportnummer, ISRN
FOI-R--0310--SE

Klassificering
Vetenskaplig rapport

Forskningsområde
5. Bekämpning

Månad, år
Juni 2002

Projektnummer
E2022

Verksamhetsgren
5. Uppdragsfinansierad verksamhet

Delområde
51 VVS med styrda vapen

Författare/redaktör
Lars Westerling

Projektledare
Ewa Lidén

Godkänd av

Uppdragsgivare/kundbeteckning

Tekniskt och/eller vetenskapligt ansvarig

Rapportens titel (i översättning)

EN IMPLICIT METOD FÖR EN TÖJNINGSHASTIGHETSBEROENDE KONSTITUTIV MODELL INFÖRD I AUTODYN SOM EN ANVÄNDARRUTIN

Sammanfattning (högst 200 ord)

En implicit numerisk metod för den töjningshastighetsberoende konstitutiva modellen av Johnson och Cook beskrivs. Vi har infört den i AUTODYN-2D som en användarrutin. Anledningen är, att standardmetoden ibland leder till högfrekventa oscillationer och underskattning av töjningshastighetens inflytande på flytspänningen. Standard-metoden, vår implicita metod liksom även en ny metod (där också de nämnda problemen är åtgärdade), införd i senaste versionen av Autodyn (v. 4.2 utgiven i oktober 2001), jämförs.

Nyckelord
Hydrokod; Töjningshastighetsberoende; Johnson och Cook; Implicit; Autodyn;

Övriga bibliografiska uppgifter

Språk engelska

ISSN 1650-1942

Antal sidor: 42 s.

Distribution enligt missiv

Pris: Enligt prislista

CONTENTS

1. INTRODUCTION 5

2. THE STANDARD METHOD 6
   2.1. Basic algorithm for elastic-plastic materials 6
   2.2. Plastic strain 7
   2.3. Expression for the yield stress 7

3. THE IMPROVED METHOD 7

4. TEST RUNS AND THEIR RESULTS 8
   4.1. Simulations of tensile tests 8
   4.2. Simulations of projectile penetration 10

5. DISCUSSION 11
   5.1. Comparison of the three methods 11
   5.2. Further tests of the old method 12
   5.3. Explanation of the oscillations 12
   5.4. Strain hardening and temperature dependence 13
   5.5. Motivation of the improved version 13

6. CONCLUSIONS 13

ACKNOWLEDGEMENTS 14

REFERENCES 14

APPENDIX A. THE USER MODEL 21
   A1. Overview 21
   A2. Miscellaneous details 21
   A3. The subroutine `STRENGTH` 24
   A4. The subroutine `JC` 25
   A5. The subroutine `SOLVE` 25

APPENDIX B. A SIMPLE ONE-DIMENSIONAL PROBLEM 27
   B1. Formulation of the problem for constant friction 27
   B2. Solution of discrete problem for constant friction 27
   B3. Formulation of the problem for non-constant friction 28
   B4. Solution of the problem for non-constant friction 28
   B5. Method I, the traditional method 29
   B6. Method II, the improved method 30
   B7. Differential equation 30
   B8. Method III, Euler backwards 31
   B9. Method IV, Euler forwards 32
   B10. The equivalence of methods II and III 33
   B11. Stability analysis for method I 34
   B12. Relaxation times in method I and the differential equation 36
   B13. Stability analysis of a simplified version of method II 38
   B14. Stability analysis of method IV 39
   B15. Discussion of the case with no rate effects 40
   B16. Summary and conclusions of this appendix 40

1. INTRODUCTION

The commonly used constitutive model for metals by Johnson and Cook [1] includes strain hardening, strain rate dependence, and thermal softening. When we performed simulations of tensile tests with AUTODYN-2D version 4.1 [2-4] using that model, we sometimes encountered oscillations in the stress response and also too little increase of yield stress due to strain rate effects. S. A. Silling [5] reports problems with numerical instability when implementing the same model in an Euler code called CTH [6]. Silling performs stability analysis for a few methods, both stable and unstable. He chooses an implicit scheme (Euler backwards), which is stable, and describes how that was implemented in CTH. L. Nilsson [7] described problems with oscillating solutions and erroneous relaxation times when using the old versions of strain rate dependent constitutive models in DYNA [8]. The problem was solved by using a fully viscoplastic [9] formulation of the model. O. Wall [10] describes his implementation in a finite element code of the strain rate dependent constitutive model by Zerilli and Armstrong [11]. He uses an implicit scheme (Euler backwards) and shows how the implicitness reduces to solving a single (scalar) equation.

The problems were reported to Century Dynamics, the developer of AUTODYN, who solved them in the latest version (version 4.2, released in October 2001) of the code by an optional "strain rate correction" for Johnson-Cook's constitutive model.

Constitutive models used in hydrocodes have developed over the years and now take more and more phenomena into account. When implementing these models, a frequently occurring problem is that, at some point in the algorithm a value of a variable is needed that has not yet been calculated. A common practice in these cases seems to be to simply use *the most recent known value* of that variable, although it would have made more sense to solve an equation for that variable. The reason for this practice is that since the basic time integration in hydrocodes is explicit, it is preferred to have an explicit implementation also of the constitutive model. Using *the most recent known value* of the strain rate in the implementation of Johnson-Cook's constitutive model is, as will be shown, the cause of the observed oscillations.

In this report an implementation of Johnson-Cook's model as a user subroutine in AUTODYN-2D is presented. The user subroutine calculates the yield stress by an implicit method every time step. The rest of the stress update algorithm is performed by AUTODYN itself. The difference from the old standard implementation in AUTODYN is that the time centring of the strain rate in the expression for the yield stress is more natural in our implementation. Some programming details are postponed to Appendix A. The old standard method in AUTODYN, our improved method (the user subroutine), and the new method implemented in the latest version of the code are compared.

Section 2 includes a description of the basic algorithm for elastic-plastic materials [2,12] used in hydrocodes together with a description of the old standard method for treating strain rate dependence. Our improved method is described in Section 3. A series of test runs are presented in Section 4, and results of these are discussed in Section 5.

In Appendix B a simple one-dimensional problem is presented, and several numerical solution methods of it are analysed. This model problem is related to the deviatoric part of the constitutive model, and we hope that it can throw some light on the numerical treatment of strain rate dependent constitutive models.

2. THE STANDARD METHOD

2.1. Basic algorithm for elastic-plastic materials

The stress tensor is divided into a hydrostatic pressure and a deviatoric stress tensor. The pressure is given as a function of specific volume and specific internal energy, while the deviator is calculated incrementally. In this description we disregard finite rotations, which AUTODYN takes care of by using Jaumann time derivative of the stress deviator. The new stress deviator for time t_{n+1} is determined by first calculating a trial stress deviator

$$s_{ij}^{*} = s_{ij}^{n} + 2Gd'_{ij} \tag{2.1}$$

from the old deviator s_{ij}^{n} at time t_n, by using the shear modulus G and the deviatoric part

$$d'_{ij} = d_{ij} - \frac{1}{3} d_{kk} \delta_{ij} \tag{2.2}$$

of the strain rate tensor

$$d_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{2..3}$$

where v_j is the particle velocity. The velocity and the strain rate component are centred at time $(t_n+t_{n+1})/2$ called $t_{n+1/2}$. If the von Mises effective stress of the trial stress tensor

$$\sigma_{eff}^{*} = \sqrt{\frac{3}{2} s_{ij}^{*} s_{ij}^{*}} \tag{2.4}$$

is less than the yield stress Y_{n+1} at time t_{n+1}, the material is elastic, and the trial stress deviator is accepted as the new one, namely,

$$s_{ij}^{n+1} = s_{ij}^{*}, \quad \text{if} \quad \sigma_{eff} \le Y_{n+1} \tag{2.5}$$

In the opposite case when

$$k = \frac{\sigma_{eff}^{*}}{Y_{n+1}} > 1, \tag{2.6}$$

the material is plastic and the new stress deviator is

$$s_{ij}^{n+1} = \frac{1}{k} s_{ij}^{*} \tag{2.7}$$

i.e. a uniformly scaled down version of the trial stress deviator. This may lead to a non-associated flow rule. (A flow rule is called associated, if the plastic flow is always perpendicular to the yield surface.)

2.2. Plastic strain

The plastic strain occurring during the time step from t_n to t_{n+1} is [2]

$$\Delta\varepsilon^p = \frac{\sigma^*_{eff} - Y_{n+1}}{3G}, \tag{2.8}$$

where G is the shear modulus, Y_{n+1} is the yield stress at time t_{n+1}, and σ^*_{eff} is given by Eq. (2.4). If the cumulated plastic strain at time t_n is called ε^p_n, we can also write

$$\Delta\varepsilon^p = \varepsilon^p_{n+1} - \varepsilon^p_n, \tag{2.9}$$

where the subscripts denote time index. The average plastic strain rate in that interval is

$$\dot{\varepsilon}^p = \frac{\Delta\varepsilon^p}{\Delta t}, \tag{2.10}$$

where

$$\Delta t = t_{n+1} - t_n. \tag{2.11}$$

2.3. Expression for the yield stress

By choosing different formulas or algorithms for the yield stress Y, different constitutive models are obtained. In both Johnson-Cook's [1] and Zerilli-Armstrong's [11] models the yield stress

$$Y = Y(\varepsilon^p, \dot{\varepsilon}^p, T), \tag{2.12}$$

is written as a function of plastic strain ε^p, plastic strain rate $\dot{\varepsilon}^p$, and temperature T. A difficulty, occurring when implementing such models, is that the yield stress for time t_{n+1} is needed in order to calculate the plastic strain increment in the time interval from t_n to t_{n+1} according to Eq. (2.8), and that this yield stress depends on the plastic flow during that time interval, see Eq. (2.12). In the old standard method, when calculating the yield stress, one uses "the most recent known" value of plastic strain and plastic strain rate, and the formula used is

$$Y_{n+1} = Y(\varepsilon^p_n, \frac{\varepsilon^p_n - \varepsilon^p_{n-1}}{\Delta t}, T_n). \tag{2.13}$$

3. THE IMPROVED METHOD

The difference between the improved method, which is introduced here, and the standard method is that different time centring is used in the expression for the yield stress. Instead of Eq. (2.13) the equation

$$Y_{n+1} = Y(\varepsilon^p_{n+1}, \frac{\varepsilon^p_{n+1} - \varepsilon^p_n}{\Delta t}, T_n). \tag{3.1}$$

is used. It looks better to use the value of plastic strain at the same time as the time at which we are going to calculate the yield stress. The plastic strain rate in Eq. (3.1) is centred at $t_{n+1/2}$, which seems better than at $t_{n\text{-}1/2}$ as it was in the standard method, Eq. (2.13). Of course, it would have been preferred to have the temperature centred as T_{n+1} in Eq. (3.1) (in the same way as the plastic strain), but this is more difficult to implement and has therefore been avoided.

The improved treatment of plastic flow is implicit, since ε^p_{n+1} in the right member of Eq. (3.1) is not known when Y_{n+1} is needed. Eq. (3.1) together with Eqs. (2.8) and (2.9) must be regarded as a system of equations. It can be simplified to one equation. From Eqs. (2.8) and (2.9) we, namely, obtain

$$\varepsilon^p_{n+1} - \varepsilon^p_n = \frac{\sigma^*_{eff} - Y_{n+1}}{3G}, \tag{3.2}$$

and this equation together with Eq. (3.1) form a system of equations for the unknowns ε^p_{n+1} and Y_{n+1}. If Y_{n+1} is eliminated, we get the single equation

■

$$\sigma^*_{eff} - 3G(\varepsilon^p_{n+1} - \varepsilon^p_n) = Y(\varepsilon^p_{n+1}, \frac{\varepsilon^p_{n+1} - \varepsilon^p_n}{\Delta t}, T_n) \tag{3.3}$$

with ε^p_{n+1} as unknown. With Eq. (2.9), we can write Eq. (3.3) as

■

$$\sigma^*_{eff} - 3G\Delta\varepsilon^p = Y(\varepsilon^p_n + \Delta\varepsilon^p, \frac{\Delta\varepsilon^p}{\Delta t}, T_n), \tag{3.4}$$

where $\Delta\varepsilon^p$ is the unknown.

The method is implicit but only one (scalar) equation has to be solved. This method has been implemented in AUTODYN as a user subroutine, where Eq. (3.4) is solved by iterations using Newton-Raphson's method. See Appendix A for more details.

4. TEST RUNS AND THEIR RESULTS

Three implementations of the Johnson-Cook's model are tested and compared. The first is the method used in version 4.1 and earlier versions of AUTODYN. We are going to call that the *old method*. The second method is our *improved method*, described in Section 3 and implemented as a user-subroutine. The third method is the one that has become the default option for Johnson-Cook's model in the latest version 4.2 of AUTODYN (both 2D and 3D). In that method, which we call the *new method*, the plastic flow algorithm has been modified in order to reduce the high frequency oscillations. The old and improved methods are run on version 4.1.13 and the new method on version 4.2.02 of AUTODYN-2D.

4.1. Simulations of tensile tests

Simulations of tensile tests of a cylindrical steel bar were carried out. The length and radius of the cylinder was 4 mm and 1 mm, respectively. It was divided into 20 cells axially and

5 radially. The nodes at one end of the bar were fixed, while the nodes at the other end were pulled with constant velocity in the axial direction. In most cases this velocity was 4 m/s, which lead to a strain rate of 1000 s^{-1}. Nodes at both ends were fixed in the radial direction.

Table 1. Material parameters for two types of steel

Material	Density	Bulk modulus	Spec. heat	Melting temp.	Shear modulus	JC-parameters in Eq. (4.1)					
	ρ	K	c_v	T_{m}	G	A	B	n	C	$\dot{\varepsilon}_0$	m
	(Mg/m^3)	(GPa)	(J/kgK)	(K)	(GPa)	(GPa)	(GPa)			(s^{-1})	
4340 Steel	7.83	159	477	1793	81.8	0.792	0.510	0.26	0.014	1.0	1.03
HNS Steel	7.83	159	477	1793	81.8	0.699	1.148	0.595	0.030	1.0	0.65

The constitutive model was Johnson-Cook's model, where the yield stress was given by

$$Y = Y(\varepsilon^p, \dot{\varepsilon}^p, T) = \left(A + B(\varepsilon^p)^n\right)\left(1 + C\ln(\dot{\varepsilon}^p/\dot{\varepsilon}_0)\right)\left(1 - T_{\mathrm{H}}^m\right), \tag{4.1}$$

where A, B, n, C, $\dot{\varepsilon}_0$, and m are parameters, and T_{H} is the homologous temperature (normalised so that it is zero for room temperature and unity for melting temperature). The remaining variables are the same as in Eq. (2.12). The strain rate dependence is defined by the strain rate coefficient C and the threshold value $\dot{\varepsilon}_0$ for strain rate effects (in most cases equal to 1 s^{-1}). The parameters were taken from the material library in AUTODYN for 4340 steel in the base line case, see Table 1.

Table 2. Simulated tensile tests

Run ID	(Figure)	Strain Rate $\dot{\varepsilon}$ (s^{-1})	Time Step Δt (μs)	Some JC-parameters		
				Strain hard. B (GPa)	Strain rate C	Thermal soft. m
The old method:						
TT13ST	(1)	1000	0.016	0.510	0.014	1.03
TT03ST	(4a)	1000	0.016	0.000	0.014	0
TT10ST	(4b)	1000	0.016	0.510	0	0
TT12ST	(5)	1000	0.016	0.510	0.001	1.03
TT14ST	(6a)	1000	0.016	0.510	0.002	1.03
TT22ST	(6b)	500	0.016	0.510	0.001	1.03
TT92ST	(6c)	1000	0.008	0.510	0.001	1.03
The improved method (implicit strain rate and strain hardening):						
TT13IM	(2)	1000	0.016	0.510	0.014	1.03
Simplified version (only implicit strain rate):						
TT43IM	(7)	1000	0.016	0.510	0.014	1.03
TT40IM	(8)	1000	0.016	0.510	0	0
The new method (default in version 4.2):						
TT13VS	(3)	1000	0.016	0.510	0.014	1.03

The results consist of time plots of the effective stress (von Mises) and the yield stress in a target point (target#4) in a cell at the middle of the bar (axially) and adjacent to the outermost cell (radially). The plots are shown in Figures 1-8. All of these diagrams include a magnification showing details of the curves at or just after onset of plastic deformation. The simulated cases are summarized in Table 2.

The important results are presented in Figures 1-3, which show results from the tensile tests with strain rate 1000 s^{-1}, run with the old method, the improved method, and the new method. The intention behind Figures 4-6 was mainly to investigate the nature of the problem with the oscillations.

We also implemented a slightly *simplified version of the improved method*, where only the strain rate sensitivity was treated implicitly but the strain hardening was treated explicitly. For this method the Eq. (3.1) was changed to

$$Y_{n+1} = Y(\varepsilon_n^p, \frac{\varepsilon_{n+1}^p - \varepsilon_n^p}{\Delta t}, T_n), \tag{4.2}$$

i.e. the first argument in the yield stress function is evaluated at time t_n instead of t_{n+1}. Corresponding changes have to be done in Eqs. (3.3) and (3.4). Results from simulations with this simplified method are shown in Figures 7-8.

4.2. Simulations of projectile penetration

Simulations of a rigid projectile (modelled as elastic with a high yield strength) penetrating a steel plate has been performed by Hansson and Skoglund [13]. The old method and the improved method for Johnson-Cook's model were tested. Parameters were taken from the material library in AUTODYN for "4340 steel", see Table 1. They also ran a simulation with the old method and the strain rate coefficient set to zero (implying no strain rate effects). The results are summarised in Table 3.

Table 3. Results from simulations of projectile penetration into 4340 steel, Ref. [13]

Method	Strain rate coeff. C	Impact velocity (m/s)	Residual velocity (m/s)
The old method	0.014	800	442
The old method	0.000 (no strain rate effect)	800	429
The improved method	0.014	800	375

Table 4. Simulations of projectile penetration in HNS-steel, Ref. [13]

Method	Strain rate coeff. C	In Figure 9 curve number from left to right
The old method	$C = 0$	1
The old method	$C = 0.030$	2
The improved method	$C = 0.030$	3
The new method	$C = 0.030$	4

Simulations using the same geometry as in the preceding penetration simulations but with a different steel quality (HNS) for the target plate were carried out in Ref. [13]. Four different combinations of numerical method and strain rate coefficient were simulated, see Table 4. In order to obtain the ballistic limit, the residual velocity was plotted against impact velocity in Figure 9.

5. DISCUSSION

5.1. Comparison of the three methods

Figures 1-3 show results from simulations of a tensile test of 4340 steel with strain rate equal to 1000 s^{-1}. In Figure 1 the effective stress (`MIS.STRESS`) and yield stress (`YLD.STRESS`) are plotted versus time from a simulation using the old implementation of the Johnson-Cook's model. The second diagram is a magnification of the first one, showing details of the curves near and after onset of plastic deformation. High frequency oscillations are seen in the first diagram. In the second, the time scale of these oscillations is resolved, and it can be seen that the yield stress oscillates every time step (compare with the time step, which is about 16 ns = $16 \cdot 10^{-6}$ ms). The effective stress oscillates with the same frequency, but with lower amplitude. It is always less than or equal to the yield stress (as it should), and every second time step these two stresses have minima and coincide. These curves are, however, expected to coincide all the time after plastic deformation has begun, because the specimen is never unloaded in this simulation. Elastic waves in the bar could lead to local unloading, but not of the kind that is observed in Figure 1. Further analysis shows that the yield stress oscillates roughly around the correct mean value. The effective stress, on the other hand, oscillates closely to the common minima, which correspond to the yield stress for zero strain rate. This causes the increase of the effective stress due to strain rate effects to be too small.

In Figure 2 the same problem as in Figure 1 is simulated using our own improved method. There are no oscillations and the effective stress remains equal to the yield stress after onset of plastic deformation.

The new method, which is default in version 4.2 of AUTODYN, was also tested. As expected the effective stress and yield stress coincided as they did for the improved method. Therefore we plotted only the effective stress from that simulation. Figure 3 shows the effective stress from that simulation and also from the case in the previous figure (the improved method). The two curves overlap, even if a small difference can be seen in the magnification.

The results from the simulation in Table 3 of a projectile penetrating a plate show a significant difference in residual velocity (15%) between the simulations with the old method and our improved method (442 and 375 m/s, respectively). The target material appears weaker with the old method, because this method underestimates the increase of the yield stress due to strain rate. This explanation is supported by the case with no strain rate effect (C = 0), which gave a residual velocity close to that of the simulation with the old method (3% difference).

The results shown in Figure 9 point in the same direction. The curves number two, three, and four from the left show results from simulations with the same material parameters, but different implementations of the constitutive models, namely, the old method, the improved method, and the new method, respectively, see Table 4. The ballistic limit obtained from the old method is significantly lower than those obtained from the improved and new methods. The two latter methods give results that are fairly close to each other. When evaluating the difference between the last two curves, one must consider that they come from different versions of AUTODYN, and that the difference, anyway, is much smaller than the difference between these two curves and the curve from the old method. The two leftmost curves, corresponding to simulations with the old method, are close together despite the fact that one of them corresponds to a case with the strain rate coefficient equal to zero and the other to the case with the normal value for that coefficient.

5.2. Further tests of the old method

The constitutive model we use takes strain hardening, strain rate sensitivity, and thermal softening into account. In Figure 4 the constitutive parameters have been changed so that only one of these effects is present in each case. In Figure 4(a) the constitutive model has only strain rate sensitivity, and in Figure 4(b) there is only strain hardening. The fact that oscillations occur in Figure 4(a) but not in 4(b) suggests that the oscillations have to do with the strain rate dependence in the material model and not with strain hardening.

The case presented in Figure 5 is the same as that in Figure 1 with the exception that the strain rate coefficient is lowered to $C = 0.001$ (from 0.014). In Figure 5 the small oscillations that occur just after onset of plastic deformations are damped out very soon, and after that the simulation appears to be stable. Figure 6 contains three variations of the stable case in the preceding figure. In all three there are un-damped oscillations. Since we know that oscillations will occur if C becomes as large as 0.014 we increase C from 0.001 to 0.002 in order to see if this is sufficient to cause oscillations. It turns out to be sufficient. It is, however, surprising that the remaining two variations also lead to oscillations. In Figure 6(b) the strain rate has been lowered from 1000 to 500 s^{-1} and in Figure 6(c) the time step has been decreased from 16 to 8 ns (by changing the time step safety factor from the default 0.6666 to the new value 0.3333.).

5.3. Explanation of the oscillations

As we have seen, the material deforms plastically only every second time step when oscillations occur in the old method. This means that the plastic strain rate during these cycles must be twice the average strain rate, which is 1000 s^{-1} in our baseline case. Therefore the strain rate causes the yield stress to increase by the factor $1 + C\ln(2000) = 1.11$, if $C = 0.014$; cf. Eq. (4.1). By looking at Figure 1, one may see that the maxima and minima of the oscillating yield stress differ by that factor.

What happens is the following: The trial stress tensor breaks through the yield surface and is then projected back to it. This back projection means plastic deformation, and the strain rate will cause the yield surface to move outwards, so that the stress tensor will lie inside of the yield surface. In the next time step the trial stress tensor will move outwards again. The key question is, *does it move far enough to cross the yield surface*. In the situation, where oscillations of the type shown in Figur 1 occur, the answer is no. The trial stress tensor will be inside of the yield surface, and the material considered elastic, leading to zero plastic deformation and zero strain rate. This zero strain rate is used in the next time step for calculating the yield stress, which will therefore fall back to its static position (disregarding for the moment strain hardening and thermal softening). In the next time step the trial stress tensor will be outside of the yield surface and process will repeat itself.

Returning to the key question in the preceding paragraph, one sees that the difficulties occur if the influence of strain rate on the yield stress is large and the difference between the effective stress of the trial stress tensor and that of the old stress tensor is small. That difference is proportional to the strain increment during a time step. Therefore small strain rates and small time steps contributes to the difficulties. It is not so surprising that large strain rate sensitivity (large *C*) may cause problems, but it is surprising (at least to me) that also small strain rates and small time steps may do so, cf. the three cases in Figure 6. A method that does not work for small time steps can, of course, not converge when the time step tends to zero.

The velocities and strain rates occurring in ballistic problems have not change so much over the years (at least not orders of magnitude), but the time steps have, due to finer grids, possible because of the development of the computers. Since the old implementation of strain rate dependence may work well in some cases (Figure 5) and works better for larger time steps, it may very well be the case that the oscillations were not seen when strain rate dependent models were introduced.

5.4. Strain hardening and temperature dependence

The simplified version of the improved method, where the strain rate sensitivity is treated implicitly but strain hardening is not, is still implicit. The gain in computational efficiency is therefore marginal. The reason for studying the simplified method is to investigate the necessity of treating all history variables implicitly. Figure 7 shows results from simulations with the original parameters for 4340 steel, whereas in Figure 8 both the strain rate and temperature dependence are ignored by setting $C = 0$ and $m = 0$ (indicating no temperature dependence). These figures should be compared to Figure 2 and Figure 4(b), respectively, where the same cases are simulated with the improved method itself. The similarity is very good, indicating negligible influence of the simplification on the results. Since an explicit treatment of the strain hardening is sufficient one may hope that the same could be true for other history variables like the temperature.

5.5. Motivation for the improved version

The improved method has shown to behave stable even in situations were the old method gave rise to erroneous oscillations. A common sense motivation for the improved method is that the time centring of the strain rate in the formula for the yield stress looks better than in the old method, where the strain rate is at least one time step too old. A way of finding a theoretical motivation would be to carry out a stability analysis for the method as in Ref. [5]. Instead of doing that we have studied numerical algorithms for solving a very simple problem with a Maxwell type material in Appendix B. Our improved method, applied to that problem, turns out to be equivalent to an established numerical method, namely, the Euler backwards scheme. This could perhaps give some support for the soundness of the method.

6. CONCLUSIONS

Our implicit method for treating the Johnson-Cook’s constitutive model in the hydrocode AUTODYN-2D leads to smooth curves for the material response. The yield stress is correctly increased due to the influence from strain rate. The new “strain rate corrected” option in the latest version of AUTODYN (v. 4.2) was also tested and found to give almost identical result as our implicit method.

The erroneous oscillations that were present in the old standard implementation of the model are due to the treatment of strain rate, because this implementation works well if the strain rate influence is removed from the constitutive model and only strain hardening and thermal softening are kept. The good behaviour of the simplified version of the improved method suggests that it might not be necessary to treat all history variables implicitly.

ACKNOWLEDGEMENTS

The authors of Ref. [13], Håkan Hansson and Peter Skoglund, are acknowledged for having pointed out the seriousness of the problem addressed in this report, and for giving permission to use a preprint of their report as reference and to use one of their diagrams (Figure 9).

REFERENCES

1. G. R. Johnson and W. H. Cook. A constitutive model and data for metals subjected to large strains, high strain rates, and high temperatures. *Proc. 7th. Int. Symp. on Ballistics,* Holland*,* pp. 541-547, 1983.
2. AUTODYN – Interactive Non-Linear Dynamic Analysis Software, Theory Manual. Century Dynamics Inc., USA, 1998.
3. N. K. Birnbaum, M. S. Cowler, M. Itoh, M. Katayama, and H. Obata. AUTODYN - an interactive non-linear dynamic analysis program for microcomputers through supercomputers. *9th Int. Conf. on Structural Mechanics in Reactor Technology,* August 1987, Lausanne, Switzerland.
4. N. K. Birnbaum and M. S. Cowler. Numerical simulation of impact phenomena in an interactive computing environment. In *Impact Loading and Dynamic Behavior of Materials* , Vol. 2, pp. 881-888 (Edited by C. Y. Chiem, H.-D. Kunze and L. W. Meyer), DGM Informationsgesellschaft mbH, Oberursel, 1988.
5. S. A. Silling. Stability and accuracy of differencing methods for viscoplastic models in wavecodes. *J. Comp. Phys*. Vol. 104, pp. 30-40, 1993.
6. J. M. McGlaun, S. L. Thompson, and M. G. Elrick. CTH: A three-dimensional shock wave physics code. *Int. J. Impact Engng* Vol. 10, pp. 351-360, 1990.
7. L. Nilsson. Material behaviour in dynamic loading situations – plasticity and visco-plasticity. In *Material test procedures in support of dynamic material modelling* (Edited by L. Ågård), *FOI-report*, FOA-R--99-01227-311--SE, 1999.
8. J. O. Hallquist. LS-DYNA3D Theoretical manual. Livermore Software Technology corp., 1991.
9. P. Perzyna. Fundamental problems in viscoplasticity. *Advances in applied mechanics*, 9 (1966), pp. 243-377.
10. O. Wall. Viscoplastic material behaviour in the transition region. Licentiate Theses, Solid Mechanics, Lund Institute of Technology, 1998.
11. F. J. Zerilli and R. W. Armstrong. Dislocation-Mechanics based constitutive relation for material dynamics calculations. *J. Appl. Phys*., Vol. 61, No. 5, 1987.
12. M. L. Wilkins. Calculation of Elastic-Plastic Flow. In *Methods of Computational Physics*, Vol. 3, pp. 211-263 (Edited by B. Alder, S. Fernback and M. Rotenberg). Academic Press, New York, 1964.
13. P. H. Hansson and P. Skoglund. Modeling of steel behaviour with application to armour penetration. *FOI-report,* FOI-R--0201--SE, 2001.

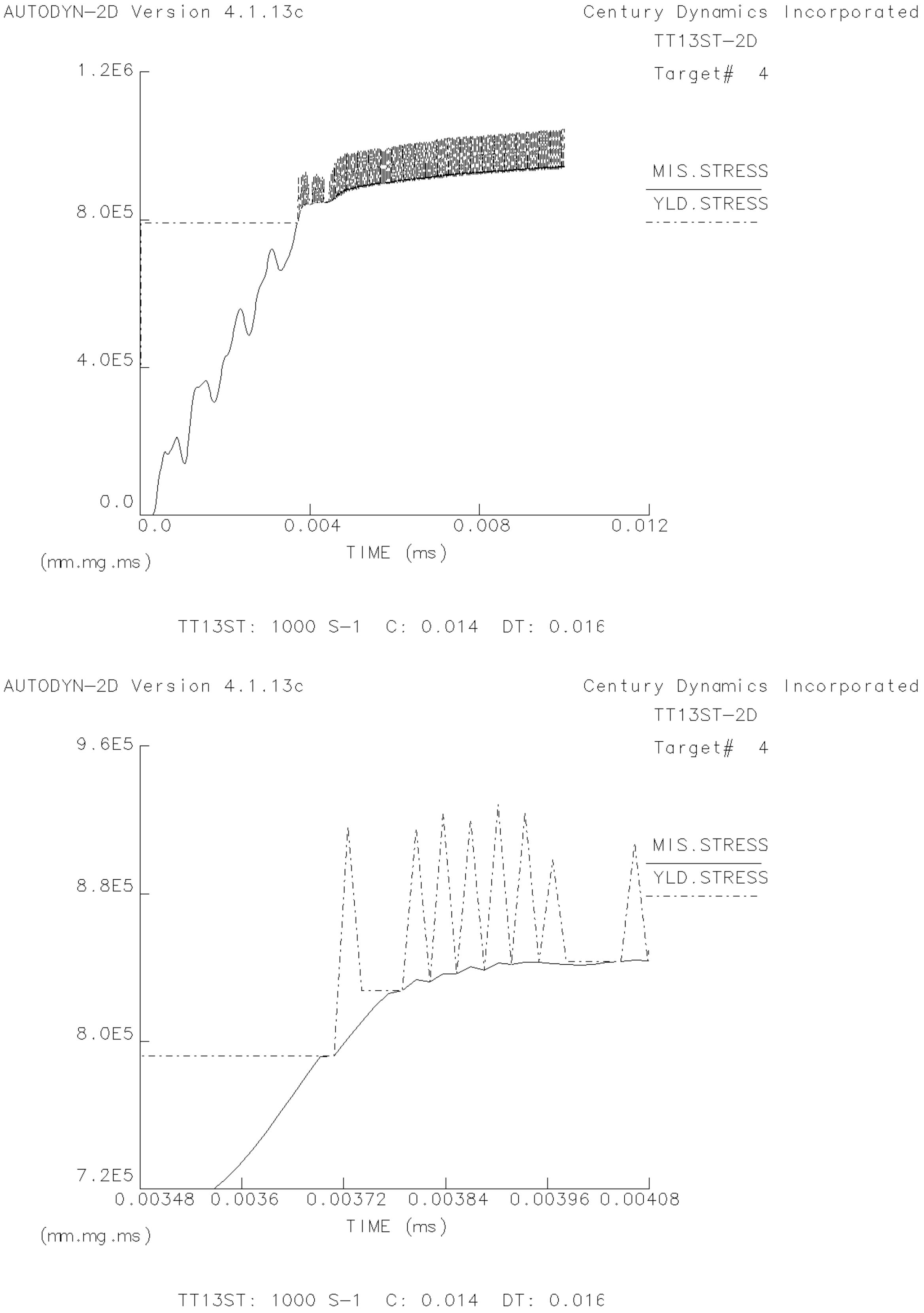
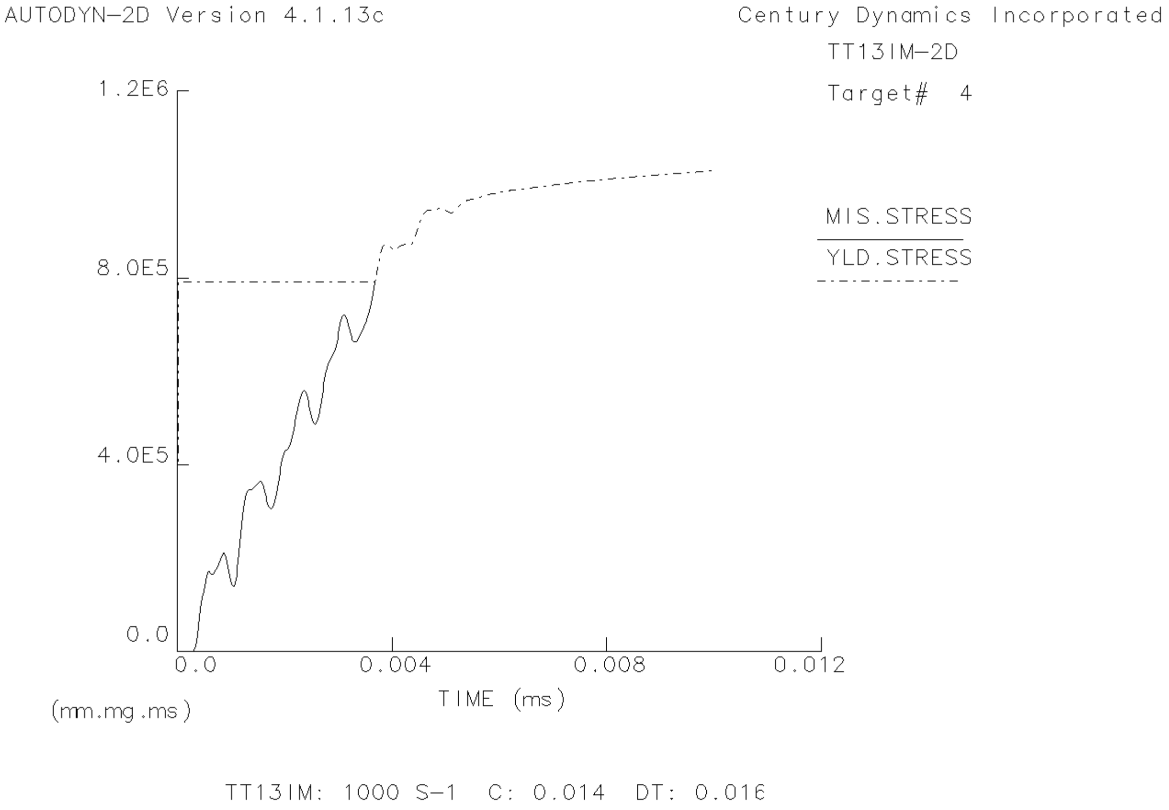


Figure 1. Yield stress and von Mises effective stress vs. time. The old method. Parameters for 4340 steel.



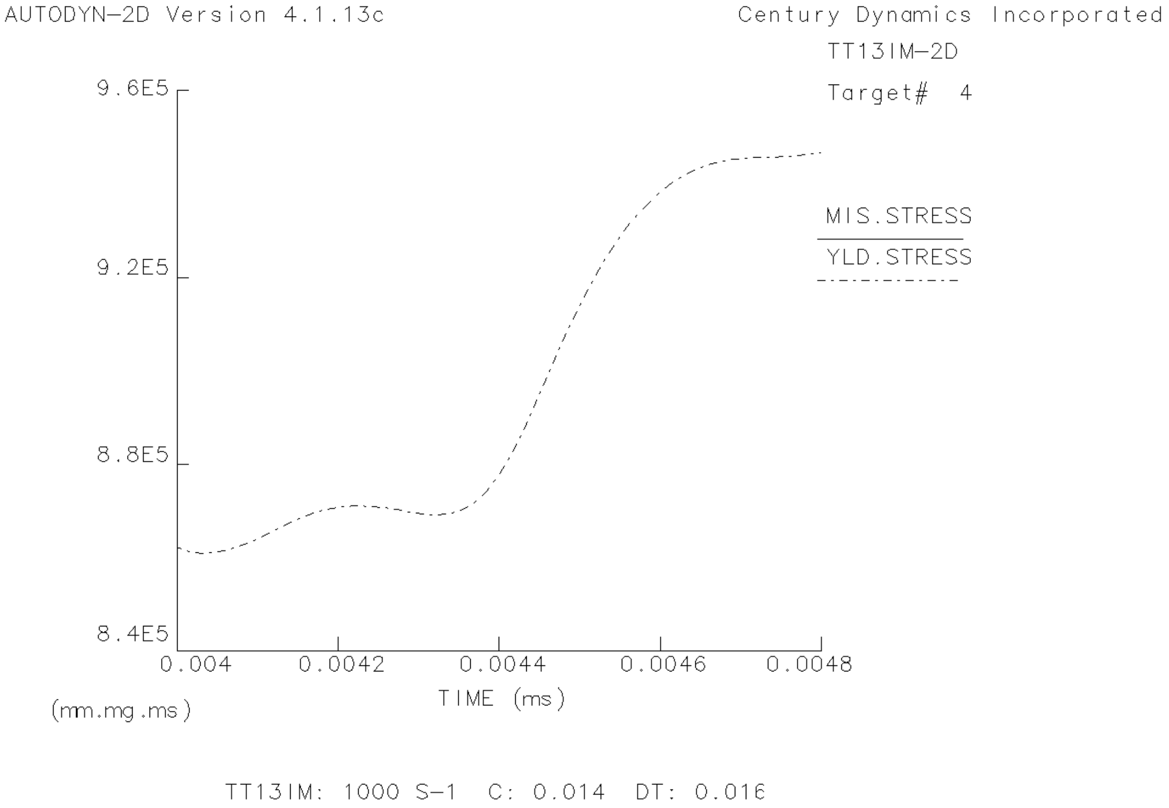
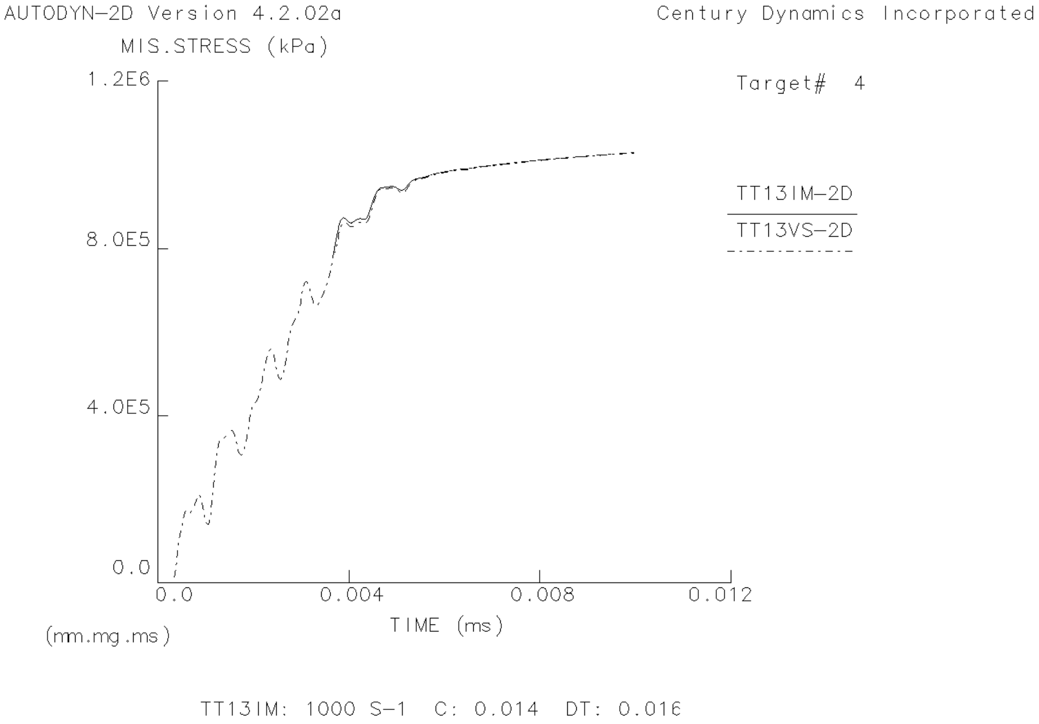


Figure 2. Yield stress and von Mises effective stress vs. time. The improved method, implemented as a user-subroutine in AUTODYN. Parameters for 4340 steel.



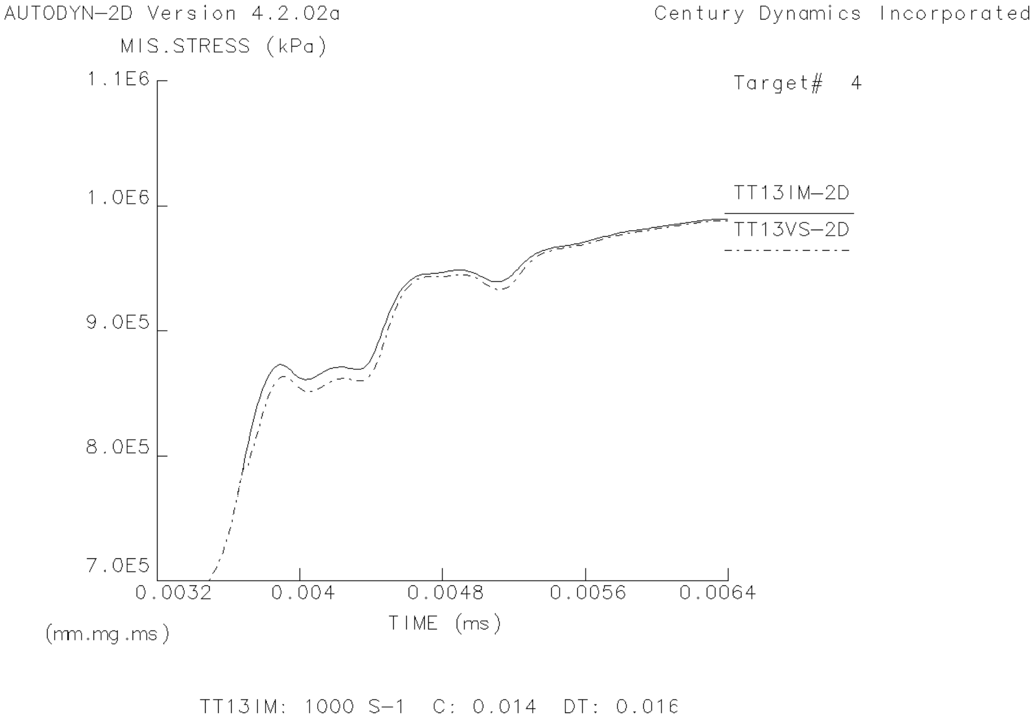


Figure 3. Yield stress and von Mises effective stress vs. time for 4340 steel. Two methods are compared, namely, the improved method (solid curve), implemented as a user-subroutine in AUTODYN, and the new method in AUTODYN version 4.2 (dashed curve).

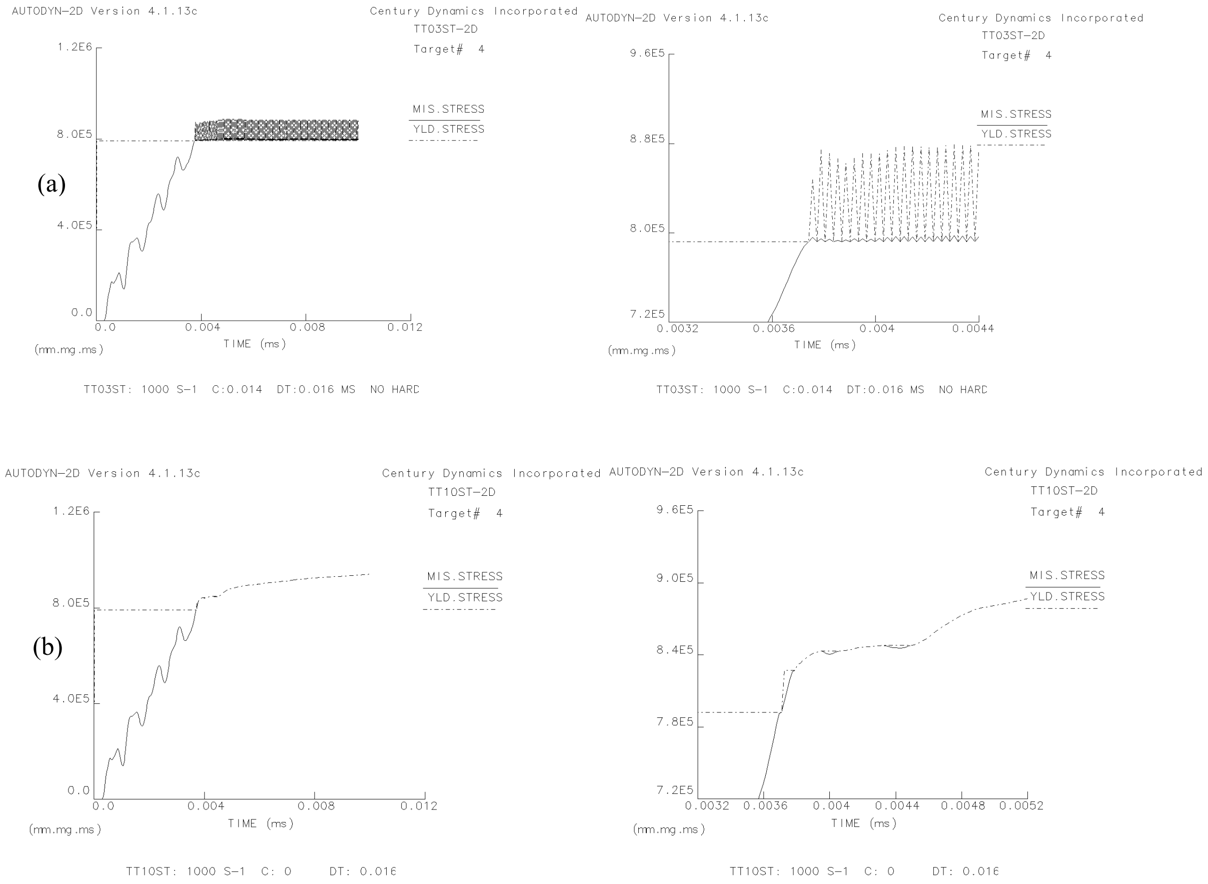


Figure 4. Yield stress and von Mises effective stress vs. time. The old method. Parameters for 4340 steel, except
(a) that strain hardening and thermal softening are turned off (i.e. only strain rate sensitivity is on)
(b) that strain rate sensitivity and thermal softening are turned off (i.e. only strain hardening is on).

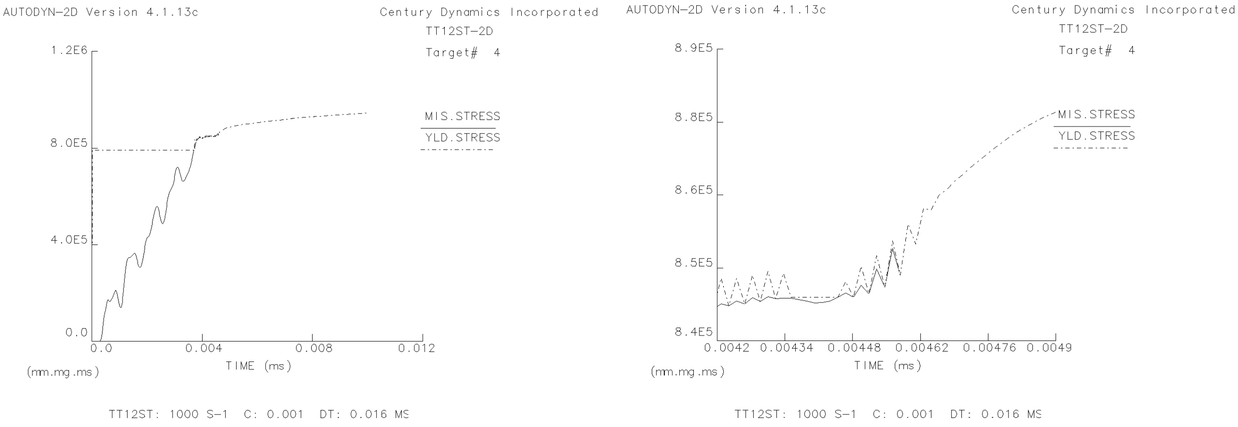


Figure 5. Yield stress and von Mises effective stress vs. time. The old method. Parameters for 4340 steel, except for the strain rate coefficient, which is set to a very low value C = 0.001, so that un-damped oscillations do not occur.

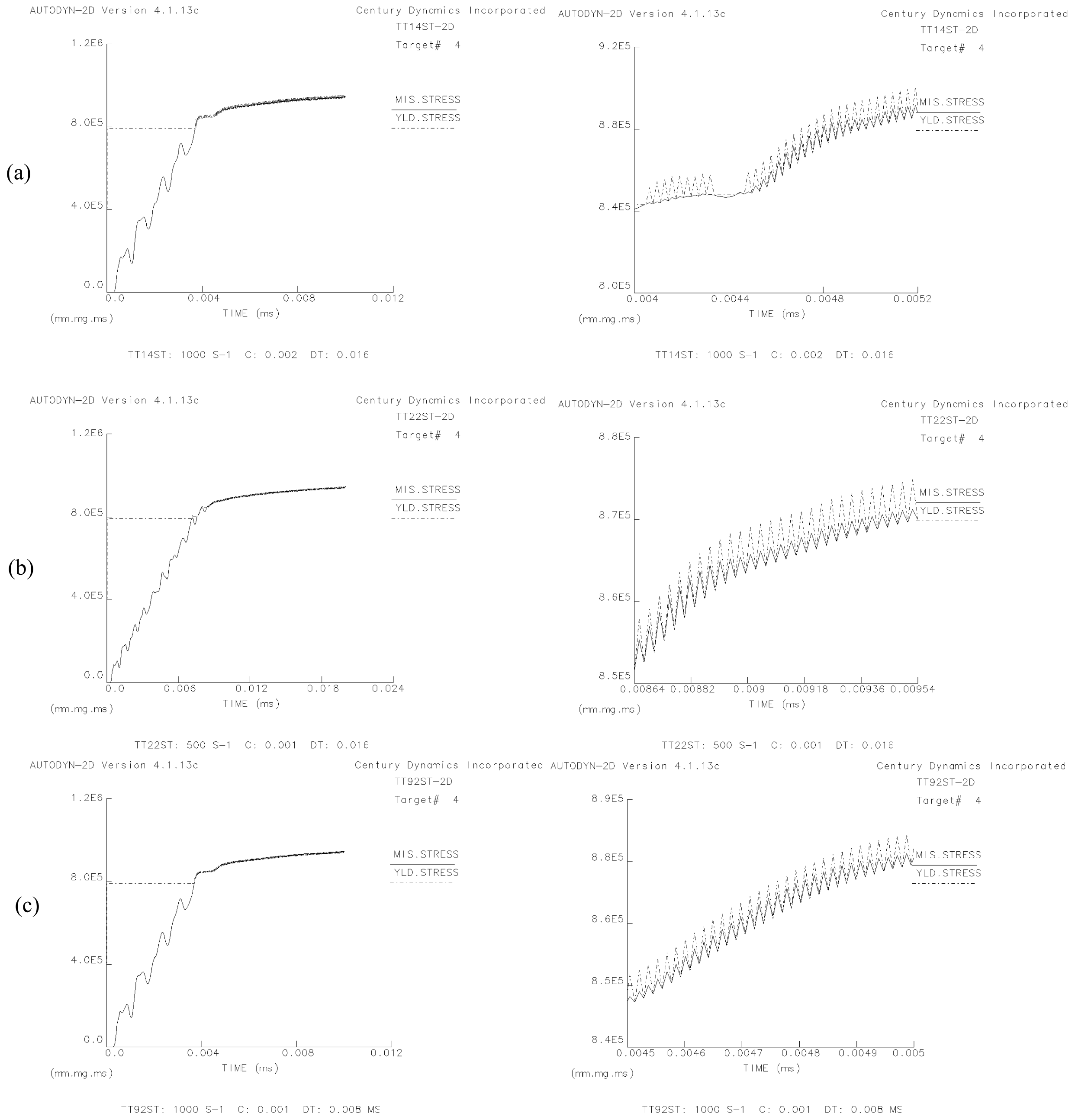


Figure 6. Yield stress and von Mises effective stress vs. time. The old method. Parameters as in figure 5, except for

(a) strain rate coefficient, which is doubled to C = 0.002,

(b) applied strain rate, which is decreased from 1000 to 500 s^{-1},

(c) time step, which is decreased from 16 to 8 ns

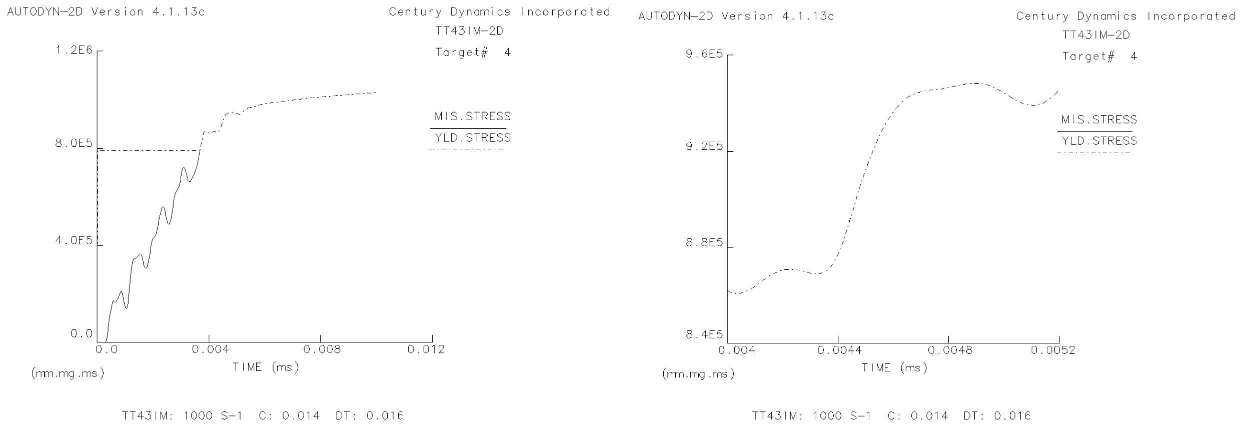


Figure 7. Yield stress and von Mises effective stress vs. time. The simplified version of the improved method. Parameters for 4340 steel.

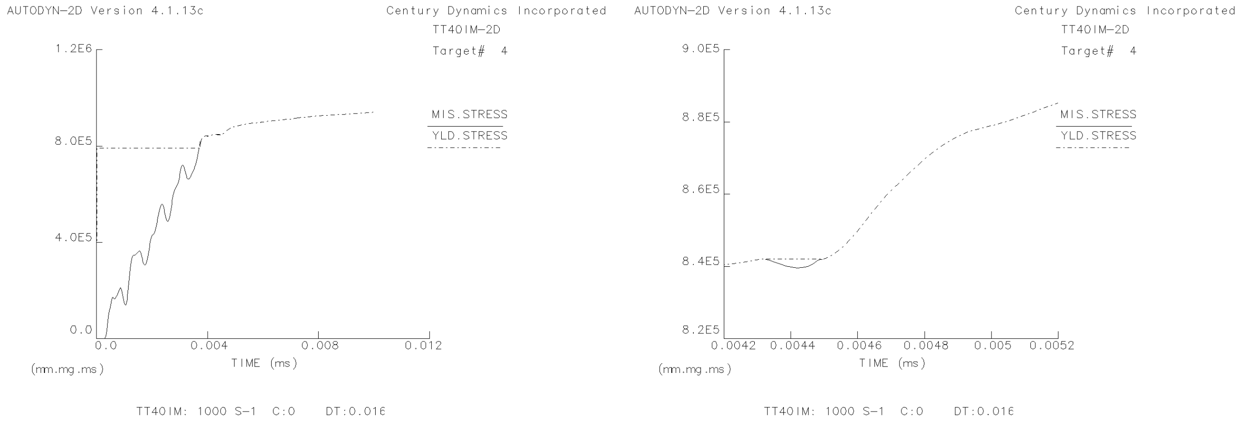


Figure 8. Yield stress and von Mises effective stress vs. time. The simplified version of the improved method. Parameters for 4340 steel, except that strain rate effects and thermal softening are turned off, i.e. only strain hardening remains.

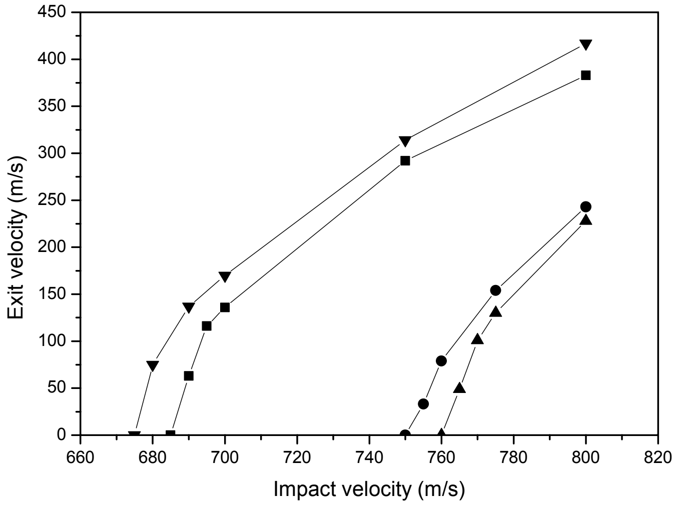


Figure 9. Residual velocity (or exit velocity) vs. impact velocity for a rigid projectile penetrating a plate of HNS steel. The left curve represents simulations with the old method and no strain rate dependence (C = 0), whereas the remaining three curves correspond to the data for HNS steel (according to Table 1) but to different methods, namely, the old method, the improved method, and the new method, respectively, from left to right. See also Table 4. The diagram is taken from Ref. [13].

APPENDIX A. THE USER MODEL

A1. Overview

A file with user-subroutine skeletons are supplied with Autodyn. Our implementation of Johnson-Cook's constitutive model in Autodyn is done by completing the user subroutine `EXYLD`, in which the yield limit is calculated. The Fortran 90 technique of having internal subroutines is used, and `EXYLD` contains several subroutines. The most important are `STRENGTH`, `JC`, and `SOLVE`. The purpose of the subroutine `STRENGTH` is to evaluate a function, namely, the yield stress as a function of plastic strain, plastic strain rate, and temperature, and also to calculate the partial derivative of this function with respect to the first two variables. `STRENGTH` is an administrative routine, which calls `JC` if Johnson-Cook (model = 1) is selected and some other routine (not yet implemented) if another constitutive model is used. For the time being `ZA` is only a dummy routine. In `SOLVE` the Newton-Raphson iterations are carried out.

Table A1. Input parameters to the user implementation of Johnson-Cook's model

Strength constant array	Fortran variable	Physical variable	Description
`SC(1)`	`g`	G	Shear modulus
`SC(2)`	`aa`	A	Static yield stress
`SC(5)`	`bb`	B	Strain hardening coefficient
`SC(6)`	`nn`	n	Strain hardening exponent
`SC(7)`	`cc`	C	Strain rate coefficient
`SC(8)`	`mm`	m	Thermal softening exponent
`SC(9)`	`tempref`	T_0	Reference temperature
`SC(10)`	`tempsm`	T_m	Melting temperature
`SC(11)`	`epsdot0`	$\dot{\varepsilon}_0$	Strain rate threshold
`SC(12)`	`TOL`		Tolerance for the relative error in strain rate
`SC(20)`	`method`		= 1 Only strain rate is treated implicitly = 2 Also strain hardening is treated implicitly
`SC(21)`	`model`		= 1 Johnson-Cook = 2 (Not implemented)

The material parameters for Johnson-Cook's model, which are input data to the user subroutine, are supplied via the `SC(.)` array. The details are found in Table A1. The four last parameters in Table A1 might be confusing to the user. Therefore in some versions of the user routine, the user's input values are overridden or changed to a proper default value if the user has left them as zero. In the current version, the user inputs for `SC(20)` and `SC(21)` are ignored and they are set internally to 2 and 1, respectively. If the tolerance `SC(12)` is left as zero by the user, the program will set it equal to a variable called `EPSPP5`, which is supplied by Autodyn in the module `KINDEF` and has the value 10^{-5} in single precision and 10^{-10} in double.

A2. Miscellaneous details.

The declarative part of `EXYLD` including descriptions of input and output parameters is found in Table A2.

In order to simplify the notation here, we write the function, expressing the yield stress as $Y = Y(u,v)$, where u is the plastic strain and v is the plastic strain rate. Its partial derivatives are $\partial Y/\partial u$ and $\partial Y/\partial v$. The temperature argument in the function *Y* is not written explicitly since it is always T_n, the temperature at the time t_n. The function *Y* is defined as (for Johnson-Cook's model)

$$Y(u,v) = \left(A + Bu^n\right)\left(1 - T_H^m\right), \qquad v \le \dot{\varepsilon}_0^p \qquad \text{(A2.1)}$$

$$Y(u,v) = \left(A + Bu^n\right)\left(1 + C\ln\left(v/\dot{\varepsilon}_0^p\right)\right)\left(1 - T_H^m\right). \qquad v \ge \dot{\varepsilon}_0^p \qquad \text{(A2.1)}$$

Table A2. Formal parameters and declarations in subroutine EXYLD

```
SUBROUTINE EXYLD (PRES,TT1,TT2,TT3,XMU1,EPS1,EPSD,TEMP1,YIELD1,IFAIL)
USE KINDEF
USE CYCVAR
USE ...

! INPUT PARAMETER

      !    PRES    PRESSURE
      !    Tnn     PRINCIPAL STRESSES
      !    XMU     COMPRESSION
      !    EPS1    EFFECTIVE PLASTIC STRAIN
      !    EPSD    EFFECTIVE PLASTIC STRAIN RATE
      !    TEMP1   TEMPERATURE
      !    IFAIL   STRESS STATE INDICATOR
      !            = 0   HYDRO
      !            = 1   ELASTIC
      !            = 2   PLASTIC
      !            = 3   BULK FAILURE (WITH HEAL)
      !            > 3   BULK FAILURE (NO HEAL)
! OUTPUT PARAMETERS

      !    YIELD1     YIELD STRESS FOR CURRENT MATERIAL
      !    IFAIL     STRESS STATE INDICATOR (SEE ABOVE)
...
...
...
      END SUBROUTINE EXYLD
```

In order to calculate the yield stress for time t$_{n+1}$, we have to solve Eq. (3.4) (in Sec. 3 of the main text) for the plastic strain increment. The yield stress is then found to be equal to the right member of that equation. We reformulate the equation in terms of plastic strain rate, which we call *x* in this context and obtain

$$\sigma_{eff}^{*p} - (3G\Delta t)x = Y(\varepsilon_n^p + x\Delta t, x)\,. \qquad \text{(A2.3)}$$

Here $\Delta t = t_{n+1} - t_n$ is the time step called DLTH in the Fortran code of Autodyn and is supplied in the module CYCVAR. The shear module *G* and the material parameters in the expression for *Y* are taken from the array SC(.) in module MATDEF, see Table A1. The effective stress for the trial stress tensor σ_{eff}^{*} is calculated from the principal stresses in the usual way

$$\sigma_{eff}^{*} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}, \qquad \text{(A2.4)}$$

which is easy, since the principle stresses for the trial stress tensor are in the parameter list of EXYLD. They are called TT1, TT2, and TT3.

We write Eq. (A2.3) as

$$F(x) \equiv \sigma_{eff}^{*} - (3G\Delta t)x - Y(\varepsilon_n^p + x\Delta t, x) = 0 \tag{A2.6}$$

and solve it by Newton-Raphson's iterative method

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} \tag{A2.7}$$

where

$$F'(x) = -3G\Delta t - \Delta t \frac{\partial Y}{\partial u}(\varepsilon_n^p + x\Delta t, x) - \frac{\partial Y}{\partial v}(\varepsilon_n^p + x\Delta t, x) \tag{A2.8}$$

and $F(x)$ is given by Eq. (A2.6).

There are some difficulties that are addressed below. We first list them.

(i) $F(x)$ is not differentiable for $x = \dot{\varepsilon}_0^p$, since $Y(u,v)$ is not differentiable with respect to v for $v = \dot{\varepsilon}_0^p$, which is a consequence of the fact that $Y(u,v)$ is defined by different expressions depending on whether v is below or above the threshold value for strain rate effects, $\dot{\varepsilon}_0^p$, see Eqs (A2.1-2).

(ii) $F'(0)$ may be non-existent, since $\partial Y/\partial u$ does not exist for $u = 0$ in the common situation when the strain hardening exponent n is less than 1. This problem occurs only if the cumulated plastic strain $\varepsilon_n^p = 0$, see Eq. (A2.8). It is interesting to note that a slight problem exists also for n = 1, despite that the derivative certainly exists in that case. If one tries to evaluate the derivative of x^n, namely, nx^{n-1}, for $x = 0$ in the special case when $n = 1$, a run time error will result.

(iii) $F(x)$ is defined only for $x \geq 0$, so x-values during the iteration must stay in that interval, or rather because of (ii) be positive. A related problem is that Newton-Raphsons's method need not generally converge if the initial guess is too bad.

(iv) In the spirit of AUTODYN programming, we would like the user subroutine to fit into both the single and double precision versions of the code without any changes.

The left member of Eq. (A2.3) is a decreasing linear function of x, and the right member is an increasing function of x. Therefore $F(x)$ is a decreasing function of x, and the routine EXYLD distinguishes between the following three cases:

Case 1: $$F(0) \equiv \sigma_{eff}^{*} - Y(\varepsilon_n^p, 0) \leq 0 \tag{A2.9}$$

Case 2: $F(0) \equiv \sigma_{eff}^{*} - Y(\varepsilon_n^p, x) > 0$, and

$$F(\dot{\varepsilon}_0^p) \equiv \sigma_{eff}^{*} - (3G\Delta t)\dot{\varepsilon}_0^p - Y(\varepsilon_n^p + \dot{\varepsilon}_0^p \Delta t, \dot{\varepsilon}_0^p) \leq 0 \tag{A2.10}$$

Case 3: $$F(\dot{\varepsilon}_0^p) \equiv \sigma_{eff}^{*} - (3G\Delta t)\dot{\varepsilon}_0^p - Y(\varepsilon_n^p + \dot{\varepsilon}_0^p \Delta t, \dot{\varepsilon}_0^p) > 0 \tag{A2.11}$$

In case 1 the trial stress tensor is on or inside the yield surface that corresponds to zero strain rate. This means that no plastic strain will occur during the time step and therefore $x = 0$ is considered as the solution. In case 2 the solution satisfies

$$0 < x \leq \dot{\varepsilon}_0^p, \tag{A2.12}$$

i.e. the strain rate x is below the threshold for strain rate effects. In case 3 the solution x is above the threshold,

$$x > \dot{\varepsilon}_0^p \, . \qquad \text{(A2.13)}$$

Difficulty (i) is solved by the possibility to distinguish between these cases before the iteration starts. The trick is to use only one of the expressions for $Y(u,v)$ during the iteration. In case 2, when the solution is known to be below the threshold, we use the expression (A2.1), for $Y(u,v)$ when calculating $F(x_k)$ and $F'(x_k)$ even if x_k occasionally should be above the threshold. Analogously expression (A2.2) is used in case 3.

Difficulty (ii) and (iii). By analysing the graph of $F(x)$ or alternatively the curves represented by the two members of Eq. (A2.3), it is easily seen (at least for reasonable values of the constitutive parameters) that F apart from being a decreasing function has a positive second derivative. From this it follows that the iteration converges monotonically from all values less than the true solution, and that all values greater than the true solution will in the next step of the iteration lead to a value less than the true solution.

Iterations are performed only in case 2 and 3, and in these cases the solution is known to be positive, i. e. $x > 0$. Zero can not be used as an initial value because of (ii). Instead we use the threshold value $x = \dot{\varepsilon}_0^p$ as initial value for the iterations, when the cell becomes plastic for the first time. In the later iterations the preceding value of the plastic strain rate is used as initial value. If the x-value becomes negative during the iteration it is natural to throw it back on the positive side, but how far? The following procedure is adopted: Assume $x_k > 0$ and $x_{k+1} \le 0$. Then we set x_{k+1} equal to $0.01 x_k$, which is positive. Should the negative values persist this procedure eventually leads to a value less than the true solution, and from that the convergence will be monotone.

Difficulty (iv). In Autodyn floating point variables have declarations of the type

```
REAL(REAL8):: ZZ
```

where `REAL8` is set to an appropriate value in a module called `KINDEF`, which is different in the single and double precision versions of the code. Therefore the declarations can be the same in both versions. Floating point constants have to be avoided in subroutine calls, since they are written differently in single and double precision, for instance 1.0 and 1.0D0, respectively. Use of wrong type will pass a totally wrong value to the subroutine. However,

```
USE  KINDEF
REAL(REAL8):: ZZ
...
ZZ = 1.D0
CALL SUB(ZZ)
```

does work for both precisions, since a proper type conversion is done in the assignment statement in single precision.

A3. The subroutine `STRENGTH`

This subroutine is purely administrative and is listed in its entirety in Table A3. The arguments are passed along directly to the next routine. This routine is needed when more than one constitutive model is included in the user routine.

Table A3. Subroutine strength

```
subroutine Strength(isw,ideriv1,ideriv2,ideriv3, &
                    u,v,      sig,dsigdu,dsigdv)

integer(int4):: isw   !INPUT: isw = 0 below threshold
                      !       isw = 1 above threshold

integer(int4):: ideriv1,ideriv2,ideriv3
!        INPUT: ideriv1 = 1, Calculate the derivative dsig/du
!               ideriv1 = 0, Do not calculate the derivative
!               ideriv2 = 1, Calculate the derivative dsig/dv
!               ideriv2 = 0, Do not calculate the derivative
!               ideriv3 = dummy

real(real8):: u, v    !INPUT:  plastic strain, plastic stain rate

real(real8):: sig, dsigdu, dsigdv
  !OUTPUT: Yield limit,
  !        derivative of yield limit with respect to plastic strain
  !        derivative of yield limit with respect to plastic strain rate

      if (imodel .eq. 1) then
         call JC(isw,ideriv1,ideriv2,ideriv3,u,v,     sig,dsigdu,dsigdv)
      else if (imodel .eq.2) then
          call ZA(isw,ideriv1,ideriv2,ideriv3,u,v,    sig,dsigdu,dsigdv)
      endif
      end subroutine Strength
```

The first parameter ISW is used as a switch to choose between the two expressions (A2.1) and (A2.2) for the yield stress. The next parameters IDERIV1 and IDERIV2 are used to indicate whether or not the partial derivatives shall be calculated. IDERIV3 is a dummy parameter. The parameters U and V are plastic strain and plastic strain rate, respectively. The three last arguments are output parameters: SIG is the yield stress, and DSIGDU, DSIGDV are the partial derivatives.

A4. The subroutine JC

The subroutine JC, see Table A4, is the routine that is specific to the Johnson and Cook's model. It is called from STRENGTH, which passes along all the parameters with the same meaning and declarations. They are therefore explained only in connection with that routine in the preceding section.

Table A4. Formal parameters and declarations in subroutine JC

```
subroutine JC(isw,ideriv1,ideriv2,ideriv3,u,v,   sig,dsigdu,dsigdv)
!  The parameters have the same meaning as in the calling
!  subroutine STRENGTH
…
…
…
end subroutine JC
```

A5. The subroutine SOLVE

In the subroutine SOLVE, see Table A5, the iteration is carried out. The first four parameters are passed unchanged to the call of STRENGTH, but with slightly changed meaning for three of them. The first, ISWITCH selects whether the solution is going to be searched below or above the threshold. IDERIV1 selects if the strain hardening is going to be treated in the more accurate implicit way or the explicit way. IDERIV2 does the same thing for the strain rate. IDERIV3 is a dummy parameter (intended for the temperature). The intention with IDERIV1=IDERIV2=0, is that SOLVE should calculate the same yield stress as the standard model in Autodyn would do. However, IDERIV2=0 is not implemented. In order to obtain the improved method, SOLVE should be called with IDERIV1=IDERIV2=1. It is easy to implement IDERIV1=0 (explicit treatment of strain hardening), by simply cancelling the term $x\Delta t$ in the first argument of Y in Eq. (A2.3) and follow up the consequences of that in subsequent equations.

If XSTART is less than or equal to zero it is reset to the threshold value. During the iteration (when x_2 is calculated from x_1) x_1 is guarantied to be positive by a procedure described in the end of Sec. A2 of this appendix.

A maximum of 25 (for the time being) iterations is carried out. The iteration is stopped when the relative difference between two consecutive values of the strain rate is less than a tolerance TOL, i. e.

$$\left|x_2 - x_1\right| < (\mathrm{TOL})\left|x_2\right| \tag{A5.1}$$

Here TOL = SC(12) is an input value, see Table 1. The yield stress is then calculated by

$$Y = Y(\varepsilon_n^p + x\Delta t, x)\,. \tag{A5.2}$$

Table A5. Formal parameters and declarations in subroutine SOLVE

```
subroutine solve(iswitch,ideriv1,ideriv2,ideriv3,  &
                 eps_old,SigTrial,xstart,   x,sig)

real(real8):: eps_old, SigTrial,xstart
                 !INPUT: old strain,
                 !       effective stress for trial stress tensor
                 !       strat value for x (strain rate)

integer(int4):: iswitch,ideriv1,ideriv2,ideriv3
                 !INPUT: iswitch = 0 below strain rate threshold
                 !       iswitch = 1 above strain rate threshold
                 !       Switches for impliciteness.
                 !       ideriv1 = 0 strain hardening explicit
                 !       ideriv1 = 1 strain hardening implicit
                 !       ideriv2 = strain rate explicit (not allowed)
                 !       ideriv2 = strain rate implicit
                 !       ideriv3 = dummy

real(real8):: x,sig
                 !OUTPUT: strain rate,
                 !        yield limit
…
…
end subroutine solve
```

APPENDIX B. A SIMPLE ONE-DIMENSIONAL PROBLEM

B1. Formulation of the problem for constant friction

A box standing on a table can be pushed and pulled with a spring which is attached to the box at one end, see Fig. B1. The friction force between the box and the table is first assumed to be constant. Later on we will consider non-constant friction. We assume an *x*-axis along which the box moves. Let the position of the end of the spring, where the pulling force is applied (point A), be $x = f(t) + L$ and the position of a reference point on the box be $x = u(t)$. By choosing the constant L appropriately the tensional force in the spring can be written as

$$p(t) = k(f(t) - u(t)), \tag{B1.1}$$

where k is a positive constant. For time $t = 0$ we assume $f(0) = u(0)$. As a first example, we assume a constant friction force Y between the box and the table.

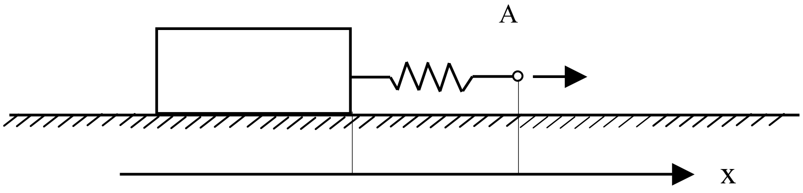


Figure B1.

The problem that we are going to study can be formulated as follows: *Determine the force $p(t)$ in the spring when the position of point A, namely $f(t)$, is given for $t \geq 0$.* Because of the relation (B1.1) between $p(t)$ and $u(t)$, the problem could equivalently be formulated as: determine the position $u(t)$ of the box when $f(t)$ is given.

B2. Solution of the discrete problem for constant friction

We are going to study a time discrete version of this problem, and we therefore assume that f is given at times $0 = t_0 < t_1 < t_2 < t_3 < \ldots$ as $f_n = f(t_n)$, and we seek $p_n = p(t_n)$ and $u_n = u(t_n)$. For simplicity of notation the time step is assumed constant and equal to Δt. This restriction could easily be removed in the algorithms presented here. The discrete counterpart of Eq. (B1.1) is

$$p_n = k(f_n - u_n). \tag{B2.1}$$

At time zero we have $p_0 = 0$ and $u_0 = 0$. When p_n is known, p_{n+1} is calculated in the following way: First, we carry out the movement of point A and constrain the box to stand still, so that the force in the spring becomes $p^* = p_n + k(f_{n+1} - f_n)$. If this trial force p^* is less than the friction force Y in absolute value, i.e. $|p^*| \leq Y$, then the trial force is accepted as the new force and the box does not change its position during the time step: $p_{n+1} = p^*$ and $u_{n+1} = u_n$. In the opposite case when $|p^*| > Y$ the box is moved slightly, so that the absolute value of the force in the spring is decreased to Y, and we get $p_{n+1} = Y\,\mathrm{sgn}(p^*)$. From Eq. (B1.1) it is seen that in order to accomplish this, the box has to be moved the distance $(|p^*| - Y)/k$, in positive direction if p^* is positive and in negative direction if p^* is negative. This can be written as $u_{n+1} = u_n + (1/k)(|p^*| - Y)\,\mathrm{sgn}(p^*)$.

Here we have had to use the sign function defined as sgn(x) = 1 if $x > 0$, sgn(0) = 0, and sgn(x) = -1 if x < 0. If the attention had been restricted to the special case, where the box is pulled only in one direction all the time, the complication of having to use the sign function and absolute signs in several places in the formulas, would have disappeared.

B3. Formulation of the problem for non-constant friction

Let us first introduce some notations. The velocity of the box is

$$v(t) = \frac{du}{dt} = u'(t)\,, \tag{B3.1}$$

and the absolute values of this velocity and the force are

$$V(t) = |v(t)| \tag{B3.2}$$

$$P(t) = |p(t)|\,, \tag{B3.3}$$

respectively. The total distance U travelled by the box since time zero is

$$U(t) = \int_0^t |u'(\tau)| d\tau = \int_0^t |v(\tau)| d\tau\,. \tag{B3.4}$$

It is worthwhile noting that

$$\frac{dU}{dt} = V \tag{B3.5}$$

which is seen by differentiating Eq. (B3.4) and use Eq. (B3.2). Observe that in the special case where the box is constantly pulled to the right, we have $U = u$, $V = v$, and $P = p$.

We will now generalise the problem and assume the friction force to be dependent of U and V according to

$$Y = Y(U,V) = a + \psi(U) + \varphi(V)\,, \tag{B3.6}$$

where a is a positive constant. The functions φ and ψ have to be defined only for non-negative arguments, since U and V are always non-negative. We assume that both φ and ψ are monotonically increasing functions and that $\varphi(0) = 0$ and $\psi(0) = 0$. However, occasionally we will relax the restriction that ψ has to be increasing. Most of the detailed analysis will be performed for the linear friction force

$$Y = a + bU + cV\,, \tag{B3.7}$$

where the constants satisfy $a > 0$, $c \geq 0$, and $|b| < k$.

B4. Solution of the problem for non-constant friction

Here we are going to describe a method for solving the case with non-constant friction force Y given by Eq. (B3.6). We use the notation Y_n for the value of that force at time t_n. The method outlined in Sec. B2 ”A first solution of the discrete problem” can be applied almost literary. The only change is that Y_{n+1} has to be substituted for Y. We write the formulas with that modification:

$$p^* = p_n + k(f_{n+1} - f_n) \tag{B4.1}$$

$$P^* = |p^*| \tag{B4.2}$$

If

$$\gamma = \frac{P^*}{Y_{n+1}} > 1 \tag{B4.3}$$

then

$$p_{n+1} = \frac{1}{\gamma} p^* \tag{B4.4}$$

$$u_{n+1} = u_n + \frac{P^* - Y_{n+1}}{k} \operatorname{sgn}(p^*) \tag{B4.5}$$

else

$$p_{n+1} = p^* \tag{B4.6}$$

$$u_{n+1} = u_n \,. \tag{B4.7}$$

One equation is still missing, namely, an expression for Y_{n+1}, which is a discrete counterpart of Eq. (B3.6). We will give two alternative formulas for that in the following sections.

Let us note that in order to calculate p_{n+1} we do not have to reference u_n and u_{n+1}, because Eq. (B4.5) and (B4.7) could be left out if we were not interested in calculating $u(t)$.

For future references we derive a formula for the increment $\Delta U = U_{n+1} - U_n$, which is equal to the distance that the box moves during the corresponding time step, i.e. $|u_{n+1} - u_n|$. In the case $\gamma > 1$, this is the absolute value of the last term in Eq. (B4.5). In the other case, $\gamma < 1$, we have $\Delta U = 0$ according to Eq. (B4.7). Both cases can be written with one formula, namely,

$$\Delta U = U_{n+1} - U_n = \frac{1}{k} \langle P^* - Y_{n+1} \rangle \,, \tag{B4.8}$$

where we have used the symbol $\langle \quad \rangle$, defined as $\langle x \rangle = x$ if $x \geq 0$, $\langle x \rangle = 0$ if $x < 0$. Note also that the absolute value of the velocity of the box, V, during the time interval from t_n to t_{n+1} is approximated by the difference quotient $V = \Delta U / \Delta t$.

B5. Method I, the traditional method

We use the algorithm described in the previous section together with the following discrete version of Eq. (B3.6):

■ $$Y_{n+1} = a + \psi(U_n) + \varphi\left(\frac{U_n - U_{n-1}}{\Delta t} \right) . \tag{B5.1}$$

We recall that Y_{n+1} is used when integrating over the time step from t_n to t_{n+1}, so the values of U seem to be a bit old. The reason for using them is that when Y_{n+1} is needed in the algorithm, Eq. (B4.3), U_n is the most recent known U-value. The value U_{n+1} is calculated towards the end of the cycle by Eq. (B4.8), where in fact Y_{n+1} is used.

Method I will be further analysed in Sec. B11, but a few problems with the expression (B5.1) can be pointed out already at this stage. The time derivative in the last term is centred at the midpoint of the interval from t_{n-1} to t_n, which is outside of the time interval over which we integrate, namely, the interval from t_n to t_{n+1}. Let us look at the special case where there is no rate effect, i.e., $\varphi \equiv 0$. If the box has not been standing still during the time step, the new values P_{n+1} and Y_{n+1} will be equal, so we have $P_{n+1} = Y_{n+1} = a + \psi(U_n)$. The exact solution in this case satisfies $P(t) = a + \psi(U(t))$, and a good discrete solution would therefore satisfy $P_{n+1} = a + \psi(U_{n+1})$. This small mismatch of the time centring of U and V in Eq. (B5.1) is not harmless in presence of rate effects, since it causes instabilities, as will be seen in Sec. B11.

B6. Method II, the improved method

In the improved method we also use the algorithm in Sec. B4, but instead of using Eq. (B5.1) we use

■

$$Y_{n+1} = a + \psi(U_{n+1}) + \varphi\left(\frac{U_{n+1} - U_n}{\Delta t}\right), \tag{B6.1}$$

which have an improved centring of the U-values. The difference quotient in the last term is centred at the midpoint of the time interval over which we integrate. If we use $\Delta U = U_{n+1} - U_n$, Eq. (B6.1) can be written

$$Y_{n+1} = Y\left(U_n + \Delta U, \frac{\Delta U}{\Delta t}\right) = a + \psi(U_n + \Delta U) + \varphi\left(\frac{\Delta U}{\Delta t}\right). \tag{B6.2}$$

With this choice of Y_{n+1} the algorithm in Sec. B4 will be implicit. Equations (B6.2) and (B4.8) constitute a non-linear system of equations with two unknowns Y_{n+1} and ΔU. However, Y_{n+1} can be eliminated. By substituting the expression (B6.2) for Y_{n+1} into Eq. (B4.8), we obtain

$$\Delta U = \frac{1}{k}\left\langle P^* - Y\left(U_n + \Delta U, \frac{\Delta U}{\Delta t}\right)\right\rangle \tag{B6.3}$$

or

$$\Delta U = \frac{1}{k}\left\langle P^* - a - \psi(U_n + \Delta U) - \varphi\left(\frac{\Delta U}{\Delta t}\right)\right\rangle. \tag{B6.4}$$

This equation is solved, for instance by iteration, for ΔU. Then $U_{n+1} = U_n + \Delta U$ and Y_{n+1} is given by Eq. (B6.2).

B7. Differential equation

Method I and *II* were formulated without having formulated a differential equation. They were derived directly from the mechanical problem. In this section we are going to write down the governing differential equation and then, in a following section, solve this equation with an established numerical method. In this section we assume that the function φ is strictly monotonically increasing, so that its inverse exists.

Although we are more interested in $p(t)$ than in $u(t)$, it is probably easier to construct the differential equation in terms of $u(t)$ than $p(t)$, and then use Eq. (1.1) to get $p(t)$. If the force in the spring is not strong enough to overcome the friction force, the box is standing still and we have $\mathrm{d}u/\mathrm{d}t = 0$. On the other hand, if the force is strong enough to move the box, the absolute value of the force, P, will be equal to the friction force Y. From Eq. (B3.6) we get the relation

$$P = a + \psi(U) + \varphi(V) \tag{B7.1}$$

between P, U, and $V=dU/dt$. We might regard this as a relation between the velocity V and an "excess" force $P - a - \psi(U)$. We recall that we have assumed $\varphi(V)$ to be a monotonically increasing function, defined for $V \geq 0$, with $\varphi(0) = 0$. Therefore, we can solve for V, and express V as a function of the excess force. A positive excess force gives rise to a positive V. If the excess force is zero or negative, the box stands still and $V = 0$. All this can be expressed in the formula

$$V = \Phi\left(\left\langle P - a - \psi(U) \right\rangle\right), \tag{B7.2}$$

where we again have used the symbol $\langle \quad \rangle$, and also introduced the inverse function to φ, namely Φ. Here $V = |du/dt|$ and $P = |p|$, so we need a sign rule in order to obtain du/dt. The box, of course, moves in the same direction as it is pulled, so the sign rule is that du/dt and p should have the same sign. Therefore we obtain

$$\frac{du}{dt} = \Phi\left(\left\langle |p| - a - \psi(U) \right\rangle\right) \operatorname{sgn}(p)\,. \tag{B7.3}$$

By substituting the expression (B1.1) for p into Eq. (B7.3), we finally obtain the differential equation for u

■
$$\frac{du}{dt} = \Phi\left(\left\langle k|f(t) - u(t)| - a - \psi(U) \right\rangle\right) \operatorname{sgn}(f(t) - u(t))\,, \tag{B7.4}$$

where U is given by Eq. (B3.4) as

■
$$U(t) = \int_0^t |u'(\tau)| d\tau \tag{B7.5}$$

B8. Method III, Euler backwards

Euler backwards is an implicit method for solving a differential equation of the type

$$\frac{dy}{dt} = G(t, y) \tag{B8.1}$$

by the difference scheme

$$\frac{y_{n+1} - y_n}{\Delta t} = G(t_{n+1}, y_{n+1})\,. \tag{B8.2}$$

Here y may be a vector.

If we define the function g as the right member of Eq. (B7.4), namely, as

$$g(t,u,U)=\Phi\left(\left\langle k\left|f(t)-u\right|-a-\psi(U)\right\rangle\right)\operatorname{sgn}(f(t)-u), \tag{B8.3}$$

the differential equation (B7.4) can be written as

$$\frac{du}{dt}=g(t,u(t),U(t)), \tag{B8.4}$$

which is not quite of the type (B8.1) because of the appearance of U in the right member. If we disregard this complication, for the time being, the Euler backwards scheme for the differential equation (B8.4) would be

■ $$\frac{u_{n+1}-u_n}{\Delta t}=g(t_{n+1},u_{n+1},U_{n+1})=\Phi\left(\left\langle k\left|f_{n+1}-u_{n+1}\right|-a-\psi(U_{n+1})\right\rangle\right)\operatorname{sgn}(f_{n+1}-u_{n+1}) \tag{B8.5}$$

■ $$U_{n+1}-U_n=\left|u_{n+1}-u_n\right|, \tag{B8.6}$$

where the last equation is a difference approximation of (B7.5). Alternatively (B8.5) may be written

$$\varphi\left(\left|\frac{u_{n+1}-u_n}{\Delta t}\right|\right)=\left\langle k\left|f_{n+1}-u_{n+1}\right|-a-\psi(U_{n+1})\right\rangle, \tag{B8.7}$$

in which case a sign rule has to be given, namely that u_{n+1} - u_n must have the same sign as f_{n+1} - u_{n+1}).

In order to be more convinced that (B8.5-6) is a sound numerical scheme that deserves to be called Euler backwards, we first observe that Eq. (B7.5) after differentiating can be written as

$$\frac{dU}{dt}=\left|\frac{du}{dt}\right|=\left|g(t,u,U)\right|, \tag{B8.8}$$

where Eq. (B8.4) has been used for the last equal sign. Introducing a vector $y=(u,U)^{\mathrm{T}}$, the system of differential equations, consisting of Eqns (B8.4) and (B8.8), is represented by Eq. (B8.1) if

$$G(t,y)=\begin{pmatrix} g(t,u,U) \\ \left|g(t,u,U)\right| \end{pmatrix}, \qquad y=\begin{pmatrix} u \\ U \end{pmatrix}. \tag{B8.9}$$

With these notations it is easily seen that the Euler backwards scheme (B8.2) is the same as the system (B8.5-6).

B9. Method IV Euler forwards

Euler forward is an explicit method which you get by substituting n for n+1 in the right member of Eq. (B8.2). This changes Eqs (B8.5-6) to

■ $$\frac{u_{n+1}-u_n}{\Delta t}=\Phi\left(\left\langle k\left|f_n-u_n\right|-a-\psi(U_n)\right\rangle\right)\operatorname{sgn}(f_n-u_n) \tag{B9.1}$$

■ $$U_{n+1}-U_n=\left|u_{n+1}-u_n\right|, \tag{B9.2}$$

Alternatively (B9.1) can be written

$$\varphi\left(\left|\frac{u_{n+1}-u_n}{\Delta t}\right|\right)=\left\langle k\left|f_n-u_n\right|-a-\psi(U_n)\right\rangle, \tag{B9.3}$$

where a sign rule is needed, namely that u_{n+1} - u_n should have the same sign as f_n - u_n.

B10. The equivalence between methods II and III

The implicit *method II* is derived by physical reasoning without formulating the differential equation. We will show that *method III*, which is an established numerical method, is in fact equivalent to *method II*. *Method II* consists of two steps. First point A is moved, and then, if the trial force P* exceeds the friction force, the box is moved the distance ΔU so that the force is relaxed to P_{n+1}. Therefore

$$P_{n+1}=P^*-k\Delta U\,, \tag{B10.1}$$

which is obvious, if you think of the "physical interpretation" of *method II*. It also follows from Eq. (B4.8) and the observation that P_{n+1} = Y_{n+1} in the case where the box is moving. In the other case where the box stands still we have P_{n+1} = P^*, so Eq. (B10.1) still holds.

Another interesting derivation of this might be as follows. From Eqs (B4.1-2), where P^* is defined, we obtain

$$P^*=\left|p_n+k(f_{n+1}-f_n)\right|=\left|k(f_n-u_n)+k(f_{n+1}-f_n)\right|=k\left|f_{n+1}-u_n\right|=$$
$$=k\left|f_{n+1}-u_{n+1}+u_{n+1}-u_n\right|$$

The sign rule says that f_{n+1} - u_{n+1} and u_{n+1} - u_n must have the same sign; the box moves in the same direction as it is being pulled. Therefore we get

$$P^*=k\left|f_{n+1}-u_{n+1}\right|+k\left|u_{n+1}-u_n\right|=k\left|p_{n+1}\right|+k\left|u_{n+1}-u_n\right|=kP_{n+1}+k\Delta U$$

The difference equation (B8.5), or rather the form (B8.7), of *method III* can be written as

$$\varphi\left(\frac{\Delta U}{\Delta t}\right)=\left\langle P_{n+1}-a-\psi(U_n+\Delta U)\right\rangle. \tag{B10.2}$$

By substituting the expression for P_{n+1} from Eq. (B10.1) into Eq. (B10.2) we obtain

$$\varphi\left(\frac{\Delta U}{\Delta t}\right)=\left\langle P^*-k\Delta U-a-\psi(U_n+\Delta U)\right\rangle. \tag{B10.3}$$

This Eq. (B10.3), derived from *method III*, should be compared to Eq. (B6.4) of *method II*, and it is immediate that they are equivalent if we disregard the $\langle\ \rangle$-brackets. The expression within the brackets in the right member of both equations (B6.4) and (B10.3) are decreasing functions of ΔU, which for $\Delta U = 0$ has the value $P^*-a-\psi(U_n)$. It is easily seen that if that quantity is positive, both equations have solutions $\Delta U > 0$, and the brackets does not affect the equations in this case. Therefore the solutions are the same. In the opposite case when $P^*-a-\psi(U_n)\le 0$ the quantities within the brackets is less than or equal to zero for all positive ΔU and the solutions to both equations are $\Delta U = 0$. Therefore the equations (B6.4) and (B10.3) have the same solution ΔU, and therefore the updated U_{n+1} will be the same for both *methods II and III*. Since the sign rule is the same in both methods, also u_{n+1} will be the same, and it follows that p_{n+1} will be the same because of the relation (B1.1). Therefore, *methods II* and *III* are equivalent (assuming that the function φ is strictly monotonically increasing, so that its inverse exists, cf. Sec. B7).

Of the four methods we are considering, we have shown that the implicit *methods II* and *III* are equivalent. A natural question is if the two explicit *methods*, *I* and *IV*, are equivalent. The answer is no, because *method I* involves three time-levels, see Eq. (B5.1), while *method IV* involves only two.

B11. Stability analysis for method I

In order to simplify the analysis we assume that both hardening and rate effects are linear, i.e. $\psi(U) = bU$ and $\varphi(V) = cV$. For the four constants *k*, *a*, *b*, *c*, we assume that $k > 0$, $a > 0$, $|b| < k$, and $c \geq 0$. With these assumptions Eq. (B5.1) becomes

$$Y_{n+1} = a + bU_n + c\left(\frac{U_n - U_{n-1}}{\Delta t}\right) \tag{B11.1}$$

To simplify matters even more we assume that the box is pulled to the right (in positive *x*-direction) and that the box actually moves. Then $U_n = u_n$ for all *n*, and the force in the spring will be equal to the friction force, i.e. $p_{n+1} = P_{n+1} = Y_{n+1}$, so from Eq. (B11.1) it follows that

$$p_{n+1} = a + bu_n + c\left(\frac{u_n - u_{n-1}}{\Delta t}\right). \tag{B11.2}$$

Using the expression for p_{n+1} given by Eq. (B1.1), we obtain

$$k(f_{n+1} - u_{n+1}) = a + bu_n + c\left(\frac{u_n - u_{n-1}}{\Delta t}\right), \tag{B11.3}$$

which can be written

$$ku_{n+1} + (b + \frac{c}{\Delta t})u_n - c\left(\frac{u_{n-1}}{\Delta t}\right) = kf_{n+1} - a. \tag{B11.4}$$

Here we assume f_{n+1} to be increasing with *n* and so large that the right member is positive, because that is, namely, the condition for the box to be moving.

Equation (B11.4) is a linear difference equation for u_n with f_{n+1} as the "driving force" in the right member. The solution is a sum of a particular solution and the general solution to the homogenous equation, which is

$$ku_{n+1} + (b + \frac{c}{\Delta t})u_n - \frac{c}{\Delta t}u_{n-1} = 0. \tag{B11.5}$$

By substituting the expression $u_n = \lambda^n$ into Eq. (B11.5) we obtain the characteristic equation

$$k\lambda^2 + (b + \frac{c}{\Delta t})\lambda - \frac{c}{\Delta t} = 0 \tag{B11.6}$$

for the unknown λ. The two roots are

$$\lambda = -\frac{1}{2k}(b+\frac{c}{\Delta t}) \pm \frac{1}{2k}\sqrt{\left(b+\frac{c}{\Delta t}\right)^2 + \frac{4kc}{\Delta t}} \qquad \text{(B11.7)}$$

The difference equation (B11.4) is stable if both roots satisfy $|\lambda| \le 1$. Since $k > 0$ and both roots are real, this means that the following two inequalities must both hold:

$$\lambda_1 = -\frac{1}{2k}(b+\frac{c}{\Delta t}) + \frac{1}{2k}\sqrt{\left(b+\frac{c}{\Delta t}\right)^2 + \frac{4kc}{\Delta t}} \le 1 \qquad \text{(B11.8)}$$

$$\lambda_2 = -\frac{1}{2k}(b+\frac{c}{\Delta t}) - \frac{1}{2k}\sqrt{\left(b+\frac{c}{\Delta t}\right)^2 + \frac{4kc}{\Delta t}} \ge -1 . \qquad \text{(B11.9)}$$

After multiplication by $2k\Delta t$ these inequalities can be written

$$\sqrt{(b\Delta t + c)^2 + 4kc\Delta t} \le (2k+b)\Delta t + c \qquad \text{(B11.10)}$$

$$\sqrt{(b\Delta t + c)^2 + 4kc\Delta t} \le (2k-b)\Delta t - c , \qquad \text{(B11.11)}$$

respectively. Squaring and collecting terms leads to

$$4k(k+b)\Delta t^2 \ge 0 \qquad \text{(B11.12)}$$

$$4k(k-b)\Delta t^2 - 8kc\Delta t \ge 0 , \qquad \text{(B11.13)}$$

respectively. Under our assumption $|b| < k$, the first inequality is true for all positive Δt (only these are of interest) while the second holds if and only if

$$\Delta t \ge \frac{2c}{k-b} , \qquad \textit{“Stable time step”, method I} \qquad \text{(B11.14)}$$

(again under the assumption that only positive Δt values are considered). In order to verify that the last sentence is true, not only for (B11.12-13), but also for (B11.10-11) and (B11.8-9), one must verify that the right member of (B11.10) is non-negative for all positive Δt and that the right member of (B11.11) are non-negative for Δt satisfying (B11.14). That is easily verified.

The stability criterion (B11.14) is somewhat strange, since it requires the time step to be larger than a certain value. A numerical method with such a stability criterion can never converge, because it will be unstable when the time step tends to zero. The analysis has been performed assuming that the box is moving to the right during all time steps. It is, however, sufficient to study a special case, when showing that the numerical method does not work properly.

The simulations in the main text, cf. Fig. 1, suggests that *method I* will lead to bounded oscillations for small time steps. This is not contradicted by the fact that for small time steps Eq. (B11.4) contains an exponentially growing and oscillating component of type λ^n with $\lambda <$ -1, because this component will eventually destroy the conditions under which Eq. (B11.4) represents the algorithm (namely that the box is *moving* to the right). In order to better understand the oscillations, let us investigate a steady state solution in the special case of no hardening, $b = 0$. Assume that the box is pulled via the spring with the constant velocity df/dt. The correct physical solution (after steady state is attained) is, of course, that the box also moves with the same velocity, i.e. $du/dt = df/dt$, and that the friction force is $Y = a + c\,df/dt$. This steady state solution is permitted by the algorithm, but a completely different oscillating solution is also possible, namely, a solution where the box stands still every second time step and moves every second time step with the velocity $2df/dt$. The friction force oscillates between $Y = a$ and $Y = a + 2c\,df/dt$. The smaller friction force is used during the time steps when the box is moving, and, the larger during the other time steps. (This is consistent with the algorithm, since the friction force used in one time step is based on the velocity of the box during the preceding time step.) During the time steps when the box is standing still the computed trial force p* must be less then or equal to the friction force. It is fairly easy to see that this is possible if the time step satisfies the inequality

$$\Delta t \le \frac{2c}{k}. \qquad \text{(B11.15)}$$

From Eq. (B4.1) it is, namely, seen that the trial force is $p^* = a + k(df/dt)\Delta t$, since in the preceding time step the smaller of the two friction forces was used, namely $Y = a$, and therefore also $p_n = a$. The condition that this trial force should be less than or equal to the larger of the two friction forces is in fact the inequality (B11.15).

B12. Relaxation times in method I and the differential equation

We assume here, as we did in the preceding section, that the friction force

$$Y = a + bU + c\frac{dU}{dt} \qquad \text{(B12.1)}$$

depends linearly on the distance U travelled by the box and velocity dU/dt and we also assume that the box is pulled constantly to the right. Then $u = U$, $p = Y$ and by using (B1.1) we get

$$k(f(t) - u(t)) = a + bu(t) + c\frac{du}{dt}. \qquad \text{(B12.2)}$$

We write this differential equation as

$$c\frac{du}{dt} + (b+k)u(t) + a = kf(t) - a \qquad \text{(B12.3)}$$

(it could also have been found from (B7.4)). The homogenous equation has a solution equal to an exponential function with a relaxation time equal to $c/(b+k)$. This solution is, however, difficult to interpret physically, since the assumption that the box is being pulled to the right is not consistent with a vanishing right member. After differentiation the differential equation is

$$c\frac{dv}{dt} + (b+k)v(t) = kf'(t) \qquad \text{(B12.4)}$$

and here a vanishing right member corresponds to zero pulling rate. A particular solution to Eq. (B12.4) is

$$v(t) = \frac{k}{k+b}f'(t), \qquad f'(t) = const. \qquad \text{(B12.5)}$$

The homogenous solution is

$$v(t)=e^{-(k+b)t/c}=e^{-t/\tau}, \tag{B12.6}$$

where the relaxation time is

$$\tau=\frac{c}{k+b}, \qquad \textit{From the differential equation} \tag{B12.7}$$

(which is the same as that for (B12.3)).

Now, we are going to investigate the corresponding properties of the numerical scheme *method I.* For the same reason as we differentiated the differential equation, we carry out the corresponding procedure with Eq. (B11.4) and get

$$k\Delta u_{n+1}+(b+\frac{c}{\Delta t})\Delta u_n-\frac{c}{\Delta t}\Delta u_{n-1}=k\Delta f_{n+1}, \tag{B12.8}$$

where $\Delta u_n=u_n-u_{n-1}$ and $\Delta f_n=f_n-f_{n-1}$. For constant pulling rate, i.e. constant Δf_n, we seek a solution Δu_n that is constant. Eq. (B12.8) gives

$$k\Delta u+(b+\frac{c}{\Delta t})\Delta u-\frac{c}{\Delta t}\Delta u=k\Delta f, \tag{B12.9}$$

and we get the relation

$$\Delta u=\frac{k}{k+b}\Delta f, \tag{B12.10}$$

which is consistent with that for the differential equation (B12.5). It means that under steady state conditions (constant pulling rate) and if the stability criterion (B11.14) is fulfilled, *method I* gives a correct answer.

Equation (B12.8) has the same characteristic equation for λ as (B11.5), so the homogenous solution to Eq. (B12.8) is $\Delta u_n=C_1\lambda_1^n+C_2\lambda_2^n$, where λ_1 and λ_2 are already calculated in Sec. B11 and given by (B11.8-9). The difference equation in *method I* has higher order than the differential equation, which is the reason why it has two independent homogenous solutions while the differential equation has only one. As we have seen in Sec. B11, λ_2 becomes negative and larger than one in absolute value for small time steps. That will of course destroy any good behaviour in the other solution that might be present. But still, because of curiosity, let us investigate whether or not $\Delta u_n=C_1\lambda_1^n$ tends to a decaying exponential function with the correct relaxation time, i.e. the relaxation time (B12.7) for the differential equation.

From Eq. (B11.8) we get

$$\begin{aligned}\lambda_1&=-\frac{1}{2k\Delta t}(b\Delta t+c)+\frac{1}{2k\Delta t}\sqrt{(b\Delta t+c)^2+4kc\Delta t}=-\frac{1}{2k\Delta t}(b\Delta t+c)+\frac{b\Delta t+c}{2k\Delta t}\sqrt{1+\frac{4kc\Delta t}{(b\Delta t+c)^2}}\approx\\&\approx-\frac{1}{2k\Delta t}(b\Delta t+c)+\frac{b\Delta t+c}{2k\Delta t}\left(1+\frac{2kc\Delta t}{(b\Delta t+c)^2}\right)=-\frac{1}{2k\Delta t}(b\Delta t+c)+\left(\frac{b\Delta t+c}{2k\Delta t}+\frac{c}{(b\Delta t+c)}\right)=\\&=\frac{c}{(b\Delta t+c)}=\frac{1}{1+(b/c)\Delta t}\approx1-(b/c)\Delta t\end{aligned}$$

Using the relation $t = n\Delta t$, we get from the last expression

$$(\lambda_1)^n \approx (1-\frac{b}{c}\Delta t)^n = (1-\frac{b/c}{n/t})^{(n/t)t} \approx e^{-(b/c)t} \tag{B12.11}$$

i.e. the relaxation time is

$$\tau_I = \frac{c}{b}, \qquad \textit{Method I} \tag{B12.12}$$

which differs significantly from that given by Eq. (B12.7).

B13. Stability analysis of a simplified version of method II

Method II is equivalent to *method III*, which is a backward Euler scheme, and therefore is known to be unconditionally stable (stable for all time steps). However, we are also interested in a simplification of *method II*, where the friction force is given by

■ $$Y_{n+1} = a + \psi(U_n) + \varphi\left(\frac{U_{n+1} - U_n}{\Delta t}\right) \qquad \textit{Simplified method II} \tag{B13.1}$$

instead of by Eq. (B6.1). The difference is that in the hardening term the old cumulated motion of the box (plastic strain) U_n is used instead of the new value U_{n+1}. The simplification achieved by treating the hardening in an explicit way while retaining the implicit treatment of the rate effect is very minor, since the method will still be implicit. There is, however, another reason for studying it. If the error caused by treating hardening in an explicit way is negligible, that fact will indicate that the same could be true also for other history variables.

We assume, as usual, that the box is constantly drawn to the right so that it never stops. Then the pulling force is equal to the friction force, given by Eq. (B13.1) for the simplified version, i.e.

$$k(f_{n+1} - u_{n+1}) = a + bu_n + c\left(\frac{u_{n+1} - u_n}{\Delta t}\right), \tag{B13.2}$$

which can be written

$$(k + \frac{c}{\Delta t})u_{n+1} + (b - \frac{c}{\Delta t})u_n = kf_{n+1} - a \tag{B13.3}$$

The characteristic equation is

$$(k + \frac{c}{\Delta t})\lambda + (b - \frac{c}{\Delta t}) = 0, \tag{B13.4}$$

and the solution is

$$\lambda = \frac{\frac{c}{\Delta t} - b}{\frac{c}{\Delta t} + k} = \frac{c - b\Delta t}{c + k\Delta t} = \frac{1 - (b/c)\Delta t}{1 + (k/c)\Delta t} \qquad \textit{Simplified method II} \tag{B13.5}$$

which is always less than unity in absolute value under our assumptions $k > 0$, $c > 0$, and $|b| < k$. We conclude that the simplified method is also unconditionally stable. This analysis does not hold when $c = 0$, which is discussed below in Sec. B15.

For the true *method II* the hardening term in the difference equation (B13.2) is bu_{n+1}. By carrying out the modifications of Eqs (B13.3-4) the solution

$$\lambda = \frac{1}{1+\Delta t(k+b)/c} \qquad \textit{Method II} \qquad \text{(B13.6)}$$

is found to the characteristic equation for *method II* and *III*. Both solutions (B13.5) and (B13.6) have the same asymptotic expression for small time steps, namely,

$$\lambda \approx 1 - \frac{k+b}{c}\Delta t\,, \qquad \text{(B13.7)}$$

which gives the correct relaxation time, given by Eq. (B12.7).

B14. Stability analysis of method IV

We start by making the same assumption as was done for the stability analysis in Sec. B11, namely, that $\psi(U) = bU$ and $\varphi(V) = cV$, $k > 0$, $a > 0$, $|b| < k$, $c \ge 0$, and that the box is pulled in positive x-direction. With these assumptions the difference equation (B9.1) for *method IV* becomes

$$\frac{u_{n+1}-u_n}{\Delta t} = \frac{1}{c}\left(k\left(f_n - u_n\right) - a - bu_n\right). \qquad \text{(B14.1)}$$

This may be written

$$\frac{c}{\Delta t}u_{n+1} + (k+b-\frac{c}{\Delta t})u_n = kf_n - a\,. \qquad \text{(B14.2)}$$

By substituting the expression $u_n = \lambda^n$ into the homogenous equation one obtains the characteristic equation

$$\frac{c}{\Delta t}\lambda + (k+b-\frac{c}{\Delta t}) = 0\,, \qquad \text{(B14.3)}$$

which has the solution

$$\lambda = 1 - \frac{k+b}{c}\Delta t\,. \qquad \text{(B14.4)}$$

The requirement $|\lambda| \le 1$ leads to the stability condition

$$\Delta t \le \frac{2c}{k+b}\,. \qquad \textit{"Stable time step", method IV} \qquad \text{(B14.5)}$$

B15. Discussion of the case with no rate effects

Here we are going to discuss the special case where no rate effects are present, i.e. the function $\varphi(V)$ vanishes identically and the constant $c = 0$, in formulas where we have assumed $\varphi(V) = cV$. This case should be easier to treat, but since the methods and analysis above does not always hold in this case, it deserves a discussion. The case with no deformation hardening, $b = 0$, does not present the same change of character of the problem as does the case with no rate effects.

Methods I and *II*, which are based on the algorithm in Sec. B4 hold also in the case with no rate effects. However, the differential equation formulation in Sec. B7 and *methods III* and *IV*, which are based on that formulation, does not hold. In fact, it is the derivative that drops out of the differential equation, which thereby is transformed into an ordinary equation. It is interesting to note that although *methods II* and *III* are equivalent in the case with rate effects, only *method II* works when rate effects vanish completely.

In the case with no rate effects the simplified version of *method II* is equivalent to *method I*, for which the stability analysis in Sec. B11 is simplified. The order of the difference equation (B11.4) is reduced by one to first order (which is still one unit too much since the corresponding "differential equation" is of zero order). The root to the characteristic equation will be

$$\lambda = -\frac{b}{k}, \tag{B15.1}$$

which is seen from Eq. (B11.6) by setting $c = 0$. The solution to the homogenous equation is therefore

$$u_n = \lambda^n = \left(-\frac{b}{k}\right)^n, \tag{B15.2}$$

which is decaying because of our assumption $|b| < k$. The solution (B15.2) represents an error, a false relaxation mode, present because, in a way, we are solving an ordinary equation by a first order difference equation. The good part is that it decays very fast, because normally the deformation hardening modulus is much smaller than the elastic modulus, compare Table B.1. It also decays with a certain factor every time step (not every time unit), so the relaxation time will tend to zero with the time step.

B16. Summary and Conclusions of this appendix

We have formulated a simple one-dimensional problem, see Fig. B1, which may serve as a model problem for constitutive models of elastic-plastic materials with strain hardening and strain rate sensitivity. The correspondence of concepts in the model problem with those in the constitutive model is outlined in Table B1. The aim of this appendix was to investigate different numerical treatments of constitutive models, by applying them to the model problem.

Four methods were introduced. *Method I* corresponds to the "old method" described in the main text of this report, Sec. 2, especially Eq. (2.13). *Method II* corresponds to the "improved method", Sec. 3, Eq. (3.1). Both these numerical methods are derived directly from the physical problem. The difference lies in the time centring of the velocity of the box (strain rate). *Method I*, which is explicit, uses a value from the previous time step for that quantity, while *method II* uses a value from the current time step, which is better, but causes *method II* to be implicit. *Methods III* and *IV* are derived from a differential equation that describes the physical problem. *Method III* is the Euler backwards scheme, which is implicit, and *method IV* is the Euler forwards scheme, which is explicit.

Table B1. Correspondence between parameters in the model problem and in the constitutive model

Model problem		Constitutive model by Johnson and Holmquist	
Parameter	Notation	Notation	Parameter
Position of point A	f	ε_{ij}	Strain tensor
Tensional force in the spring	p	σ_{ij}	Stress tensor
Position of the box	u	ε_{ij}^{p}	Plastic strain tensor
Velocity of the box	v	$\dot{\varepsilon}_{ij}^{p}$	Plastic strain rate tensor
Friction force	Y	Y	Yield stress
Absolute value of the force p	P	σ_{eff}	Effective stress
Distance travelled by the box	U	ε^{p}	Effective plastic strain
Absolute value of the velocity v	V	$\dot{\varepsilon}^{p}$	Effective plastic strain rate
Constant term in friction force	a	A	Static Yield limit
"Hardening" term in friction force	$\psi(U) = bU$		
Hardening coeff.	b	B	Strain hardening coeff.
Rate sensitivity term in friction force	$\varphi(V) = cV$		
Rate sensitivity coeff.	c	C	Strain rate coeff.*
Spring constant	K	E, $2G$, $3G$	Elastic moduli

* Poor correspondence, since *C* is coefficient of a logarithmic term.

The stability of the numerical methods were analysed assuming a linear expression for the friction force $Y = a + bU + cV$, corresponding to linear dependence of yield stress on plastic strain and plastic strain rate. We assume $a > 0$, $c > 0$, $b < |k|$, where *k* is the stiffness of the spring, Eq. (B1.1). *Method I* is, strangely enough, stable only if the time step is larger than a critical value, Eq. (B11.14), which of course prevents it from converging when the time step tends to zero. A method that is unable to solve the model problem in this appendix cannot be expected to work well in the more complicated situation with a multidimensional constitutive model.

Method II was shown to be equivalent to *method III*, when rate effects are present. Since *method III* is known to be an established numerical method, we conclude that also *method II* is a reliable method. The latter method has the advantage of working also in the special case of no rate effects. *Method III* does not work in this special case, because it is based on the differential equation, and there is no differential equation in this case; it is, namely, the term containing the derivative that drops out when rate effects vanish. The fact that the equivalent *methods II* and *III* are found to be stable, supports the soundness of the improved method, which we have implemented as a user subroutine in Autodyn.

Euler backwards schemes are known to be unconditionally stable (stable for all values of the time step), so that will also be true for *methods II* and *III*. *Method IV* is stable only for small time steps, Eq. (B14.5), and the restriction on the time step gets stronger when strain rate sensitivity gets smaller. This fact makes *method IV* less attractive, although it is explicit.

The simplified version of *method II*, see Eq. (B13.1), where the strain hardening was treated explicitly, was found to be unconditionally stable, see Sec. B13. Both this simplified method and *method II* itself are first order difference equations approximating a first order differential equation. For both methods the relaxation times for the exponential modes are the same (asymptotically for small time steps), as for the differential equation, see Sec. B13. In the special case of no rate effects, the derivative drops out of the differential equation, which thereby turns into an ordinary equation. This is correctly reflected in the “difference equation” of the true *method II*, which in this special case involves only one time level. The difference equation of the simplified version will, however, still involve two time levels and therefore have a false relaxation mode. It decays very fast according to Eq. (B15.2) and is probably harmless. The good behaviour of the simplified *method II* suggests that it might not be necessary to treat history variables implicitly; cf. the explicit treatment of the temperature in our user-routine implementation of Johnson-Cook’s model, see Sec. 3 in the main text.