Implementation of Material Models for High Strain Rate Applications as User-subroutines in Abaqus/Explicit

By

Dean Bonorchis

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree MSc (Engineering)

Department of Mechanical Engineering
University of Cape Town
October 2003

DECLARATION

1. I the undersigned declare that this dissertation contains only my own work, except where reference is made with acknowledgement to contributions from others. I also declare that this material has not been submitted for any purpose or examination to any other Department or University.

Signed this 7th day of OCTOBER 2003

Dean Bonorchis Signed by candidate

2. This half dissertation comprises 20 credits out of the necessary 40 credits required for the degree MSc (Engineering). The courses completed to make up the remaining 20 credits are as follows:

• Introduction to Finite Elements	4 credits
• Finite Element Analysis	3 credits
• Non-destructive Testing	3 credits
• Structural Impact	3 credits
• Mechanical Engineering Project	5 credits
• Continuum Mechanics	3 credits
	21 credits

ABSTRACT

The general purpose finite element program, Abaqus, has the facility to allow users to supplement its existing material model library with user-defined material models (VUMATs). This thesis involves the implementation and verification of the Johnson-Cook and Zerilli-Armstrong material models as VUMATs. The same version of the Johnson-Cook material model is contained in Abaqus and was therefore used as a benchmark. These material models are suitable for high strain, high strain rate and high temperature applications.

The implementation of the material models was verified by comparing simulation results obtained using the Abaqus version of the Johnson-Cook material model with the simulation results obtained using the VUMATs of the Johnson-Cook and Zerilli-Armstrong material models. Firstly, this verification process was followed using single and multiple element tests with varying prescribed loading conditions. The verification process was then extended by performing a more "realistic" set of Taylor test simulations. The Taylor test simulation results were also compared with published experimental results for validation purposes.

ACKNOWLEDGEMENTS

I would like to thank the following for their contributions thereby ensuring the successful completion of my thesis:

- Mr T.J. Cloete who co-supervised the thesis - for his invaluable support academically and for his role as mentor.
- Professor G.N. Nurick who co-supervised the thesis - for his guidance and support both academically and financially.
- CERECAM and the Department of Mechanical Engineering – for the use of their facilities and support staff.
- NRF and CSIR – for financial support for this thesis.
- My loving wife, Kim – for always encouraging and supporting me.

"Trust in the LORD with all your heart, And lean not on your own understanding; In all your ways acknowledge Him, And He shall direct your paths."

Proverbs 3:5,6 (NKJV)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Subscripts

old, i, t	Previous time step
new, i+1, t+Δt	End of current time step
m	Mean value

Superscripts

pl	Plastic
el	Elastic
*	Dimensionless

Notation

$\underline{\ }$	Underscore - vector
$\bar{\ }$	Bar above - equivalent value
$\dot{\ }$	Dot above - rate
ij	Indicial notation
ii	Summation convention

C_{ijkl}	Elasticity tensor
c_p	Specific heat
δ_{ij}	Kronecker delta
Δ	Increment
E	Young's modulus
E^{dissip}	Dissipated inelastic specific energy
E^{intern}	Specific internal energy
e_{ij}	Deviatoric strain
$\underline{\hat{e}}$	Total deviatoric strain
$\tilde{e}$	Total equivalent deviatoric strain
ε_{ij}	Strain

ε_{kk}	Dilatation
f	Mises yield function
G	Shear modulus
J	Invariant of deviatoric stress
K	Bulk modulus
κ	Hardening parameter
λ,μ	Lamé constants
$\underline{n}$	Normal to yield surface
η	Inelastic heat fraction
P_σ	Stress power
υ	Poisson's ratio
q	Mises equivalent stress
ρ	Density
r^{pl}	Heat flux per unit volume
S	Deviatoric stress
σ_{ij}	Stress
σ_m,p	Mean or hydrostatic stress
σ_y	Yield stress
t	Time
tol	Tolerance value
T	Temperature
T_{melt}	Melting temperature
T_{ref}	Reference temperature
V	Velocity
W^{pl}	Plastic work

CHAPTER 1
INTRODUCTION

At present several areas of research at the University of Cape Town involve the experimental and numerical modelling of materials subjected to high strain rates. These include blast loading of plates of various geometries and configurations, the use of the Hopkinson Bar in material tests and drop-weight testing. In order to accurately simulate impact events of this nature the material models used in the finite element program need to be valid over a wide range of strains, strain rates and temperatures. Typically strains can reach beyond 100%, strain rates range from quasi-static to 10^6/s and temperatures due to plastic work exceed 600°C.

Currently, due to the simplicity and ease of use of the Cowper-Symonds [1] strain rate dependence equation, it is almost exclusively used. For simulation of global phenomena, such as gross plastic deformation, this equation gives reasonable results (Chung [2]). At present the uniaxial tensile test stress-strain values are adjusted manually [3] to account for temperature effects and are inputted into Abaqus [4] in tabular form. Abaqus regularises this data and interpolates when necessary. However for research which includes localised phenomena, the material model needs to explicitly include temperature coupling effects. This will allow localised phenomena such as shear banding, necking and tearing to be studied in more detail. It is for this reason that a library of more sophisticated material models will be built up to use in simulations. The experimental work being performed at high strain rates will no doubt also lead to further development of these material models.

The general finite element program, Abaqus, has the facility to allow users to develop custom material models which are coded in FORTRAN. These coded material models are implemented as user-subroutines, referred to as VUMATs (vectorised user material models) by Abaqus. The subject of this thesis is therefore

to code a selection of popular, published material models and to verify that the user-subroutines are functioning correctly. The Johnson-Cook [5] and the Zerilli-Armstrong [6] material models were chosen for implementation.

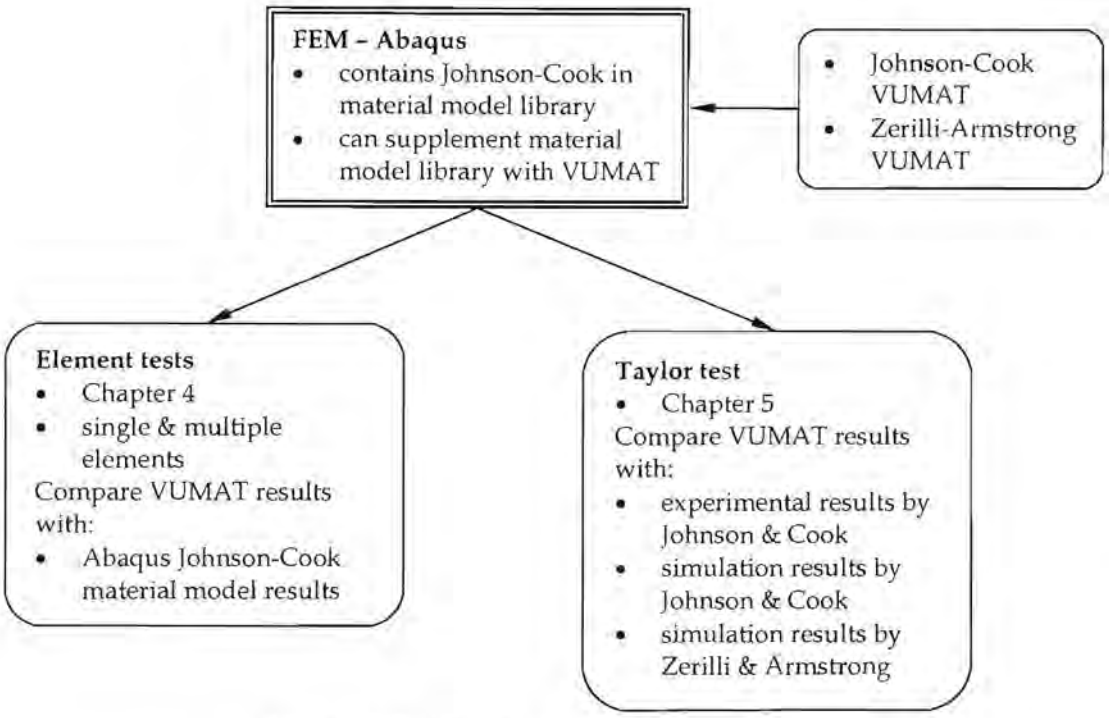


Figure 1.1: VUMAT verification strategy.

The VUMATs need to be thoroughly tested and verified by performing some type of "patch" test. Single and multiple element tests with prescribed velocity loading conditions were used (Figure 1.1). The Johnson-Cook material model was coded first because an identical version of the Johnson-Cook material model is already contained in the Abaqus material model library. The VUMAT coding was verified by running simulations using the VUMAT and comparing these results with the results obtained using the corresponding Abaqus material model in an identical simulation. The VUMAT Johnson-Cook material model results are compared quantitatively with the Abaqus Johnson-Cook model results as the results should be identical. The results obtained using the Zerilli-Armstrong VUMAT are compared to the Johnson-Cook results in a qualitative sense because slightly different results are expected since the material models are different.

The VUMATs were further verified by comparing selected results obtained from simulations of the Taylor test with published simulation results by Johnson and Cook [5] and by Zerilli and Armstrong [6] (Figure 1.1). The Taylor test is used for high strain rate material testing whereby a deformable cylinder impacts a rigid surface at a prescribed velocity (Chapter 2 & 5). The deformation results of the Taylor test simulations (using the VUMATs) were validated by comparing them with published experimental results by Johnson and Cook [5].

In summary, the outcomes of this thesis were:

- the background and theory required in order to be able to implement the material models as VUMATs was researched
- the Johnson-Cook and Zerilli-Armstrong material models were implemented as VUMATs
- the implementation of the VUMATs was verified using element tests with prescribed loading conditions
- the implementation of the VUMATs was further verified and the Johnson-Cook and Zerilli-Armstrong were validated by comparing simulation results of the Taylor test with published experimental results
- additional material models are now able to be implemented to extent the material model library at the University of Cape Town.

CHAPTER 2
BACKGROUND AND THEORY

2.1 INTRODUCTION

A constitutive equation, in this context, is an equation (or set of equations) which is used to describe the relationship between the stress and the strain in a material. These equations describe a materials response to load and are thus also known as material models. The material models described in this chapter are valid for metals subjected to large strains, high strain rates and high temperatures. In particular the subject of this thesis is the Johnson-Cook [5] and the Zerilli-Armstrong [6] material models. Any additional theories used in the implementation of these particular material models (such as linear elasticity) will also be discussed.

2.2 DEVIATORIC STRESS AND STRAIN

2.2.1 DEVIATORIC STRESS TENSOR

The deviatoric stress tensor is defined as the difference between the stress and the mean stress [7, 8]:

$$\begin{aligned} S_{ij} &= \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij} \\ &= \sigma_{ij} - \sigma_{m}\delta_{ij} \end{aligned} \tag{2.1}$$

where $\sigma_{m} = \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$ is the mean or hydrostatic or equivalent pressure stress.

2.2.2 Deviatoric Strain Tensor

In a similar way the deviatoric strain tensor is defined as the difference between the strain and the mean strain:

$$\begin{aligned} e_{ij} &= \varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\delta_{ij} \\ &= \varepsilon_{ij} - \varepsilon_m \delta_{ij} \end{aligned} \tag{2.2}$$

where $\varepsilon_m = \frac{1}{3}(\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33})$ is the mean strain and ε_{kk} is the volumetric strain or dilatation.

2.3 Linear Elasticity

An elastic material is one in which the material returns completely to its original, unloaded state after removal of the load [9]. It can therefore be classified as being non-dissipative. The small strain behaviour of most metals can satisfactorily be described by linear elasticity. As the name implies, in this form of elasticity the stress is directly proportional to the strain. This is a simple statement of Hooke's law, which is as follows [7]:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \tag{2.3}$$

where ε_{kl} is the elastic strain tensor and C_{ijkl} is a fourth order tensor of elastic components.

An isotropic material has the property that the response to loading is not dependent on direction. For an isotropic material the elasticity tensor has only two independent material parameters [10]. Two possible parameters are Lamé's constants, λ and μ. The parameter μ is also referred to as the Shear modulus G. The stress tensor can then be written in terms of the two constants as follows:

$$\sigma_{ij} = \lambda\varepsilon_{kk}\delta_{ij} + 2G\varepsilon_{ij} \tag{2.4}$$

The elastic constitutive equation can also be written in terms of the deviatoric and mean stress and strain values. This is especially useful when dealing with metal plasticity (see section 2.3). The bulk modulus is used in the deviatoric formulation:

$$K = \frac{\sigma_{ii}}{3\varepsilon_{kk}} = \frac{\sigma_m}{\varepsilon_{kk}} \tag{2.5}$$

The elastic deviatoric constitutive formulation is thus:

$$\begin{aligned} \sigma_{ij} &= S_{ij} + \sigma_m \delta_{ij} \\ &= 2Ge_{ij} + K\varepsilon_{kk}\delta_{ij} \end{aligned} \tag{2.6}$$

2.4 Metal Plasticity

The term "plastic" comes from a Greek word, πλάσσειν, which means "to shape" [10]. In this context it is used to describe ductile metals whose shape changes under application of sufficient force. This deformation is treated as a constant volume process where the deformation mechanism is predominantly due to slip or shear. It is for this reason that deviatoric stresses and strains are used when describing plasticity. The plastic deformation is assumed to be independent of the hydrostatic or mean stress [7].

Incremental plasticity theory assumes that the rate of deformation can be described as the sum of an elastic component and an inelastic (plastic) component [7, 8, 11]. The total strain rate is therefore:

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}^e_{ij} + \dot{\varepsilon}^p_{ij} \tag{2.7}$$

where $\dot{\varepsilon}^e_{ij}$ is the elastic component and $\dot{\varepsilon}^p_{ij}$ is the plastic component. The integrated form of this additive decomposition is [11]:

$$\varepsilon_{ij} = \varepsilon^e_{ij} + \varepsilon^p_{ij} \tag{2.8}$$

In order for incremental plasticity models to differentiate between the elastic and plastic responses the formulation of the following three relations is required:

- *Yield surface:* defines when yielding occurs.
- *Flow rule:* defines the direction of the inelastic deformation.
- *Hardening law:* defines how the yield and/or flow definitions vary with inelastic deformation.

2.4.1 YIELD SURFACE

The yield surface defines the onset of plastic deformation and separates the elastic and plastic responses. Yield surface plasticity relies on the fact that it is possible to clearly define an initial yield point. This value of stress is typically taken to be the value of stress for which a 0.2% value of plastic strain is produced.

The yield surface is described by a yield function which has the following general form [7, 11]:

$$f(\sigma_{ij}, \varepsilon_{ij}^{p}, T, \kappa) = 0 \tag{2.9}$$

The yield surface is therefore a function of the stress state, the plastic strain state, the temperature and one or more hardening parameters. Stress states for which $f < 0$ indicate that the material is within the elastic region. For stress states which produce $f = 0$ (i.e. on the yield surface) both elastic and plastic conditions are possible. By definition the stress states cannot be outside the yield surface. The evolution of the yield surface upon yielding, such that the stress state is never outside the yield surface, will be described in section 2.4.3.

If it is assumed that the initial yield surface of a material is isotropic then the yield function can be expressed in terms of the principal stresses $\sigma_1, \sigma_2, \sigma_3$ or the principal stress invariants. If this is combined with the further assumption that the yield state is independent of the hydrostatic pressure then the yield function can be written in the following form [7]:

$$f_1(J_2, J_3) = 0 \tag{2.10}$$

where $J_2 = \frac{1}{2} S_{ij} S_{ij}$ and $J_3 = \frac{1}{3} S_{ij} S_{jk} S_{ki}$ are the second and third invariants of the deviatoric stress tensor, respectively.

The von Mises or Distortion Energy criterion for initial yield states that yielding takes place when the maximum shear strain energy of the multi-axial stress state is equal to the maximum shear strain energy of the system at the yield point in simple tension [7, 9, 12]. Mathematically this can be written as:

$$\sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]} = \sigma_y \tag{2.11}$$

This can be re-written in terms of the second invariant of the deviatoric stress tensor (and is often called **J_2 flow theory**):

$$\begin{aligned} \sigma_y - \sqrt{3J_2} &= 0 \\ \therefore \sigma_y - \sqrt{\frac{3}{2} S_{ij} S_{ij}} &= 0 \end{aligned} \tag{2.12}$$

This yield stress is therefore a scalar quantity which is known as the **Mises equivalent stress, q.** Therefore,

$$q = \sqrt{\frac{3}{2} S_{ij} S_{ij}} \tag{2.13}$$

The von Mises yield criterion can be viewed graphically in the Haigh-Westergard principal stress space [7. Ch 5]. The three-dimensional stress state is represented as a vector using the principal stresses. The von Mises yield criteria can therefore be viewed as a circular cylinder with its centreline along the $\sigma_1 = \sigma_2 = \sigma_3$ line (hydrostatic stress line) in principal stress space.

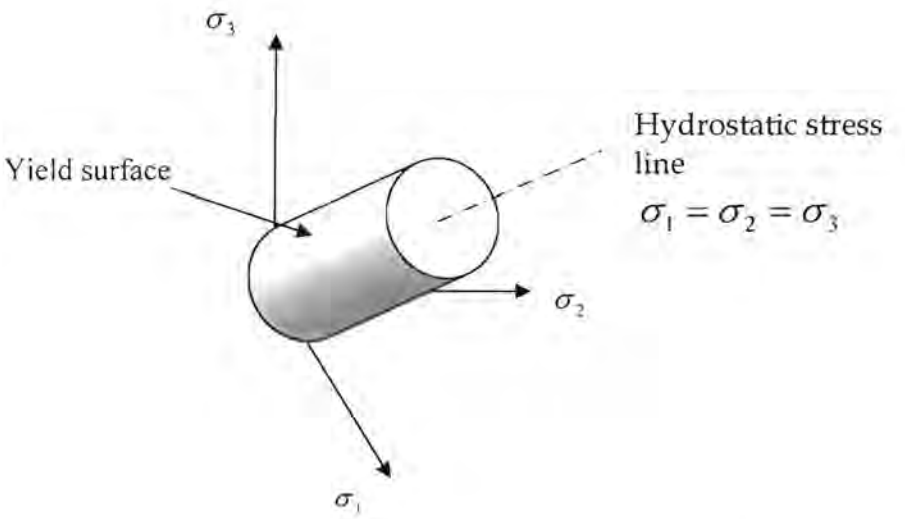


Figure 2.1: von Mises criteria in Haigh-Westergard stress space [7].

The plane normal to the hydrostatic stress line and passing through the origin is called the Pi plane and if the three stress axes are projected onto this plane then their values are $\sqrt{2/3}$ of their nominal values [7, 8]. The von Mises yield criteria as represented in the Pi plane is shown below.

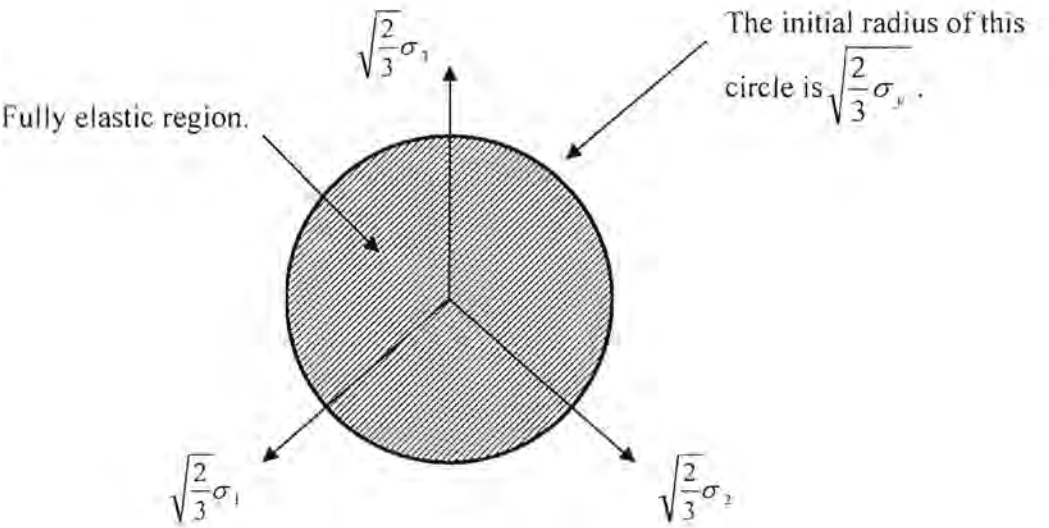


Figure 2.2: von Mises yield criteria in the Pi plane [7, 8].

2.4.2 Flow Rule

The flow rule defines the direction of plastic deformation (or flow) by defining the direction of plastic strain. In 1870 Saint-Venant [8] proposed that the directions of the principal axes of the strain increment tensor coincided with the directions of the principal axes of the stress tensor. Lévy and Mises independently developed a flow rule which relates the total strain increments to the total deviatoric stresses known as the Lévy-Mises equations [8]:

$$d\varepsilon_{ij} = S_{ij} d\lambda \tag{2.14}$$

where $d\lambda$ is a positive scalar which varies during the loading history. This flow rule assumes that the total strain is equal to the plastic strain (i.e. zero elastic strains). Prandtl and Reuss extended this flow rule to include both elastic and plastic strains and the resulting relations are known as the Prandtl-Reuss equations [8]:

$$d\varepsilon_{ij}^{p} = S_{ij} d\lambda \tag{2.15}$$

von Mises also developed the idea that the plastic strain increments can be obtained from a plastic flow potential, g. The form of this relation is [7]:

$$d\varepsilon_{ij}^{p} = d\lambda \frac{\partial g}{\partial \sigma_{ij}} \tag{2.16}$$

If the plastic flow potential is equal to f (the yield function), then the plastic strain increments are associated with the yield function and the flow rule is known as an associated flow rule. By differentiating the von Mises yield function with respect to the total stress it is possible to show that the Prandtl-Reuss equation is the flow rule associated with the von Mises yield function [7. Ch 5].

To find the magnitudes of the plastic strain increment components the scalar value $d\lambda$ needs to be found. This can be achieved by beginning with equation (2.15) [7, 12]:

$$\frac{1}{2}d\varepsilon_{ij}^{p}(d\varepsilon_{ij}^{p}) = \frac{1}{2}d\varepsilon_{ij}^{p}(S_{ij}d\lambda) = \frac{1}{2}S_{ij}S_{ij}d\lambda^{2} \tag{2.17}$$

By using the definition of the Mises equivalent stress, q (equation (2.13)), and defining the equivalent plastic strain increment as:

$$d\varepsilon_{eq}^{p} = \sqrt{\tfrac{2}{3}d\varepsilon_{ij}^{p}d\varepsilon_{ij}^{p}}\,, \tag{2.18}$$

equation (2.17) can be solved for $d\lambda$. Therefore the plastic strain increment is:

$$d\varepsilon_{ij}^{p} = \frac{3d\varepsilon_{eq}^{p}}{2q}S_{ij} \tag{2.19}$$

The seemingly arbitrary factors of $\sqrt{2/3}$ in the equivalent plastic strain increment and $\sqrt{3}$ in the Mises equivalent stress are traditionally chosen to give agreement with uniaxial tension experiments [12].

2.4.3 Hardening Rule

The Mises yield criterion presented in section 2.4.1 is only valid for initial yield. If there is no material hardening then there is no increase in strength after yield and the material is said to be elastic-perfectly plastic. If this is the case then the yield surface has a fixed radius in stress space. There are three general models for hardening [7, 10]:

- *Isotropic*: the centre of the yield surface is fixed while the surface expands uniformly (Odqvist 1933) – Figure 2.3 a).
- *Kinematic*: the yield surface translates with no change of shape (Prager 1956) – Figure 2.3 b).
- *Combined*: involves both a yield surface expansion and translation (Hodge 1957) – Figure 2.3 c).

Isotropic hardening is the simplest model to implement and produces satisfactory results but is not valid for cyclic loading. Experimentally it is found that any strain

hardening in tension actually reduces the subsequent yield stress in compression (known as the Bauschinger effect). Kinematic hardening was developed to account for the Bauschinger effect in cyclic loading. The material models to be considered in this thesis are based on the isotropic hardening model due to the majority of loading conditions studied being monotonic. The hardening models can be viewed graphically in the pi plane as shown in Figure 2.3 below.

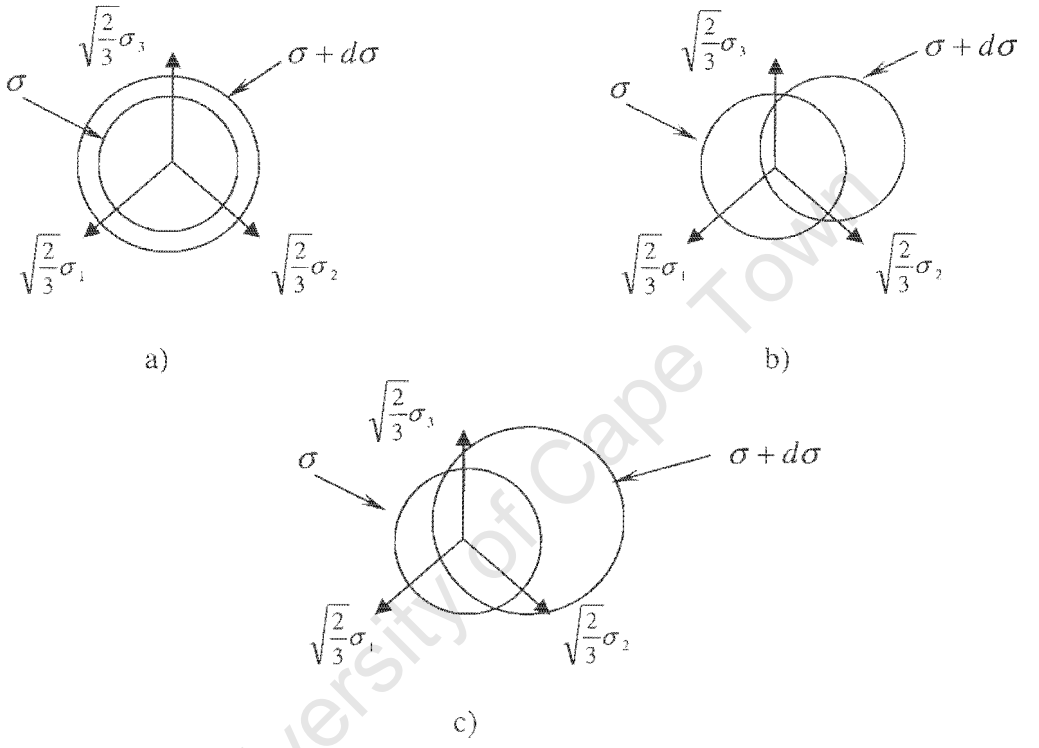


Figure 2.3: a) Isotropic, b) kinematic and c) combined hardening.

Metals are usually assumed to be strain hardening materials which means that the yield surface is a function of the integrated plastic strain. The loading history is therefore captured in this integration along the stress-strain path. The integration along the stress-strain path may therefore be expressed as an equivalent plastic strain increment [7]:

$$\bar{e}^{pl} = \int \sqrt{\frac{2}{3} d\varepsilon_{ij}^{p} d\varepsilon_{ij}^{p}} \tag{2.20}$$

2.5 Specific Forms of the Equivalent Flow Stress

The loading conditions and material type determine which form of equivalent flow stress, $\overline{\sigma}$, is used with the Mises yield criteria. If a metal is deformed at low strain rate and at ambient temperature then it is possible to use a simple power law flow stress equation such as the Ludwik (1909) equation [13, 14]:

$$\overline{\sigma} = K\varepsilon^n \tag{2.21}$$

A suitable flow stress equation needs to be able to describe a material's dependence on strain rate, temperature, strain and strain rate history and its strain hardening behaviour [15]. It is extremely difficult to include all these dependencies and it is usually the strain, strain rate and temperature dependence that is included.

Flow stress equations can usually be split into two types: phenomenological and dislocation-mechanics-based. The phenomenological types usually describe how the flow stress changes with plastic strain, temperature and rate of deformation. The loading history is not explicitly taken into account but is captured, in a sense, in the plastic strain [13]. With this approach both the form and the magnitude of the curve are obtained experimentally (i.e. a mathematical representation of the experimental results). It is usually quite accurate when strain hardening is the main driving factor. However, if the internal state of the material is important then a model which describes the physics of the deformation will give more accurate results. Examples of this are: if thermal softening is significant or if the deformation history very much determines the internal state of the material. In these cases the evolution of the internal state of the material is very important in order to accurately determine the stress. The material models to be described fall into the two categories of flow stress equations. The Johnson-Cook model [5] is of the phenomenological type while the Zerilli-Armstrong model [6] is of the dislocation-mechanics-based type.

2.5.1 JOHNSON-COOK

In 1983 Johnson and Cook [5] proposed a material model suitable for materials subjected to large strains, high strain rates and high temperatures. The model was intended primarily for numerical computations. It was stated that at the time finite element codes had developed to the point that the restricting factor in obtaining accurate results for intensely impulsive events was due to the material model. To solve this problem the simulations were repeated with small changes to the material parameters until the results were in agreement with experiments.

Due to its simplicity and the relative ease (in comparison to dislocation-mechanics-based models) of obtaining its constants, the Johnson-Cook material model has been very popular. The variables used in the model are also available in most commercial finite element codes and it is therefore easy to implement. Johnson and Cook conceded that the use of more complicated models (e.g. Follansbee and Kocks [16]) could indeed give more accurate representations of material behaviour. It was also stated that the use of different models tailored for individual materials could also give more accurate results. These more complicated material models are, however, often much more difficult to implement and it is a much more tedious process to obtain accurate material parameters for these models.

The Johnson-Cook material model takes the form of a product of dependencies. The material behaviour is function of the multiplicative effects of the strain, strain rate and temperature [15]. There is no representation of thermal or strain rate history effects. The von Mises equivalent flow stress can be written as [5]:

$$\overline{\sigma} = \left[A + B\varepsilon^{n}\right] \cdot \left[1 + C \ln \dot{\varepsilon}^{*}\right] \cdot \left[1 - T^{*m}\right] \tag{2.22}$$

where ε is the equivalent plastic strain, $\dot{\varepsilon}^* = \dot{\varepsilon}/\dot{\varepsilon}_0$ is the dimensionless plastic strain rate for $\dot{\varepsilon}_0 = 1.0s^{-1}$ and T^* is the homologous temperature. This temperature is defined as

$$T^* = \frac{T - T_{ref}}{T_{melt} - T_{ref}} \tag{2.23}$$

where T_{ref} is a reference temperature taken as room temperature and T_{melt} is the melting temperature of the material. The five material constants are therefore A, B, n, C and m. The homologous temperature is zero for temperatures less than the reference temperature and equal to one for temperatures above the melting temperature. For temperatures above the melting temperature the flow stress is zero and there is no resistance to flow. As can be seen from equation (2.22) the term in the first set of brackets represents the dependence on strain, the term in the second set of brackets represents the dependence on strain rate and the term in the third set of brackets represents the dependence on temperature.

In order to see the effect of strain rate and temperature in the Johnson-Cook model two sets of graphs are shown for Armco-iron. The material constants are given in Chapter 3.4, Table 3.1. In Figure 2.4 the temperature is held constant at 293K (assumed room temperature) and the strain rate is increased from $1000s^{-1}$ to $10{,}000s^{-1}$ to $100{,}000s^{-1}$, while in Figure 2.5 the strain rate is held constant at $1.0s^{-1}$ and the temperature is increased from 293K to 500K to 800K.

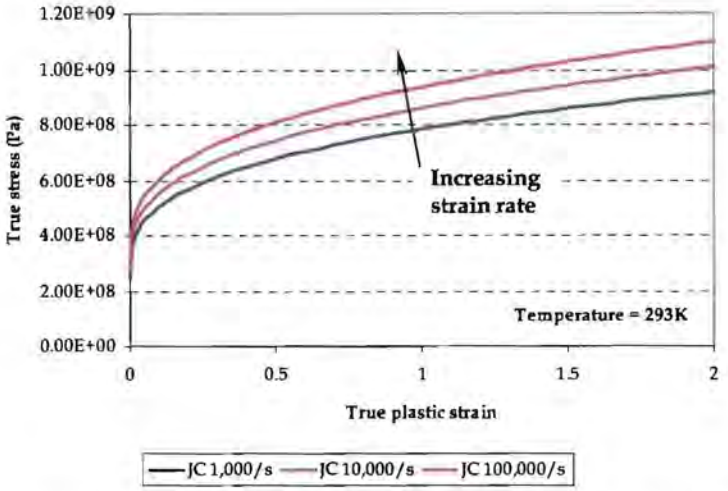


Figure 2.4: Effect of strain rate in the Johnson-Cook material model (Armco-iron [5]).

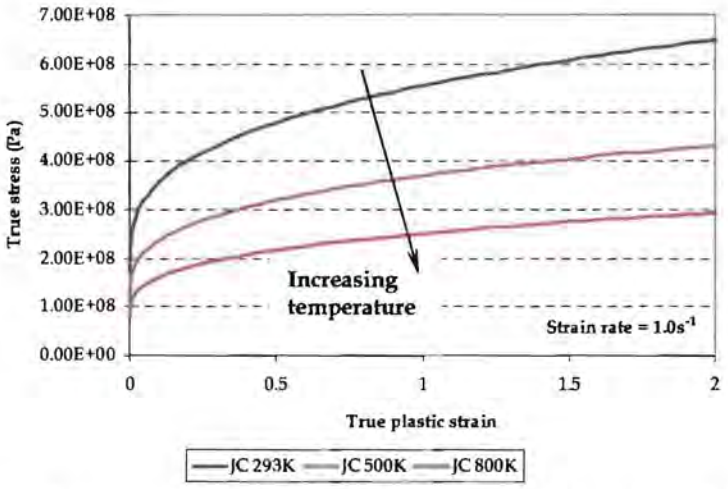


Figure 2.5: Effect of temperature in the Johnson-Cook material model (Armco-iron).

To have something to compare the form of the Johnson-Cook strain rate hardening and thermal softening to, the Cowper-Symonds strain rate equation [reported in 1] and the Masui et al [3] temperature relationship for mild steel will be used. The Cowper-Symonds equation is

$$\frac{\sigma_0'}{\sigma_0} = 1 + \left(\frac{\dot{\varepsilon}}{D}\right)^{1/q} \tag{2.24}$$

where $\sigma_0^{\cdot}$ is the dynamic flow stress at a uniaxial plastic strain rate $\dot{\varepsilon}$, σ_0 is the static flow stress and D and q are material specific constants. For mild steel these constants are taken as 40.4 s^{-1} and 5 respectively.

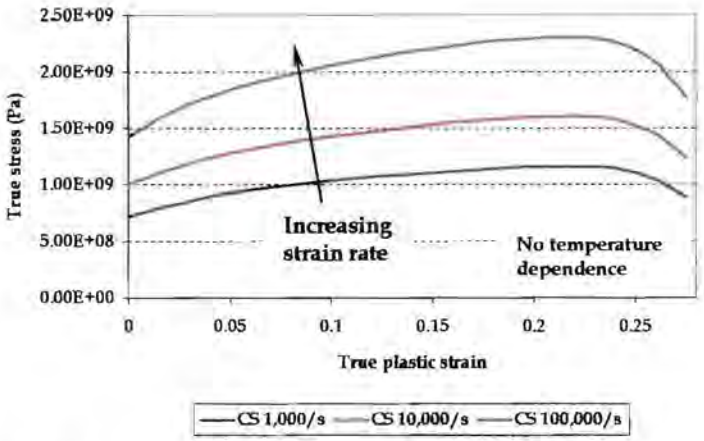


Figure 2.6: Effect of strain rate in the Cowper-Symonds material model (mild steel [1]).

Comparing the general form of the graphs in Figures 2.4 and 2.6 shows that the effect of strain rate hardening using the Cowper-Symonds equation is more severe but this might be due to mild steel being much more strain rate sensitive than Armco-Iron. Unfortunately the constants for the Johnson-Cook equation for mild steel are not available and therefore a direct comparison of the curves cannot be made.

The temperature dependence of mild steel as reported by Masui et al [3] is:

$$E = 207e9 - 58.34e6 \cdot \tau \qquad \text{for } \tau \le 600^{\circ}C$$

$$E = 3.1e5 \cdot (\tau - 1100)^2 + 97e9 \qquad \text{for } 600^{\circ}C < \tau \le 1100^{\circ}C \qquad (2.25)$$

$$\sigma_o = \sigma_{yo} \qquad \text{for } \tau \le 200^{\circ}C$$

$$\sigma_o = \sigma_{yo} \cdot [1 - 0.00178 \cdot (\tau - 200)] \qquad \text{for } 200^{\circ}C < \tau < 700^{\circ}C$$

$$\sigma_o = \sigma_{yo} \cdot [0.133 - 3.884e{-}4 \cdot (\tau - 700)] \qquad \text{for } 700^{\circ}C \le \tau \le 1000^{\circ}C \qquad (2.26)$$

where E is Young's Modulus, τ is the material temperature, σ_o is the static yield stress and σ_{yo} is the static yield stress at the reference temperature. The effect of temperature on the flow stress for mild steel is shown in Figure 2.7.

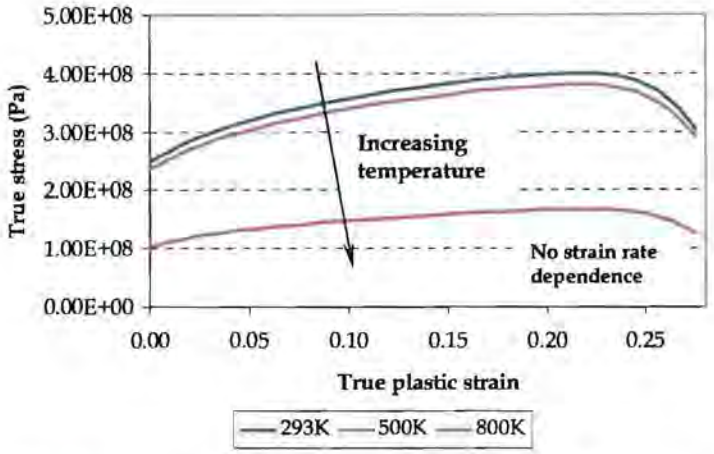


Figure 2.7: Effect of temperature by Masui et al [3] for mild steel.

For temperatures below 473K there are no temperature effects using the Masui temperature relationship. In Figure 2.5 the flow stress shows dependence on temperature for any increased temperature. This means that for a temperature of 500K (see Figures 2.5 and 2.7) the Johnson-Cook flow stress will be much lower than the flow stress using the Masui relationship. The Cowper-Symonds strain rate equation and the Masui et al temperature relationship operates on a set of tabulated stress-strain data from uniaxial tensile tests. Figures 2.6 & 2.7 show that the data is valid up to about a strain of 0.28 whereas the Johnson-Cook (and any explicit constitutive equation) is valid for any strain range used. This is important to note since Abaqus will use the stress value associated with the maximum strain value for all strains outside the tabulated range in a simulation.

It was shown by Liang and Khan [15] that the Johnson-Cook material model is appropriate for most work hardening metals. It fails to predict sequential loading experiments such as strain rate "jump" tests. It was also shown to be inappropriate for metals whose work hardening rate, $d\sigma/d\varepsilon$, decreases with increasing strain rate

such as tantalum. The multiplicative form of the temperature dependence was deemed to be appropriate for most metals [15].

A slightly modified form of the Johnson-Cook equation was later presented by Holmquist and Johnson [17] in order to better represent the effect of strain rate. It has been shown that the effect of strain rate on material strength is not a linear function of the natural log as the Johnson-Cook material model suggests. The modified form of the equation is shown here for completeness even though it will not be implemented.

$$\bar{\sigma} = \left[A + B\varepsilon^{n}\right] \cdot \left[\dot{\varepsilon}^{*C}\right] \cdot \left[1 - T^{*m}\right] \tag{2.27}$$

As can be seen in equation (2.27) the strain rate dependence has been replaced by an exponential function.

It has been observed that many ductile metals exhibit a sudden increase in strength at strain rates greater than $10^4 s^{-1}$. It was for this reason that a revised version of the Johnson-Cook model was presented by Rule and Jones [18]. Essentially this revision is in the form of additional coefficients to more accurately represent the effects of strain rate. This revision gives more accurate results for high strain rate applications such as the Taylor test. This version of the Johnson-Cook equation is not implemented. The form of the equation is given as:

$$\bar{\sigma} = \left(C_1 + C_2\varepsilon^{n}\right)\left[1 + C_3 \ln \varepsilon^{*} + C_4\left(\frac{1}{C_5 - \ln \varepsilon^{*}} - \frac{1}{C_5}\right)\right]\left(1 - T^{*m}\right) \tag{2.28}$$

where C_4 and C_5 are additional material coefficients.

It seems clear from the revision above (equation 2.28) that in order to accurately represent complicated material behaviour with phenomenological models a large number of coefficients is needed at increased numerical and experimental cost.

2.5.2 ZERILLI-ARMSTRONG

The Zerilli-Armstrong [6] constitutive relations were proposed in order to better describe the individual material responses during the simulation of Taylor impact tests. The constitutive relations are based on the thermal activation analysis and incorporate the effects of strain hardening, strain rate hardening and thermal softening. The effect of initial grain size is also included.

The Zerilli-Armstrong relations are among the more simple dislocation-mechanics-based constitutive equations. A major aspect of this set of constitutive equations is that each material structure type (BCC, FCC etc.) has its own constitutive equation based on that materials particular rate-controlling mechanism [15]. This is a very important feature because the different materials have different temperature dependent strain rate effects. For high strain rate and high temperature simulations this difference will obviously become important. Zerilli and Armstrong [6] state that the simulation of the Taylor test provides a good test of the material model. This is particularly valid if the material parameters were not obtained from the Taylor test but from quasi-static material tests.

The FCC constitutive equation is written as follows [6]:

$$\overline{\sigma} = C_0 + C_2 \varepsilon^{1/2} \exp\left(-C_3 T + C_4 T \ln \dot{\varepsilon}\right) \tag{2.29}$$

where $C_0 = \Delta\sigma_G^{'} + kl^{-1/2}$. The additional component of stress, $\Delta\sigma_G^{'}$, occurs potentially due to the influence of solute and the original dislocation density on the yield stress. The term $kl^{-1/2}$ takes into account the increase in flow stress at low temperatures due to the requirement of slip band-stress concentrations at grain boundaries being needed for the transmission of plastic flow. T is the absolute temperature. The FCC model therefore has 4 material constants. From equation (2.29) it is seen that the dependence of the flow stress on thermal softening and strain rate hardening increases with increasing strain hardening. There are again no strain rate history effects.

The **BCC** model is written as follows:

$$\bar{\sigma} = C_0 + C_1 \exp\left(-C_3 T + C_4 T \ln \dot{\varepsilon}\right) + C_5 \varepsilon^n \tag{2.30}$$

The constant C_0 has the same decomposition as for the FCC material. For this model there are 6 material constants. In this case the strain hardening is not coupled to the temperature and strain rate dependence. This is seen by Liang and Khan [15] as the major downfall of this model since most BCC metals have work-hardening behaviours which are dependent on strain rate and temperature.

In order to see the effect of strain rate and temperature in the Zerilli-Armstrong model two sets of graphs are plotted for Armco-iron (BCC) and two sets for OFHC (oxygen free high conductivity) copper (FCC). The material constants are given in Chapter 3.5, Tables 3.3 and 3.4. In the first set of graphs (Figures 2.8 & 2.10) the temperature is held constant at 293K (assumed room temperature) and the strain rate is increased from 1000s^{-1} to 10,000s^{-1} to 100,000s^{-1}. In the second set of graphs (Figures 2.9 & 2.11) the strain rate is held constant at 1.0s^{-1} and the temperature is increased from 293K to 500K to 800K.

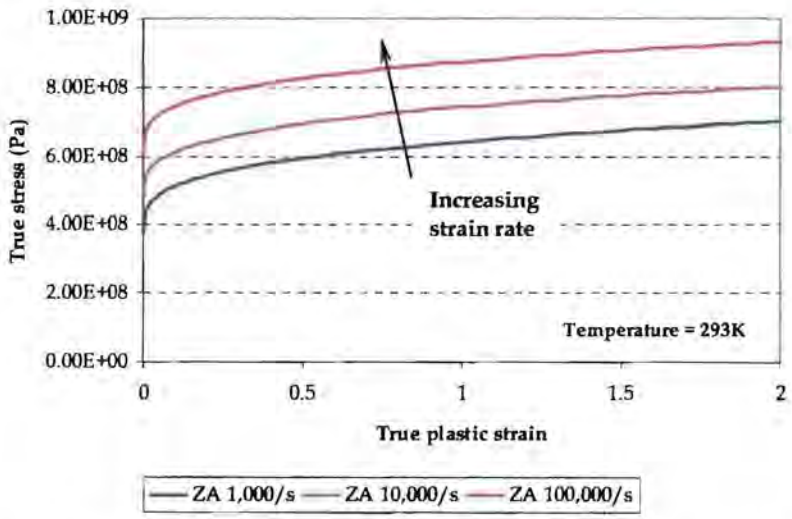


Figure 2.8: Effect of strain rate in the Zerilli-Armstrong material model (Armco-iron [6]).

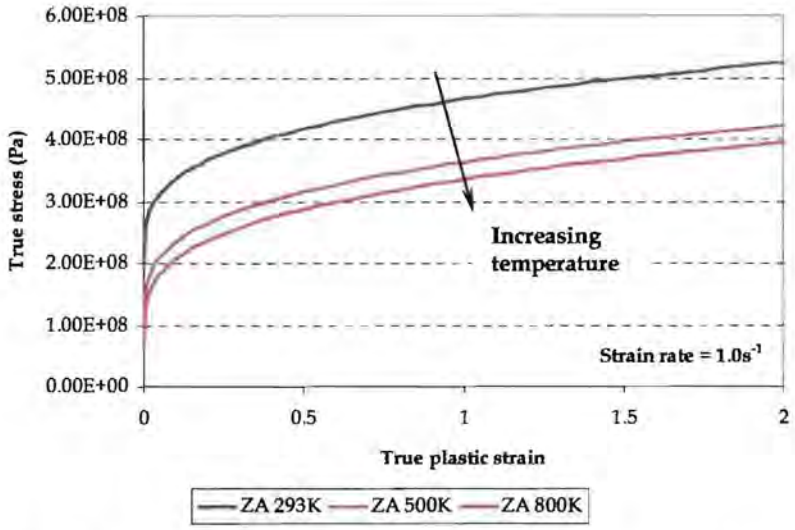


Figure 2.9: Effect of temperature in the Zerilli-Armstrong material model (Armco-iron).

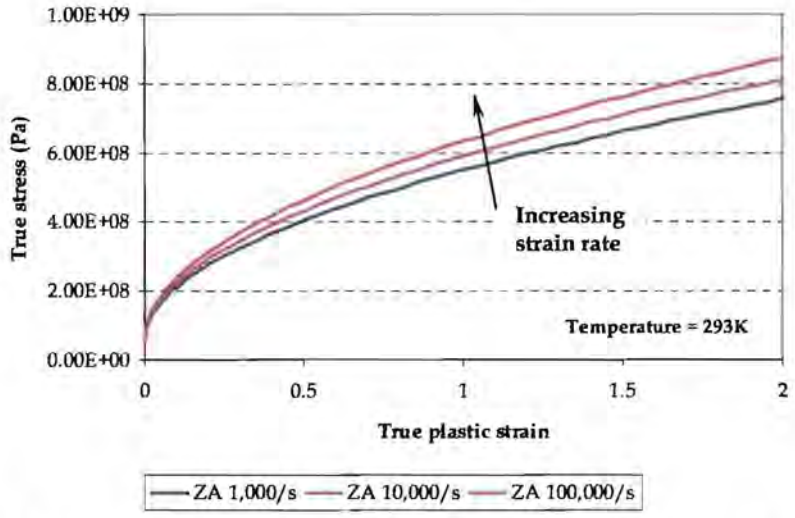


Figure 2.10: Effect of strain rate in the Zerilli-Armstrong material model (OFHC copper [6]).

For the Armco-iron (BCC) material model Figures 2.8 & 2.9 show that the strain rate hardening and the thermal softening "shifts" the flow stress curve up and down respectively. By contrast, from Figures 2.10 & 2.11 it is seen that for the OFHC copper (FCC) the strain rate hardening and the thermal softening are coupled to the

strain hardening and the curves therefore tend to "fan" out. In other words, the strain rate hardening and the thermal softening are strain hardening dependent.

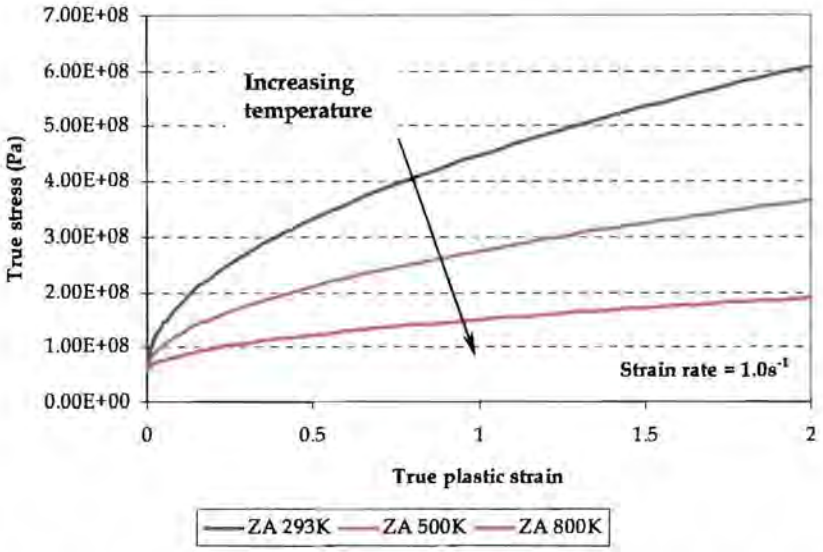


Figure 2.11: Effect of temperature in the Zerilli-Armstrong material model (OFHC copper).

2.6 ADIABATIC ANALYSIS

An adiabatic analysis is one in which any heat conduction is neglected [4]. This is typically true for high speed processes which include large inelastic strains. The event happens so quickly that the heat has no time to diffuse. This type of analysis is especially important for materials whose material properties are temperature dependent. Any significant changes in temperature for these materials can cause large changes (usually lowering) of flow stress and hence increased deformation. The temperature in this type of analysis is solved as part of the constitutive equations and will therefore be discussed in Chapter 3.4.3. It is not an additional degree of freedom as in a coupled temperature-displacement analysis.

Traditionally the value of the inelastic heat fraction has been assumed to be about 0.9 i.e. 90% of the plastic work of deformation is converted to heat (Abaqus default value [4], [14] etc.). In 1998 Kapoor and Nemat-Nasser [18] reported the details of the experimental measurement of temperature rise during high strain rate deformation. An infra-red system was used and it was found that this system underestimates the temperature rise in a specimen. It was however concluded that for the purpose of predicting flow stresses it is valid to assume that close to 100% of the work done during high strain rate deformation is converted to heat. The simulations of the Taylor test in Chapter 5 of this thesis will however use the traditional value of 0.9 in order to be able to compare the results with previously published results.

2.7 TAYLOR TEST

In the 1940's, Taylor [19, 20] used a flat-ended cylindrical projectile striking a flat rigid target to determine the dynamic yield stress of the projectile. This test is one of the simplest methods to achieve high strain rates in material testing, typically in the range $10^4 - 10^6$ s^{-1}. During the test the impact end of the projectile deforms plastically while the free end is undeformed. The impact end experiences very high stresses and once the elastic limit is reached, a plastic front moves towards the free end of the projectile. As more of the elastic rear portion of the material becomes plastic the cylinder shortens as material flows out radially [21].

The Taylor test is often used to obtain material strength parameters for high strain rate models and there is an assortment of techniques to optimise these coefficients by numerically simulating the Taylor test [18, 23, 24, 25]. The Taylor test is also often used to validate high strain rate material models by comparing the simulation of the test to the experimental results. The coefficients used in these simulations are those determined using lower strain rate material testing techniques and then the Taylor

test simulation is used to test the extrapolation of the material model to high strain rates [5, 6, 17, 26].

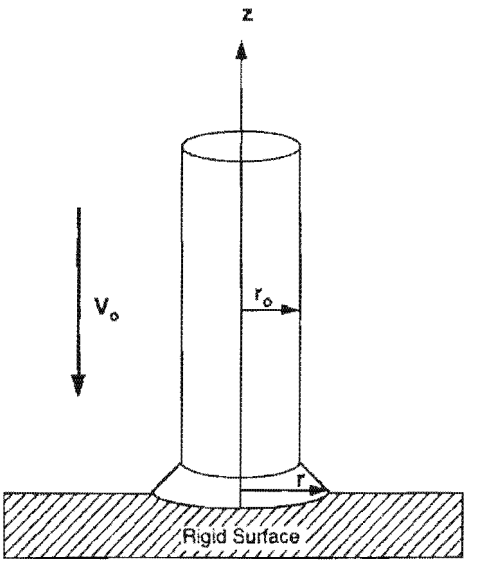


Figure 2.12: Schematic of Taylor test [22].

Johnson and Holmquist [24] used the Taylor test to determine the constants for the Johnson-Cook and Zerilli-Armstrong material models. Their method used three parameters from the deformed specimen and therefore could only predict three of the material constants [23]. They defined an average error:

$$\overline{\Delta} = \frac{1}{3}\left(\frac{|\Delta L|}{L} + \frac{|\Delta D|}{D} + \frac{|\Delta W|}{W}\right) \tag{2.31}$$

where L, D and W are the deformed length, diameter and bulge (diameter at 20% of the deformed length from the impact end) of the test specimen. The numerators are the differences between the test specimen and the computed results. A method was later developed which uses the entire specimen profile for optimising material constants [23].

2.8 Solution Techniques for Nonlinear Equations

The material models in this thesis are metal plasticity models. When the model is implemented (see Chapter 3) the plasticity component is found by solving the uniaxial form of the Mises equivalent stress function. Solving this equation gives the increment in equivalent deviatoric plastic strain. The flow stress equation to be solved is a nonlinear equation in the equivalent deviatoric plastic strain increment and therefore a nonlinear equation solving technique is required.

2.8.1 Bisection Method

One of the oldest and most simple methods of solving nonlinear equations is the Bisection or interval halving method [27]. This method begins with choosing two values of x namely x_1 and x_2 which bracket a root of $f(x) = 0$. The method then repeatedly halves the interval and replaces one endpoint with the midpoint so that in each case the root is bracketed. The endpoints bracket the root if the function values at the respective endpoints are of opposite sign: $f(x_1)*f(x_2) < 0$. This can be seen graphically in Figure 2.13.

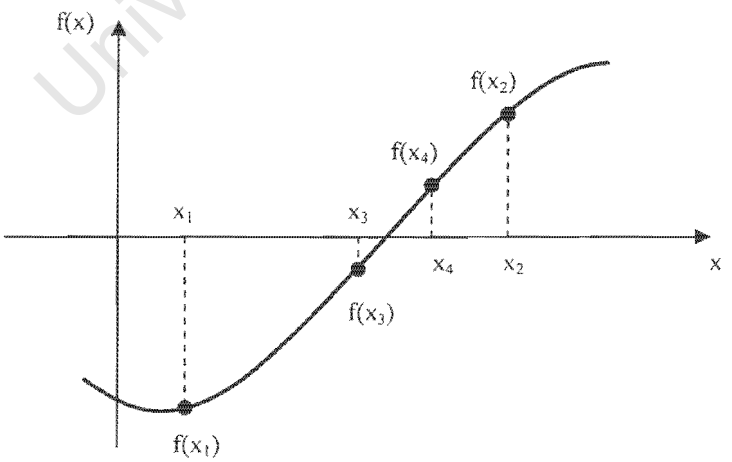


Figure 2.13: The Bisection method [27].

An algorithm for the Bisection method is presented in Figure 2.14 [27].

Given values x_1 and x_2 such that $f(x_1)*f(x_2) < 0$:

REPEAT

Set $x_3 = (x_1 + x_2)/2$

IF $f(x_3)*f(x_1) < 0$

Set $x_2 = x_3$

ELSE

Set $x_1 = x_3$

ENDIF

UNTIL

$|x_1 - x_2| < Tol_1$ or $f(x_3) < Tol_2$

Figure 2.14: Algorithm for the Bisection method [27].

An advantage of the Bisection method is that it is guaranteed to work if the function is continuous within the endpoints. Another advantage is that the number of iterations required to achieve a specified accuracy is known in advance when using the first convergence criterion shown above. The disadvantages are that this method requires many function evaluations and that it is relatively slow to converge. With modern computers and for specific applications these disadvantages can often be tolerated.

2.8.2 Newton's Method

This is one of the most widely used methods of solving equations and is the method used by Abaqus [11, 27]. This method requires that an initial point is chosen in the proximity of the root. The slope of the function is evaluated at the current point. The position where this slope intersects with the x-axis is used as the next point.

This is continued until a suitable convergence criterion is met. This method is shown in Figure 2.15.

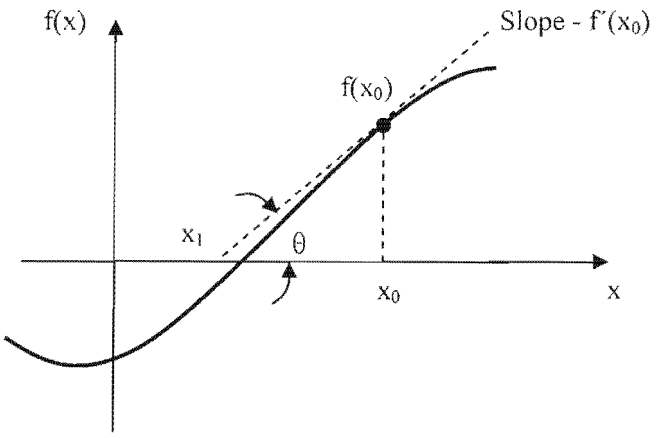


Figure 2.15: Newton's method [27].

From Figure 2.15 an expression for the slope of the function is given as:

$$\tan(\theta) = f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \tag{2.32}$$

From this the general form of Newton's method is [27]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.33}$$

An algorithm for Newton's method is given in Figure 2.16 [27]. The major advantage of Newton's method is that it converges quadratically [27]. In other words, the accuracy doubles after each iteration. The disadvantages are that it requires two function evaluations per iteration, it will not converge if the initial guess is not sufficiently close to the root and sometimes the exact form of the equation of the slope of the function is difficult to formulate.

Given a value x_0 close to the root:

Calculate $f(x_0)$ and $f'(x_0)$

Set $x_1 = x_0$

IF $f(x_0) \neq 0$ and $f'(x_0) \neq 0$

REPEAT

Set $x_0 = x_1$

Set $x_1 = x_0 - f(x_0)/f'(x_0)$

UNTIL

$|x_0 - x_1| < Tol_1$ or $f(x_1) < Tol_2$

Figure 2.16: Algorithm for Newton's method [27].

2.9 SUMMARY

The background and theory required to implement the Johnson-Cook [5] and Zerilli-Armstrong [6] material models has been presented in this chapter. The linear elastic component of the material model was discussed followed by the plasticity component. The discussion of the plasticity component involved the form and function of the yield surface, flow rule and hardening law which together define the plasticity component. Some comparisons were made between the Johnson-Cook and Zerilli-Armstrong models and these were in turn compared (in form) with the Cowper-Symonds strain rate model and the Masui et al temperature dependence relationship. The definition and applicability of an adiabatic analysis was highlighted and the background to the Taylor test and its usefulness were discussed. The chapter concluded with a discussion of solution techniques for nonlinear equations, in particular the Bisection and Newton methods.

CHAPTER 3

IMPLEMENTATION

3.1 INTRODUCTION

Abaqus has a facility which allows users to implement a user-defined material model as a VUMAT. The VUMAT is then used to supplement the existing material models in Abaqus. The implementation of the Johnson-Cook [5] and Zerilli-Armstrong [6] material models will be discussed in this chapter. A VUMAT needs to solve for the stress state at the end of each time step. The solution scheme to be used is the elastic predictor-radial return method [26, 28 - 32].

3.2 ELASTIC PREDICTOR-RADIAL RETURN METHOD

The elastic predictor-radial return method is begun by calculating an incremental change in the stress state assuming that the deformation is purely elastic. This stress increment is added to the previous stress state (point A in Figure 3.1) to find the trial stress (point D^E). The Mises equivalent stress (equation (2.13)) is then found and compared to the yield (flow) stress as found using a constitutive equation such as the Johnson-Cook equation. If the equivalent stress is less than the yield stress the material is still elastic and the trial stress is set as the new stress state. If the equivalent stress is greater than the yield stress the material has yielded. The deviatoric stresses are then reduced such that the equivalent stress is equal to the yield stress. The stress state is therefore returned to the yield surface (point D). This reduction in the deviatoric stress is accomplished by reducing the total equivalent deviatoric strain by the equivalent plastic deviatoric strain increment. The uniaxial form of the Mises equivalent stress is used to do this (see Appendix A). The return of the stress state to the yield surface is radial because the plastic strain increment

correction is in the direction of the deviatoric stress (associated flow rule - the normal is perpendicular to the yield surface). A graphical representation of this solution strategy can be seen in Figure 3.1 [28].

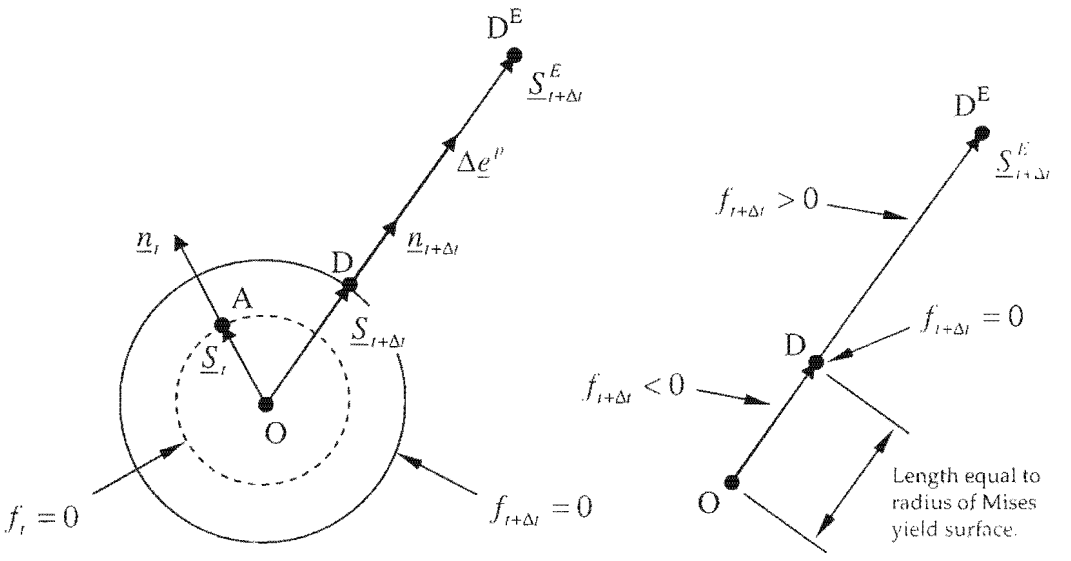


Figure 3.1: Graphical representation of elastic predictor-radial return method [28].

The unit normal to the yield surface is $\underline{n}$ and the yield function at the end of the previous time step is f_t while the yield function at the end of the current time step is $f_{t+\Delta t}$.

3.3 User-defined Material Models in Abaqus

Implementing material models in Abaqus involves evaluating the state of the material at an integration point over the time increment during a nonlinear analysis [11]. Each material integration point is treated separately and therefore material behaviour can be assumed to be defined locally. The element behaviours in Abaqus are formulated in terms of the updated Lagrangian or material description [4]. In this formulation the element deforms with the material. The state of the element at the beginning of each time step is treated as the reference state and hence small strain assumptions are valid throughout the analysis due to the small time steps.

The total strain increment components for each time step are passed to the user subroutine for each integration point and these are then used to update the stresses and any solution dependent state variables (Figure 3.2). Care should be taken when coding the material model because the model is written in tensor form but Abaqus passes the tensors to the VUMAT as vectors.

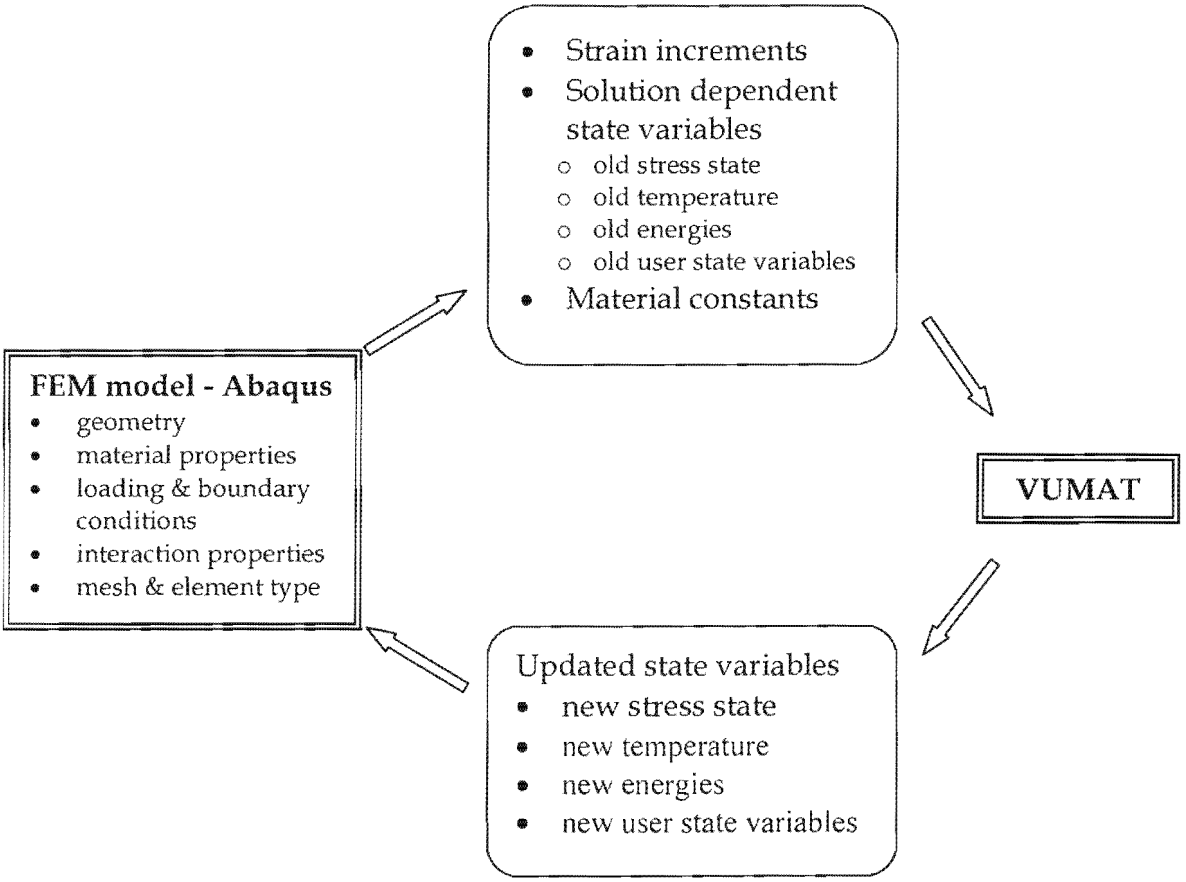


Figure 3.2: Abaqus - VUMAT interaction.

As discussed in Chapter 2 an adiabatic analysis excludes any heat conduction. The built-in material models in Abaqus can be used in conjunction with the adiabatic analysis option (also a built-in function). A VUMAT cannot however be used with the built-in adiabatic analysis option and therefore the adiabatic heat generation due to any plastic straining needs to be calculated in the VUMAT if it is to be included. The corresponding increase in temperature in an increment is added to the temperature from the previous time step and in this way it is possible to include

adiabatic heating in the VUMAT. The details of this implementation will be described later.

Abaqus sends a set of fictitious strains to the VUMAT at time = 0.0 seconds in order to check the VUMAT and to calculate initial material properties. The initial elastic wave speeds can then be found (which are used to find the stable time increment). Abaqus uses a co-rotational coordinate system in which the basis system rotates with the material. All variables (stresses, strains, state variables etc.) are therefore orientated according to the local material axes and it is not necessary to rotate any tensors. Abaqus uses the "true" or Cauchy stress which is defined as the force per current area.

For the 3D case, symmetric tensor components are passed to the VUMAT in the following order: 11, 22, 33, 12, 23 and 31. It should be noted that Abaqus uses different component ordering when reporting components elsewhere for example in the visualisation module.

The VUMAT is required to define the stresses and any solution dependent state variables at the end of the time increment. It is also possible to update the internal energy and the dissipated inelastic energy. Even though the documentation states that the temperature at the end of the increment is passed in to the VUMAT for information only and that it should not be altered, the author mistakenly used this variable to store the updated temperatures. Despite the warning in the documentation, there does not appear to be any adverse consequences by doing this and this method seems to work correctly.

3.4 ISOTROPIC LINEAR ELASTICITY

As mentioned in Chapter 2 the total strain rate can be decomposed into the elastic and plastic rate components respectively:

$$\dot{\underline{\varepsilon}} = \dot{\underline{\varepsilon}}^{el} + \dot{\underline{\varepsilon}}^{pl} \tag{3.1}$$

The isotropic, linear elasticity can be written in terms of two independent material parameters. For the purpose of the current material model these are chosen as the bulk modulus, K, and the shear modulus, G, defined as follows:

$$K = \frac{E}{3(1-2\nu)}$$
$$G = \frac{E}{2(1+\nu)} \tag{3.2 a, b}$$

where E is Young's modulus and ν is Poisson's ratio which are inputted by the user.

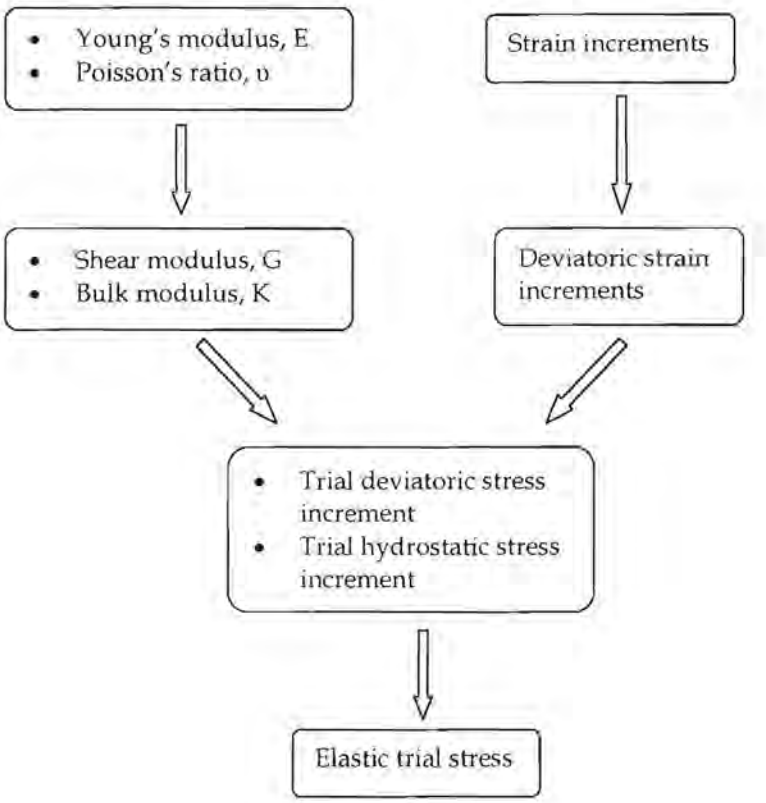


Figure 3.3: Linear elasticity algorithm.

The material model is developed in terms of the strain increments (Figure 3.3) as this is what is passed to the VUMAT by Abaqus. The development of the material model is also described in the order in which it is implemented in the user subroutine. The volumetric strain increment is defined as:

$$\Delta\varepsilon_{vol} = trace(\Delta\underline{\varepsilon}) \tag{3.3}$$

which is essentially an elastic volumetric strain since the plastic volumetric strain is zero (Mises yield function with associated flow). The deviatoric strain increment is therefore:

$$\Delta\underline{e} = \Delta\underline{\varepsilon} - \frac{1}{3}\varepsilon_{vol}\underline{I} \tag{3.4}$$

The elasticity is written in volumetric and deviatoric form. The trial deviatoric stress increment is therefore:

$$\Delta\underline{S}^{trial} = 2G\Delta\underline{e} \tag{3.5}$$

The volumetric part is written in terms of the hydrostatic or equivalent pressure stress increment (note the negative used in the definition):

$$\Delta p^{trial} = -K\Delta\varepsilon_{vol} \tag{3.6}$$

It is possible to find the total elastic trial stress which will be set as the new stress if it is later found that the yield condition has not been exceeded:

$$\underline{\sigma}_{new}^{trial} = \underline{\sigma}_{old} + \Delta\underline{S}^{trial} - \Delta p^{trial} \tag{3.7}$$

3.5 JOHNSON-COOK PLASTICITY

The Johnson-Cook constants are entered as user defined material constants in the Property Module in Abaqus. For Armco-Iron the Johnson-Cook constants are given in Table 3.1 [5].

The order in which the constants are entered (in the Property Module in Abaqus) is important since the constants are sent to the VUMAT as a sequence of material properties. These properties are assigned to the appropriate variables in the

VUMAT. The density is also entered in the Property Module as 7890 kg/m^3 for Armco-Iron.

NO.	CONSTANT	VALUE	UNIT
1	**E** – Young's modulus	200	GPa
2	**υ** – Poisson's ratio	0.3	-
3	**A** – static yield stress	175	MPa
4	**B** – work hardening coefficient	380	MPa
5	**n** – work hardening exponent	0.32	-
6	**m** – thermal softening exponent	0.55	-
7	T_{ref} – room temperature	300	K
8	T_{melt} – melting temperature	1811	K
9	**η** – inelastic heat fraction	0.9	-
10	c_p – specific heat	452	J/kg.K
11	**C** – strain rate coefficient	0.06	-

Table 3.1: Johnson-Cook material constants for Armco-Iron [5].

For OFHC copper the Johnson-Cook constants are given in Table 3.2 [5]. The density is taken as 8960 kg/m^3.

NO.	CONSTANT	VALUE	UNIT
1	E - Young's modulus	70	GPa
2	υ - Poisson's ratio	0.3	-
3	**A** - static yield stress	90	MPa
4	**B** - work hardening coefficient	292	MPa
5	**n** - work hardening exponent	0.31	-
6	**m** - thermal softening exponent	1.09	-
7	T_{ref} - room temperature	300	K
8	T_{melt} - melting temperature	1356	K
9	η - inelastic heat fraction	0.9	-
10	c_p - specific heat	383	J/kg.K
11	C - strain rate coefficient	0.025	-

Table 3.2: Johnson-Cook material constants for OFHC copper [5].

3.5.1 YIELD CRITERION

In order to use the yield criterion the Mises equivalent stress is needed (Figure 3.4). For this to be calculated the total deviatoric stresses need to be found. The total hydrostatic or equivalent pressure stress can be found from the new trial stress (again note the negative in the definition):

$$p = -\frac{1}{3} trace\left(\underline{\sigma}_{new}^{trial}\right) \tag{3.8}$$

The total deviatoric stress can now be found:

$$\underline{S}^{trial} = \underline{\sigma}_{new}^{trial} + p\underline{I} \tag{3.9}$$

An alternative way of doing this (although requiring 7 extra state variables) would be to save the total equivalent pressure stress and the total deviatoric stress components as solution dependent state variables. When the respective increments

are calculated (equations 3.5 & 3.6) they could be added to the old values of the state variables and the new total values could be found. This method is not used due to the additional memory usage and the associated increase in computation time for retrieving stored state variables.

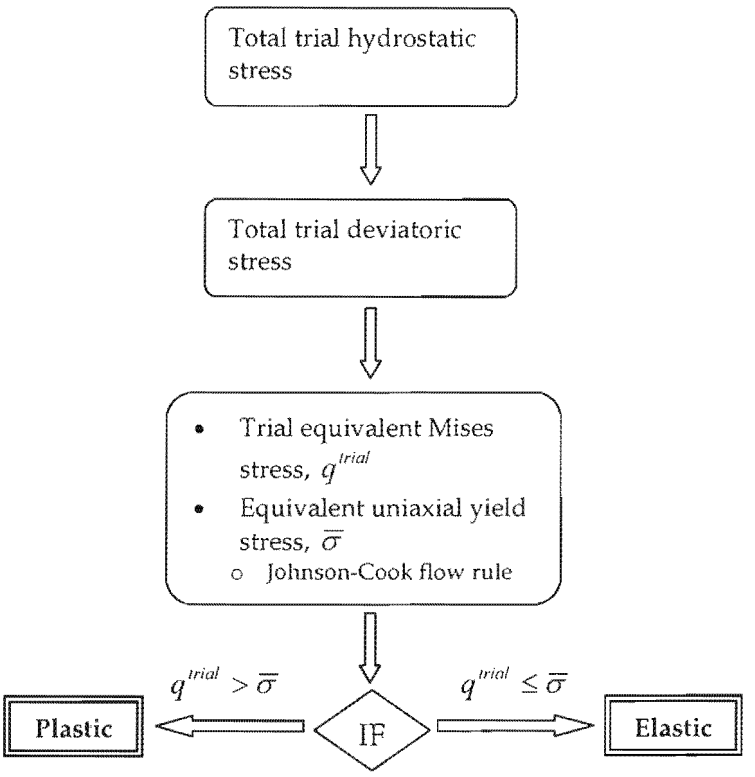


Figure 3.4: Yield criterion algorithm.

Using the total deviatoric stress, the equivalent Mises stress (for an assumed purely elastic response) can be found:

$$q^{trial} = \sqrt{\frac{3}{2}\underline{S}^{trial} : \underline{S}^{trial}} \tag{3.10}$$

The equivalent uniaxial yield stress is found using the Johnson-Cook flow rule as described in Chapter 2:

$$\overline{\sigma} = \left[A + B\left(\overline{e}^{pl}\right)^{n}\right] \cdot \left[1 + C \ln\left(\dot{\overline{e}}^{pl^*}\right)\right] \cdot \left[1 - T^{*m}\right] \tag{3.11}$$

where $\overline{e}^{pl}$ is the equivalent deviatoric plastic strain, T^* is the homologous temperature as defined by equation (2.23) and $\dot{\overline{e}}^{pl^*}$ is the equivalent dimensionless

plastic strain rate (defined as the equivalent deviatoric plastic strain divided by the current time increment). This was initially how the equivalent uniaxial yield stress was calculated (for the element tests – Chapter 4) but this method was altered by using the total equivalent deviatoric strain rate instead for the Taylor tests (Chapter 5). The reason for this is discussed at the end of section 3.5.3.

If $q^{trial} \leq \overline{\sigma}$, then yielding has not occurred and the material response is purely elastic. The equivalent deviatoric plastic strain increment is zero and the old equivalent deviatoric plastic strain is saved as the new value. The new stress is then equal to the new trial stress; $\underline{\sigma}_{new} = \underline{\sigma}^{trial}_{new}$. There is also therefore no increase in temperature. Since there is no plasticity the material model concludes by updating the solution dependent state variables and the internal and dissipated energies (see section 3.5.6).

3.5.2 YIELDING

The material is deemed to have yielded if the equivalent Mises stress, q^{trial}, exceeds the equivalent uniaxial yield stress. If the material has yielded then the material model needs to solve for the plasticity state. To initiate the iterations, the old deviatoric stresses are needed. These can be found by first calculating the old total hydrostatic stress:

$$p_{old} = -\frac{1}{3} trace(\underline{\sigma}_{old}) \tag{3.12}$$

and then the old deviatoric stress is:

$$\underline{S}_{old} = \underline{\sigma}_{old} + p_{old}\,\underline{I} \tag{3.13}$$

The total deviatoric strain is then calculated as:

$$\begin{aligned} \underline{\hat{e}} &= \underline{e}^{el}_{old} + \Delta\underline{e} \\ &= \frac{\underline{S}_{old}}{2G} + \Delta\underline{e} \end{aligned} \tag{3.14}$$

The total equivalent deviatoric strain can now be found:

$$\tilde{e} = \sqrt{\frac{2}{3}\underline{\hat{e}} : \underline{\hat{e}}} \tag{3.15}$$

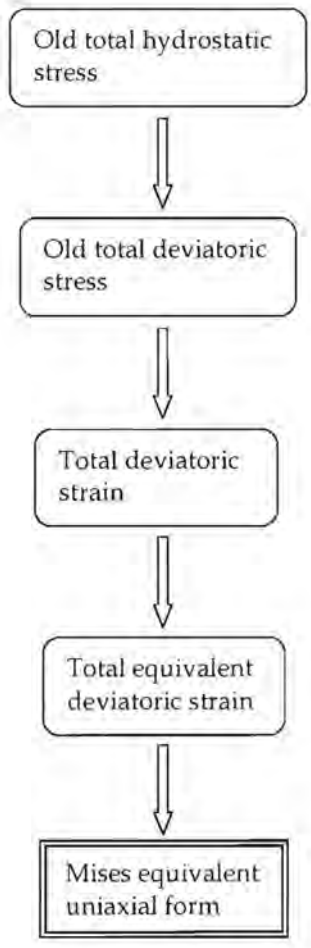


Figure 3.5: Algorithm to obtain the Mises equivalent uniaxial form.

The Mises equivalent stress, q, must satisfy the following uniaxial form (see derivation in Appendix A):

$$\begin{aligned} 3G(\tilde{e} - \Delta\bar{e}^{pl}) - q = 0 \\ \therefore 3G(\tilde{e} - \Delta\bar{e}^{pl}) - \bar{\sigma} = 0 \end{aligned} \tag{3.16}$$

Equation (3.16) is a nonlinear equation in $\Delta\bar{e}^{pl}$, the equivalent deviatoric plastic strain increment. A nonlinear solution technique is therefore required to solve it. This relationship is in fact ensuring that the stress state remains on the yield surface ($q = \bar{\sigma}$ on the yield surface).

3.5.3 Solving for the Equivalent Deviatoric Plastic Strain Increment

Abaqus uses Newton's method to solve equation (3.16) [11]. As described in Chapter 2, this requires the calculation of the differential of the stress with respect to the equivalent strain which, for complicated material models, is not always a simple task. The use of the Bisection method is suggested [33] as an alternative to Newton's method due to its ease of use and stability, but slower convergence. The Bisection method was therefore attempted first due to its inherent stability. Section 3.5.4 discusses the implementation of Newton's method and section 3.5.5 discusses an alternative non-iterative method.

In order to use the Bisection method some initial values need to be defined. The Bisection method then uses an iterative scheme to solve for the equivalent deviatoric plastic strain increment (Figure 3.6). The iterations are continued until the convergence criterion is met. In this case the criterion is met when:

$$abs\left(f_{deqpl}\right) = abs\left(3G(\tilde{e} - \Delta\bar{e}^{pl}) - q\right) \leq tol \tag{3.17}$$

Equation (3.17) implies that the stress state is not returned to the yield surface exactly but only to within a specified tolerance.

The tolerance value used is an arbitrary choice of the user and perhaps the most important factor in determining the accuracy of the constitutive model solution. A value of 10^5 was used because for a typical material response the Mises equivalent stress is in the order of at least 200 MPa. The tolerance therefore ensures that the solution at the end of the time step is within about: $10^5/2\text{x}10^8 = 5\text{x}10^{-4} = 0.05\%$ of the radius of the yield surface. Helmut Bowles* [34] recommended setting the tolerance value to at least the yield stress divided by 10^3.

* Mr Bowles is an employee of FEAS, the Abaqus representatives in South Africa.

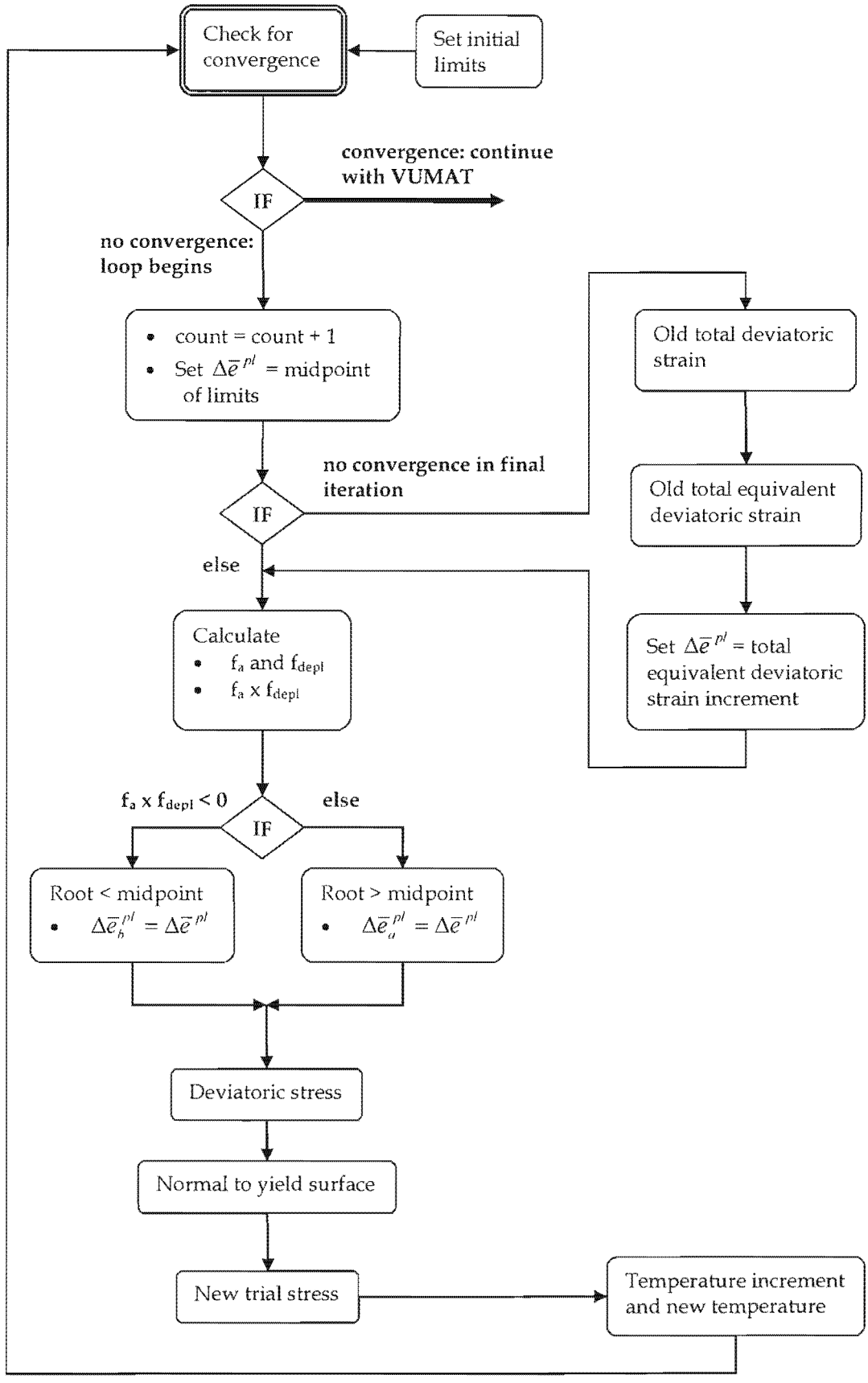


Figure 3.6: Algorithm to solve for equivalent deviatoric plastic strain increment.

Initially the convergence criterion was based on the successive difference in the equivalent deviatoric plastic strain increment within the iterations:

$$abs\left(\Delta\bar{e}_{i+1}^{pl} - \Delta\bar{e}_{i}^{pl}\right) \leq tol \tag{3.18}$$

After consultation with Helmut Bowles [34] (see footnote, pg 41) this convergence criterion was changed to equation (3.17). This was done because it was felt that due to the small values of the equivalent deviatoric plastic strain increments the tolerance used in equation (3.18) would have to be in the order of 1×10^{-8}, at the highest. This does not however give any indication as to how close the stress state is to the yield surface and was thus not considered to be as reliable.

The range of interest for the equivalent deviatoric plastic strain increment is set by the limits chosen for the Bisection method. If the limits are set such that the equivalent deviatoric plastic strain increment to be solved for is outside the limits then the iterations will not converge. The lower limit is set at zero since the plastic strain is irrecoverable and therefore cannot be negative. The upper limit is set at 10^{-1}. The closer this value is to the lower value (i.e. the narrower the range) the faster the iterations will converge. Through experience it was found that the equivalent deviatoric plastic strain increment value is usually in the order of 10^{-4} due to the small time steps in the explicit method. The highest value seen was about 10^{-2}.

Once the initial limits are set, the convergence criterion is used to check if the average (midpoint) of the initial values causes convergence. If the convergence criterion is met using the initial limits then it is not necessary to iterate. If not then the iteration loop is entered into to solve for the equivalent deviatoric plastic strain increment. This loop involves a number of equations because the Mises equivalent stress is a function of the equivalent deviatoric plastic strain increment, the equivalent plastic strain rate and the temperature.

At the beginning of the loop the limits are averaged in order to find the midpoint value:

$$\Delta\bar{e}^{pl} = \frac{\Delta\bar{e}_a^{pl} + \Delta\bar{e}_b^{pl}}{2} \tag{3.19}$$

Equation (3.17) is then used to find the uniaxial form of the Mises stresses for the lower limit and the midpoint value:

$$\begin{aligned} f_a &= 3G(\tilde{e} - \Delta\bar{e}_a^{pl}) - q_a \\ f_{depl} &= 3G(\tilde{e} - \Delta\bar{e}^{pl}) - q_{depl} \end{aligned} \tag{3.20}$$

where

$$\begin{aligned} q_a &= \left[A + B\left(\bar{e}_{old}^{pl} + \Delta\bar{e}_a^{pl}\right)^n\right]\cdot\left[1 + C\ln\left(\dot{\bar{e}}_a^{pl^*}\right)\right]\cdot\left[1 - T^{*m}\right] \\ q_{depl} &= \left[A + B\left(\bar{e}_{old}^{pl} + \Delta\bar{e}^{pl}\right)^n\right]\cdot\left[1 + C\ln\left(\dot{\bar{e}}^{pl^*}\right)\right]\cdot\left[1 - T^{*m}\right] \end{aligned} \tag{3.21}$$

The signs of the functions (equations (3.20)) are then used to see if the zero of the uniaxial form of the equivalent Mises stress is between the lower limit and the midpoint or between the midpoint and the upper limit. The appropriate limit is then adjusted accordingly to iterate towards the correct position of the zero (see Chapter 2).

The deviatoric stress can then be found (Appendix A):

$$\underline{S} = \frac{2G}{\left(1 + \frac{3G}{q_{depl}}\Delta\bar{e}^{pl}\right)}\underline{\hat{e}} \tag{3.22}$$

From this the normal to the yield surface, which is in the direction of the deviatoric stress, is found:

$$\underline{n} = \frac{3}{2}\frac{\underline{S}}{q_{depl}} \tag{3.23}$$

In this case the deviatoric plastic strain increment components are not needed explicitly (they are found implicitly in equation (3.26)) but it is nevertheless possible to calculate them at this point if required:

$$\Delta\underline{e}^{pl} = \Delta\bar{e}^{pl}\,\underline{n} \tag{3.24}$$

After calculating the deviatoric stress it is possible to find the new trial stress:

$$\underline{\sigma}_{new}^{trial} = \underline{S} - p\underline{I} \tag{3.25}$$

The increase in temperature can be formulated for an adiabatic analysis. In an adiabatic analysis the inelastic deformation produces a heat flux per unit volume [4]:

$$r^{pl} = \eta \underline{\sigma} : \underline{\dot{\varepsilon}}^{pl} \tag{3.26}$$

where η is the inelastic heat fraction (the fraction of the heat generated due to plastic work that is added to the thermal energy balance) and $\underline{\dot{\varepsilon}}^{pl}$ is the rate of plastic straining. The plastic strain increment in rate form is:

$$\underline{\dot{\varepsilon}}^{pl} = \dot{\varepsilon}^{pl}\underline{n} \tag{3.27}$$

At the end of the increment, after using the backward Euler method to integrate the plastic strain, the heat flux is given as:

$$r^{pl} = \frac{1}{2\Delta t}\eta\Delta\varepsilon^{pl}\underline{n} : \left(\underline{\sigma}_{old} + \underline{\sigma}_{new}\right) \tag{3.28}$$

Now the heat equation solved at each integration point is,

$$\rho c_p \dot{T} = r^{pl} \tag{3.29}$$

where ρ is the material density and c_p is the specific heat (which is assumed constant).

From equation (3.29) and substituting equation (3.28) the increase in temperature can be found [16, 18]:

$$\begin{aligned} \rho c_p \frac{\Delta T}{\Delta t} &= r^{pl} \\ \therefore \Delta T &= \Delta t \frac{r^{pl}}{\rho c_p} \\ \Delta T &= \frac{\Delta t}{\rho c_p}\frac{1}{2\Delta t}\eta\Delta\varepsilon^{pl}\underline{n} : \left(\underline{\sigma}_{old} + \underline{\sigma}_{new}\right) \\ \therefore \Delta T &= \frac{\eta\Delta\varepsilon^{pl}}{2\rho c_p}\underline{n} : \left(\underline{\sigma}_{old} + \underline{\sigma}_{new}\right) \end{aligned} \tag{3.30}$$

The new temperature is then:

$$T_{new} = T_{old} + \Delta T \tag{3.31}$$

The homologous temperature, T^* (see Chapter 2), is now found which will be used in the Johnson-Cook equation if the iteration loop is begun again.

$$T^* = \frac{T_{new} - T_{ref}}{T_{melt} - T_{ref}} \tag{3.32}$$

The iteration loop (equations (3.19) to (3.32)) is continued until the convergence criterion is met. Once the solution has converged the new trial stress (from equation (3.25)) is set as the new stress: $\underline{\sigma}_{new} = \underline{\sigma}^{trial}_{new}$. This concludes the plasticity section.

If the solution does not converge (e.g. choice of upper limit should be greater than 0.1) then the equivalent deviatoric plastic strain increment is set to the total equivalent deviatoric strain increment. The equivalent deviatoric plastic strain increment should always be less than or equal to the total equivalent deviatoric strain increment. For large deformations it is usually of similar magnitude since the elastic component is very small in comparison. The total equivalent deviatoric strain increment is now formulated. If the maximum number of iterations is reached and the solution is yet to converge then the old total deviatoric strain is found:

$$\underline{\hat{e}}_{old} = \frac{\underline{S}_{old}}{2G} \tag{3.33}$$

The old total equivalent deviatoric strain is found:

$$\tilde{e}_{old} = \sqrt{\frac{2}{3}\underline{\hat{e}}_{old} : \underline{\hat{e}}_{old}} \tag{3.34}$$

The equivalent deviatoric plastic strain increment is then set equal to the total equivalent deviatoric strain increment:

$$\Delta\bar{e}^{pl} = \tilde{e} - \tilde{e}_{old} \tag{3.35}$$

If this formulation is not used then in the case of non-convergence the equivalent deviatoric plastic strain increment will be set to one of the initial limits by the

Bisection method. The analysis is not simply stopped if the solution does not converge because in certain situations it is unavoidable (see next paragraph) to have non-convergence unless the initial upper limit is set very high. Setting this limit high to account for severe cases will be computationally expensive, since many more iterations will be required for each time increment.

If the equivalent deviatoric plastic strain rate is used to find the equivalent uniaxial yield stress then a sudden increase in yield stress occurs when the material initially goes plastic, causing non-convergence. This non-convergence is due to the fact that the state variables from the previous time step are used to test whether the material should enter the plasticity loop and for the first plastic increment the equivalent deviatoric plastic strain rate from the previous time step is zero (equation 3.11). If this initial non-convergence can be tolerated, while still keeping the solution relatively accurate, then the yield stress for the next time steps will have taken the equivalent plastic strain rate into account. The solution will then self-regulate and yielding will occur at the correct stress level thereafter.

As mentioned in section 3.5.1, an alternative is to simply use the total equivalent deviatoric strain rate when calculating the initial equivalent uniaxial yield stress. The total equivalent deviatoric strain rate is a reasonable approximation to the equivalent plastic deviatoric strain rate due to the small elasticity component [26]. This approach was implemented and there was no noticeable difference in the results obtained when using both methods in a sample Taylor test simulation. This approach is useful to speed up run times but since it doesn't alter the results it was not used.

3.5.4 Newton's Method

Newton's method is only implemented for the Johnson-Cook VUMAT. Newton's method is implemented using the total equivalent deviatoric strain rate to find the initial yield stress which allows the plasticity loop to be entered (as discussed in

3.5.3). The tolerance used for Newton's method is the same as for the Bisection method - 10^5. The total equivalent deviatoric strain rate is also used in the plasticity calculation and the homologous temperature from the previous time step is used. This means that the slope of the stress-strain curve is based on the previous time step but due to the small time steps in the explicit method this assumption is reasonable.

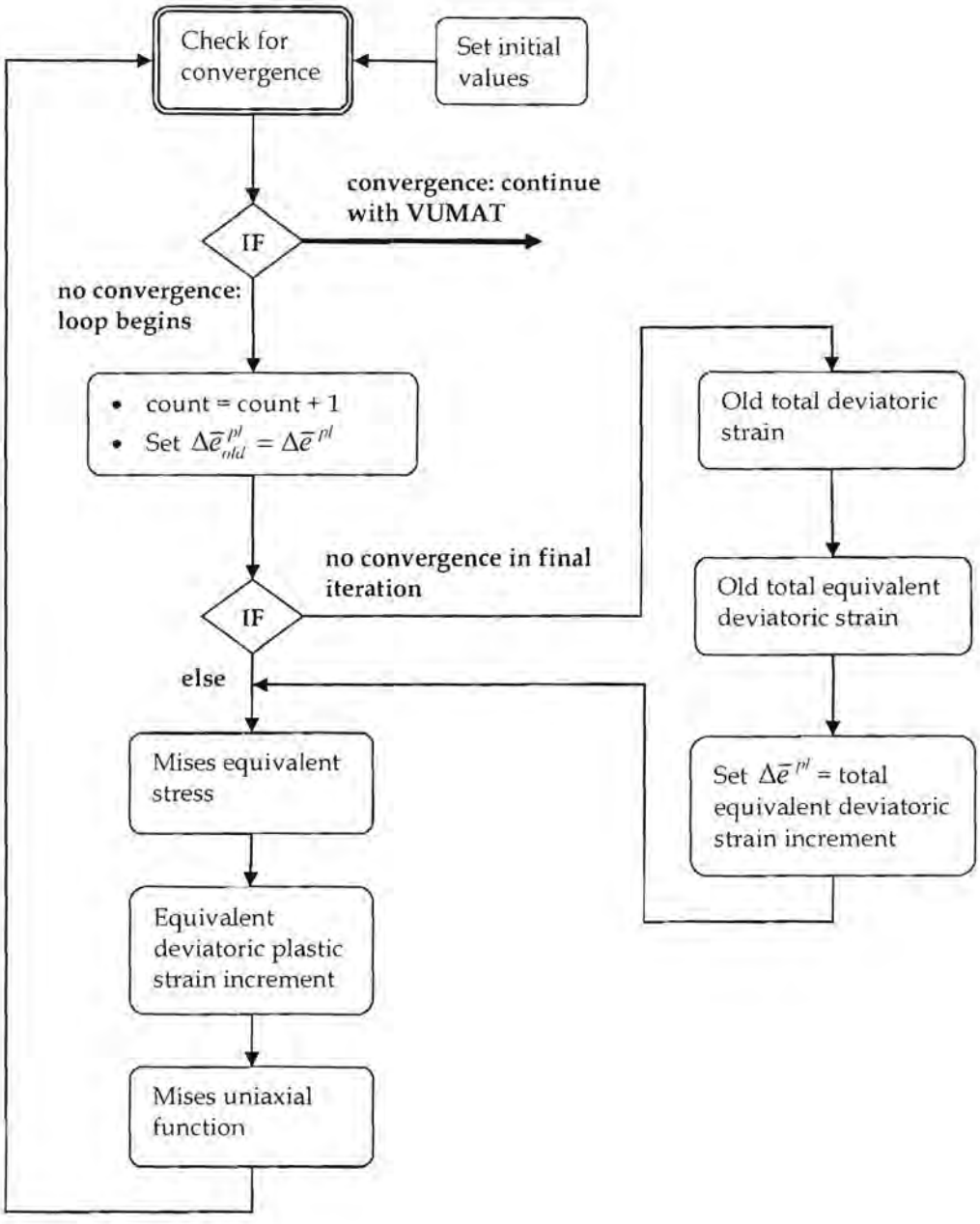


Figure 3.7: Algorithm for Newton's method plasticity loop.

To initialise the Newton loop the initial guess for the root (Chapter 2.8.2) is chosen arbitrarily to be 10^{-3} and the initial old value of equivalent deviatoric plastic strain increment is set to zero. Upon entering the Newton's method loop the old deviatoric plastic strain increment is set to the new value (which is the initial guess for the first iteration) (Figure 3.7):

$$\Delta\bar{e}^{pl}_{old} = \Delta\bar{e}^{pl} \tag{3.36}$$

The Mises equivalent stress is calculated as:

$$q = \left[A + B\left(\bar{e}^{pl}_{old} + \Delta\bar{e}^{pl}\right)^{n}\right] \cdot \left[1 + C\ln\left(\dot{\bar{e}}^{*}\right)\right] \cdot \left[1 - T^{*m}\right] \tag{3.37}$$

The increment in equivalent deviatoric plastic strain is found:

$$\Delta\bar{e}^{pl} = \Delta\bar{e}^{pl} + \frac{3G\left(\tilde{e} - \Delta\bar{e}^{pl}\right) - q}{3G + Bn\left(\bar{e}^{pl}_{old}\right)^{n-1}\left(1 + C\ln\left(\dot{\bar{e}}^{*}\right)\right)\left(1 - T^{*m}\right)} \tag{3.38}$$

The Mises uniaxial function value is calculated to test for convergence:

$$f_{depl} = 3G(\tilde{e} - \Delta\bar{e}^{pl}) - q \tag{3.39}$$

Equations 3.36 to 3.39 form the Newton's method loop which is repeated until convergence is achieved or the maximum number of iterations is exceeded. If there is no convergence then the total equivalent deviatoric strain increment is set as the equivalent deviatoric plastic strain increment as for the Bisection method.

3.5.5 Non-Iterative Method

The non-iterative or conventional radial return method was used in the EPIC implementation of the Johnson-Cook equation [24]. EPIC is an explicit finite element code. This is a non-iterative method because the deviatoric stresses are simply scaled in order to make the Mises equivalent stress equal to the yield stress (Figure 3.8). As with Newton's method, the non-iterative method is implemented using the total equivalent strain rate as an approximation to the equivalent deviatoric plastic strain rate and only for the Johnson-Cook VUMAT. Although there is no tolerance value, the results are an approximation and therefore not necessarily more accurate than the iterative methods. The non-iterative method is valid because the time

increments in the explicit method are very small. There is no appreciable difference in results between the iterative and non-iterative methods as shown in Figure 4.19.

The plasticity section is initiated by calculating the scaling factor:

$$\beta = \frac{\sigma^{yield}}{q^{trial}} \tag{3.40}$$

The trial deviatoric stresses are adjusted using the scaling factor:

$$\underline{S}_{new}^{trial} = \beta \underline{S}^{trial} \tag{3.41}$$

The elastic deviatoric strain increments are found:

$$\Delta \underline{e}^{el} = \frac{\underline{S}_{new}^{trial} - \underline{S}_{old}}{2G} \tag{3.42}$$

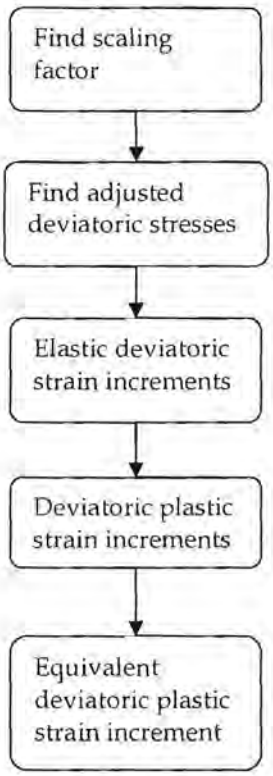


Figure 3.8: Algorithm for the non-iterative radial return method.

The deviatoric plastic strain increments are calculated as:

$$\Delta \underline{e}^{pl} = \Delta \underline{e} - \Delta \underline{e}^{el} \tag{3.43}$$

The equivalent deviatoric plastic strain increment can now be found:

$$\Delta\bar{e}^{pl} = \sqrt{\frac{2}{3}\Delta\underline{e}^{pl} : \Delta\underline{e}^{pl}} \tag{3.44}$$

The remainder of the implementation is the same as for the other methods.

3.5.6 Updating State Variables

Once the elastic and plastic sections are complete, the new values of the solution dependent state variables as well as the temperature are stored. Lastly, the specific internal energy as well as the specific dissipated inelastic energy is updated [4] (Figure 3.9). The stress-power increment is first calculated:

$$\Delta P_{\sigma} = \frac{1}{2}\left(\underline{\sigma}_{old} + \underline{\sigma}_{new}\right) : \Delta\underline{\varepsilon} \tag{3.45}$$

Then the new specific internal energy can be found:

$$E_{new}^{intern} = E_{old}^{intern} + \frac{\Delta P_{\sigma}}{\rho} \tag{3.46}$$

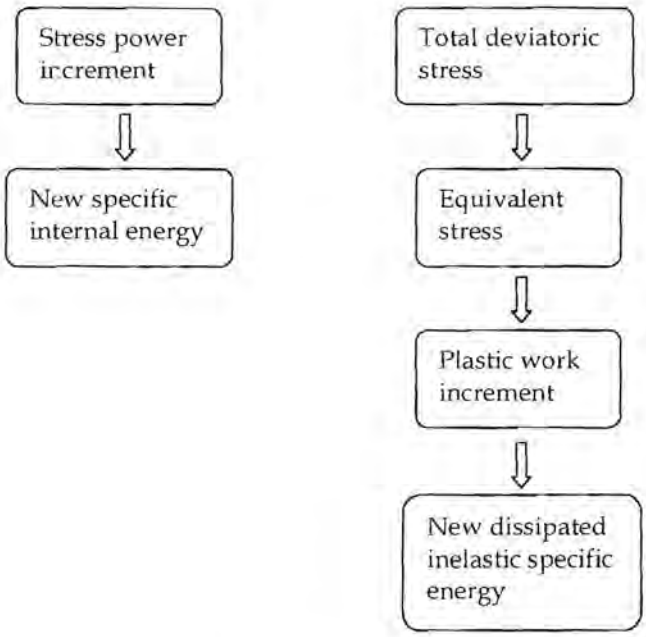


Figure 3.9: Algorithm to solve for the new specific energies.

In order to find the dissipated inelastic specific energy the equivalent stress is needed. Therefore the total deviatoric stress is (as before):

$$\underline{S}_{new} = \underline{\sigma}_{new} + p_{new}\underline{I} \tag{3.47}$$

From this the equivalent stress is found:

$$\bar{\sigma} = \sqrt{\frac{3}{2}\underline{S}_{new} : \underline{S}_{new}} \tag{3.48}$$

The plastic work increment is defined as:

$$\Delta W^{pl} = \Delta \bar{e}^{pl}\bar{\sigma} \tag{3.49}$$

Finally, the new dissipated inelastic specific energy is:

$$E_{new}^{dissip} = E_{old}^{dissip} + \frac{\Delta W^{pl}}{\rho} \tag{3.50}$$

3.6 ZERILLI-ARMSTRONG PLASTICITY

The Zerilli-Armstrong material constants for Armco-Iron (BCC) are given in Table 3.3 below [6]. The density is taken as 7890 kg/m^3 [6].

NO.	CONSTANT	VALUE	UNIT
1	**E** - Young's modulus	200	GPa
2	**υ** - Poisson's ratio	0.3	-
3	C_0 - Zerilli-Armstrong constant	65	MPa
4	C_1 - Zerilli-Armstrong constant	1033	MPa
5	C_3 - Zerilli-Armstrong constant	0.00698	K^{-1}
6	C_4 - Zerilli-Armstrong constant	0.000415	K^{-1}
7	C_5 - Zerilli-Armstrong constant	266	MPa
8	**n** - Zerilli-Armstrong constant	0.289	-
9	**η** - inelastic heat fraction	0.9	-
10	c_p - specific heat	452	J/kg.K

Table 3.3: Zerilli-Armstrong material constants for Armco-Iron [6].

The Zerilli-Armstrong material constants for OFHC copper (FCC) are given in Table 3.4 below [6]. The density is taken as 8960 kg/m^3 [6].

NO.	CONSTANT	VALUE	UNIT
1	E - Young's modulus	200	GPa
2	υ - Poisson's ratio	0.3	-
3	C_0 - Zerilli-Armstrong constant	65	MPa
4	C_2 - Zerilli-Armstrong constant	890	MPa
5	C_3 - Zerilli-Armstrong constant	0.0028	K^{-1}
6	C_4 - Zerilli-Armstrong constant	0.000115	K^{-1}
7	η - inelastic heat fraction	0.9	-
8	c_p - specific heat	383	J/kg.K

Table 3.4: Zerilli-Armstrong material constants for OFHC copper [6].

The Zerilli-Armstrong material model is implemented in exactly the same way as the Johnson-Cook model. The only difference being that the flow stress equations are now set as the Zerilli-Armstrong equations. The Johnson-Cook material model has one equation valid for all materials whereas the Zerilli-Armstrong model has separate equations for the different material types. The specific differences are shown for BCC materials first.

- The flow stress used as the yield criterion in Equation (3.11) becomes:

$$\bar{\sigma} = C_0 + C_1 \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}^{pl}\right) + C_5 \left(\bar{e}^{pl}\right)^n \tag{3.51}$$

- The uniaxial form of the Mises stresses for the lower limit and the midpoint value from Equation (3.20) and (3.21) become:

$$\begin{aligned} f_a &= 3G(\tilde{e} - \Delta\bar{e}_a^{pl}) - q_a \\ f_{depl} &= 3G(\tilde{e} - \Delta\bar{e}^{pl}) - q_{depl} \end{aligned} \tag{3.52}$$

where

$$q_a = C_0 + C_1 \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}_a^{pl}\right) + C_5 \left(\bar{e}_{old}^{pl} + \Delta \bar{e}_a^{pl}\right)^n$$
$$q_{dcpl} = C_0 + C_1 \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}^{pl}\right) + C_5 \left(\bar{e}_{old}^{pl} + \Delta \bar{e}^{pl}\right)^n \tag{3.53}$$

And secondly the differences for FCC materials are shown.

- The flow stress used as the yield criterion in Equation (3.11) becomes:

$$\bar{\sigma} = C_0 + C_2 \left(\bar{e}^{pl}\right)^{0.5} \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}^{pl}\right) \tag{3.54}$$

- The uniaxial form of the Mises stresses for the lower limit and the midpoint value from Equation (3.20) and (3.21) become:

$$f_a = 3G(\tilde{e} - \Delta \bar{e}_a^{pl}) - q_a$$
$$f_{dcpl} = 3G(\tilde{e} - \Delta \bar{e}^{pl}) - q_{dcpl} \tag{3.55}$$

where

$$q_a = C_0 + C_2 \left(\bar{e}_{old}^{pl} + \Delta \bar{e}_a^{pl}\right)^{0.5} \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}_a^{pl}\right)$$
$$q_{dcpl} = C_0 + C_2 \left(\bar{e}_{old}^{pl} + \Delta \bar{e}^{pl}\right)^{0.5} \exp\left(-C_3 T + C_4 T \ln \dot{\bar{e}}^{pl}\right) \tag{3.56}$$

The only other difference is that it is not necessary to calculate the homologous temperature for the Zerilli-Armstrong models.

3.7 SUMMARY

In this chapter the detailed formulation of the various material models for the VUMAT implementations have been presented. The material models were formulated using the variables available to the VUMAT and the VUMAT coding is therefore easy to understand. Newton's method and a non-iterative method were presented as alternatives to the Bisection method for solving for the equivalent deviatoric plastic strain increment in the Johnson-Cook VUMAT. These methods can easily be extended to the Zerilli-Armstrong VUMAT.

CHAPTER 4
VERIFICATION

4.1 INTRODUCTION

In order to address the concerns over the reliability of computer simulations, a technological area labelled validation and verification was created [33]. This technological area deals with "the creation, study and documentation of tools for assessing the predictability of CM-based methods and programs" [33]. CM stands for computational mechanics. It is important to note the difference between verification and validation [33].

- **Validation:** determining the appropriateness of the principles and mathematical methods used to develop a simulation tool e.g. comparing simulation results with experimental results.
- **Verification:** determining if the simulation tool can correctly produce results consistent with the models on which it was based e.g. using benchmark tests to find coding or software errors.

The implementation of the Johnson-Cook [5] and Zerilli-Armstrong [6] constitutive equations as discussed in Chapter 3 is verified in this chapter. The strategy employed in this verification process is to first test the material models by using single element tests. These single element tests are performed using various loading conditions. The results obtained from the Abaqus Johnson-Cook equation are compared to the results obtained from the VUMATs of the Johnson-Cook and Zerilli-Armstrong equations. This same procedure is then repeated, but with an increasing number of elements being used in the test configuration. The results obtained using the Johnson-Cook VUMAT should be identical to the results obtained

using the Abaqus Johnson-Cook equation since the same version of the equation is used. Abaqus does not contain the Zerilli-Armstrong material model as part of its material library and therefore the Zerilli-Armstrong material model response is compared to the Johnson-Cook response in a more qualitative sense. For the single element tests all the Mises equivalent stress results are reported while for the multiple element tests only the results of exceptional cases are reported. The remainder of the results are reported in Appendix C.

In the multiple element tension test results presented in this chapter an instability or localisation phenomena becomes evident. The focus of this work is the implementation and verification of the VUMATs. For verification purposes it is sufficient to show that the results obtained using the Johnson-Cook VUMAT are identical to the results obtained using the Abaqus Johnson-Cook material model. Therefore, when using the Johnson-Cook material model, as long as the VUMAT results and the Abaqus results both capture the instability the VUMAT implementation can be verified. Due to different tolerances possibly being used in the material models it is very unlikely that the moment of initiation of the instability is identical for the Johnson-Cook VUMAT and the Abaqus Johnson-Cook material model. The results of the multiple element tension tests which contain an instability are presented in this chapter but any detailed discussion concerning the instability is omitted. A simple theoretical analysis of instability in the tension test is presented in Appendix E together with some discussion on the performance of the material models in capturing the initiation of the instability.

4.2 ELEMENT TEST SET-UP

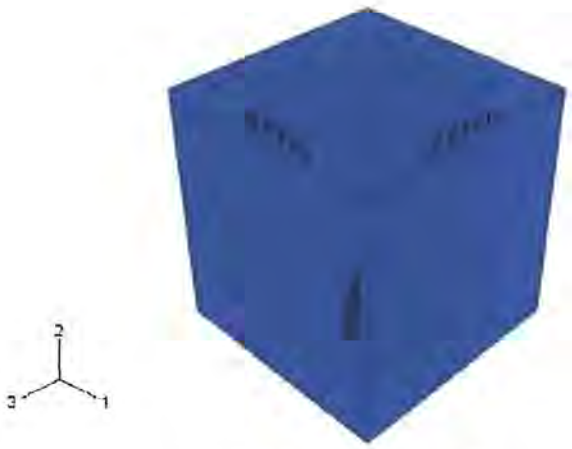


Figure 4.1: Material cube dimensions.

The element tests are performed using a cube of material with dimensions as shown in Figure 4.1. These dimensions are chosen arbitrarily. The same geometry is used for the single as well as the multiple element tests. The cube is meshed using C3D8R elements (Figure 4.2): continuum, 8 node "brick" elements with reduced integration and hourglass control. These elements are linear strain elements whose active degrees of freedom are the displacements in the 1, 2 and 3 directions respectively (u_x, u_y, u_z).

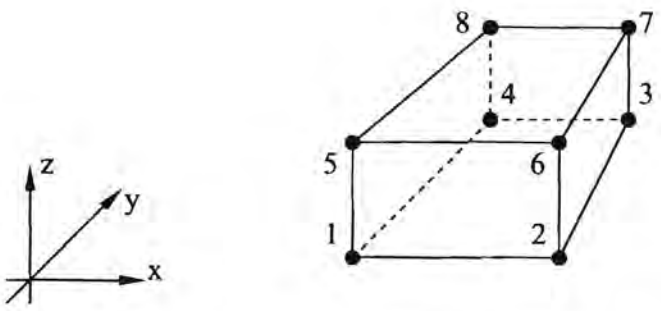


Figure 4.2: Node numbering of the C3D8R brick element [4].

The loading is applied to the cube using prescribed velocity conditions. The particular loading condition for each test will be shown in the respective sub-chapter. Two velocities are used for each loading condition, 0.1 m/s and 10 m/s. The higher value of velocity is 100 times greater than the lower value, creating a lower and a higher strain rate loading condition.

4.3 SINGLE ELEMENT TESTS

4.3.1 COMPRESSION

The loading condition used for the compression tests is shown in Figure 4.3. The prescribed velocity is applied to the entire geometric face on which it acts and therefore is independent of the number of elements. When the mesh is applied or altered, the velocity will be assigned to whichever nodes are on the geometric face.

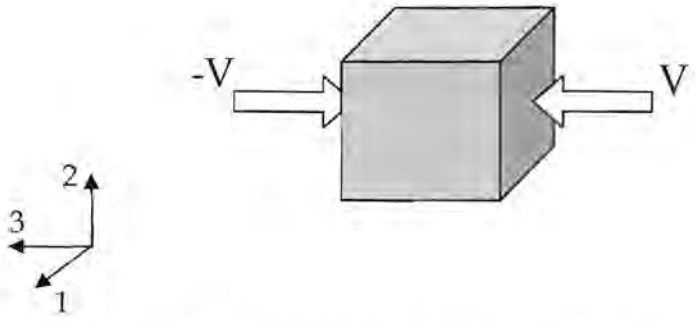


Figure 4.3: Compression loading conditions.

There are no other loading conditions apart from the prescribed velocities and therefore the cube is free to deform in compression. The time over which the velocity is applied needs to be calculated and limited so that severe deformation is avoided. This point should be noted for all prescribed velocity (displacement) loading conditions because, using the compression test as an example, if the time is too long then the opposite faces will close in on each other to the point where they will try to pass through each other. This is physically not possible but the software will try to do what it is told to do. The simulation results will be incorrect through

no fault of the finite element program or the material model. The times used for each simulation can be seen in the graphs of the results. An example of the amount of deformation allowed in the simulations is shown in Figure 4.4. There is sufficient deformation to test the material model but not too much so as to cause incorrect results due to severe element distortion.

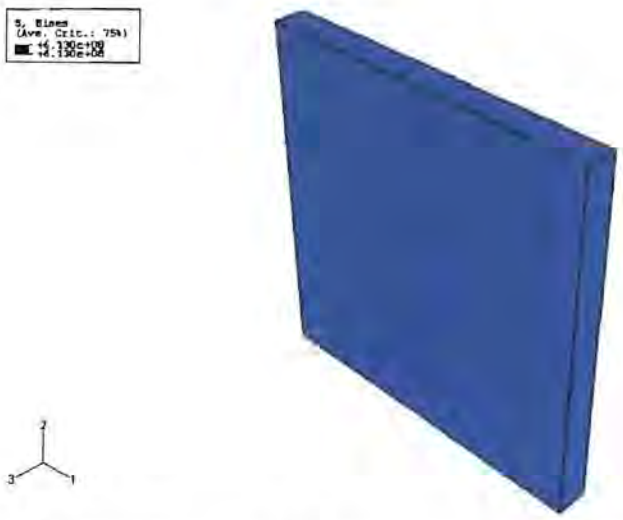
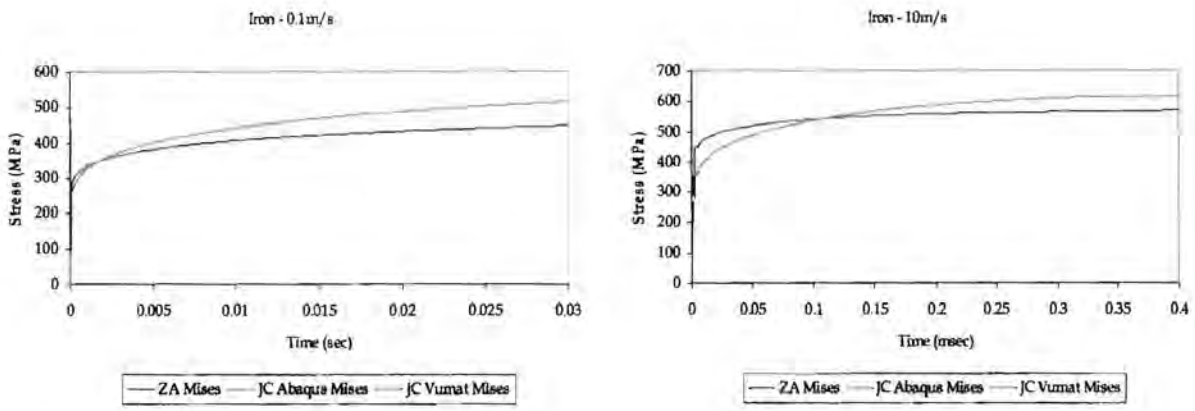


Figure 4.4: Compression - Armco-Iron, 10 m/s using Abaqus Johnson-Cook.

The values of the Mises equivalent stress are reported for the different material models and velocities. In each of the graphs shown in this chapter, results are shown for the Abaqus Johnson-Cook, VUMAT Johnson-Cook and VUMAT Zerilli-Armstrong material models. The term "collective results" will therefore refer to results obtained from all three material models. The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.5 for Armco-Iron.

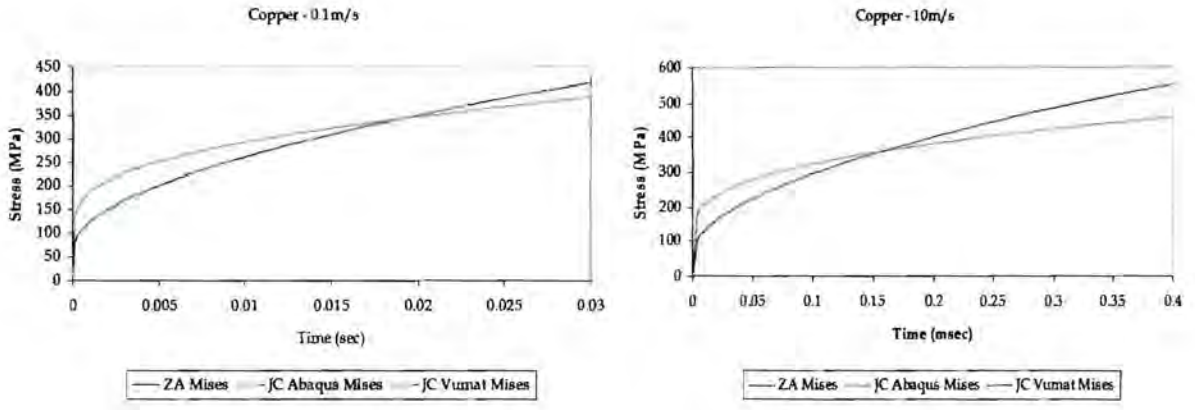
In Figure 4.5 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and therefore the VUMAT is functioning correctly for this loading condition.



a) b)

Figure 4.5: Compression - Mises equivalent stress for Armco-Iron a) 0.1 m/s, b) 10 m/s.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.6 for OFHC Copper.



a) b)

Figure 4.6: Compression - Mises equivalent stress for OFHC Copper a) 0.1 m/s, b) 10 m/s.

In Figure 4.6 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and therefore the VUMAT is functioning correctly for this loading condition.

4.3.2 SHEAR

The loading conditions for the shear test are shown in Figure 4.7.

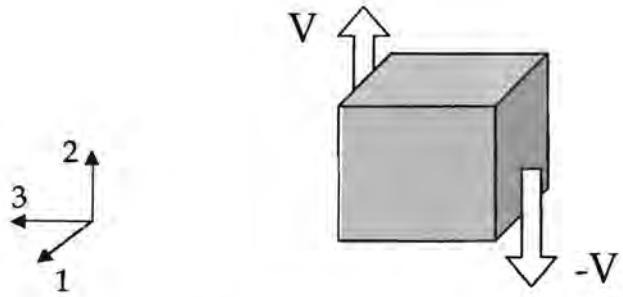


Figure 4.7: Shear test loading conditions.

The velocity loading is applied to the entire geometric face to cause the shear. The faces to which the velocities are applied are restrained from moving towards each other i.e. the displacement in the 3 direction is zero causing the faces to remain parallel to each other. An example of the final deformation is shown in Figure 4.8.

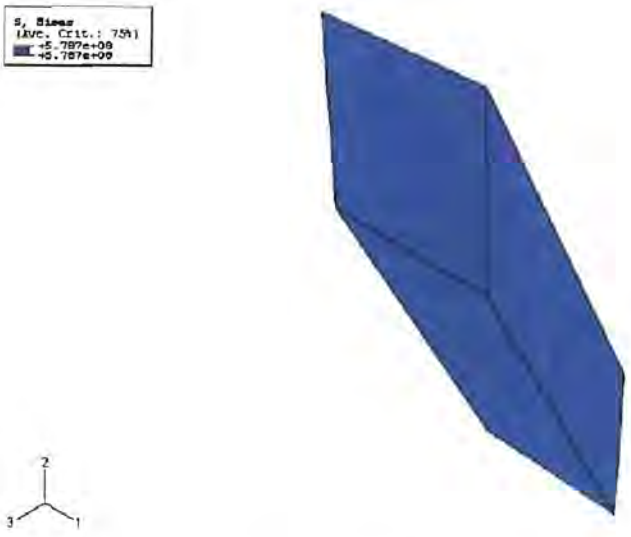
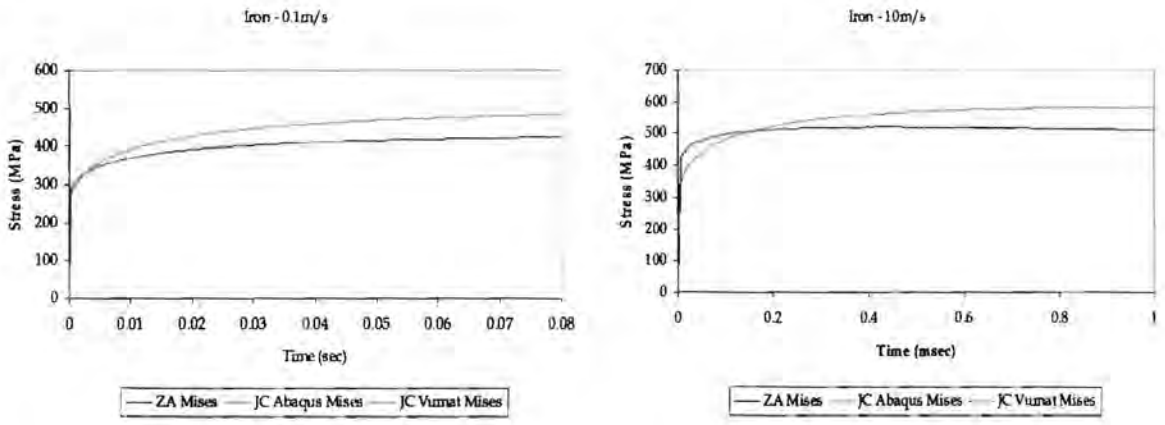


Figure 4.8: Shear - Armco-Iron, 10 m/s using Johnson-Cook VUMAT.

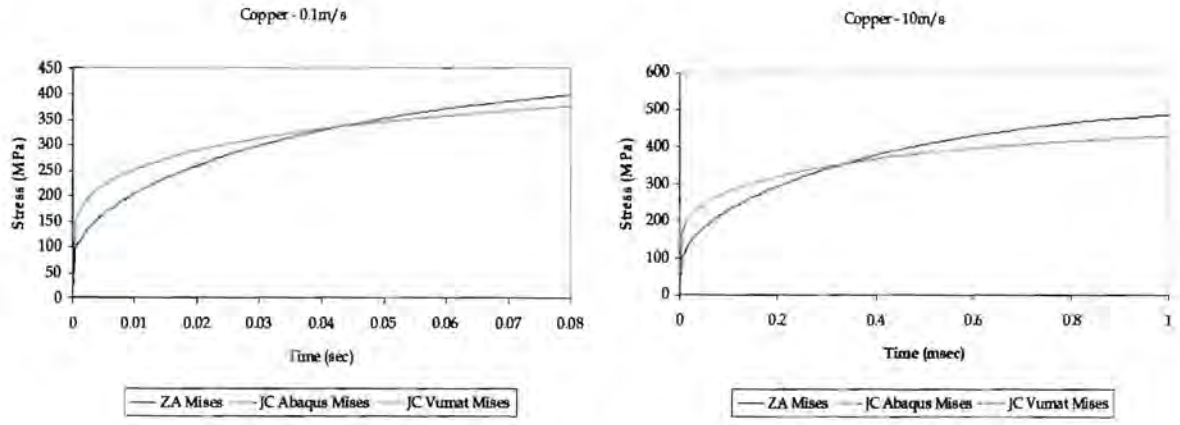
The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.9 for Armco-Iron.



a) b)

Figure 4.9: Shear - Mises equivalent stress for Armco-Iron a) 0.1 m/s, b) 10 m/s.

In Figure 4.9 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and the VUMAT is therefore functioning correctly for this loading condition. The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.10 for OFHC Copper.



a) b)

Figure 4.10: Shear - Mises equivalent stress for OFHC Copper a) 0.1 m/s, b) 10 m/s.

In Figure 4.10 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and therefore the VUMAT is functioning correctly for this loading condition.

4.3.3 TENSION

The loading conditions for the tensile test are shown in Figure 4.11.

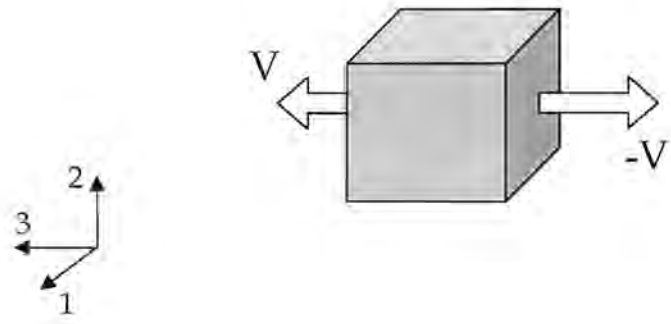


Figure 4.11: Tension test loading conditions.

The velocity loading is applied to the entire geometric face to cause the tension. There is no other loading applied to the element. An example of the final deformation is shown in Figure 4.12.

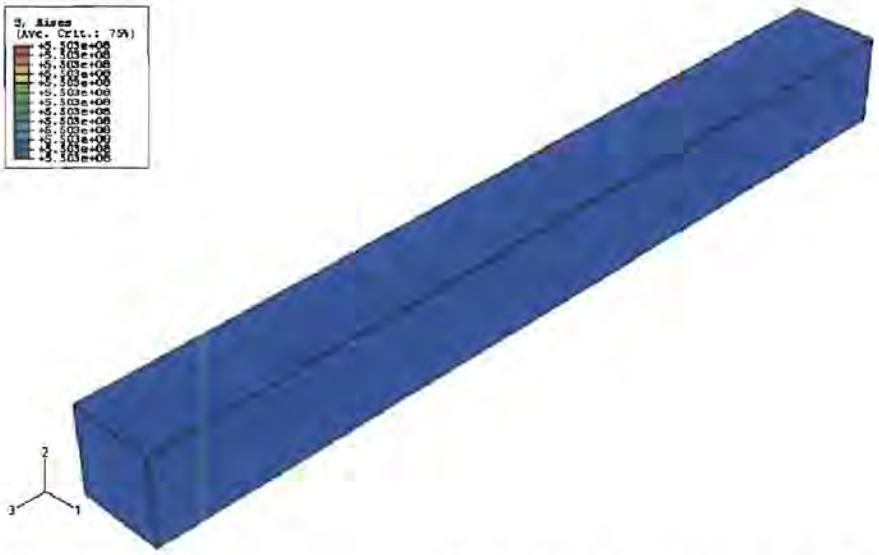
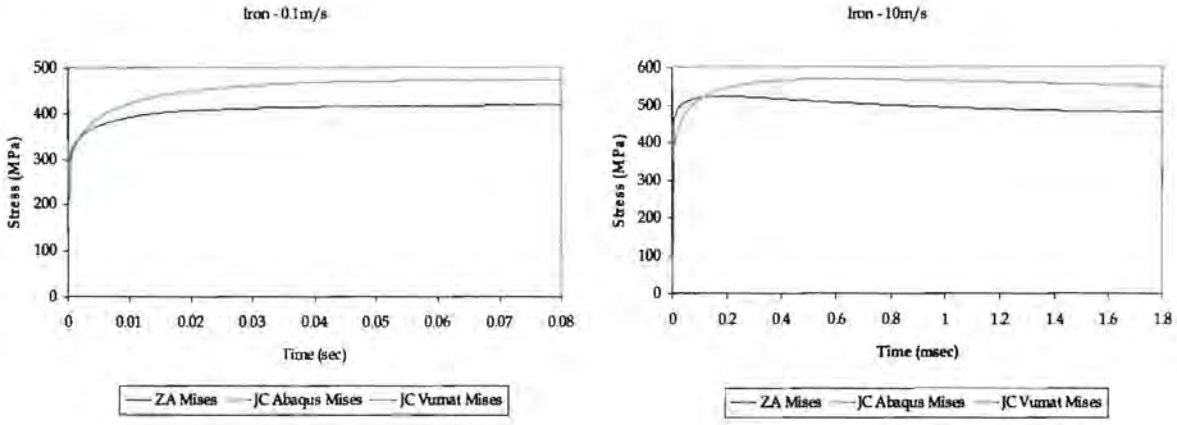


Figure 4.12: Tension - Armco-Iron, 10 m/s using Johnson-Cook VUMAT.

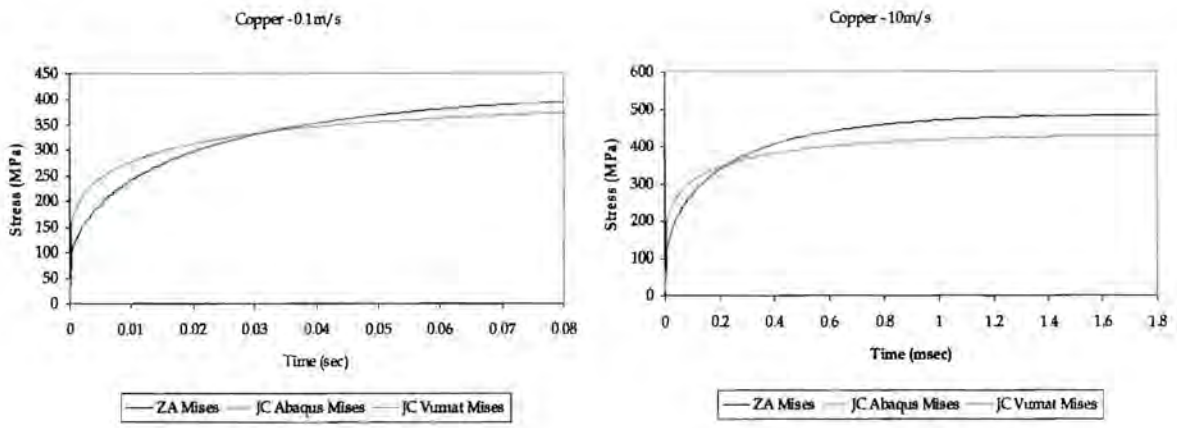
The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.13 for Armco-Iron.



a) b)

Figure 4.13: Tension - Mises equivalent stress for Armco-Iron a) 0.1 m/s, b) 10 m/s.

In Figure 4.13 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and therefore the VUMAT is functioning correctly for this loading condition. The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are reported in Figure 4.14 for OFHC Copper.



a) b)

Figure 4.14: Tension - Mises equivalent stress for OFHC Copper a) 0.1 m/s, b) 10 m/s.

In Figure 4.14 a) and b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are superimposed. The results are indistinguishable and therefore the VUMAT is functioning correctly for this loading condition.

4.3.4 Combined Loading

In order to have a set of results which are based on multi-axial loading conditions a combination of tension and shear was used. This arbitrary choice of multi-axial loading is shown in Figure 4.15 and is hereafter referred to as "combined loading".

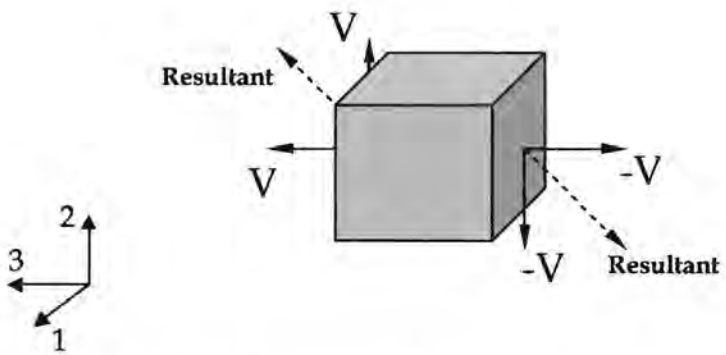
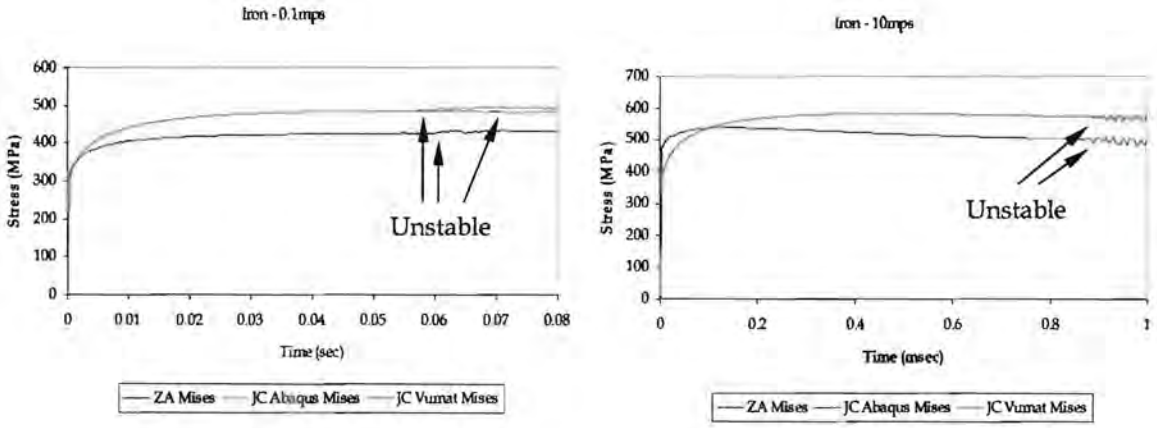


Figure 4.15: Combined loading conditions.

The velocity loading is applied to the entire geometric face to cause the combined loading. There is no other loading applied to the element. An example of the final deformation produced is shown in Figure 4.16.

Figure 4.16: Combined loading - Armco-Iron, 10 m/s using Johnson-Cook VUMAT.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.17 for Armco-Iron.



a) b)

Figure 4.17: Combined loading - Mises equivalent stress for Armco-Iron a) 0.1 m/s, b) 10 m/s.

Figure 4.17 a) shows that the solution for the Mises equivalent stress begins to become unstable from about 0.06 seconds for the Zerilli-Armstrong VUMAT and from about 0.055 seconds for the Johnson-Cook VUMAT for 0.1 m/s. The Abaqus Johnson-Cook solution becomes unstable from about 0.068 seconds. The solutions for all the material models seem to correct themselves and remain close to the correct solution. The VUMAT solutions also become unstable for the 10 m/s loading at about 0.9 milliseconds as shown by Figure 4.17 b). The other variables such as temperature do not become unstable (Appendix C).

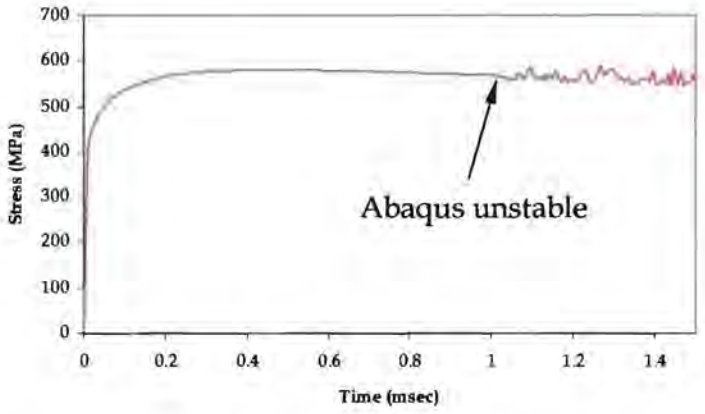


Figure 4.18: Combined loading - Abaqus Johnson-Cook Mises stress for Armco-Iron at 10 m/s.

Initially it was thought that for the 10 m/s loading the Abaqus Johnson-Cook solution does not become unstable. However, when the simulation was allowed to run longer the Abaqus Johnson-Cook solution also became unstable at about 1.01 milliseconds (Figure 4.18). The unstable results seem to be numerical inaccuracies probably due to the choice of tolerance in the plasticity calculations. It should be noted that this inaccuracy is not the same as the instability in the tension test.

In order to check whether the early initiation of unstable results in the VUMAT simulations was due to the Bisection method, the simulations for Armco-Iron at 10 m/s were also performed using the VUMATs which incorporate Newton's method and the non-iterative method (only for the Johnson-Cook material model) (Figure

4.19). The non-iterative method eliminates the need for a tolerance but it is an approximation and is therefore not necessarily more accurate. The tolerance used for Newton's method was the same as for the Bisection method.

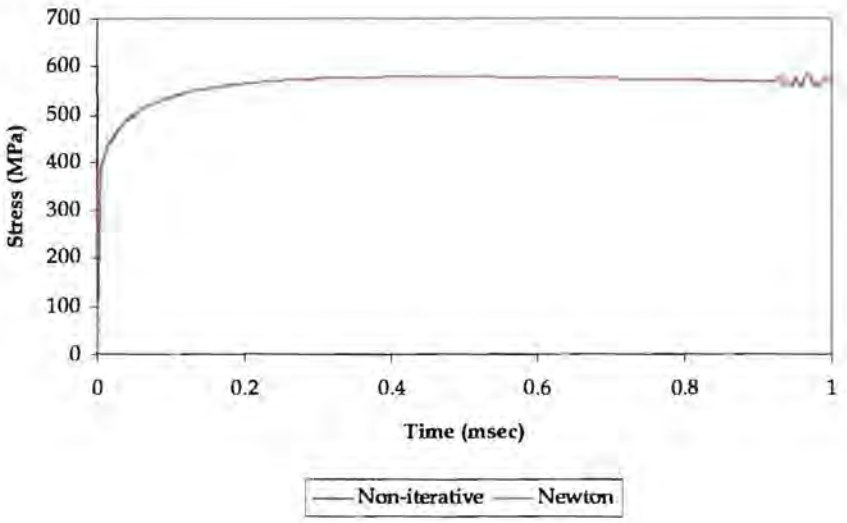
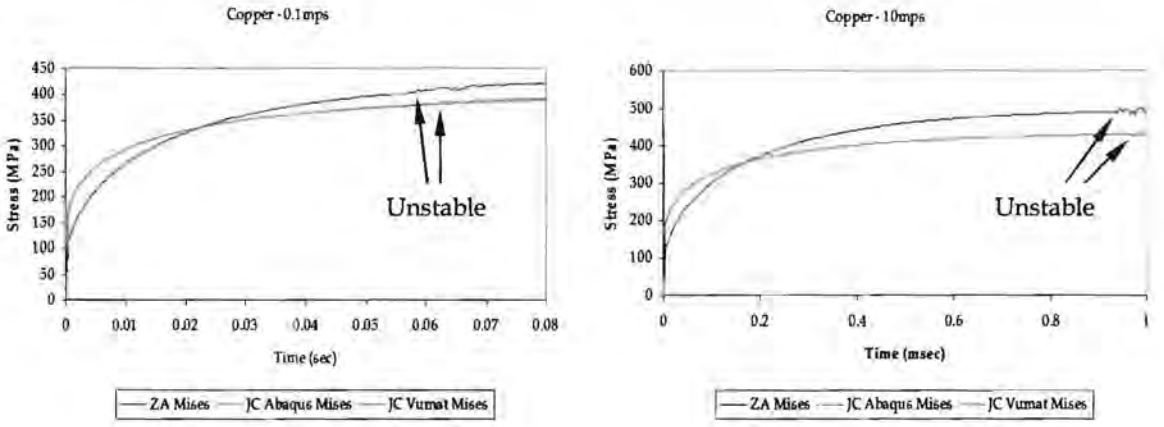


Figure 4.19: Combined loading - Mises stress for Armco-Iron at 10 m/s using Newton and non-iterative methods.

Figure 4.19 show that the Mises stress curves obtained using Newton's method and the non-iterative method are indistinguishable. The unstable results initiate at about 0.92 milliseconds using Newton's method and the non-iterative method. This is a slight improvement over the Bisection method results in Figure 4.17 b) which shows the unstable solution beginning at 0.91 milliseconds (for the Johnson-Cook VUMAT). For this loading condition the VUMAT results become unstable slightly before the Abaqus results.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s and a velocity of 10 m/s are shown in Figure 4.20 for OFHC Copper.



a) b)

Figure 4.20: Combined loading - Mises equivalent stress for OFHC Copper a) 0.1 m/s, b) 10 m/s.

The solution becomes slightly unstable at about 0.06 seconds using the VUMATs for OFHC Copper at 0.1 m/s but once again the other variables such as temperature do not become unstable (Appendix C). The simulations using the Abaqus Johnson-Cook material model need to run longer in order for the solution to become unstable.

4.4 8 (2X2X2) ELEMENTS

For the multiple element tests only the tension and shear loading conditions will be used (an arbitrary choice). This is done to reduce the number of simulations performed due to time constraints.

4.4.1 TENSION

The velocity loading is applied in the same way as for the single element test to produce the tensile load (Figure 4.11). Figure 4.21 shows the mesh used to obtain 8 elements and which elements the results for section 4.4 are reported from. The internal Abaqus labelling of these elements cannot be altered by the user and

therefore the different element labelling scheme for the different models is unavoidable.

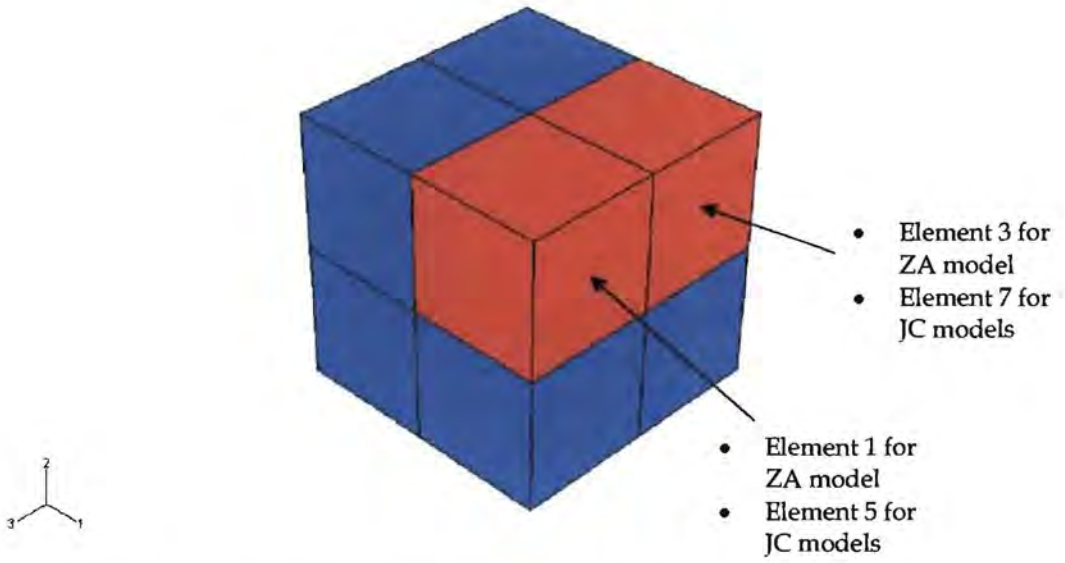


Figure 4.21: Position of elements used in extracting results for 8 element test.

An example of the typical final deformation produced is shown in Figure 4.22.

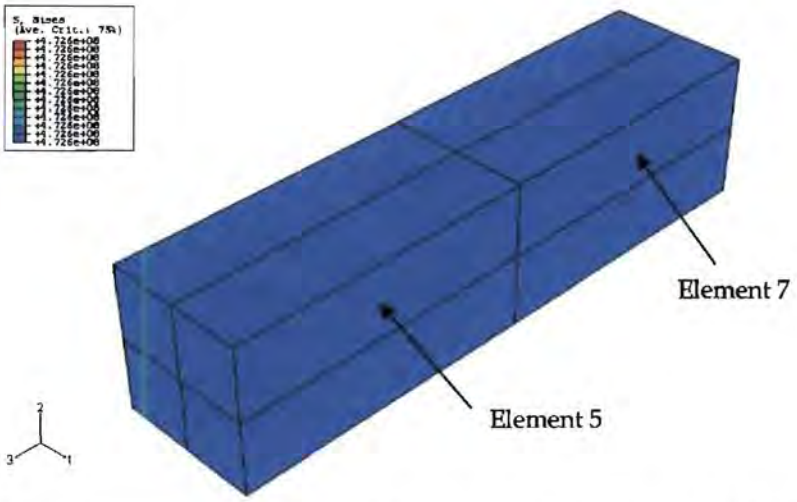
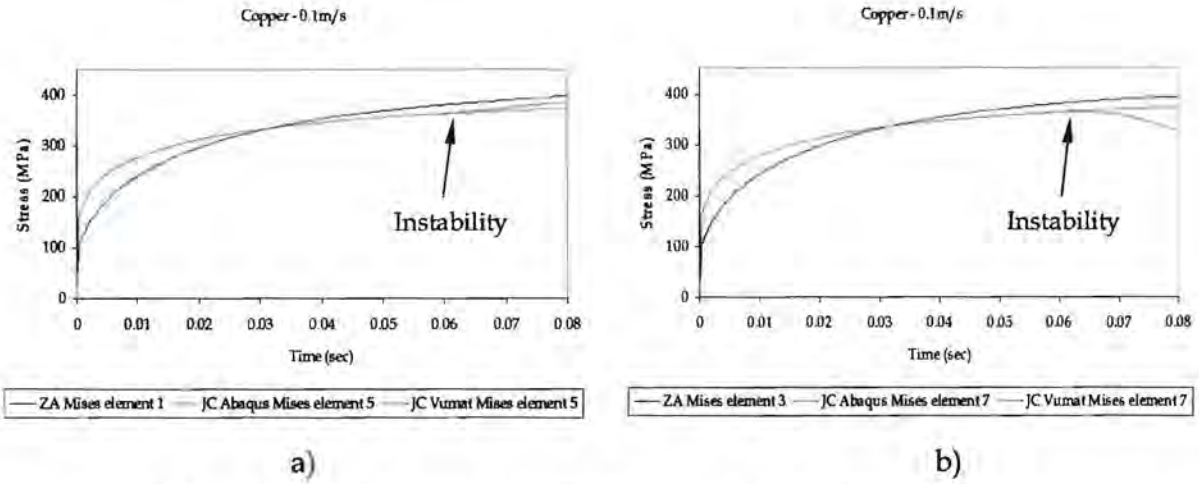


Figure 4.22: Tension (8 elements) - Armco-Iron, 10 m/s using Abaqus Johnson-Cook.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s are shown in Figure 4.23 for OFHC Copper. The remainder of the results can be found in Appendix C.



a) b)

Figure 4.23: Tension - Mises equivalent stress for OFHC Copper for 0.1 m/s a) element 1 & 5, b) element 3 & 7.

Figure 4.23 shows that the results for the VUMAT and Abaqus version of Johnson-Cook are indistinguishable until about 0.06 seconds. Thereafter the Johnson-Cook VUMAT results begin to deviate from the Abaqus results signifying the beginning of instability (Appendix E). It is not clear why the Johnson-Cook VUMAT produces an instability while the Zerilli-Armstrong VUMAT does not. For multiple element tension tests this instability or localisation usually becomes evident at large deformations. The VUMAT material models usually show earlier manifestations of the instability than the Abaqus Johnson-Cook material model. For the remainder of the 8 element tensile tests, the 10 m/s tests for Armco-Iron and OFHC Copper do not show any signs of instability while the 0.1 m/s test for Armco-Iron shows very slight instability (Appendix C).

4.4.2 SHEAR

The velocity loading is applied in the same way as for the single element test to produce the shear load (Figure 4.7). An example of the typical final deformation is shown in Figure 4.24.

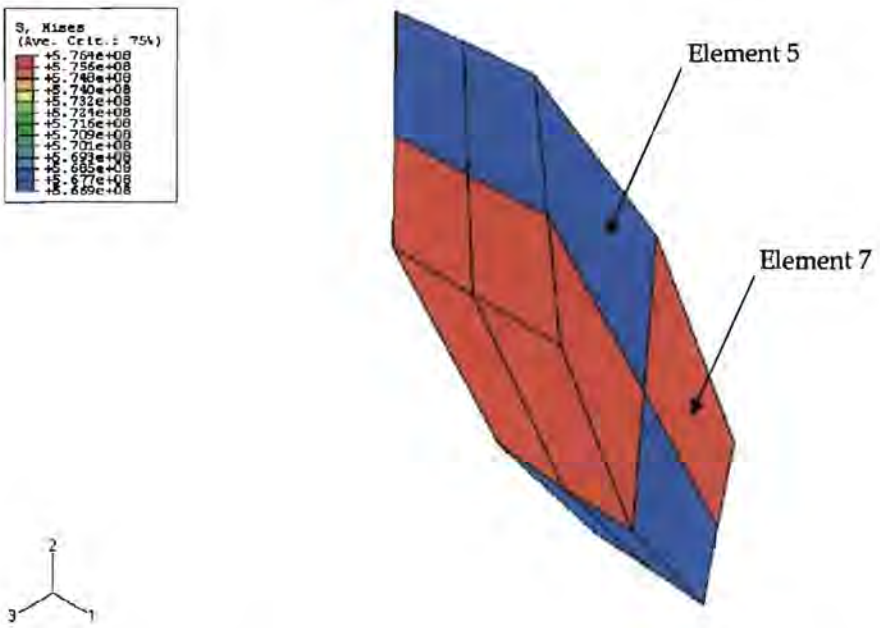
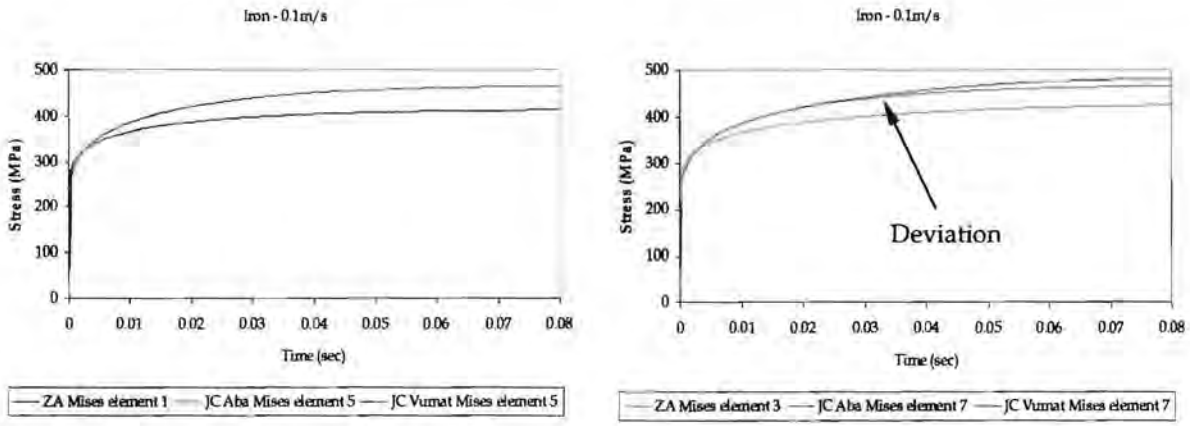


Figure 4.24: Shear (8 elements) - Armco-Iron, 10 m/s using Abaqus Johnson-Cook.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s are shown in Figure 4.25 for Armco-Iron.

The element numbering is the same as for the tension test (Figure 4.21). In Figure 4.25 a) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress are indistinguishable. In Figure 4.25 b) the curves for the Abaqus Johnson-Cook Mises stress and the VUMAT Johnson-Cook Mises stress deviate slightly as the simulation progresses. This deviation between the VUMAT and Abaqus results is not seen in the remainder of the results for Armco-Iron at 0.1 m/s such as the temperature (Appendix C). The reason for the deviation is not known.



a) b)

Figure 4.25: Shear - Mises equivalent stress for Armco-Iron for 0.1 m/s a) element 1 & 5, b) element 3 & 7.

The results for the Mises stress for Johnson-Cook at 10 m/s show the same deviation between the VUMAT and Abaqus results for element 7 (Figure 4.21). The remainder of the results do not show this deviation except the S23 stress component for the 10 m/s tests which show a similar (though less pronounced) deviation (Appendix C).

4.5 64 (4X4X4) ELEMENTS

4.5.1 TENSION

The velocity loading is applied in the same way as for the single element test to produce the tensile load (Figure 4.11). Figure 4.26 shows which elements the results are reported from and the internal Abaqus labelling of these elements as well as the mesh used to obtain 64 elements.

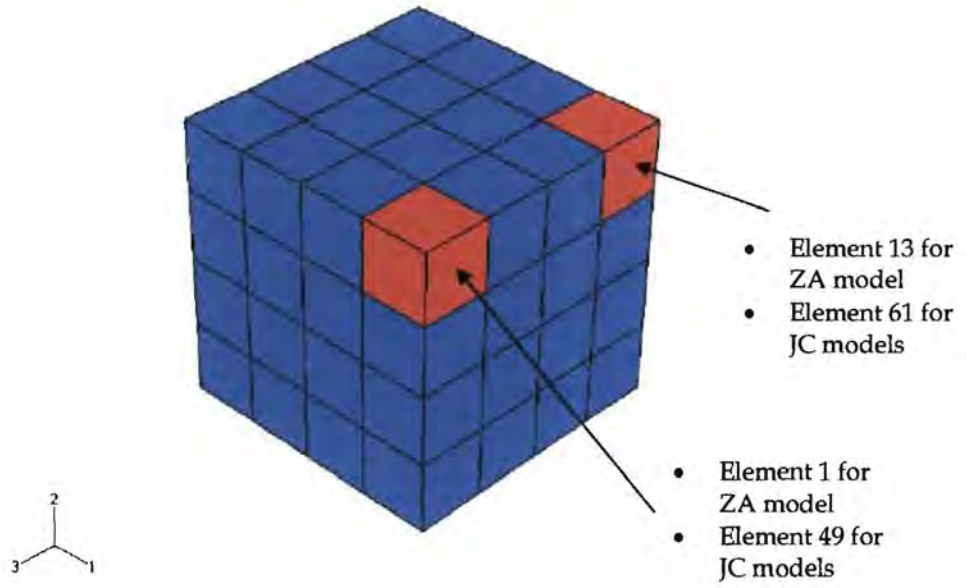


Figure 4.26: Position of elements used in extracting results for 64 element test.

An example of the typical final deformation is shown in Figure 4.27. The stress contour in Figure 4.27 is unsymmetrical indicating the beginning of an instability or localisation (Appendix E). The increased local deformation causes an associated increase in stress.

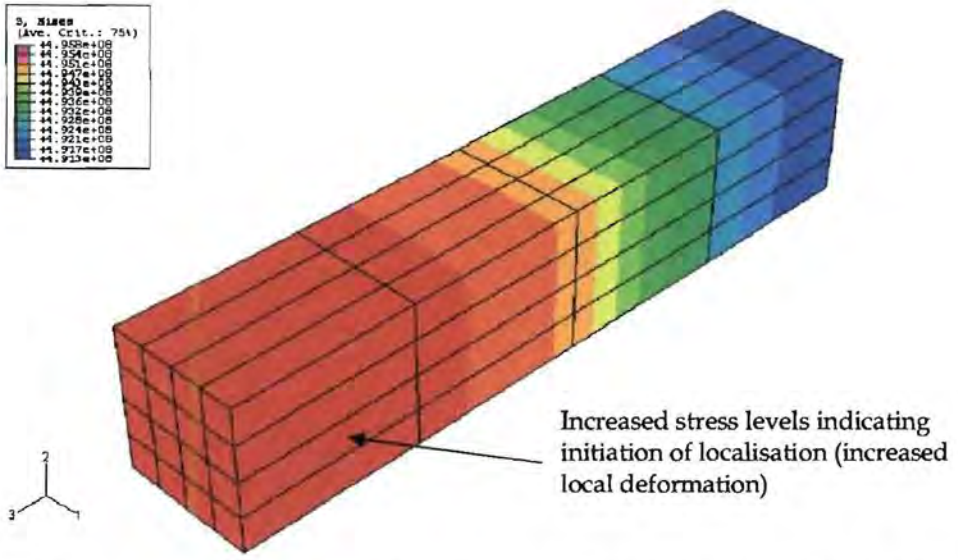
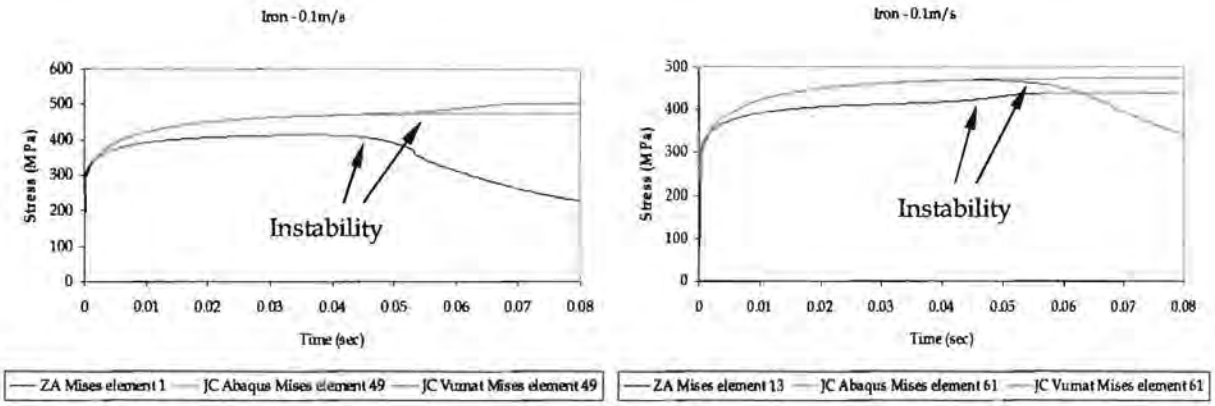


Figure 4.27: Tension (64 elements) - Armco-Iron, 10 m/s using Zerilli-Armstrong.

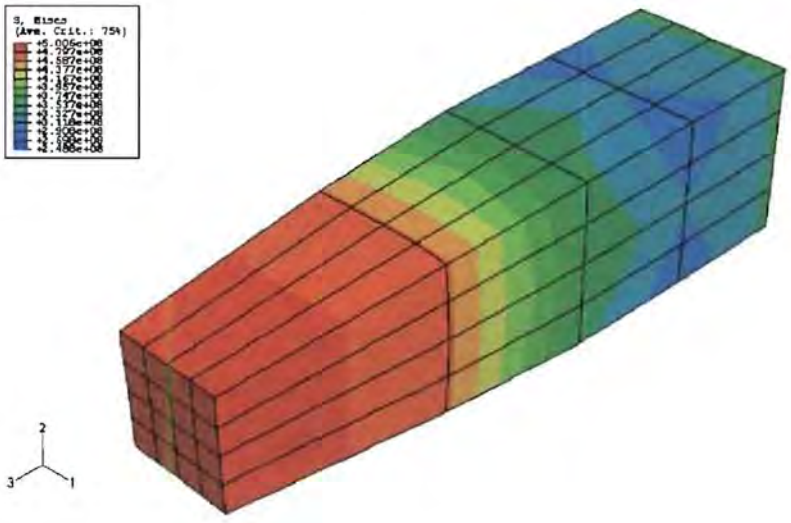
The collective results for the Mises equivalent stress for a velocity of 0.1 m/s are shown in Figure 4.28 for OFHC Copper. The remainder of the results can be found in Appendix C.



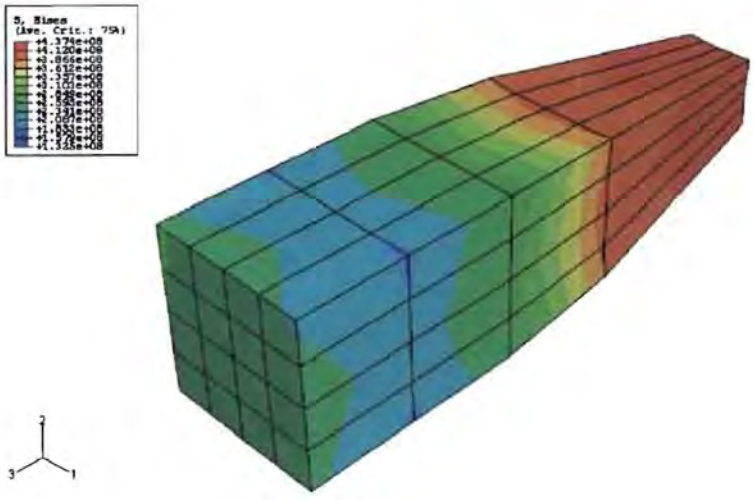
a) b)

Figure 4.28: Tension - Mises equivalent stress for Armco-Iron at 0.1 m/s a) element 1 & 49, b) element 13 & 61.

In Figure 4.28 the VUMAT material model results for Mises stress show instabilities beginning at about 0.045 seconds for the Zerilli-Armstrong results and about 0.055 seconds for the Johnson-Cook results. Once the instability occurs, the deformation throughout the model is non-uniform and the results from the material models (such as stress) therefore become non-uniform. This is seen in Figure 4.27 where the stress distribution is no longer symmetrical and more dramatically in Figure 4.29 where the deformation is clearly non-uniform. Figures 4.29 a) and b) show the final deformation of the tension test corresponding to the VUMAT results in Figure 4.28 for Johnson-Cook and Zerilli-Armstrong respectively.



a)



b)

Figure 4.29: Instability of Armco-Iron tension test at 0.1 m/s for 64 elements a) Johnson-Cook VUMAT, b) Zerilli-Armstrong VUMAT.

4.5.2 SHEAR

The velocity loading is applied in the same way as for the single element test to produce the shear load (Figure 4.7). An example of the typical final deformation is shown in Figure 4.30.

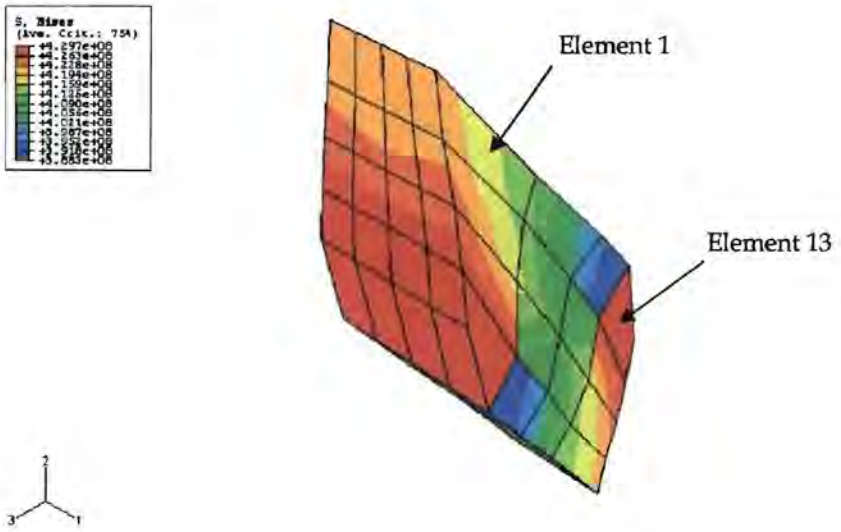
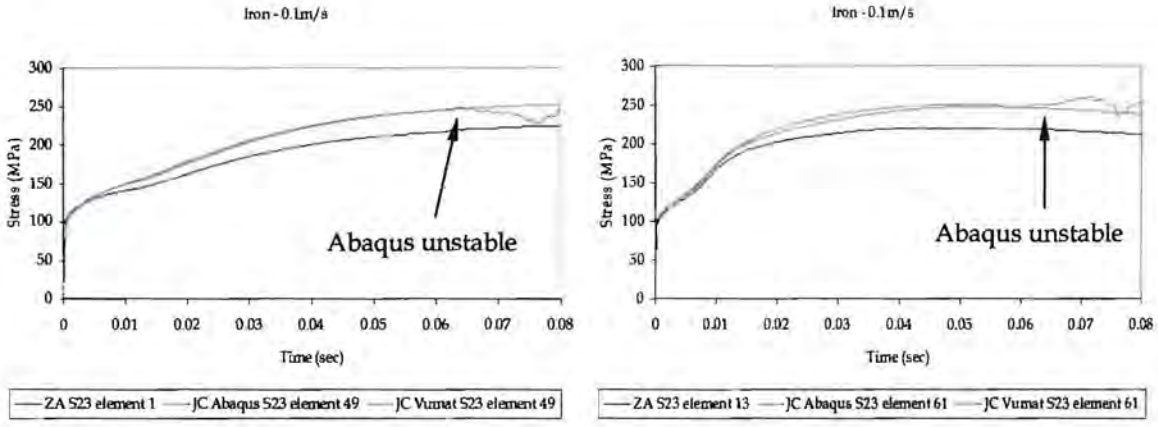


Figure 4.30: Shear (64 elements) - Armco-Iron, 0.1 m/s using Zerilli-Armstrong.

The collective results for the S23 stress component for a velocity of 0.1 m/s are shown in Figure 4.31 for Armco-Iron. The remainder of the results can be found in Appendix C.



a) b)

Figure 4.31: Shear - S23 stress component for Armco-Iron at 0.1 m/s
a) element 1 & 49, b) element 13 & 61.

In Figure 4.31 a) and b) it is the Abaqus Johnson-Cook solution which becomes unstable first. The unstable results of the S23 stress component are not as pronounced in the Mises stress solution and the remainder of the Abaqus results

such as the temperature are not unstable. The reason for the unstable Abaqus results is not known. There is a slight deviation between the Abaqus and VUMAT Johnson-Cook solutions in Figure 4.31 b). There is no deviation between the Abaqus and VUMAT Johnson-Cook solutions for OFHC Copper but the remainder of the Armco-Iron results show a slight deviation for the temperature and S23 stress components (Appendix C).

4.6 512 (8X8X8) ELEMENTS

4.6.1 TENSION

The velocity loading is applied in the same way as for the single element test to produce the tensile load (Figure 4.11). Figure 4.32 shows which elements the results are reported from and the internal Abaqus labelling of these elements.

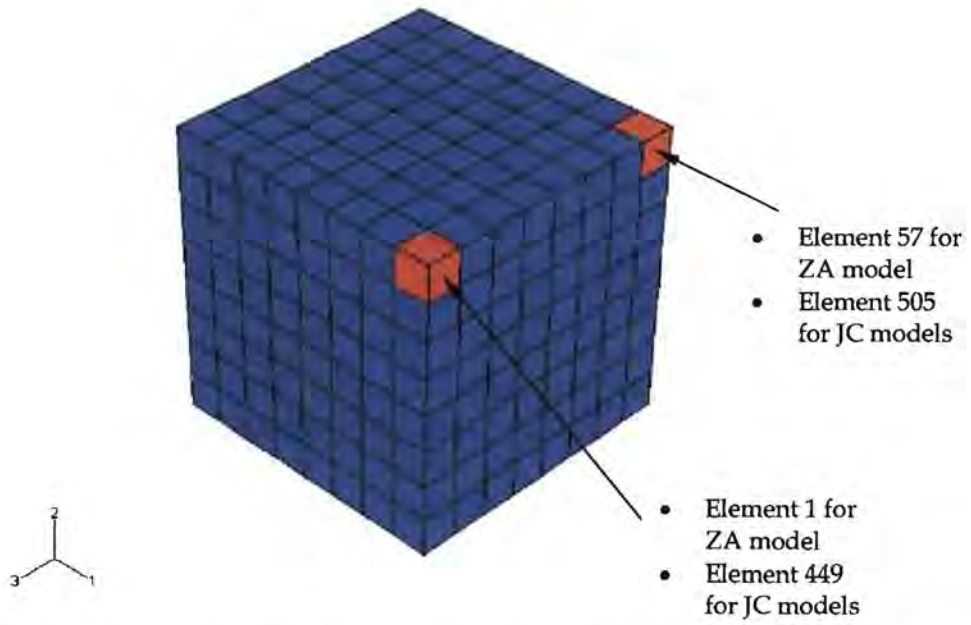
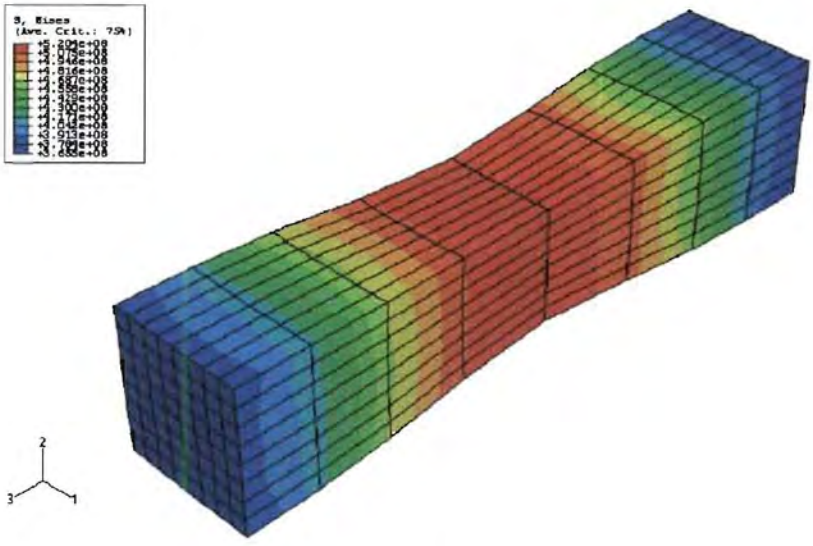
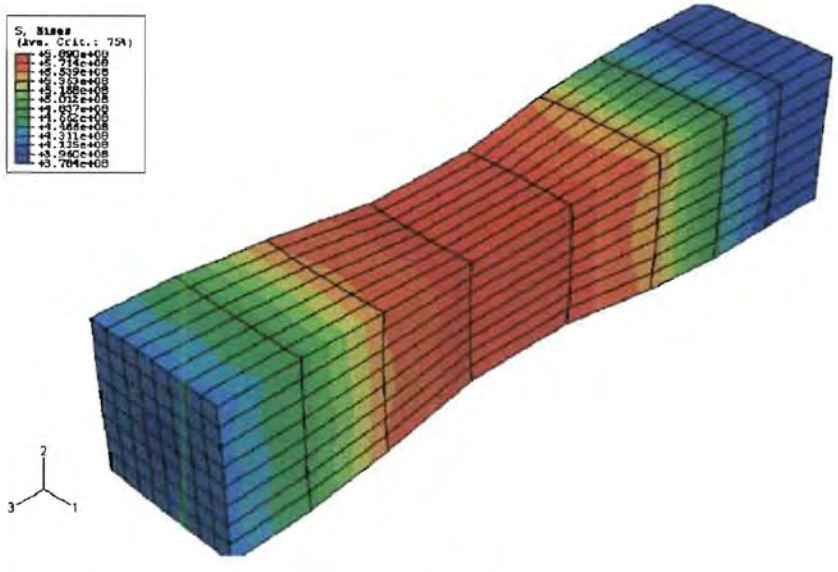


Figure 4.32: Position of elements used in extracting results for 8x8x8 element test.

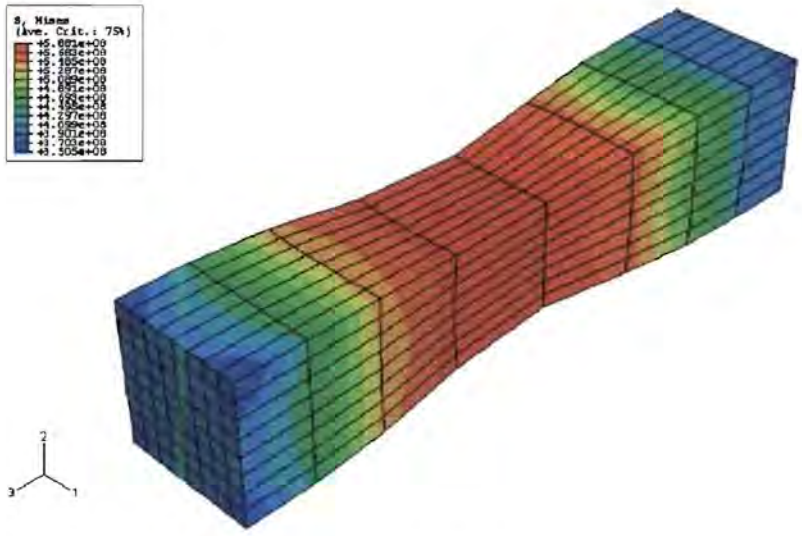
The final deformations for the 10 m/s Armco-Iron tension tests are shown in Figures 4.33 a) - c).



a)



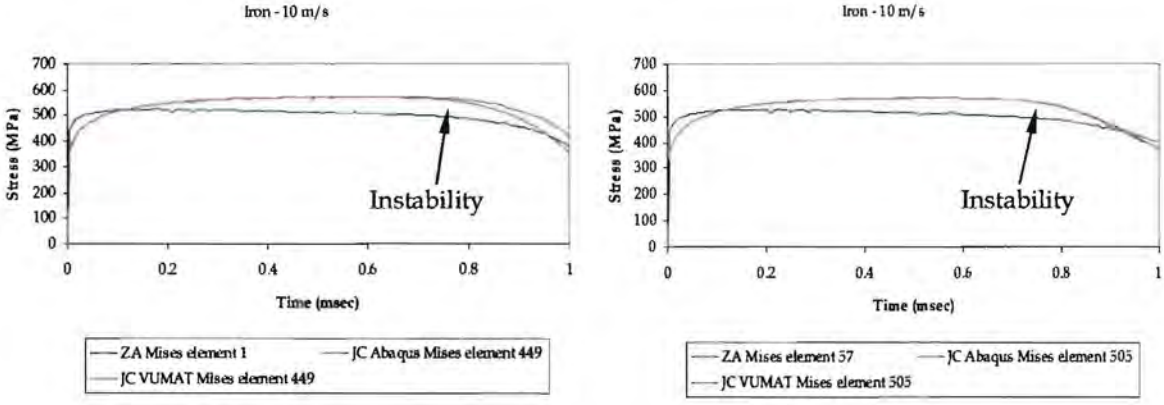
b)



c)

Figure 4.33: 8x8x8 element tensile results for Armco-Iron at 10 m/s
a) Abaqus JC, b) VUMAT JC and c) VUMAT ZA

The collective results for the Mises equivalent stress for a velocity of 10 m/s are shown in Figure 4.34 for Armco-Iron. The remainder of the results can be found in Appendix C.



a) b)

Figure 4.34: Tension – Mises equivalent stress for Armco-Iron at 10 m/s
a) element 1 & 449, b) element 57 & 505.

In Figure 4.34 a) and b) the Mises equivalent stress results for the Abaqus and VUMAT Johnson-Cook models are indistinguishable until the instability occurs (Appendix E). The moment of initiation of the instability, in this case, is the same for the VUMAT Johnson-Cook and the Abaqus Johnson-Cook material models. The results of the Mises stress using the VUMAT Johnson-Cook and Abaqus Johnson-Cook for element 449 deviate after the instability. The deformation results in Figure 4.33 show that the instability occurs at the same place geometrically for the Armco-Iron simulations. The instability was however observed to occur in different geometric positions as shown in Figure 4.35 for OFHC Copper at 10 m/s using the Abaqus version of Johnson-Cook.

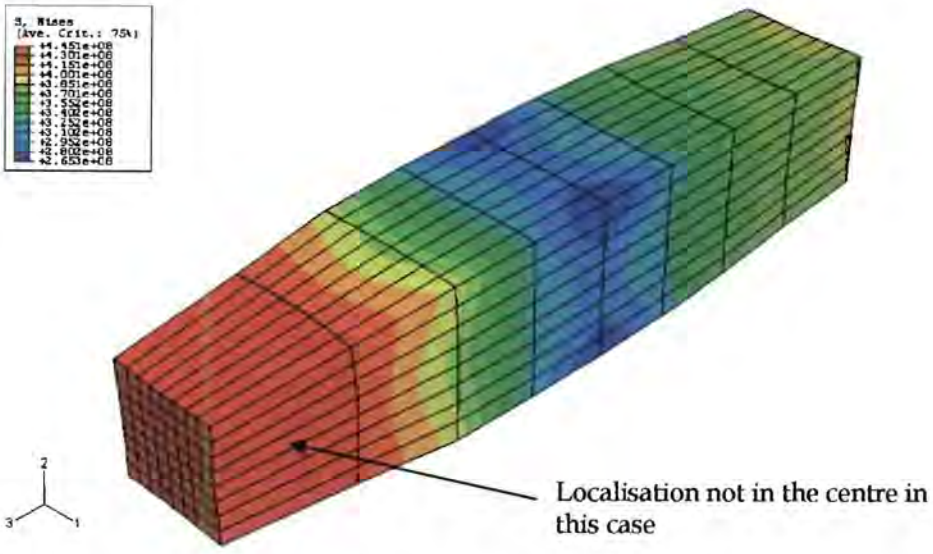


Figure 4.35: Alternative geometric positions for instability.

All the simulation results using the VUMAT Johnson-Cook material model are indistinguishable from the Abaqus Johnson-Cook results until the instability occurs. The instability occurs earlier for the VUMATs than for the Abaqus Johnson-Cook model and the results diverge thereafter (Appendix C).

4.6.2 SHEAR

The velocity loading is applied in the same way as for the single element test to produce the shear load (Figure 4.7). The elements from which the results are extracted are shown in Figure 4.32. An example of the typical final deformation is shown in Figure 4.36.

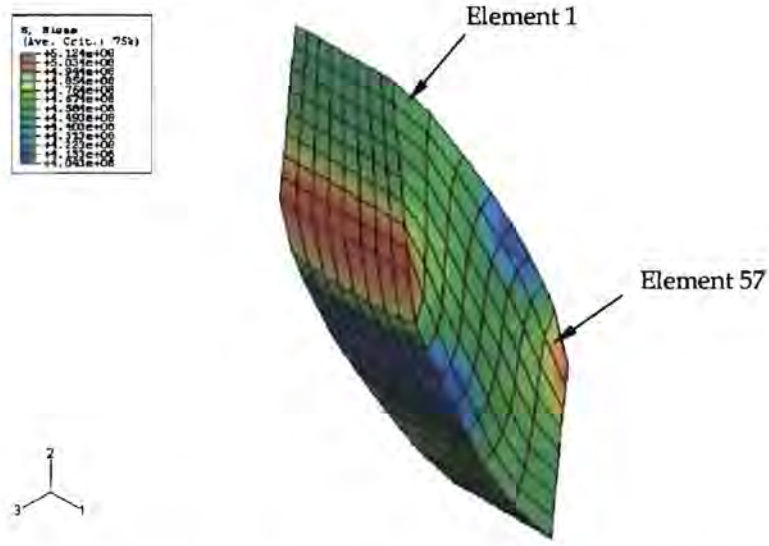
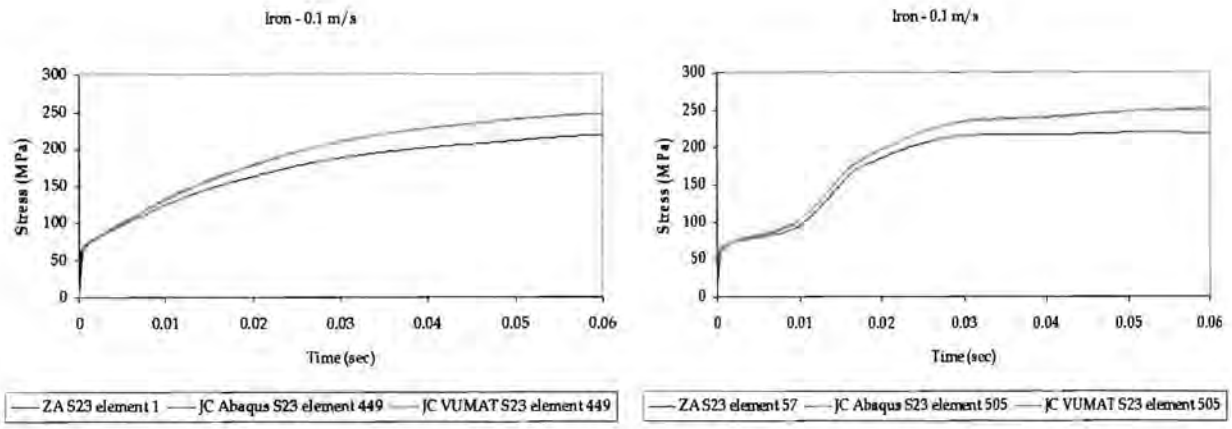


Figure 4.36: Shear (8x8x8 elements) test for OFHC Copper at 0.1 m/s using Zerilli-Armstrong.

The collective results for the Mises equivalent stress for a velocity of 0.1 m/s are shown in Figure 4.37 for Armco-Iron. The remainder of the results can be found in Appendix C.

All the simulation results using the VUMAT Johnson-Cook material model are indistinguishable from the Abaqus Johnson-Cook results for the shear simulations. There is only a slight divergence between the temperature and PEEQ results for the Johnson-Cook models at 10 m/s (Appendix C).



a) b)

Figure 4.37: Shear – S23 stress component for Armco-Iron at 0.1 m/s
a) element 1 & 449, b) element 57 & 505.

4.7 SUMMARY

The element tests for different loading conditions allowed each component of the material model solutions to be examined. For the VUMATs, each of these components was verified by comparing them to the results obtained using the Abaqus Johnson-Cook equation. The VUMATs eventually become unstable numerically, but after the point of physical instability (Appendix E) and therefore any modelling performed with the VUMAT is not affected. In certain instances the VUMAT remains stable longer than the Abaqus material model but the reason for this is unclear.

Now that the VUMATs have been verified thoroughly using the controlled element tests, they can be used with confidence in simulating more complicated events such as the Taylor test (see Chapter 5). The results of the verification process detailed in this chapter show that the implementation of the Johnson-Cook and Zerilli-Armstrong VUMATs has been successful and that the foundation has been laid for more complicated material models to be attempted.

CHAPTER 5

TAYLOR TEST

5.1 INTRODUCTION

The background to the Taylor test was discussed in Chapter 2. As stated the Taylor test is often used for material model validation at high strain rates. In this chapter the Taylor test will be used both for verification and validation (Chapter 4) purposes. The implementation of the Johnson-Cook and Zerilli-Armstrong material models as VUMATs will be verified further by comparing the results of simulations using the Abaqus version of the Johnson-Cook equation with the results obtained using the VUMAT implementation. This verification is used to extend the element test verification in Chapter 4 to a more "realistic" simulation. Experimental results of the Taylor test by Johnson and Cook [5] will be used to validate the simulation results obtained using the VUMAT implementation of the Johnson-Cook and Zerilli-Armstrong material models.

5.2 FEM MODEL

The finite element model of the Taylor test consists of a deformable 3D cylinder with an initial prescribed velocity impacting an analytical rigid surface as shown in Figure 5.1. The cylinder was meshed using C3D8R "brick" elements with an element size of 0.5 mm x 0.5 mm x 0.5 mm. This is an axi-symmetric problem but the VUMAT, in its present form, is not able to be used for an axi-symmetric analysis. A 3-dimensional, ¼ symmetric model was used with all the material models in order to reduce computational expense. "Hard" contact with separation after contact (the default setting) was used as the normal interaction property between the deformable cylinder and the rigid surface. The tangential behaviour between the cylinder and

rigid surface was modelled as frictionless [23]. The cylinder was given initial conditions of temperature (300 K - assumed room temperature) and velocity (test specific).

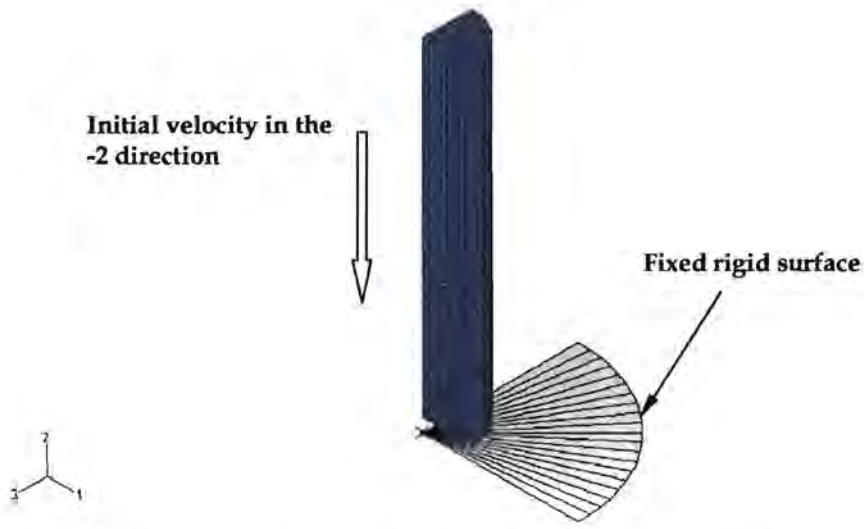


Figure 5.1: ¼ symmetric model of Taylor test.

A stable time increment can be defined as the time required for a dilatational wave to propagate across the smallest element dimension [4]. The time increment used in an analysis must be smaller than the stability limit [4]. If the time increment is not sufficiently small the solution will become unstable. If the solution becomes unstable then solution variables such as displacements will usually oscillate with increasing amplitudes [4].

Initially the simulations were run using the default global stable time increment estimator as opposed to element-by-element estimation. If the global procedure is specified Abaqus begins with the element-by-element estimation and then changes to the global estimation method if the algorithm determines that the accuracy of the global method is acceptable. The global estimation algorithm allows greater time increments than the element-by-element algorithm. This gave unstable results especially for the lower velocities. An example of the severe deformation (with equivalent deviatoric plastic strains in excess of 10) which resulted from this is

shown in Figure 5.2. This particular result was obtained using the Abaqus version of the Johnson-Cook equation and therefore cannot be attributed to an incorrect implementation of the VUMAT. In order to try and combat these excessive distortions the time increment was manually reduced and fixed. This solved the problem and the time increment estimator was returned to the automatic mode but with element-by-element estimation. This also solved the problem and this method was then used for all the Taylor test simulations.

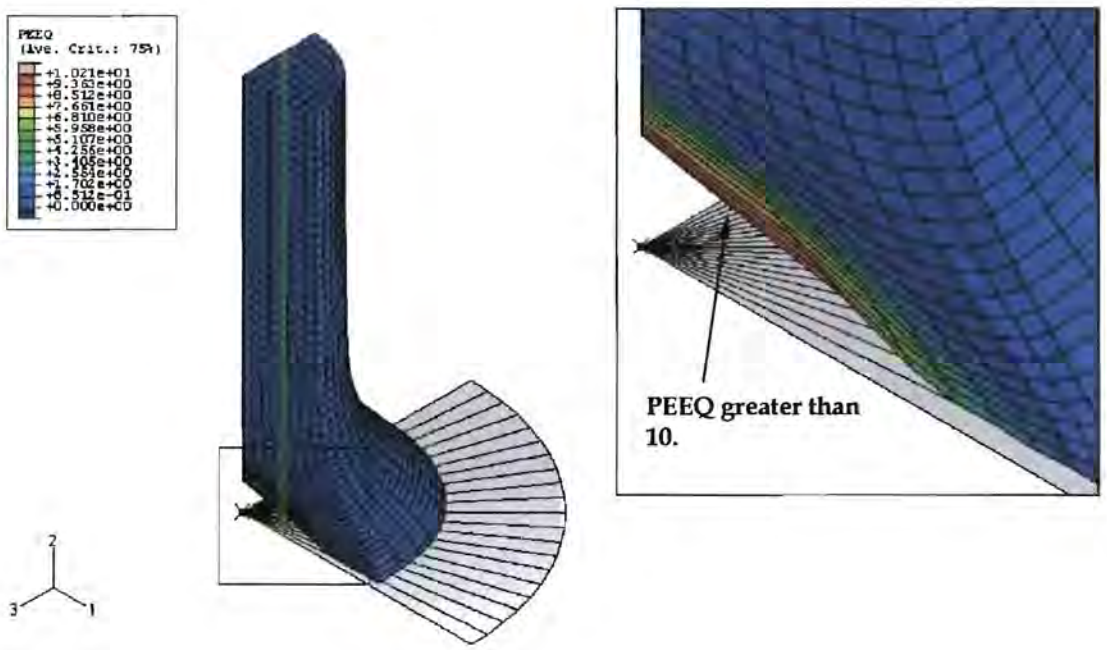


Figure 5.2: Unstable solution when using global time estimation.

The material constants used for the element tests are given in Chapter 3 for the Johnson-Cook and Zerilli-Armstrong material models. For the Taylor test simulations the values for Armco-Iron and OFHC Copper reported by Johnson and Cook [34] are used. All the material constants used in the element tests are valid for the Taylor test simulation except for Young's modulus and Poisson's ratio. These become 207 GPa and 0.29 for Armco-Iron and 124 GPa and 0.34 for OFHC Copper respectively. The material constants used in the simulations of the Taylor test need to correspond to the experimental material constants used by Johnson and Cook [34] to be useful for validation purposes.

To analyse the deformation results, the final length and radial ratios (at the impacted end) of the cylinder are extracted from the simulation results and are compared to experimentally measured results by Johnson and Cook [5]. For an initial length L_0 and a final length L_1, a length ratio can be defined as L_1/L_0 (Figure 5.3). Similarly, for an initial radius R_0 and a final radius R_1, a radial ratio can be defined as R_1/R_0. The initial radius for all tests is $R_0 = 3.81$ mm. The initial length is $L_0 = 25.4$ mm for all tests except the Armco-Iron test of 279 m/s which has $L_0 = 12.6$ mm. Therefore the length ratio will be less than unity and the radial ratio will be greater than unity. For time history graphs of these ratios see Appendix D.

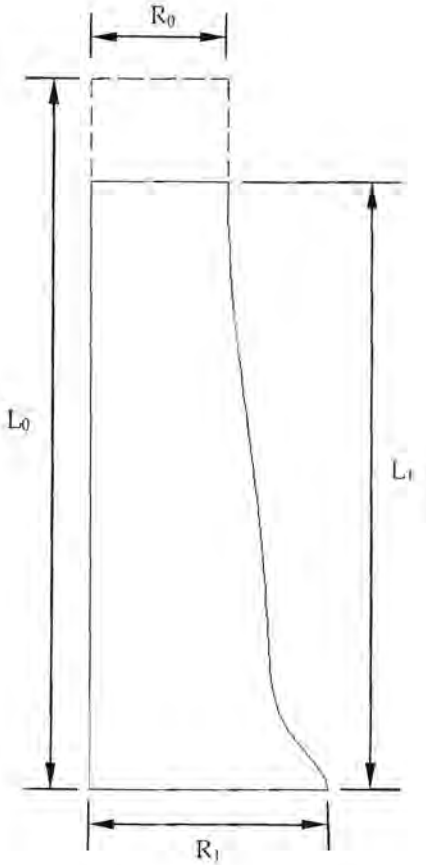


Figure 5.3: Definition of deformation ratio measurements.

The equivalent deviatoric plastic strain (PEEQ) contours from the simulations for all the material models are compared to the strain contours produced by the simulations of Johnson and Cook [5] in their original paper. This comparison will

therefore be a verification of the VUMAT implementation as no experimental PEEQ contours are available.

Finally, the reaction force produced by the rigid body during the impact of the cylinder is plotted. The reaction forces from the VUMAT material model simulations are compared to the reaction forces from the Abaqus Johnson-Cook simulations. This is another form of verification because to be classified as validation the reaction force would have to be measured experimentally (using a direct impact Hopkinson bar for example).

5.3 ARMCO-IRON

5.3.1 DEFORMATION RESULTS

The deformation results for the simulations and experiments for Armco-Iron are summarised in Table 5.1 for comparison.

Velocity m/s	Experimental results [6]		JC Abaqus		JC VUMAT		ZA VUMAT	
	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0
197	0.802	1.59	0.795 (-1)	1.61 (1)	0.795 (-1)	1.61 (1)	0.814 (1)	1.63 (3)
221	0.780	1.80	0.756 (-3)	1.75 (-3)	0.756 (-3)	1.75 (-3)	0.777 (0)	1.77 (-2)
279	0.707	1.97	0.666 (-6)	1.87 (-5)	0.666 (-6)	1.87 (-5)	0.689 (-3)	1.82 (-8)

Table 5.1: Comparison of cylinder impact results for Armco-Iron. The numbers in parenthesis are the percentage deviation from the experimental results.

The results in Table 5.1 show that the VUMAT implementation of the Johnson-Cook material model gives the same results as the Abaqus version of the Johnson-Cook model. It is also clear that there is not a significant difference between the Johnson-Cook results and the Zerilli-Armstrong predictions. This result was found

surprising by Zerilli and Armstrong [6]. The simulation results obtained by Zerilli and Armstrong are shown in Table 5.2. The numbers in parenthesis (for all the remaining tables) are the percentage deviation from the experimental results.

Velocity m/s	Johnson-Cook eqn.		Zerilli-Armstrong eqn.	
	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0
197	0.795 (-1)	1.65 (4)	0.816 (2)	1.66 (5)
221	0.755 (-3)	1.80 (0)	0.779 (0)	1.81 (0)
279	0.664 (-6)	1.92 (-3)	0.691 (-2)	1.86 (-6)

Table 5.2: Simulation results obtained by Zerilli and Armstrong for Armco-Iron [6].

The results obtained using the Abaqus and VUMAT versions of the Johnson-Cook and Zerilli-Armstrong equations are similar to those obtained by Zerilli and Armstrong [6] in their simulations.

5.3.2 Contour Plots

5.3.2.1 Velocity = 197 m/s

The contour plot of equivalent deviatoric plastic strain (PEEQ) from the Johnson and Cook simulation [5] is shown in Figure 5.4 for a velocity of 197 m/s. The dots indicate the experimental outline of the deformed specimen and the strain contours are shown at intervals of 0.5 on the left half of the figure.

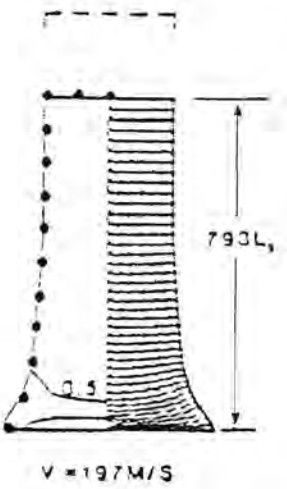


Figure 5.4: Contour plot of PEEQ for Armco-Iron - 197 m/s [5].

The dashed line indicates the undeformed length. The right half of the deformed cylinder in Figure 5.4 shows the triangular mesh used by Johnson and Cook in their simulations. The final length ratio indicated in Figure 5.4 ($0.793L_0$) deviates slightly from the ratio reported by Zerilli and Armstrong in Table 5.2 ($0.795L_0$).

A slight deviation between the Johnson and Cook [5] simulations and the Zerilli and Armstrong simulations [6] is evident for all the Taylor test results. As shown in Figure 5.5, the simulation results using the VUMAT implementation of the material models show excellent agreement with the simulation results using the Abaqus version of Johnson-Cook. In particular, Figure 5.5 d) shows that the contours obtained using the Johnson-Cook VUMAT are almost identical to the contours obtained using the Abaqus Johnson-Cook model. Qualitatively the results are similar to the contour plot shown in Figure 5.4.

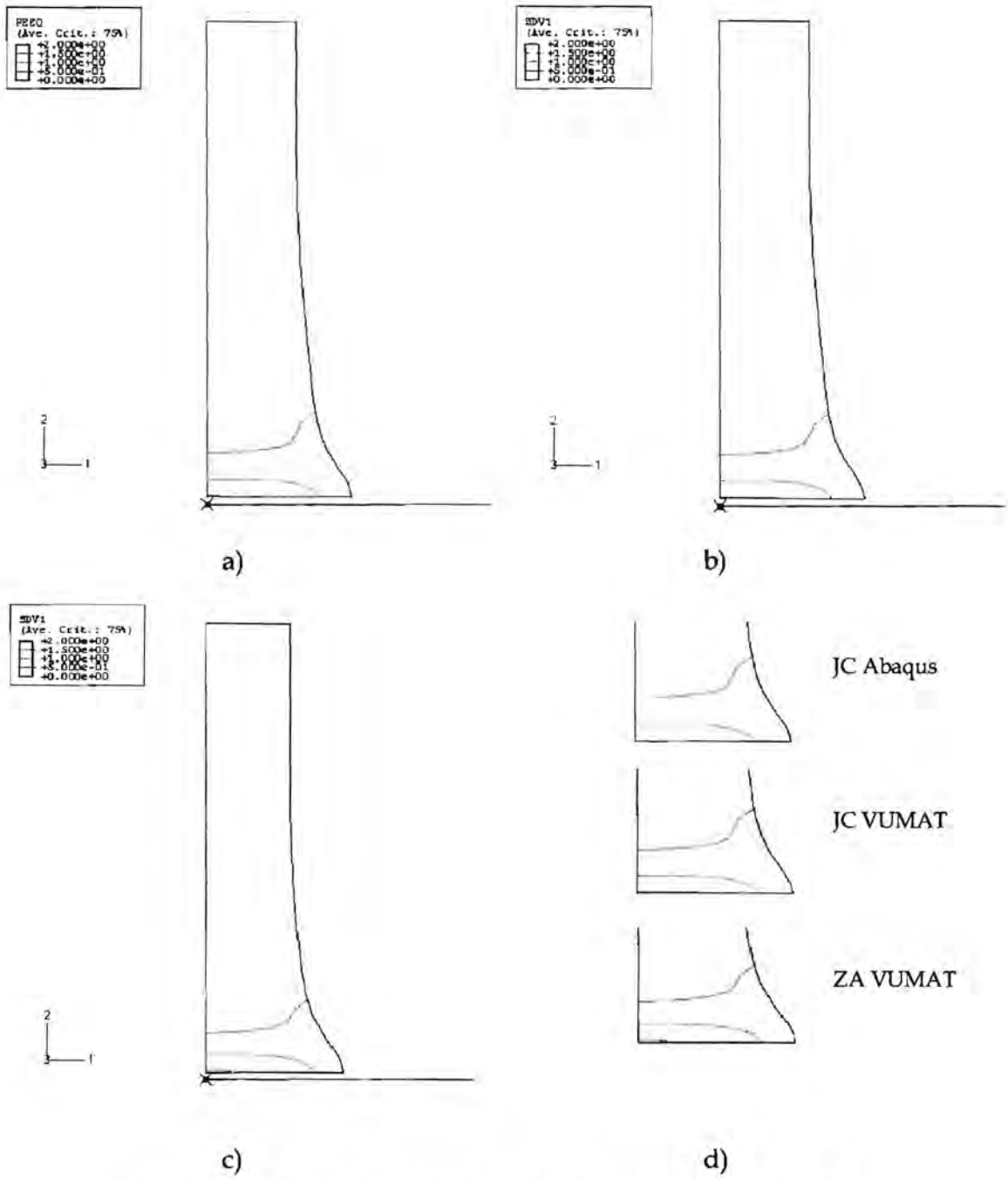


Figure 5.5: Contour plots of PEEQ for Armco-Iron (197 m/s) for a) JC Abaqus, b) JC VUMAT, c) ZA VUMAT and d) close-up of impact ends.

5.3.2.2 VELOCITY = 221 M/S

The contour plot of equivalent deviatoric plastic strain (PEEQ) from the Johnson and Cook simulation is shown in Figure 5.6 for a velocity of 221 m/s.

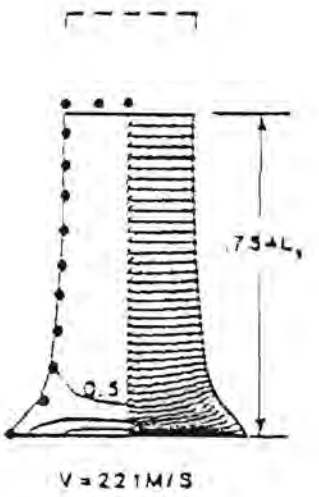


Figure 5.6: Contour plot of PEEQ for Armco-Iron - 221 m/s [5].

The simulation results in Figure 5.6 are not as accurate as the results shown in Figure 5.4 as seen by the increased distance between the simulation results and the experimental results indicated by the dots. As shown in Figure 5.7, the simulation results using the VUMAT implementation of the material models again show excellent agreement with the simulation results using the Abaqus version of Johnson-Cook. In particular, Figure 5.7 d) shows that the contours obtained using all the material models are almost identical.

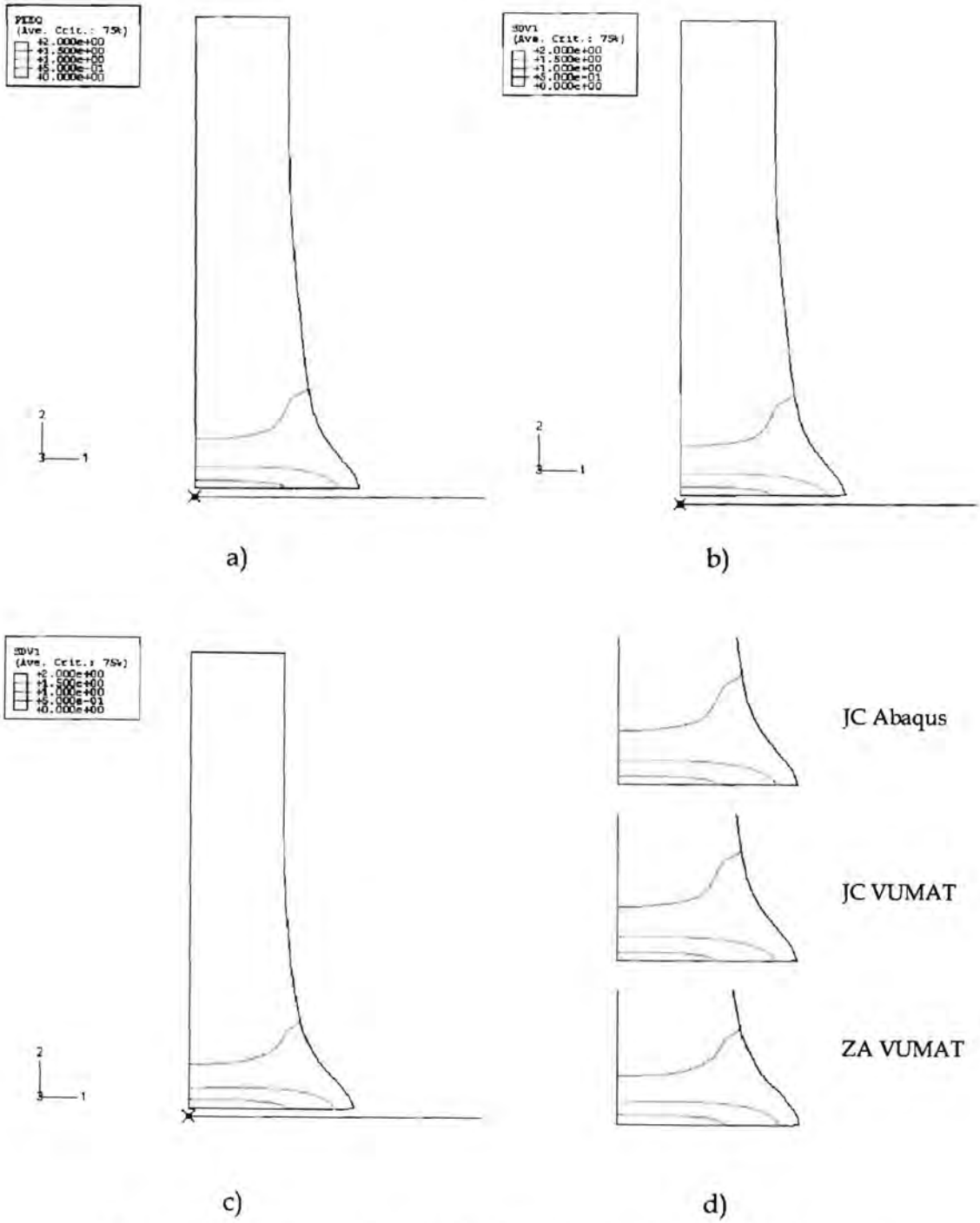


Figure 5.7: Contour plots of PEEQ for Armco-Iron (221 m/s) for a) JC Abaqus, b) JC VUMAT, c) ZA VUMAT and d) close-up of impact ends.

5.3.2.3 VELOCITY = 279 M/S

The contour plot of equivalent deviatoric plastic strain (PEEQ) from the Johnson and Cook simulation is shown in Figure 5.8 for a velocity of 279 m/s. Note that the initial length, L_0, is 12.6 mm for this specimen.

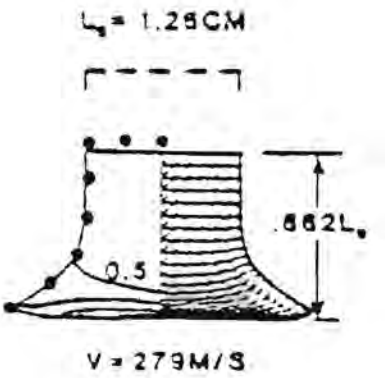
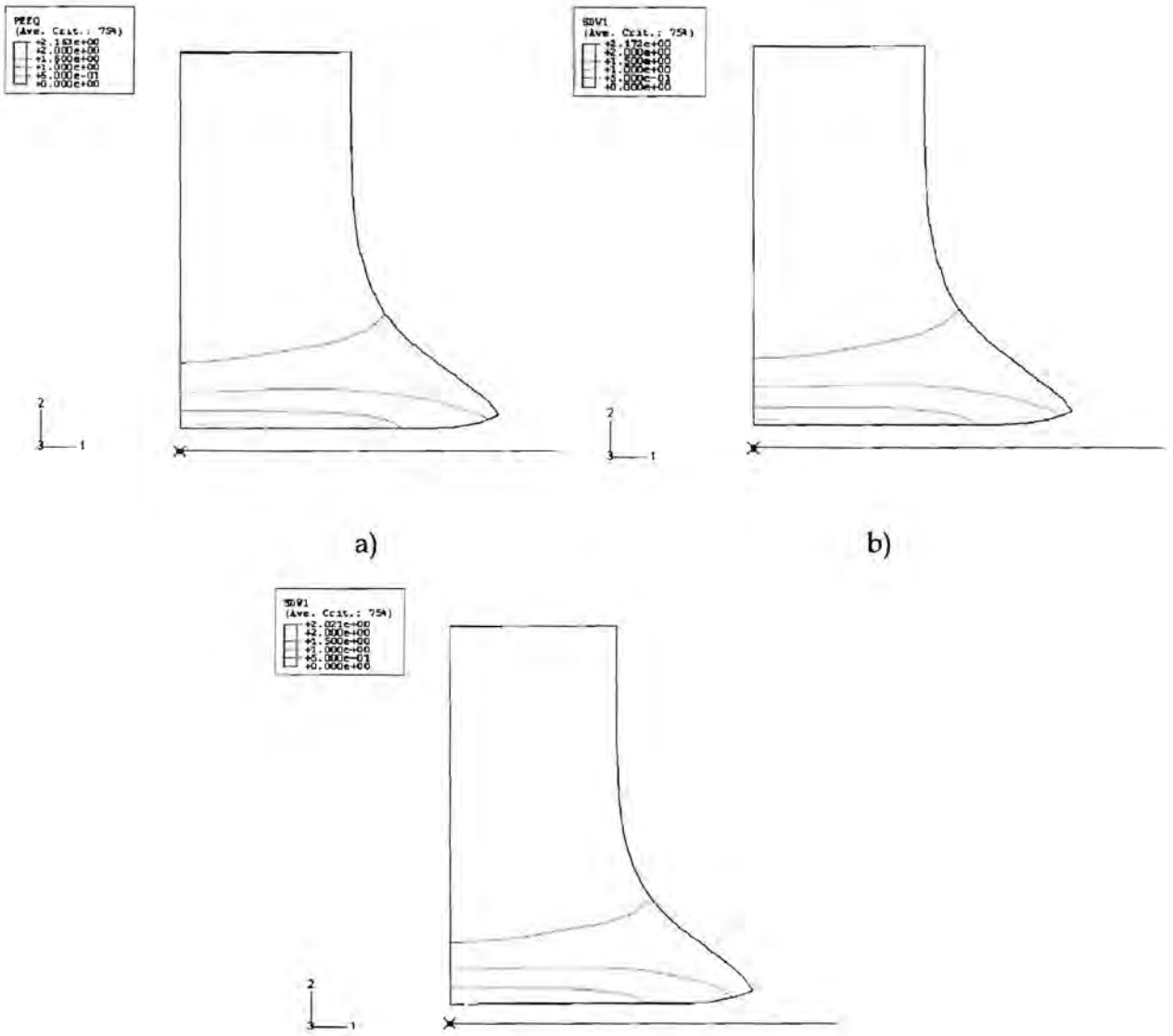


Figure 5.8: Contour plot of PEEQ for Armco-Iron - 279 m/s [5].

As shown in Figure 5.9, the simulation results using the VUMAT implementation of the material models again show excellent agreement with the simulation results using the Abaqus version of Johnson-Cook. In this case a combined close-up view is not shown because the figures are large enough for qualitative comparison as is. The results in Figure 5.8 and Figure 5.9 show a curling up of the edges of the deformed end of the cylinder. This result will not be obtained if "separation after contact" is not allowed in the contact interaction properties.



a) b)

c)

Figure 5.9: Contour plots of PEEQ for Armco-Iron (279 m/s) for a) JC Abaqus, b) JC VUMAT and c) ZA VUMAT.

5.3.3 Reaction Force

The reaction force produced by the rigid surface during the impact of the cylinder is extracted from the simulation results for the Abaqus version of the Johnson-Cook equation as well as the VUMAT implementations of the Johnson-Cook and the

Zerilli-Armstrong equations. These reaction forces are then compared for verification purposes.

5.3.3.1 VELOCITY = 197 M/S

The time histories of the reaction forces for the 197 m/s Armco-Iron simulations are shown in Figure 5.10.

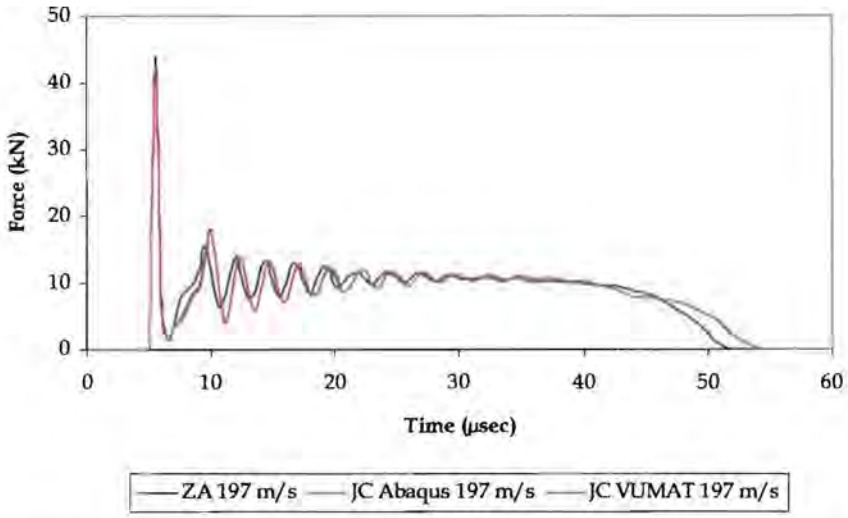


Figure 5.10: Reaction force time histories, Armco-Iron - 197 m/s.

In Figure 5.10 the Abaqus Johnson-Cook and the VUMAT Johnson-Cook results are indistinguishable thus further verifying the VUMAT implementation. Comparing the reaction histories from the simulations to experimental results could be an additional validation of the simulation results especially if the Taylor test is used to extract the constants for the material models. The reaction force returns to zero once the cylinder has lost contact with the rigid surface i.e. it has rebounded. Figure 5.10 shows that the Zerilli-Armstrong results return to zero earlier than the Johnson-Cook results. This type of detail could be captured from Hopkinson bar Taylor tests and could be used for validation purposes.

5.3.3.2 VELOCITY = 221 M/S

The time histories of the reaction forces for the 221 m/s Armco-Iron simulations are shown in Figure 5.11.

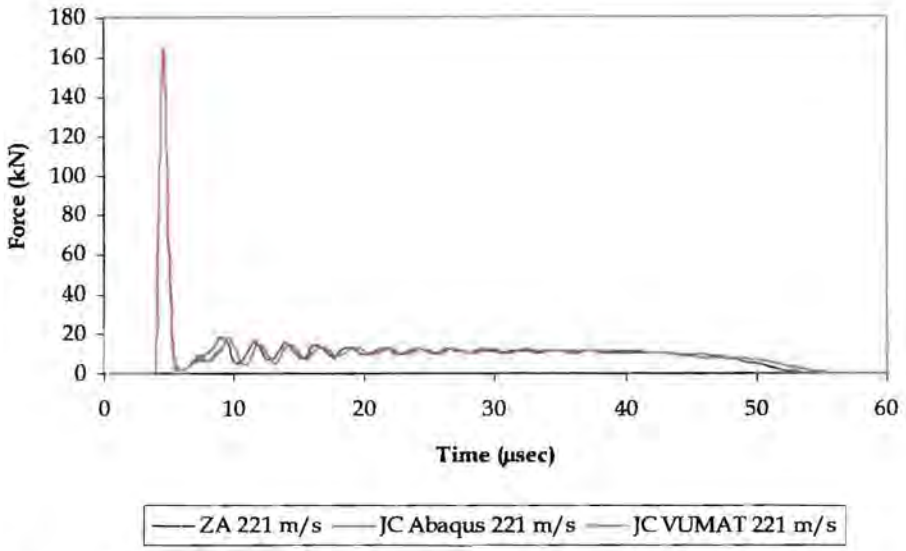


Figure 5.11: Reaction force time histories, Armco-Iron – 221 m/s.

In Figure 5.11 the Abaqus Johnson-Cook and the VUMAT Johnson-Cook results are again indistinguishable. The plateau in Figure 5.11 seems very low because of the high initial peak. It is in fact about the same height as the plateau in Figure 5.10. The Zerilli-Armstrong results in Figure 5.11 return to zero earlier than the Johnson-Cook results.

5.3.3.3 VELOCITY = 279 M/S

The time histories of the reaction forces for the 279 m/s Armco-Iron simulations are shown in Figure 5.12.

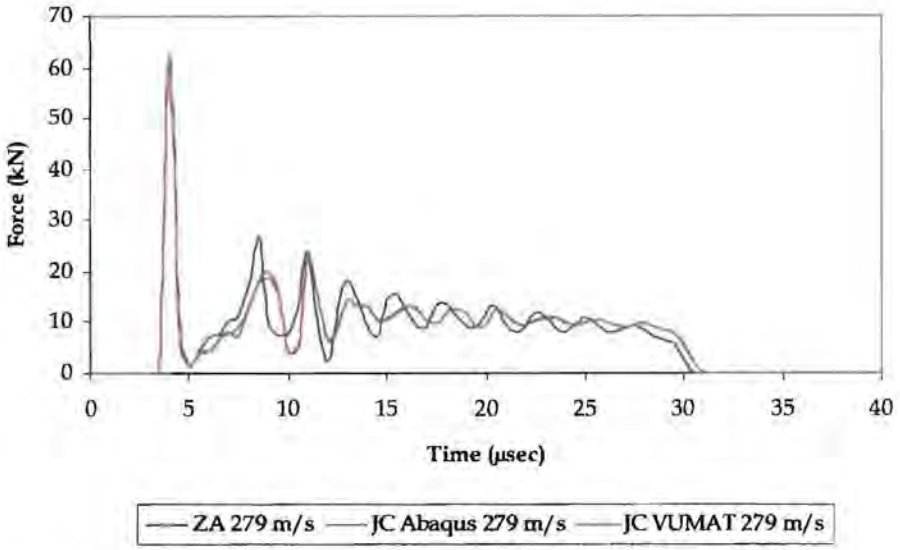


Figure 5.12: Reaction force time histories, Armco-Iron - 279 m/s.

In Figure 5.12 the Abaqus Johnson-Cook and the VUMAT Johnson-Cook results are indistinguishable thus providing further verification of the VUMAT implementation.

The reaction force-time curves were plotted by extracting 200 data points from the simulation results. In order to determine the loss of accuracy as a result of the number of data points used, the impulse obtained from integrating the reaction force-time curve was compared with the impulse obtained by calculating the change in momentum during the Taylor test simulation. This impulse comparison was performed using results from 200 data points, 1000 data points and all available data points from the Abaqus Johnson-Cook simulation for Armco-Iron at 279 m/s. The averaged rebound velocity was 8 m/s. The percentage deviation between the impulse calculations was found to be 8.2%, 1.5% and 0.5% for 200, 1000 and all data points respectively. If the simulation results are going to be validated using experimental data then all the data points should be used if possible. There is also a greater chance of capturing the true initial peak of the reaction force-time curve as the number of data points is increased. Continuing with this example of Abaqus results using Johnson-Cook at 279 m/s, the initial peak in reaction force was found

to be about 60 kN, 102 kN and 124 kN for 200, 1000 and all data points respectively. The graphs of reaction force history shown in this chapter are used for verification purposes and show the Abaqus Johnson-Cook results to be indistinguishable from the Johnson-Cook VUMAT results. The use of 200 data points is therefore adequate in this case.

5.4 OFHC Copper

5.4.1 Deformation Results

The deformation results for the simulations and experiments for OFHC Copper are summarised in Table 5.3 for comparison. The numbers in parenthesis are once again the percentage deviation from the experimental results.

Velocity m/s	Experimental results [6]		JC Abaqus		JC VUMAT		ZA VUMAT	
	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0
130	0.770	1.30	0.812 (5)	1.46 (12)	0.812 (5)	1.46 (12)	0.786 (2)	1.40 (8)
146	0.736	1.40	0.778 (6)	1.56 (11)	0.778 (6)	1.56 (11)	0.752 (2)	1.48 (6)
190	0.638	1.78	0.681 (7)	1.89 (6)	0.681 (7)	1.89 (6)	0.660 (3)	1.75 (-2)

Table 5.3: Comparison of cylinder impact results for OFHC Copper.

The simulation results obtained by Zerilli and Armstrong [6] are shown in Table 5.4. The simulation results by Zerilli and Armstrong are similar to the results reported in Table 5.3 but the simulation results obtained using the Abaqus and VUMAT versions of the Johnson-Cook equation produce slightly improved radial results for the lower velocities.

Velocity m/s	Johnson-Cook eqn.		Zerilli-Armstrong eqn.	
	L_1/L_0	R_1/R_0	L_1/L_0	R_1/R_0
130	0.812 (5)	1.48 (13)	0.784 (2)	1.41 (8)
146	0.778 (6)	1.58 (13)	0.750 (2)	1.49 (6)
190	0.682 (7)	1.92 (8)	0.658 (3)	1.77 (0)

Table 5.4: Simulation results obtained by Zerilli and Armstrong for OFHC Copper [6].

The results in Tables 5.3 and 5.4 show that the Zerilli-Armstrong material model gives slightly better results for OFHC Copper than the Johnson-Cook material model.

5.4.2 Contour Plots

5.4.2.1 Velocity = 130 m/s

The contour plot of equivalent deviatoric plastic strain (PEEQ) obtained from the Johnson and Cook simulation [1] is shown in Figure 5.13 for a velocity of 130 m/s.

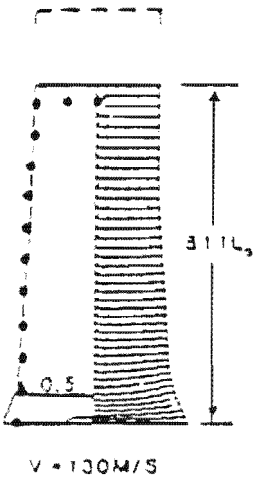


Figure 5.13: Contour plot of PEEQ for OFHC Copper - 130 m/s [5].

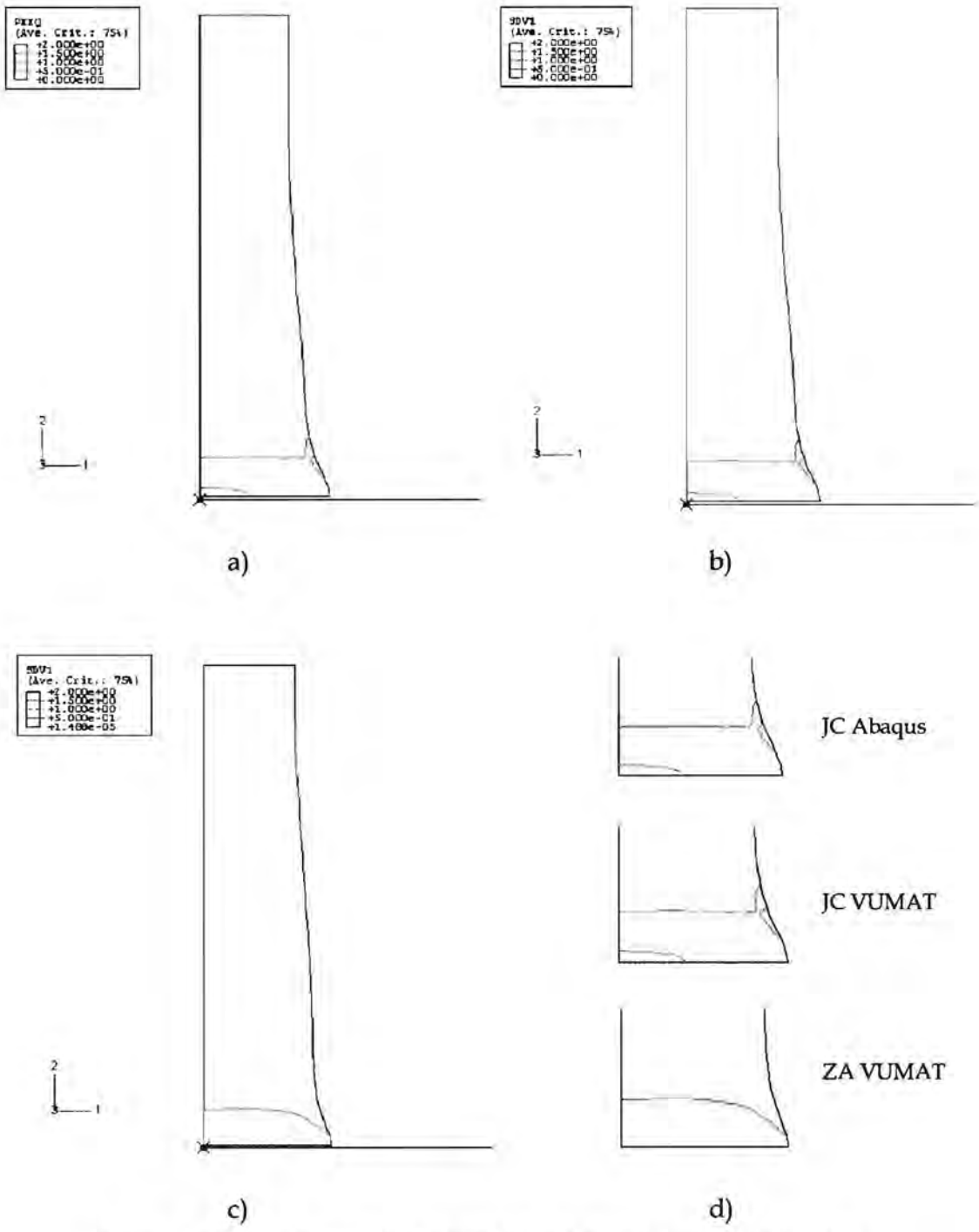


Figure 5.14: Contour plots of PEEQ for OFHC Copper (130 m/s) for a) JC Abaqus, b) JC VUMAT, c) ZA VUMAT and d) close-up of impact ends.

The results in Figure 5.14 show that in this case the Zerilli-Armstrong VUMAT produces noticeably different contours to the Abaqus and VUMAT Johnson-Cook models. The Zerilli-Armstrong VUMAT results are more accurate as shown by

Table 5.3 for the 130 m/s loading condition. Figure 5.14 d) shows that the contours obtained using the Johnson-Cook VUMAT are almost identical to the contours obtained using the Abaqus Johnson-Cook model.

5.4.2.2 VELOCITY = 146 M/S

The contour plot of equivalent deviatoric plastic strain (PEEQ) from the Johnson and Cook simulation [5] is shown in Figure 5.15 for a velocity of 146 m/s.

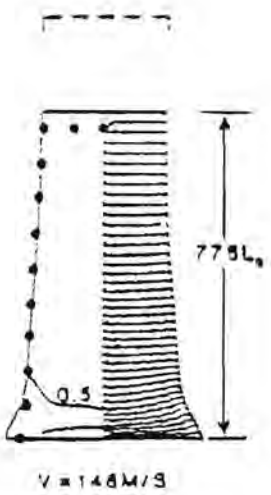


Figure 5.15: Contour plot of PEEQ for OFHC Copper - 146 m/s [5].

The large difference between the results obtained by Johnson and Cook in their simulations (using their equation - solid profile) and their experimental results (dots) is shown in Figure 5.15. The results in Figure 5.16 show that in this case the Zerilli-Armstrong VUMAT produces slightly different contours to the Abaqus and VUMAT Johnson-Cook models. The Zerilli-Armstrong VUMAT results are more accurate as shown by Table 5.3 for the 146 m/s loading condition. Figure 5.16 d) shows that the contours obtained using the Johnson-Cook VUMAT are almost identical to the contours obtained using the Abaqus Johnson-Cook model.

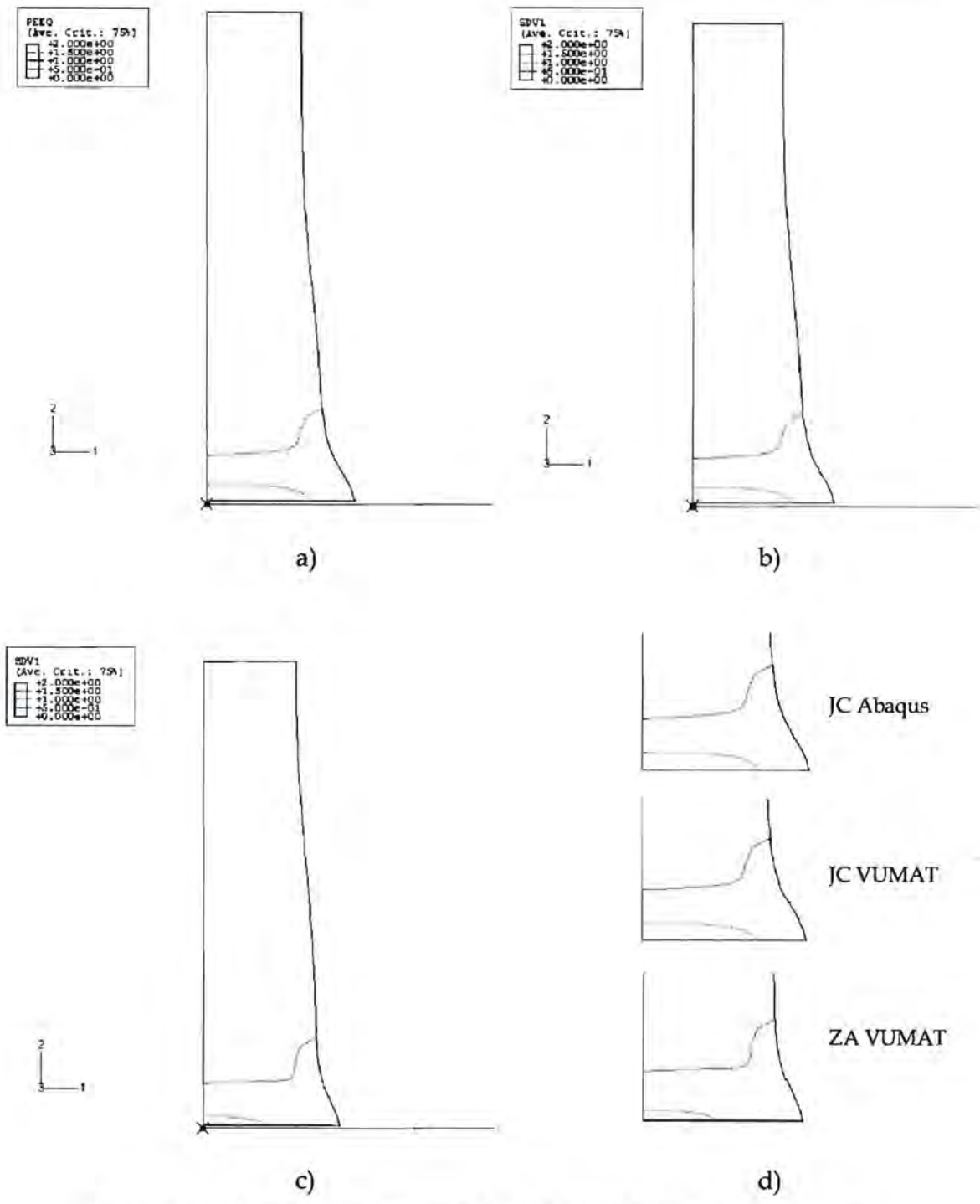


Figure 5.16: Contour plots of PEEQ for OFHC Copper (146 m/s) for a) JC Abaqus, b) JC VUMAT, c) ZA VUMAT and d) close-up of impact ends.

5.4.2.3 VELOCITY = 190 M/S

The contour plot of equivalent deviatoric plastic strain (PEEQ) obtained from the Johnson and Cook simulation [5] is shown in Figure 5.17 for a velocity of 190 m/s.

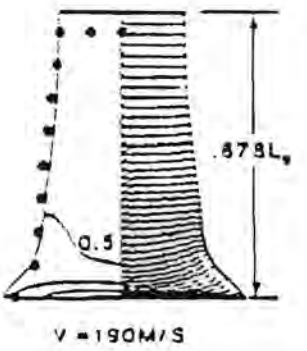


Figure 5.17: Contour plot of PEEQ for OFHC Copper - 190 m/s [5].

The large difference between the results obtained by Johnson and Cook in their simulations (using their equation) and their experimental results is again shown in Figure 5.17. The results in Figure 5.18 show that in this case the Zerilli-Armstrong VUMAT again produces slightly different contours to the Abaqus and VUMAT Johnson-Cook models. The Zerilli-Armstrong VUMAT results are more accurate as shown by Table 5.3 for the 190 m/s loading condition. The maximum equivalent deviatoric plastic strains (PEEQ) are 2.284, 2.292 and less than 2 for the simulations using the Abaqus Johnson-Cook model, the Johnson-Cook VUMAT and the Zerilli-Armstrong model respectively. Figure 5.18 d) shows that the contours obtained using the Johnson-Cook VUMAT are almost identical to the contours obtained using the Abaqus Johnson-Cook model. The Johnson-Cook contours in Figure 5.18 d) correlate very well (qualitatively) with the contours in Figure 5.17.

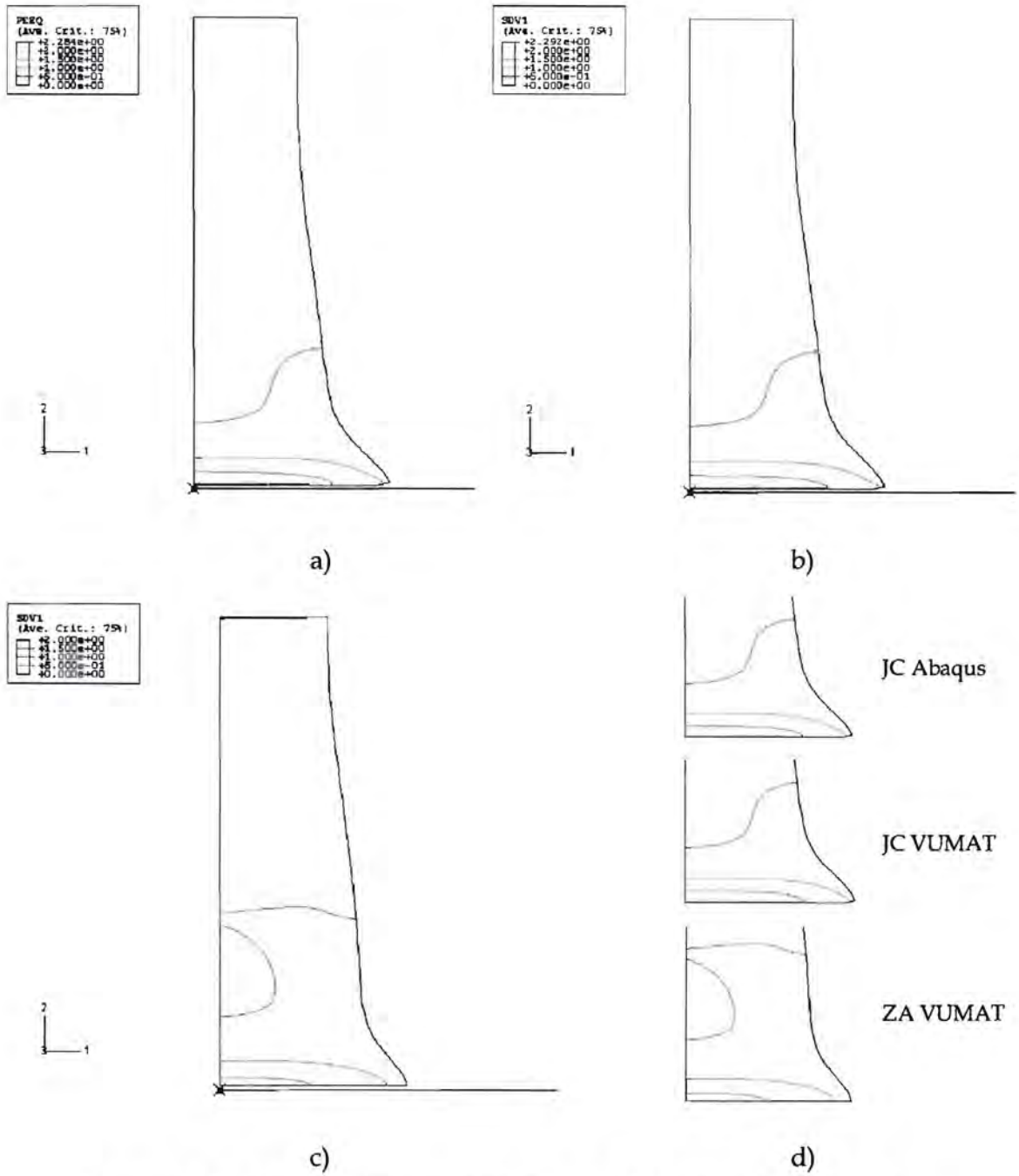


Figure 5.18: Contour plots of PEEQ for OFHC Copper (190 m/s) for a) JC Abaqus, b) JC VUMAT, c) ZA VUMAT and d) close-up of impact ends.

5.4.3 Reaction Force

5.4.3.1 Velocity = 130 m/s

The time histories of the reaction forces for the 130 m/s OFHC Copper simulations are shown in Figure 5.19.

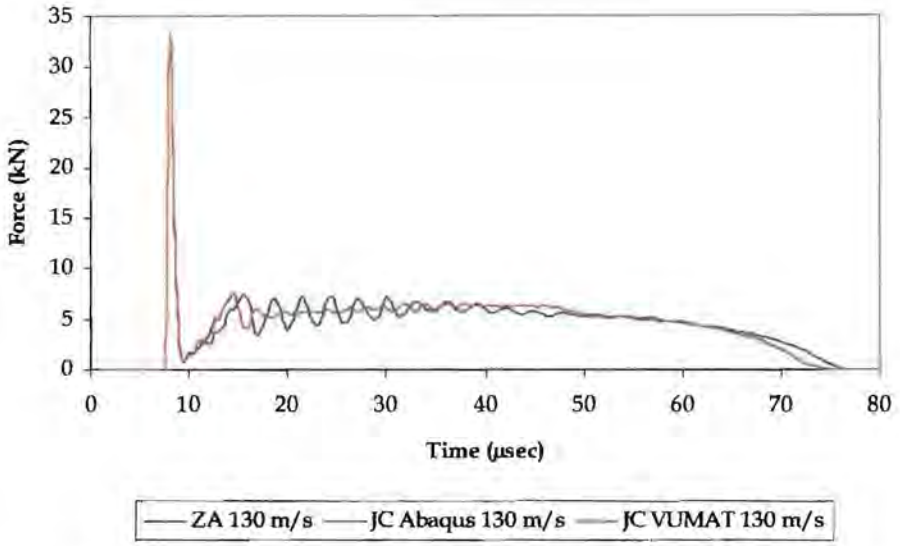


Figure 5.19: Reaction force time histories, OFHC Copper - 130 m/s.

In Figure 5.19 the Abaqus Johnson-Cook and the VUMAT Johnson-Cook results are indistinguishable which further verifies the VUMAT implementation. It is interesting to note that in this case the Zerilli-Armstrong results return to zero after the Johnson-Cook results showing that the contact lasts longer. The opposite was true for the Armco-Iron tests.

5.4.3.2 Velocity = 146 m/s

The time histories of the reaction forces for the 146 m/s OFHC Copper simulations are shown in Figure 5.20.

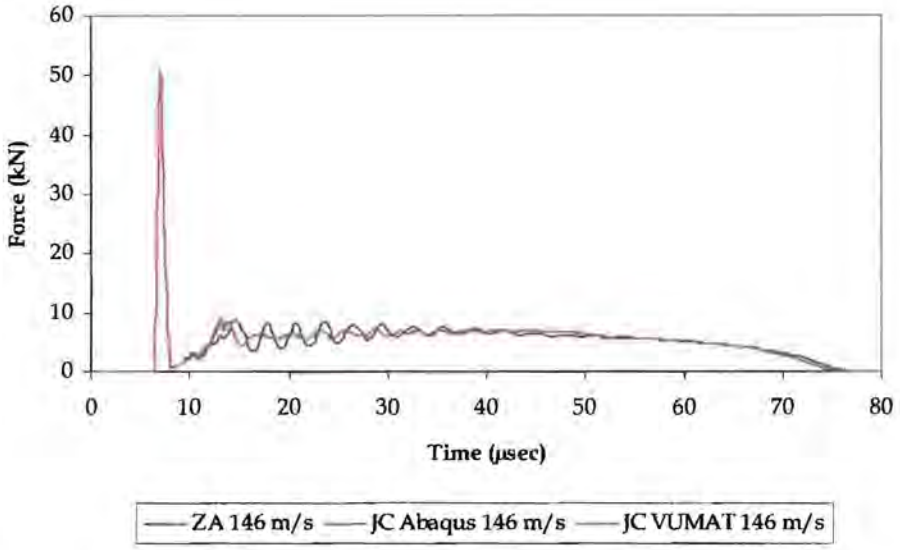


Figure 5.20: Reaction force time histories, OFHC Copper - 146 m/s.

In Figure 5.20 the Abaqus Johnson-Cook and the VUMAT Johnson-Cook results are again indistinguishable thereby further verifying the VUMAT implementation.

5.4.3.3 Velocity = 190 m/s

The time histories of the reaction forces for the 190 m/s OFHC Copper simulations are shown in Figure 5.21.

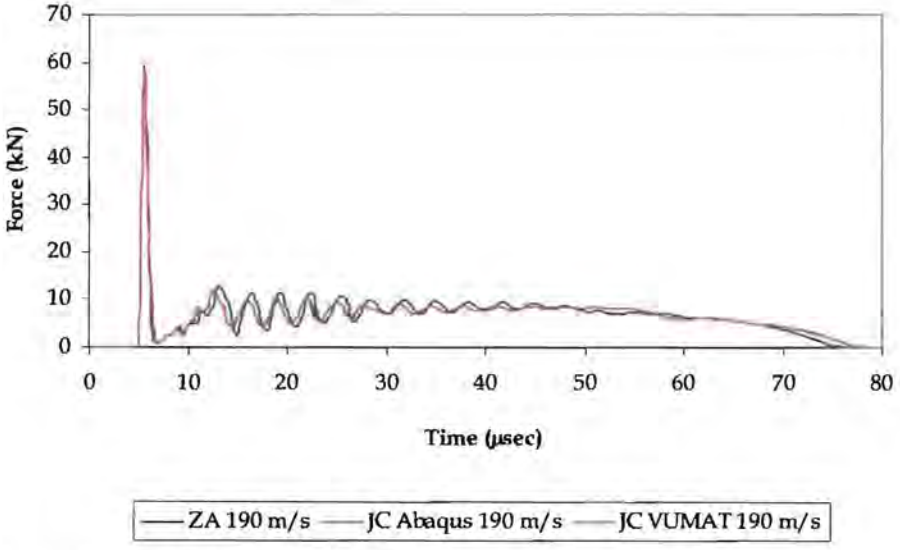


Figure 5.21: Reaction force time histories, OFHC Copper - 190 m/s.

In Figure 5.21 the Abaqus Johnson-Cook model and the Johnson-Cook VUMAT results are indistinguishable thereby verifying the VUMAT implementation.

5.5 SUMMARY

The results reported in this chapter further verify the implementation of the material models as user subroutines in Abaqus. The results obtained using the Johnson-Cook VUMAT are consistent with the results obtained using the Abaqus Johnson-Cook model. The simulation results obtained using all the material models were also compared to the simulation results obtained by Johnson and Cook [5] and by Zerilli and Armstrong [6], thereby further verifying the implementation of the VUMATs. Comparing the simulation results with the experimental results has also confirmed the validation of these material models, as reported by Johnson and Cook [5] and by Zerilli and Armstrong [6].

CHAPTER 6

CONCLUSIONS

The Johnson-Cook [5] and Zerilli-Armstrong [6] material models for BCC and FCC materials were successfully implemented as VUMATs (user-defined material models) for use with Abaqus, a general purpose finite element program. These VUMATs supplement the existing Abaqus material model library.

After presenting the background and theory of these models, the implementation of the models as VUMATs was discussed. The material models were presented in the form in which they are implemented so that the coding of the VUMAT could be easily interpreted.

In Chapter 4 the implementation of the material models as VUMATs was verified by comparing the results of element test simulations obtained using the Abaqus version of Johnson-Cook with the results obtained using the VUMAT versions of Johnson-Cook and Zerilli-Armstrong. The Johnson-Cook VUMAT material model is the same version of the Johnson-Cook equation as the Abaqus Johnson-Cook material model and the simulation results obtained using the two material models were therefore compared for exactness. The Zerilli-Armstrong VUMAT results were compared to the results obtained using the Abaqus Johnson-Cook material model in a qualitative sense.

The element tests were performed using a selection of prescribed loading conditions. The single and multiple element test results showed excellent correlation between the Abaqus Johnson-Cook results and the VUMAT results. Only in a few cases was there a deviation between the results and this deviation only occurred after large deformation at an advanced stage of the simulation. The implementation of the

VUMATs was verified in Chapter 4 which allowed them to be used in a more realistic simulation (the Taylor test) in Chapter 5.

The results of the Taylor test simulations presented in Chapter 5 were used for validation and further verification of the VUMAT implementation. The simulation results obtained using the VUMAT Johnson-Cook equation again showed excellent correlation with the Abaqus Johnson-Cook equation (verification). The simulation results also confirmed that the Zerilli-Armstrong material model gives slightly more accurate results than the Johnson-Cook material model, especially for copper when compared to published experimental results (validation).

The various issues involved in implementing and verifying user material models in ABAQUS has been assessed and the groundwork has been laid for future research into the performance of material models in high strain rate work at the University of Cape Town.

CHAPTER 7

RECOMMENDATIONS

The implementation and verification of the Johnson-Cook and the Zerilli-Armstrong material models as VUMATs has been achieved. This has laid the foundations for further research into the behaviour and significance of these material models in high strain, high strain rate and high temperature simulations. There are a few areas in which the current work on VUMATs can be enhanced:

- the performance of the VUMATs for larger simulations such as blast loading of plates should be investigated
- some form of damage model or failure criterion should be implemented (such as the Johnson-Cook damage model)
- the convergence criterion used in the plasticity formulation should be optimised
- obtain mild steel material constants for these material models for direct comparison with other material models such as Cowper-Symonds and Masui et al.
- the exact cause of the numerical instability should be investigated.

CHAPTER 8

REFERENCES

1. Jones N.: **Structural Impact.**
   Cambridge University Press, 1989, pp 348 - 352.
2. Chung Kim Yuen, S.: **Deformation and tearing of uniformly blast-loaded quadrangular stiffened plates.**
   UCT, Masters Thesis, 2000.
3. Masui T., Nunokawa T. and Hiramatsu T.: **Shape correction of hot rolled steel using an on line leveller.**
   Jrnl. Jap. Soc. Tech. Plas., (1), 1987.
4. Hibbitt, Karlsson and Sorensen: **ABAQUS/Explicit User's Manual.**
   ABAQUS 6.3-1 Documentation, 2002.
5. Johnson G. R. and Cook W. H.: **A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures.**
   Proc. 7th Intl. Symp. Ballistics, Netherlands, 1983, pp 541 - 547.
6. Zerilli F. J. and Armstrong R. W.: **Dislocation - mechanics - based constitutive relations for material dynamics calculations.**
   J. Appl. Phys. 61(5), March 1987, pp 1816 - 1825.
7. Stouffer D. C. and Dame L. T.: **Inelastic Deformation of Metals.**
   John Wiley & Sons Inc., 1996, general reference.
8. Mendelson A.: **Plasticity: Theory and Application.**
   The Macmillan Company, 1968, general reference.
9. Hearn E. J.: **Mechanics of Materials, Volume 1.**
   Second Edition, Pergamon Press, 1985, pp 1 - 4.
10. Lubliner J.: **Plasticity Theory.**
    Macmillan Publishing Company, 1990, pp 69 - 82.
11. Hibbitt, Karlsson and Sorensen: **ABAQUS Theory Manual.**
    ABAQUS 6.3-1 Documentation, 2002.

12. Thomson R.: **Plastic constitutive theory, part 2: plastic flow.**
http://www.mech.gla.ac.uk/~rthomson/teaching/lecnotes/ch30.htm

13. Gronostajski Z.: **The constitutive equations for FEM analysis.**
J. Mat. Proc. Tech., **106**, 2000, pp 40 - 44.

14. Bai Y. and Dodd B.: **Adiabatic shear localization.**
Pergamon Press, 1992, pp 104 - 129.

15. Liang R. and Khan A.S.: **A critical review of experimental results and constitutive models for BCC and FCC metals over a wide range of strain rates and temperatures.**
Intl. J. Plasticity, **15**, 1999, pp 963 - 980.

16. Follansbee P.S. and Kocks U.F.: **A constitutive description of the deformation copper based on the use of the mechanical threshold stress as an internal state variable.**
Acta metall., Vol. 36, No. 1, 1988, pp 81 - 93.

17. Holmquist T.J. and Johnson G.R.: **Determination of constants and comparison of results for various constitutive models.**
J. de Physique, III, Vol. 1, October 1991, pp 853 - 860.

18. Kapoor R. and Nemat-Nasser S.: **Determination of temperature rise during high strain rate deformation.**
Mech. of Mat., **27**, 1998, pp 1 - 12.

19. Taylor G.I.: **The use of flat-ended projectiles for determining dynamic yield stress: I. Theoretical considerations.**
Proc. Roy. Soc. Lon., Vol. 194, 1948, pp 289 - 299.

20. Taylor G.I.: **The use of flat-ended projectiles for determining dynamic yield stress: II. Tests on various metals.**
Proc. Roy. Soc. Lon., Vol. 194, 1948, pp 300 - 322.

21. Wilkins M.L. and Guinan M.W.: **Impact of cylinders on a rigid boundary.**
J. Appl. Phys., Vol. 44, No. 3, 1973, pp 1200 - 1206.

22. Maudlin P.J., Foster J.C. and Jones S.E.: **A continuum mechanics code analysis of steady plastic wave propagation in the Taylor test.**
Intl. J. Imp. Engng., Vol. 19, No. 3, 1997, pp 231 - 256.

23. Allen D.J., Rule W.K. and Jones S.E.: **Optimizing material strength constants numerically extracted from Taylor impact data.**
Exp. Mech., Vol. 37, No. 3, 1997, pp 333 – 338.

24. Johnson G.R. and Holmquist T.J.: **Evaluation of cylinder-impact test data for constitutive model constants.**
J. Appl. Phys., **64**(8), 1988, pp 3901 – 3910.

25. Rule W.K.: **A numerical scheme for extracting strength model coefficients from Taylor test data.**
Intl. J. Imp. Engng., Vol. 19, No.'s 9 – 10, 1997, pp 797 – 810.

26. Cook W.H., Rajendran A.M. and Grove D.J.: **An efficient numerical implementation of the Bodner- Partom model in the EPIC-2 code.**
Eng. Frac. Mech., Vol. 41, No. 5, 1992, pp 607 – 623.

27. Gerald C.F. and Wheatley P.O.: **Applied Numerical Analysis.**
Addison-Wesley Publishing, Fifth edition, 1994, pp 7 – 43.

28. Kojic M. and Bathe K.J.: **The 'effective-stress-function' algorithm for thermo-elasto-plasticity and creep.**
Intl. J. Num. Meth. Engng., Vol. 24, 1987, pp 1509 – 1532.

29. Schreyer H.L., Kulak R.F. and Kramer J.M.: **Accurate numerical solutions for elastic-plastic models.**
Trans. ASME J. Press. Vess. Tech., Vol. 101, 1979, pp 226 – 234.

30. Dodds R.H.: **Numerical techniques for plasticity computations in finite element analysis.**
Comp. & Struc., Vol. 26, No. 5, 1987, pp 767 – 779.

31. Fotiu P.A. and Nemat-Nasser S.: **A universal integration algorithm for rate-dependent elastoplasticity.**
Comp. & Struc., Vol. 59, No. 6, 1996, pp 1173 – 1184.

32. Belytschko T., Kam Liu W. and Moran B.: **Nonlinear Finite Elements for Continua and Structures.**
John Wiley & Sons, 2001, pp 215 – 392.

33. Oden J.T., Belytschko T., Babuska I. and Hughes T.J.R.: **Research directions in computational mechanics.**
Com. Meth. App. Mech. Engrg., **192**, 2003, pp 913 - 922.

34. Johnson G.R. and Cook W.H.: **Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures.**
Engrg. Frac. Mech., Vol. 21, No. 1, 1985, pp 31 - 48.

Bibliography

1. Smith I.M. and Griffiths D.V.: **Programming the finite element method.**
John Wiley & Sons, 3rd edition, 1998, general reference.

2. Ellis T.M.R., Philips I.R. and Lahey T.M.: **FORTRAN 90 programming.**
Addison-Wesley, 1993, general reference.

APPENDIX A

UNIAXIAL FORM OF THE EQUIVALENT MISES STRESS

From the deviatoric form of elasticity:

$$\underline{S} = 2G\underline{e}^{el} \tag{A.1}$$

The integrated (using backward Euler method [4]) form of the flow rule is:

$$\Delta\underline{e}^{pl} = \Delta\bar{e}^{pl}\,\underline{n} \tag{A.2}$$

where $\underline{n}$ is the unit normal to the yield surface. Using the integrated strain rate decomposition and equations A.1 and A.2:

$$\underline{\varepsilon} = \underline{\varepsilon}^{el} + \underline{\varepsilon}^{pl}$$
$$\therefore \underline{S} = 2G\left(\underline{e}^{el}\Big|_t + \Delta\underline{e}^{el}\right)$$
$$\underline{S} = 2G\left(\underline{e}^{el}\Big|_t + \left(\Delta\underline{e} - \Delta\underline{e}^{pl}\right)\right)$$
$$\underline{S} = 2G\left(\underline{e}^{el}\Big|_t + \left(\Delta\underline{e} - \Delta\bar{e}^{pl}\,\underline{n}\right)\right)$$
$$\therefore \underline{S} + 2G\Delta\bar{e}^{pl}\,\underline{n} = 2G\left(\underline{e}^{el}\Big|_t + \Delta\underline{e}\right)$$

Now using the Mises definition of flow direction (from equation (2.18)):

$$\underline{S} + 2G\Delta\bar{e}^{pl}\left(\frac{3}{2}\frac{\underline{S}}{q}\right) = 2G\left(\underline{e}^{el}\Big|_t + \Delta\underline{e}\right)$$
$$\underline{S}\left(1 + \frac{3G}{q}\Delta\bar{e}^{pl}\right) = 2G\left(\underline{e}^{el}\Big|_t + \Delta\underline{e}\right)$$

Now if we write:

$$\hat{\underline{e}} = \underline{e}^{el}\Big|_t + \Delta\underline{e}$$

then:

$$\underline{S}\left(1 + \frac{3G}{q}\Delta\bar{e}^{pl}\right) = 2G\hat{\underline{e}} \tag{A.3}$$

If we take the inner product of the equation A.3 with itself we get:

$$\underline{S}:\underline{S}\left(1+\frac{3G}{q}\Delta\bar{e}^{pl}\right)^2=(2G)^2\underline{\hat{e}}:\underline{\hat{e}} \tag{A.4}$$

Taking the square root of both sides of equation A.4, using the definition of the Mises equivalent stress (equation (2.12)):

$$q=\sqrt{\frac{3}{2}\underline{S}:\underline{S}}$$

and defining the total equivalent deviatoric strain as (equation (3.15)):

$$\tilde{e}=\sqrt{\frac{2}{3}\underline{\hat{e}}:\underline{\hat{e}}}\,,$$

equation A.4 can be written as:

$$\left(1+\frac{3G}{q}\Delta\bar{e}^{pl}\right)\sqrt{\frac{2}{3}}q=2G\sqrt{\frac{3}{2}}\tilde{e}$$

Simplifying this equation gives the Mises equivalent uniaxial form:

$$q+3G\Delta\bar{e}^{pl}=3G\tilde{e}$$
$$3G(\tilde{e}-\Delta\bar{e}^{pl})-q=0$$
$$\therefore 3G(\tilde{e}-\Delta\bar{e}^{pl})-\bar{\sigma}=0$$

APPENDIX B

VUMAT AND INPUT DECK CODES

The VUMATs are written in FORTRAN fixed form using Compaq Visual Fortran Standard Edition 6.6.0. The VUMATs need to be saved with a ".for" file extension when used with Abaqus/Explicit version 6.3-1. Sample input decks for the element tests and the Taylor tests are included. Unnecessary geometric and element definition detail is removed to save space but complete versions of selected input decks can be found on the compact disc version of this thesis.

B.1 JOHNSON-COOK VUMAT

```
      SUBROUTINE vumat(
C Read only -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
      INCLUDE 'vaba_param.inc'
C
C
C     Linear elastic, JC hardening (including temperature). Rate dependent.
C     The state variable is stored as:
C      STATE(*,1) = total equivalent plastic strain
C     Temperature and strain rate in DO loop.
C
C All arrays dimensioned by (*) are not used in this algorithm
      DIMENSION props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(nblock),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(nblock), enerInelasNew(nblock)
C
      CHARACTER*80 cmname
C
      PARAMETER( zero = 0.0d0, one = 1.0d0, two = 2.0d0, three = 3.0d0,
     1  third = one/three, half = .5d0, twoThirds = two/three,
     2  threeHalfs = 1.5d0 )
C Material property definitions
        e          = zero
        e          = props(1)
        xnu        = zero
        xnu        = props(2)
        A          = zero
        A          = props(3)
        B          = zero
        B          = props(4)
        xn         = zero
        xn         = props(5)
        xm         = zero
        xm         = props(6)
        T_trans    = zero
        T_trans    = props(7)
        T_melt     = zero
        T_melt     = props(8)
        xneta      = zero
        xneta      = props(9)
        s_heat     = zero
```

```
      s_heat    = props(10)
      C         = zero
      C         = props(11)
C
C Find Bulk and Shear modulii
      G = e / ( two * (one + xnu) )
      bulk = e / ( three * (one - two*xnu) )
C
C     OPEN(unit=110,FILE='e:\Thesis\test.dat')
C
C
      DO i = 1,nblock
C Find strain trace increment
      trace = zero
      trace = strainInc(i,1) + strainInc(i,2) +
     1              strainInc(i,3)
C Find deviatoric strain increments
      de1 = strainInc(i,1) - third*trace
      de2 = strainInc(i,2) - third*trace
      de3 = strainInc(i,3) - third*trace
      de4 = strainInc(i,4)
      de5 = strainInc(i,5)
      de6 = strainInc(i,6)
C Find trial elastic deviatoric stress increments
      s1_inc = two*G*de1
      s2_inc = two*G*de2
      s3_inc = two*G*de3
      s4_inc = two*G*de4
      s5_inc = two*G*de5
      s6_inc = two*G*de6
C Find equivalent pressure stress increment
      p_inc = zero
      p_inc = -bulk * trace
C Find trial stresses
      sig1 = stressOld(i,1) + s1_inc - p_inc
      sig2 = stressOld(i,2) + s2_inc - p_inc
      sig3 = stressOld(i,3) + s3_inc - p_inc
      sig4 = stressOld(i,4) + s4_inc
      sig5 = stressOld(i,5) + s5_inc
      sig6 = stressOld(i,6) + s6_inc
C For Abaqus check
      IF (stepTime + totalTime .eq. zero) THEN
          stressNew(i,1) = sig1
          stressNew(i,2) = sig2
          stressNew(i,3) = sig3
          stressNew(i,4) = sig4
          stressNew(i,5) = sig5
          stressNew(i,6) = sig6
      ELSE
C Find hydrostatic stress
          p = zero
          p = -third * (sig1 + sig2 + sig3)
C Find deviatoric stresses
          s1 = sig1 + p
          s2 = sig2 + p
          s3 = sig3 + p
          s4 = sig4
          s5 = sig5
          s6 = sig6
C Find Mises equivalent stress
          q_trial = zero
```

```
            q_trial = s1**2 + s2**2 + s3**2 + two*s4**2
     1            + two*s5**2 + two*s6**2
            q_trial = SQRT( threeHalfs*q_trial )
C Set up old equivalent plastic strain
            IF ((stepTime + totalTime) .eq. zero) THEN
                  e_pl_old = zero
            ELSE
                  e_pl_old = stateOld(i,1)
            ENDIF
C Set up temperature effects
            T = zero
            T = tempOld(i)
            IF (T .lt. T_trans) THEN
                  T_star = zero
            ELSEIF ( T_trans .ge. T .le. T_melt ) THEN
                  T_star = (T - T_trans)/(T_melt - T_trans)
            ELSE
                  T_star = one
            END IF
C Find the new total equivalent deviatoric strain increment
C and strain rate
            deeq = twoThirds*( de1**2 + de2**2 + de3**2 +
     1            two*de4**2 + two*de5**2 + two*de6**2 )
            deeq = SQRT(deeq)
            dedot = deeq/dt
C Log trap for dedot
            IF (dedot .gt. zero) THEN
                  dedot_LOG = LOG( dedot )
            ELSE
                  dedot_LOG = zero
            ENDIF
            yield = (A + B*(e_pl_old**xn))*(one - T_star**xm)
     1                  *(one + C*dedot_LOG)
C Check for yielding
            IF (q_trial .le. yield) THEN
                  stressNew(i,1) = sig1
                  stressNew(i,2) = sig2
                  stressNew(i,3) = sig3
                  stressNew(i,4) = sig4
                  stressNew(i,5) = sig5
                  stressNew(i,6) = sig6
                  depl = zero
                  dtheta = zero
                  count = 0
                  f_depl = 0.0d0
            ELSE
C Find old deviatoric stresses
                  sig_mean = zero
                  sig_mean = -third * (stressOld(i,1) +
     1                        stressOld(i,2) + stressOld(i,3))
                  s_old1 = stressOld(i,1) + sig_mean
                  s_old2 = stressOld(i,2) + sig_mean
                  s_old3 = stressOld(i,3) + sig_mean
                  s_old4 = stressOld(i,4)
                  s_old5 = stressOld(i,5)
                  s_old6 = stressOld(i,6)
C Find the new deviatoric strains
                  ehat1 = s_old1/(two*G) + de1
                  ehat2 = s_old2/(two*G) + de2
                  ehat3 = s_old3/(two*G) + de3
                  ehat4 = s_old4/(two*G) + de4
```

```
                ehat5 = s_old5/(two*G) + de5
                ehat6 = s_old6/(two*G) + de6
C Find the new total equivalent deviatoric strain
                etilde = twoThirds*( ehat1**2 + ehat2**2 + ehat3**2 +
     1                two*ehat4**2 + two*ehat5**2 + two*ehat6**2 )
                etilde = SQRT(etilde)
C Use Bisection Method to find depl (equivalent deviatoric plastic
C strain increment)
                Tol = 1.0d5
                depl_a = zero
                depl_b = 1.0d-1
                depl = (depl_a + depl_b)/two
                edot = depl/dt
                IF (edot .gt. zero) THEN
                      edot_LOG = LOG( edot )
                ELSE
                      edot_LOG = zero
                ENDIF
                q_depl = (A + B*((depl + e_pl_old)**xn))*(one-T_star**xm)
     1                     *(one + C*edot_LOG)
                f_depl = three*G*(etilde - depl) - q_depl
                count = 0
           DO WHILE ((ABS(f_depl).gt.Tol).AND.(count.le.100))
                count = count + 1
                depl = (depl_a + depl_b)/two
C If the Bisection method is not going to converge then set depl to it's
C maximum possible value i.e. the increment in total equivalent strain.
                IF (count.ge.100) THEN
C Find old deviatoric strains
                      ehat1_old = s_old1/(two*G)
                      ehat2_old = s_old2/(two*G)
                      ehat3_old = s_old3/(two*G)
                      ehat4_old = s_old4/(two*G)
                      ehat5_old = s_old5/(two*G)
                      ehat6_old = s_old6/(two*G)
C Find old total equivalent deviatoric strain
                      etilde_old = twoThirds*( ehat1_old**2 +
     1                 ehat2_old**2  + ehat3_old**2 + two*ehat4_old**2 +
     2                 two*ehat5_old**2  + two*ehat6_old**2 )
                      etilde_old = SQRT(etilde_old)
C Find the total equivalent deviatoric strain increment
                      depl_bb = (etilde - etilde_old)
                      depl = depl_bb
                END IF
                edot_a = depl_a/dt
                edot = depl/dt
                IF (edot_a .gt. zero) THEN
                      edot_LOG_a = LOG( edot_a )
                ELSE
                      edot_LOG_a = zero
                ENDIF
                IF (edot .gt. zero) THEN
                      edot_LOG = LOG( edot )
                ELSE
                      edot_LOG = zero
                ENDIF
                q_a = (A + B*((depl_a + e_pl_old)**xn))*(one-T_star**xm)
     1                     *(one + C*edot_LOG_a)
                q_depl = (A + B*((depl + e_pl_old)**xn))*(one-T_star**xm)
     1                     *(one + C*edot_LOG)
                f_a = three*G*(etilde - depl_a) - q_a
```

```
                  f_depl = three*G*(etilde - depl) - q_depl
C Find position of the root
                  IF ( (f_a*f_depl).gt.zero ) THEN
                        depl_a = depl
                  ELSE
                        depl_b = depl
                  ENDIF
C The deviatoric stresses are therefore:
                  denom = one + three*G*depl/q_depl
                  IF (denom .eq. zero) THEN
                        s1 = zero
                        s2 = zero
                        s3 = zero
                        s4 = zero
                        s5 = zero
                        s6 = zero
                  ELSE
                        s1 = (two*G / denom) * ehat1
                        s2 = (two*G / denom) * ehat2
                        s3 = (two*G / denom) * ehat3
                        s4 = (two*G / denom) * ehat4
                        s5 = (two*G / denom) * ehat5
                        s6 = (two*G / denom) * ehat6
                  END IF
C Find normals (in direction of deviatoric stresses)
                  xnormal1 = threeHalfs*s1/q_depl
                  xnormal2 = threeHalfs*s2/q_depl
                  xnormal3 = threeHalfs*s3/q_depl
                  xnormal4 = threeHalfs*s4/q_depl
                  xnormal5 = threeHalfs*s5/q_depl
                  xnormal6 = threeHalfs*s6/q_depl
C Find deviatoric plastic strain increment components if needed
C                 depl1 = depl*xnormal1
C                 depl2 = depl*xnormal2
C                 depl3 = depl*xnormal3
C                 depl4 = depl*xnormal4
C                 depl5 = depl*xnormal5
C                 depl6 = depl*xnormal6
C Find new stresses
                  s_trial1 = s1 - p
                  s_trial2 = s2 - p
                  s_trial3 = s3 - p
                  s_trial4 = s4
                  s_trial5 = s5
                  s_trial6 = s6
C Find increase in temperature
                  term = (xneta*depl)/(two*density(i)*s_heat)
                  dtheta = term*( xnormal1*(stressOld(i,1)+s_trial1) +
     1                            xnormal2*(stressOld(i,2)+s_trial2) +
     2                            xnormal3*(stressOld(i,3)+s_trial3) +
     3                            two*xnormal4*(stressOld(i,4)+s_trial4) +
     4                            two*xnormal5*(stressOld(i,5)+s_trial5) +
     5                            two*xnormal6*(stressOld(i,6)+s_trial6) )
                  T = tempOld(i) + dtheta
C Find updated temperature effect
                  IF (T .lt. T_trans) THEN
                        T_star = zero
                  ELSEIF ( T_trans .ge. T .le. T_melt ) THEN
                        T_star = (T - T_trans)/(T_melt - T_trans)
                  ELSE
                        T_star = one
```

```
                END IF
            END DO
C Find new stresses
                stressNew(i,1) = s_trial1
                stressNew(i,2) = s_trial2
                stressNew(i,3) = s_trial3
                stressNew(i,4) = s_trial4
                stressNew(i,5) = s_trial5
                stressNew(i,6) = s_trial6
            END IF
C Update the state variables
        stateNew(i,1) = stateOld(i,1) + depl
        tempNew(i) = tempOld(i) + dtheta
C
C Update the specific internal energy -
        stressPower = zero
        stressPower = half * (  ( stressOld(i,1) + stressNew(i,1) )
     1      *strainInc(i,1) +     ( stressOld(i,2) + stressNew(i,2) )
     2      *strainInc(i,2) +     ( stressOld(i,3) + stressNew(i,3) )
     3      *strainInc(i,3) + two*( stressOld(i,4) + stressNew(i,4) )
     4      *strainInc(i,4) + two*( stressOld(i,5) + stressNew(i,5) )
     5      *strainInc(i,5) + two*( stressOld(i,6) + stressNew(i,6) )
     6      *strainInc(i,6) )
C
        enerInternNew(i) = enerInternOld(i) + stressPower/density(i)
C
C Update the dissipated inelastic specific energy -
        smean = zero
        smean = third * ( stressNew(i,1) + stressNew(i,2)
     1                 + stressNew(i,3) )
        equivStress = threeHalfs *
     1    ( (stressNew(i,1)-smean)**2
     2    + (stressNew(i,2)-smean)**2
     3    + (stressNew(i,3)-smean)**2
     4    + two * stressNew(i,4)**2
     5    + two * stressNew(i,5)**2
     6    + two * stressNew(i,6)**2 )
        IF (equivStress .ne. zero) THEN
          equivStress = SQRT( equivStress )
        ENDIF
        plasticWorkInc = equivStress * depl
        enerInelasNew(i) = enerInelasOld(i) + plasticWorkInc/density(i)
        END IF
C      WRITE (unit=110, fmt=100) i,count,depl,q_trial,yield
      END DO
C  100 FORMAT (I4,f8.2,f16.14,f16.4,f16.4)
C
      RETURN
      END
```

B.2 ZERILLI-ARMSTRONG VUMAT FOR BCC MATERIALS

```
      SUBROUTINE vumat(
C Read only -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
      INCLUDE 'vaba_param.inc'
C
C
C     Linear elastic, ZA-BCC hardening.
C     The state variables are stored as:
C      STATE(*,1) = total equivalent plastic strain
C     Temperature and strain rate in DO loop.
C
C All arrays dimensioned by (*) are not used in this algorithm
      DIMENSION props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(nblock),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(nblock), enerInelasNew(nblock)
C
      CHARACTER*80 cmname
C
      PARAMETER( zero = 0.0d0, one = 1.0d0, two = 2.0d0, three = 3.0d0,
     1  third = one/three, half = .5d0, twoThirds = two/three,
     2  threeHalfs = 1.5d0 )
C Material property definitions
        e          = zero
        e          = props(1)
        xnu        = zero
        xnu        = props(2)
        C0         = zero
        C0         = props(3)
        C1         = zero
        C1         = props(4)
        C3         = zero
        C3         = props(5)
        C4         = zero
        C4         = props(6)
        C5         = zero
        C5         = props(7)
        xn         = zero
        xn         = props(8)
        xneta      = zero
        xneta      = props(9)
        t_heat     = zero
```

```
      s_heat    = props(10)
C
C Find Bulk and Shear modulii
      G = e / ( two * (one + xnu) )
      bulk = e / ( three * (one - two*xnu) )
C
C     OPEN(unit=110,FILE='e:\Thesis\test.dat')
C
C
      DO i = 1,nblock
C Find strain trace increment
      trace = zero
      trace = strainInc(i,1) + strainInc(i,2) +
     1              strainInc(i,3)
C Find deviatoric strain increments
      de1 = strainInc(i,1) - third*trace
      de2 = strainInc(i,2) - third*trace
      de3 = strainInc(i,3) - third*trace
      de4 = strainInc(i,4)
      de5 = strainInc(i,5)
      de6 = strainInc(i,6)
C Find trial elastic deviatoric stress increments
      s1_inc = two*G*de1
      s2_inc = two*G*de2
      s3_inc = two*G*de3
      s4_inc = two*G*de4
      s5_inc = two*G*de5
      s6_inc = two*G*de6
C Find equivalent pressure stress increment
      p_inc = zero
      p_inc = -bulk * trace
C Find trial stresses
      sig1 = stressOld(i,1) + s1_inc - p_inc
      sig2 = stressOld(i,2) + s2_inc - p_inc
      sig3 = stressOld(i,3) + s3_inc - p_inc
      sig4 = stressOld(i,4) + s4_inc
      sig5 = stressOld(i,5) + s5_inc
      sig6 = stressOld(i,6) + s6_inc
C For Abaqus check
      IF (stepTime + totalTime .eq. zero) THEN
          stressNew(i,1) = sig1
          stressNew(i,2) = sig2
          stressNew(i,3) = sig3
          stressNew(i,4) = sig4
          stressNew(i,5) = sig5
          stressNew(i,6) = sig6
      ELSE
C Find equivalent pressure stress
          p = zero
          p = -third * (sig1 + sig2 + sig3)
C Find deviatoric stresses
          s1 = sig1 + p
          s2 = sig2 + p
          s3 = sig3 + p
          s4 = sig4
          s5 = sig5
          s6 = sig6
C Find Mises equivalent stress
          q_trial = zero
          q_trial = s1**2 + s2**2 + s3**2 + two*s4**2
     1            + two*s5**2 + two*s6**2
```

```
            q_trial = SQRT( threeHalfs*q_trial )
C Set up old equivalent plastic strain
            IF ((stepTime + totalTime) .eq. zero) THEN
                  e_pl_old = zero
            ELSE
                  e_pl_old = stateOld(i,1)
            ENDIF
C Set up temperature effects
            T = zero
            T = tempOld(i)
C Find the new total equivalent deviatoric strain increment
            deeq = twoThirds*( de1**2 + de2**2 + de3**2 +
     1            two*de4**2 + two*de5**2 + two*de6**2 )
            deeq = SQRT(deeq)
            dedot = deeq/dt
C Log trap for edot
            IF (edot .gt. zero) THEN
                  dedot_LOG = LOG( dedot )
            ELSE
                  dedot_LOG = zero
            ENDIF
            yield = C0 + C1*EXP(-C3*T + C4*T*dedot_LOG) +
     1            C5*(e_pl_old**xn)
C Check for yielding
            IF (q_trial .le. yield) THEN
                  stressNew(i,1) = sig1
                  stressNew(i,2) = sig2
                  stressNew(i,3) = sig3
                  stressNew(i,4) = sig4
                  stressNew(i,5) = sig5
                  stressNew(i,6) = sig6
                  depl = zero
                  dtheta = zero
                  count = 0
            ELSE
C Find old deviatoric stresses
                  sig_mean = zero
                  sig_mean = -third * (stressOld(i,1) +
     1                        stressOld(i,2) + stressOld(i,3))
                  s_old1 = stressOld(i,1) + sig_mean
                  s_old2 = stressOld(i,2) + sig_mean
                  s_old3 = stressOld(i,3) + sig_mean
                  s_old4 = stressOld(i,4)
                  s_old5 = stressOld(i,5)
                  s_old6 = stressOld(i,6)
C Find equivalent deviatoric strains
                  ehat1 = s_old1/(two*G) + de1
                  ehat2 = s_old2/(two*G) + de2
                  ehat3 = s_old3/(two*G) + de3
                  ehat4 = s_old4/(two*G) + de4
                  ehat5 = s_old5/(two*G) + de5
                  ehat6 = s_old6/(two*G) + de6
C Find total equivalent deviatoric strain
                  etilde = twoThirds*( ehat1**2 + ehat2**2 + ehat3**2 +
     1                  two*ehat4**2 + two*ehat5**2 + two*ehat6**2 )
                  etilde = SQRT(etilde)
C Use Bisection Method to find depl (equivalent deviatoric plastic
C strain increment)
                  Tol = 1.0d5
                  depl_a = zero
                  depl_b = 1.0d-1
```

```
                depl = (depl_a + depl_b)/two
                edot = depl/dt
                IF (edot .gt. zero) THEN
                      edot_LOG = LOG( edot )
                ELSE
                      edot_LOG = zero
                ENDIF
                q_depl = C0 + C1*EXP(-C3*T + C4*T*edot_LOG) +
     1                       C5*((e_pl_old + depl)**xn)
                f_depl = three*G*(etilde - depl) - q_depl
                count = 0
          DO WHILE ((ABS(f_depl).gt.Tol).AND.(count.le.100))
                      count = count + 1
                      depl = (depl_a + depl_b)/two
C If the Bisection method is not going to converge then set depl to it's
C maximum possible value i.e. the increment in total equivalent strain.
                IF (count.ge.100) THEN
C Find old deviatoric strains
                      ehat1_old = s_old1/(two*G)
                      ehat2_old = s_old2/(two*G)
                      ehat3_old = s_old3/(two*G)
                      ehat4_old = s_old4/(two*G)
                      ehat5_old = s_old5/(two*G)
                      ehat6_old = s_old6/(two*G)
C Find old total equivalent deviatoric strain
                      etilde_old = twoThirds*( ehat1_old**2 +
     1                 ehat2_old**2 + ehat3_old**2 + two*ehat4_old**2 +
     2                 two*ehat5_old**2 + two*ehat6_old**2 )
                      etilde_old = SQRT(etilde_old)
C Find the total equivalent deviatoric strain increment
                      depl_bb = (etilde - etilde_old)
                      depl = depl_bb
                END IF
C If there is a plastic strain increment then q changes
                      edot_a = depl_a/dt
                      edot = depl/dt
                      IF (edot_a .gt. zero) THEN
                            edot_LOG_a = LOG( edot_a )
                      ELSE
                            edot_LOG_a = zero
                      ENDIF
                      IF (edot .gt. zero) THEN
                            edot_LOG = LOG( edot )
                      ELSE
                            edot_LOG = zero
                      ENDIF
                q_a = C0 + C1*EXP(-C3*T + C4*T*edot_LOG_a) +
     1                       C5*((e_pl_old + depl_a)**xn)
                q_depl = C0 + C1*EXP(-C3*T + C4*T*edot_LOG) +
     1                       C5*((e_pl_old + depl)**xn)
                f_a = three*G*(etilde - depl_a) - q_a
                f_depl = three*G*(etilde - depl) - q_depl
                IF ( (f_a*f_depl).gt.zero ) THEN
                      depl_a = depl
                ELSE
                      depl_b = depl
                ENDIF
C The deviatoric stresses are therefore:
                denom = one + three*G*depl/q_depl
                IF (denom .eq. zero) THEN
                      s1 = zero
```

```
                        s2 = zero
                        s3 = zero
                        s4 = zero
                        s5 = zero
                        s6 = zero
                    ELSE
                        s1 = (two*G / denom) * ehat1
                        s2 = (two*G / denom) * ehat2
                        s3 = (two*G / denom) * ehat3
                        s4 = (two*G / denom) * ehat4
                        s5 = (two*G / denom) * ehat5
                        s6 = (two*G / denom) * ehat6
                    END IF
C Find normals (in direction of deviatoric stresses)
                    xnormal1 = threeHalfs*s1/q_depl
                    xnormal2 = threeHalfs*s2/q_depl
                    xnormal3 = threeHalfs*s3/q_depl
                    xnormal4 = threeHalfs*s4/q_depl
                    xnormal5 = threeHalfs*s5/q_depl
                    xnormal6 = threeHalfs*s6/q_depl
C Find deviatoric plastic strain increment components
C                   depl1 = depl*xnormal1
C                   depl2 = depl*xnormal2
C                   depl3 = depl*xnormal3
C                   depl4 = depl*xnormal4
C                   depl5 = depl*xnormal5
C                   depl6 = depl*xnormal6
C Find new stresses
                    s_trial1 = s1 - p
                    s_trial2 = s2 - p
                    s_trial3 = s3 - p
                    s_trial4 = s4
                    s_trial5 = s5
                    s_trial6 = s6
C Find increase in temperature
                    term = (xneta*depl)/(two*density(i)*s_heat)
                    dtheta = term*( xnormal1*(stressOld(i,1)+s_trial1) +
     1                              xnormal2*(stressOld(i,2)+s_trial2) +
     2                              xnormal3*(stressOld(i,3)+s_trial3) +
     3                              two*xnormal4*(stressOld(i,4)+s_trial4) +
     4                              two*xnormal5*(stressOld(i,5)+s_trial5) +
     5                              two*xnormal6*(stressOld(i,6)+s_trial6) )
                    T = tempOld(i) + dtheta
                END DO
C Find new stresses
                    stressNew(i,1) = s_trial1
                    stressNew(i,2) = s_trial2
                    stressNew(i,3) = s_trial3
                    stressNew(i,4) = s_trial4
                    stressNew(i,5) = s_trial5
                    stressNew(i,6) = s_trial6
                END IF
C Update the state variables
            stateNew(i,1) = stateOld(i,1) + depl
            tempNew(i) = tempOld(i) + dtheta
C
C Update the specific internal energy -
            stressPower = zero
            stressPower = half * (  ( stressOld(i,1) + stressNew(i,1) )
     1          *strainInc(i,1) +     ( stressOld(i,2) + stressNew(i,2) )
     2          *strainInc(i,2) +     ( stressOld(i,3) + stressNew(i,3) )
```

```
     3      *strainInc(i,3) + two*( stressOld(i,4) + stressNew(i,4) )
     4      *strainInc(i,4) + two*( stressOld(i,5) + stressNew(i,5) )
     5      *strainInc(i,5) + two*( stressOld(i,6) + stressNew(i,6) )
     6      *strainInc(i,6) )
C
        enerInternNew(i) = enerInternOld(i) + stressPower/density(i)
C
C Update the dissipated inelastic specific energy -
        smean = zero
        smean = third * ( stressNew(i,1) + stressNew(i,2)
     1                 + stressNew(i,3) )
        equivStress = threeHalfs *
     1    ( (stressNew(i,1)-smean)**2
     2    + (stressNew(i,2)-smean)**2
     3    + (stressNew(i,3)-smean)**2
     4    + two * stressNew(i,4)**2
     5    + two * stressNew(i,5)**2
     6    + two * stressNew(i,6)**2 )
        IF (equivStress .ne. zero) THEN
          equivStress = SQRT( equivStress )
        ENDIF
        plasticWorkInc = equivStress * depl
        enerInelasNew(i) = enerInelasOld(i) + plasticWorkInc/density(i)
        END IF
C      WRITE (unit=110, fmt=100) i,count,f_depl,depl
      END DO
C  100 FORMAT (I3,f8.2,f16.8,f16.10)
C
      RETURN
      END
```

B.3 ZERILLI-ARMSTRONG VUMAT FOR FCC MATERIALS

```
      SUBROUTINE vumat(
C Read only -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
      INCLUDE 'vaba_param.inc'
C
C     Linear elastic, ZA-FCC hardening.
C     The state variables are stored as:
C      STATE(*,1) = total equivalent plastic strain
C     Temperature and strain rate in DO loop.
C
C All arrays dimensioned by (*) are not used in this algorithm
      DIMENSION props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(nblock),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(nblock), enerInelasNew(nblock)
C
      CHARACTER*80 cmname
C
      PARAMETER( zero = 0.0d0, one = 1.0d0, two = 2.0d0, three = 3.0d0,
     1  third = one/three, half = .5d0, twoThirds = two/three,
     2  threeHalfs = 1.5d0 )
C Material property definitions
        e          = zero
        e          = props(1)
        xnu        = zero
        xnu        = props(2)
        C0         = zero
        C0         = props(3)
        C2         = zero
        C2         = props(4)
        C3         = zero
        C3         = props(5)
        C4         = zero
        C4         = props(6)
        xneta      = zero
        xneta      = props(7)
        s_heat     = zero
        s_heat     = props(8)
C
C Find Bulk and Shear modulii
        G = e / ( two * (one + xnu) )
        bulk = e / ( three * (one - two*xnu) )
```

```
C
C     OPEN(unit=110,FILE='e:\Thesis\test.dat')
C
C
      DO i = 1,nblock
C Find strain trace increment
        trace = zero
        trace = strainInc(i,1) + strainInc(i,2) +
     1              strainInc(i,3)
C Find deviatoric strain increments
        de1 = strainInc(i,1) - third*trace
        de2 = strainInc(i,2) - third*trace
        de3 = strainInc(i,3) - third*trace
        de4 = strainInc(i,4)
        de5 = strainInc(i,5)
        de6 = strainInc(i,6)
C Find trial elastic deviatoric stress increments
        s1_inc = two*G*de1
        s2_inc = two*G*de2
        s3_inc = two*G*de3
        s4_inc = two*G*de4
        s5_inc = two*G*de5
        s6_inc = two*G*de6
C Find equivalent pressure stress increment
        p_inc = zero
        p_inc = -bulk * trace
C Find trial stresses
        sig1 = stressOld(i,1) + s1_inc - p_inc
        sig2 = stressOld(i,2) + s2_inc - p_inc
        sig3 = stressOld(i,3) + s3_inc - p_inc
        sig4 = stressOld(i,4) + s4_inc
        sig5 = stressOld(i,5) + s5_inc
        sig6 = stressOld(i,6) + s6_inc
C For Abaqus check
        IF (stepTime + totalTime .eq. zero) THEN
            stressNew(i,1) = sig1
            stressNew(i,2) = sig2
            stressNew(i,3) = sig3
            stressNew(i,4) = sig4
            stressNew(i,5) = sig5
            stressNew(i,6) = sig6
        ELSE
C Find equivalent pressure stress
            p = zero
            p = -third * (sig1 + sig2 + sig3)
C Find deviatoric stresses
            s1 = sig1 + p
            s2 = sig2 + p
            s3 = sig3 + p
            s4 = sig4
            s5 = sig5
            s6 = sig6
C Find Mises equivalent stress
            q_trial = zero
            q_trial = s1**2 + s2**2 + s3**2 + two*s4**2
     1            + two*s5**2 + two*s6**2
            q_trial = SQRT( threeHalfs*q_trial )
C Set up old equivalent plastic strain
            IF ((stepTime + totalTime) .eq. zero) THEN
                e_pl_old = zero
            ELSE
```

```
                e_pl_old = stateOld(i,1)
            ENDIF
C Set up temperature effects
            T = zero
            T = tempOld(i)
C Find the new total equivalent deviatoric strain increment
            deeq = twoThirds*( de1**2 + de2**2 + de3**2 +
     1           two*de4**2 + two*de5**2 + two*de6**2 )
            deeq = SQRT(deeq)
            dedot = deeq/dt
C Log trap for edot
            IF (edot .gt. zero) THEN
                dedot_LOG = LOG( dedot )
            ELSE
                dedot_LOG = zero
            ENDIF
            yield = C0 + C2*(e_pl_old**half)*EXP(-C3*T +
     1           C4*T*dedot_LOG)
C Check for yielding
            IF (q_trial .le. yield) THEN
                stressNew(i,1) = sig1
                stressNew(i,2) = sig2
                stressNew(i,3) = sig3
                stressNew(i,4) = sig4
                stressNew(i,5) = sig5
                stressNew(i,6) = sig6
                depl = zero
                dtheta = zero
                count = 0
            ELSE
C Find old deviatoric stresses
                sig_mean = zero
                sig_mean = -third * (stressOld(i,1) +
     1                      stressOld(i,2) + stressOld(i,3))
                s_old1 = stressOld(i,1) + sig_mean
                s_old2 = stressOld(i,2) + sig_mean
                s_old3 = stressOld(i,3) + sig_mean
                s_old4 = stressOld(i,4)
                s_old5 = stressOld(i,5)
                s_old6 = stressOld(i,6)
C Find equivalent deviatoric strains
                ehat1 = s_old1/(two*G) + de1
                ehat2 = s_old2/(two*G) + de2
                ehat3 = s_old3/(two*G) + de3
                ehat4 = s_old4/(two*G) + de4
                ehat5 = s_old5/(two*G) + de5
                ehat6 = s_old6/(two*G) + de6
C Find total equivalent deviatoric strain
                etilde = twoThirds*( ehat1**2 + ehat2**2 + ehat3**2 +
     1               two*ehat4**2 + two*ehat5**2 + two*ehat6**2 )
                etilde = SQRT(etilde)
C Use Bisection Method to find depl (equivalent deviatoric plastic
C strain increment)
                Tol = 1.0d5
                depl_a = zero
                depl_b = 1.0d-1
                depl = (depl_a + depl_b)/two
                edot = depl/dt
                IF (edot .gt. zero) THEN
                    edot_LOG = LOG( edot )
                ELSE
```

```
                        edot_LOG = zero
                    ENDIF
                    q_depl = C0 + C2*((e_pl_old + depl)**half)
     1                       *EXP(-C3*T + C4*T*edot_LOG)
                    f_depl = three*G*(etilde - depl) - q_depl
                    count = 0
                DO WHILE ((ABS(f_depl).gt.Tol).AND.(count.le.100))
                        count = count + 1
                        depl = (depl_a + depl_b)/two
C If the Bisection method is not going to converge then set depl to it's
C maximum possible value i.e. the increment in total equivalent strain.
                    IF (count.ge.100) THEN
C Find old deviatoric strains
                        ehat1_old = s_old1/(two*G)
                        ehat2_old = s_old2/(two*G)
                        ehat3_old = s_old3/(two*G)
                        ehat4_old = s_old4/(two*G)
                        ehat5_old = s_old5/(two*G)
                        ehat6_old = s_old6/(two*G)
C Find old total equivalent deviatoric strain
                        etilde_old = twoThirds*( ehat1_old**2 +
     1                   ehat2_old**2 + ehat3_old**2 + two*ehat4_old**2 +
     2                   two*ehat5_old**2 + two*ehat6_old**2 )
                        etilde_old = SQRT(etilde_old)
C Find the total equivalent deviatoric strain increment
                        depl_bb = (etilde - etilde_old)
                        depl = depl_bb
                    END IF
C If there is a plastic strain increment then q changes
                    edot_a = depl_a/dt
                    edot = depl/dt
                    IF (edot_a .gt. zero) THEN
                        edot_LOG_a = LOG( edot_a )
                    ELSE
                        edot_LOG_a = zero
                    ENDIF
                    IF (edot .gt. zero) THEN
                        edot_LOG = LOG( edot )
                    ELSE
                        edot_LOG = zero
                    ENDIF
                    q_a = C0 + C2*((e_pl_old + depl_a)**half)
     1                       *EXP(-C3*T + C4*T*edot_LOG_a)
                    q_depl = C0 + C2*((e_pl_old + depl)**half)
     1                       *EXP(-C3*T + C4*T*edot_LOG)
                    f_a = three*G*(etilde - depl_a) - q_a
                    f_depl = three*G*(etilde - depl) - q_depl
                    IF ( (f_a*f_depl).gt.zero ) THEN
                        depl_a = depl
                    ELSE
                        depl_b = depl
                    ENDIF
C The deviatoric stresses are therefore:
                    denom = one + three*G*depl/q_depl
                    IF (denom .eq. zero) THEN
                        s1 = zero
                        s2 = zero
                        s3 = zero
                        s4 = zero
                        s5 = zero
                        s6 = zero
```

```
                ELSE
                      s1 = (two*G / denom) * ehat1
                      s2 = (two*G / denom) * ehat2
                      s3 = (two*G / denom) * ehat3
                      s4 = (two*G / denom) * ehat4
                      s5 = (two*G / denom) * ehat5
                      s6 = (two*G / denom) * ehat6
                END IF
C Find normals (in direction of deviatoric stresses)
                xnormal1 = threeHalfs*s1/q_depl
                xnormal2 = threeHalfs*s2/q_depl
                xnormal3 = threeHalfs*s3/q_depl
                xnormal4 = threeHalfs*s4/q_depl
                xnormal5 = threeHalfs*s5/q_depl
                xnormal6 = threeHalfs*s6/q_depl
C Find deviatoric plastic strain increment components
C               depl1 = depl*xnormal1
C               depl2 = depl*xnormal2
C               depl3 = depl*xnormal3
C               depl4 = depl*xnormal4
C               depl5 = depl*xnormal5
C               depl6 = depl*xnormal6
C Find new stresses
                s_trial1 = s1 - p
                s_trial2 = s2 - p
                s_trial3 = s3 - p
                s_trial4 = s4
                s_trial5 = s5
                s_trial6 = s6
C Find increase in temperature
                term = (xneta*depl)/(two*density(i)*s_heat)
                dtheta = term*( xnormal1*(stressOld(i,1)+s_trial1) +
     1                          xnormal2*(stressOld(i,2)+s_trial2) +
     2                          xnormal3*(stressOld(i,3)+s_trial3) +
     3                          two*xnormal4*(stressOld(i,4)+s_trial4) +
     4                          two*xnormal5*(stressOld(i,5)+s_trial5) +
     5                          two*xnormal6*(stressOld(i,6)+s_trial6) )
                T = tempOld(i) + dtheta
            END DO
C Find new stresses
                stressNew(i,1) = s_trial1
                stressNew(i,2) = s_trial2
                stressNew(i,3) = s_trial3
                stressNew(i,4) = s_trial4
                stressNew(i,5) = s_trial5
                stressNew(i,6) = s_trial6
            END IF
C Update the state variables
        stateNew(i,1) = stateOld(i,1) + depl
        tempNew(i) = tempOld(i) + dtheta
C
C Update the specific internal energy -
        stressPower = zero
        stressPower = half * (  ( stressOld(i,1) + stressNew(i,1) )
     1      *strainInc(i,1) +     ( stressOld(i,2) + stressNew(i,2) )
     2      *strainInc(i,2) +     ( stressOld(i,3) + stressNew(i,3) )
     3      *strainInc(i,3) + two*( stressOld(i,4) + stressNew(i,4) )
     4      *strainInc(i,4) + two*( stressOld(i,5) + stressNew(i,5) )
     5      *strainInc(i,5) + two*( stressOld(i,6) + stressNew(i,6) )
     6      *strainInc(i,6) )
C
```

```
        enerInternNew(i) = enerInternOld(i) + stressPower/density(i)
C
C Update the dissipated inelastic specific energy -
        smean = zero
        smean = third * ( stressNew(i,1) + stressNew(i,2)
     1                  + stressNew(i,3) )
        equivStress = threeHalfs *
     1    ( (stressNew(i,1)-smean)**2
     2    + (stressNew(i,2)-smean)**2
     3    + (stressNew(i,3)-smean)**2
     4    + two * stressNew(i,4)**2
     5    + two * stressNew(i,5)**2
     6    + two * stressNew(i,6)**2 )
        IF (equivStress .ne. zero) THEN
          equivStress = SQRT( equivStress )
        ENDIF
        plasticWorkInc = equivStress * depl
        enerInelasNew(i) = enerInelasOld(i) + plasticWorkInc/density(i)
        END IF
C       WRITE (unit=110, fmt=100) i,count,f_depl,depl
      END DO
C  100 FORMAT (I3,f8.2,f16.8,f16.10)
C
      RETURN
      END
```

B.4 JOHNSON-COOK VUMAT USING NEWTON'S METHOD

```
      SUBROUTINE vumat(
C Read only -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only -
     5  stressNew, stateNew, enerInternNew, enerInelasNew )
C
      INCLUDE 'vaba_param.inc'
C
C
C     Linear elastic, JC hardening (including temperature). Rate dependent.
C     The state variables are stored as:
C      STATE(*,1) = total equivalent plastic strain
C     Uses Newton's method with the old temperature and strain rate.
C
C All arrays dimensioned by (*) are not used in this algorithm
      DIMENSION props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(nblock),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(nblock), enerInelasNew(nblock)
C
      CHARACTER*80 cmname
C
      PARAMETER( zero = 0.0d0, one = 1.0d0, two = 2.0d0, three = 3.0d0,
     1  third = one/three, half = .5d0, twoThirds = two/three,
     2  threeHalfs = 1.5d0 )
C Material property definitions
        e          = zero
        e          = props(1)
        xnu        = zero
        xnu        = props(2)
        A          = zero
        A          = props(3)
        B          = zero
        B          = props(4)
        xn         = zero
        xn         = props(5)
        xm         = zero
        xm         = props(6)
        T_trans    = zero
        T_trans    = props(7)
        T_melt     = zero
        T_melt     = props(8)
        xneta      = zero
        xneta      = props(9)
        s_heat     = zero
```

```
      s_heat     = props(10)
      C          = zero
      C          = props(11)
C
C Find Bulk and Shear modulii
      G = e / ( two * (one + xnu) )
      bulk = e / ( three * (one - two*xnu) )
C
C     OPEN(unit=110,FILE='e:\Thesis\test.dat')
C
C
      DO i = 1,nblock
C Find strain trace increment
      trace = zero
      trace = strainInc(i,1) + strainInc(i,2) +
     1              strainInc(i,3)
C Find deviatoric strain increments
      de1 = strainInc(i,1) - third*trace
      de2 = strainInc(i,2) - third*trace
      de3 = strainInc(i,3) - third*trace
      de4 = strainInc(i,4)
      de5 = strainInc(i,5)
      de6 = strainInc(i,6)
C Find trial elastic deviatoric stress increments
      s1_inc = two*G*de1
      s2_inc = two*G*de2
      s3_inc = two*G*de3
      s4_inc = two*G*de4
      s5_inc = two*G*de5
      s6_inc = two*G*de6
C Find equivalent pressure stress increment
      p_inc = zero
      p_inc = -bulk * trace
C Find trial stresses
      sig1 = stressOld(i,1) + s1_inc - p_inc
      sig2 = stressOld(i,2) + s2_inc - p_inc
      sig3 = stressOld(i,3) + s3_inc - p_inc
      sig4 = stressOld(i,4) + s4_inc
      sig5 = stressOld(i,5) + s5_inc
      sig6 = stressOld(i,6) + s6_inc
C For Abaqus check
      IF (stepTime + totalTime .eq. zero) THEN
          stressNew(i,1) = sig1
          stressNew(i,2) = sig2
          stressNew(i,3) = sig3
          stressNew(i,4) = sig4
          stressNew(i,5) = sig5
          stressNew(i,6) = sig6
      ELSE
C Find equivalent pressure stress
          p = zero
          p = -third * (sig1 + sig2 + sig3)
C Find deviatoric stresses
          s1 = sig1 + p
          s2 = sig2 + p
          s3 = sig3 + p
          s4 = sig4
          s5 = sig5
          s6 = sig6
C Find Mises equivalent stress
          q_trial = zero
```

```
            q_trial = s1**2 + s2**2 + s3**2 + two*s4**2
     1            + two*s5**2 + two*s6**2
            q_trial = SQRT( threeHalfs*q_trial )
C Setup old equivalent plastic strain
            IF ((stepTime + totalTime) .eq. zero) THEN
                  e_pl_old = zero
            ELSE
                  e_pl_old = stateOld(i,1)
            ENDIF
C Setup old temperature effects
            T = zero
            T = tempOld(i)
            IF (T .lt. T_trans) THEN
                  T_star = zero
            ELSEIF ( T_trans .ge. T .le. T_melt ) THEN
                  T_star = (T - T_trans)/(T_melt - T_trans)
            ELSE
                  T_star = one
            END IF
C Find the new total equivalent deviatoric strain increment
            deeq = twoThirds*( de1**2 + de2**2 + de3**2 +
     1            two*de4**2 + two*de5**2 + two*de6**2 )
            deeq = SQRT(deeq)
            dedot = deeq/dt
C Log trap for edot
            IF (dedot .gt. zero) THEN
                  dedot_LOG = LOG( dedot )
            ELSE
                  dedot_LOG = zero
            ENDIF
            yield = (A + B*(e_pl_old**xn))*(one - T_star**xm)
     1                  *(one + C*dedot_LOG)
C Check for yielding
            IF (q_trial .le. yield) THEN
                  stressNew(i,1) = sig1
                  stressNew(i,2) = sig2
                  stressNew(i,3) = sig3
                  stressNew(i,4) = sig4
                  stressNew(i,5) = sig5
                  stressNew(i,6) = sig6
                  depl = zero
                  dtheta = zero
                  count = 0
            ELSE
C Find old deviatoric stresses
                  sig_mean = zero
                  sig_mean = -third * (stressOld(i,1) +
     1                        stressOld(i,2) + stressOld(i,3))
                  s_old1 = stressOld(i,1) + sig_mean
                  s_old2 = stressOld(i,2) + sig_mean
                  s_old3 = stressOld(i,3) + sig_mean
                  s_old4 = stressOld(i,4)
                  s_old5 = stressOld(i,5)
                  s_old6 = stressOld(i,6)
C Find equivalent deviatoric strains
                  ehat1 = s_old1/(two*G) + de1
                  ehat2 = s_old2/(two*G) + de2
                  ehat3 = s_old3/(two*G) + de3
                  ehat4 = s_old4/(two*G) + de4
                  ehat5 = s_old5/(two*G) + de5
                  ehat6 = s_old6/(two*G) + de6
```

```
C Find total equivalent deviatoric strain
                  etilde = twoThirds*( ehat1**2 + ehat2**2 + ehat3**2 +
     1                   two*ehat4**2 + two*ehat5**2 + two*ehat6**2 )
                  etilde = SQRT(etilde)
C Use Newtons Method to find depl (equivalent deviatoric plastic
C strain increment
                  Tol = 1.0d5
                  depl_old = zero
                  depl = 1.0d-3
                  f_depl = three*G*(etilde - depl) - yield
                  count = 0
            DO WHILE ((ABS(f_depl).gt.Tol).AND.(count.le.15))
                        count = count + 1
                        depl_old = depl
                        q = (A + B*((depl + e_pl_old)**xn))*
     1                        (one-T_star**xm)*(one + C*dedot_LOG)
                        depl = depl + (three*G*(etilde - depl) - q)
     1                              /(three*G + B*xn*((e_pl_old)**(xn-one))
     2                              *(one-T_star**xm)*(one + C*dedot_LOG))
                  IF (count.ge.15) THEN
C Find old deviatoric strains
                        ehat1_old = s_old1/(two*G)
                        ehat2_old = s_old2/(two*G)
                        ehat3_old = s_old3/(two*G)
                        ehat4_old = s_old4/(two*G)
                        ehat5_old = s_old5/(two*G)
                        ehat6_old = s_old6/(two*G)
C Find old total equivalent deviatoric strain
                        etilde_old = twoThirds*( ehat1_old**2 +
     1                    ehat2_old**2 + ehat3_old**2 + two*ehat4_old**2 +
     2                    two*ehat5_old**2 + two*ehat6_old**2 )
                        etilde_old = SQRT(etilde_old)
C Find the total equivalent deviatoric strain increment
                        depl_bb = (etilde - etilde_old)
                        depl = depl_bb
                  END IF
                        f_depl = three*G*(etilde - depl) - q
            END DO
C The deviatoric stresses are therefore:
                  denom = one + three*G*depl/q
                  s1 = (two*G / denom) * ehat1
                  s2 = (two*G / denom) * ehat2
                  s3 = (two*G / denom) * ehat3
                  s4 = (two*G / denom) * ehat4
                  s5 = (two*G / denom) * ehat5
                  s6 = (two*G / denom) * ehat6
C Find normals (in direction of deviatoric stresses)
                  xnormal1 = threeHalfs*s1/q
                  xnormal2 = threeHalfs*s2/q
                  xnormal3 = threeHalfs*s3/q
                  xnormal4 = threeHalfs*s4/q
                  xnormal5 = threeHalfs*s5/q
                  xnormal6 = threeHalfs*s6/q
C Find deviatoric plastic strain increment components
C                 depl1 = depl*xnormal1
C                 depl2 = depl*xnormal2
C                 depl3 = depl*xnormal3
C                 depl4 = depl*xnormal4
C                 depl5 = depl*xnormal5
C                 depl6 = depl*xnormal6
C Find new stresses
```

```
                  s_trial1 = s1 - p
                  s_trial2 = s2 - p
                  s_trial3 = s3 - p
                  s_trial4 = s4
                  s_trial5 = s5
                  s_trial6 = s6
C Find increase in temperature
                  term = (xneta*depl)/(two*density(i)*s_heat)
                  dtheta = term*( xnormal1*(stressOld(i,1)+s_trial1) +
     1                      xnormal2*(stressOld(i,2)+s_trial2) +
     2                      xnormal3*(stressOld(i,3)+s_trial3) +
     3                      two*xnormal4*(stressOld(i,4)+s_trial4) +
     4                      two*xnormal5*(stressOld(i,5)+s_trial5) +
     5                      two*xnormal6*(stressOld(i,6)+s_trial6) )
C Find new stresses
                  stressNew(i,1) = s_trial1
                  stressNew(i,2) = s_trial2
                  stressNew(i,3) = s_trial3
                  stressNew(i,4) = s_trial4
                  stressNew(i,5) = s_trial5
                  stressNew(i,6) = s_trial6
               END IF
C Update the state variables
         stateNew(i,1) = stateOld(i,1) + depl
         tempNew(i) = tempOld(i) + dtheta
C
C Update the specific internal energy -
         stressPower = zero
         stressPower = half * (  ( stressOld(i,1) + stressNew(i,1) )
     1       *strainInc(i,1) +     ( stressOld(i,2) + stressNew(i,2) )
     2       *strainInc(i,2) +     ( stressOld(i,3) + stressNew(i,3) )
     3       *strainInc(i,3) + two*( stressOld(i,4) + stressNew(i,4) )
     4       *strainInc(i,4) + two*( stressOld(i,5) + stressNew(i,5) )
     5       *strainInc(i,5) + two*( stressOld(i,6) + stressNew(i,6) )
     6       *strainInc(i,6) )
C
         enerInternNew(i) = enerInternOld(i) + stressPower/density(i)
C
C Update the dissipated inelastic specific energy -
         smean = zero
         smean = third * ( stressNew(i,1) + stressNew(i,2)
     1                   + stressNew(i,3) )
         equivStress = threeHalfs *
     1     ( (stressNew(i,1)-smean)**2
     2     + (stressNew(i,2)-smean)**2
     3     + (stressNew(i,3)-smean)**2
     4     + two * stressNew(i,4)**2
     5     + two * stressNew(i,5)**2
     6     + two * stressNew(i,6)**2 )
         IF (equivStress .ne. zero) THEN
           equivStress = SQRT( equivStress )
         ENDIF
         plasticWorkInc = equivStress * depl
         enerInelasNew(i) = enerInelasOld(i) + plasticWorkInc/density(i)
         END IF
C       WRITE (unit=110,fmt=100) i,count,f_depl,depl,yield,q_trial
       END DO
  100 FORMAT (I4,f8.2,f16.4,f16.14,f16.2,f16.2)
C
       RETURN
       END
```

B.5 JOHNSON-COOK VUMAT USING NON-ITERATIVE METHOD

```
      SUBROUTINE vumat(
C Read only -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
      INCLUDE 'vaba_param.inc'
C
C     Linear elastic, JC hardening (including temperature). Rate dependent.
C     The state variables are stored as:
C      STATE(*,1) = total equivalent plastic strain
C
C All arrays dimensioned by (*) are not used in this algorithm
      DIMENSION props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(nblock),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(nblock),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(nblock), enerInelasNew(nblock)
C
      CHARACTER*80 cmname
C
      PARAMETER( zero = 0.0d0, one = 1.0d0, two = 2.0d0, three = 3.0d0,
     1  third = one/three, half = .5d0, twoThirds = two/three,
     2  threeHalfs = 1.5d0 )
C Material property definitions
        e           = zero
        e           = props(1)
        xnu         = zero
        xnu         = props(2)
        A           = zero
        A           = props(3)
        B           = zero
        B           = props(4)
        xn          = zero
        xn          = props(5)
        xm          = zero
        xm          = props(6)
        T_trans     = zero
        T_trans     = props(7)
        T_melt      = zero
        T_melt      = props(8)
        xneta       = zero
        xneta       = props(9)
        s_heat      = zero
        s_heat      = props(10)
        C           = zero
```

```
      C         = props(11)
C
C Find Bulk and Shear modulii
      G = e / ( two * (one + xnu) )
      bulk = e / ( three * (one - two*xnu) )
C
C     OPEN(unit=110,FILE='e:\Thesis\test.dat')
C
      DO i = 1,nblock
C Find strain trace increment
      trace = zero
      trace = strainInc(i,1) + strainInc(i,2) +
     1              strainInc(i,3)
C Find deviatoric strain increments
      de1 = strainInc(i,1) - third*trace
      de2 = strainInc(i,2) - third*trace
      de3 = strainInc(i,3) - third*trace
      de4 = strainInc(i,4)
      de5 = strainInc(i,5)
      de6 = strainInc(i,6)
C Find trial elastic deviatoric stress increments
      s1_inc = two*G*de1
      s2_inc = two*G*de2
      s3_inc = two*G*de3
      s4_inc = two*G*de4
      s5_inc = two*G*de5
      s6_inc = two*G*de6
C Find equivalent pressure stress increment
      p_inc = zero
      p_inc = -bulk * trace
C Find trial stresses
      sig1 = stressOld(i,1) + s1_inc - p_inc
      sig2 = stressOld(i,2) + s2_inc - p_inc
      sig3 = stressOld(i,3) + s3_inc - p_inc
      sig4 = stressOld(i,4) + s4_inc
      sig5 = stressOld(i,5) + s5_inc
      sig6 = stressOld(i,6) + s6_inc
C For Abaqus check
      IF (stepTime + totalTime .eq. zero) THEN
          stressNew(i,1) = sig1
          stressNew(i,2) = sig2
          stressNew(i,3) = sig3
          stressNew(i,4) = sig4
          stressNew(i,5) = sig5
          stressNew(i,6) = sig6
      ELSE
C Find equivalent pressure stress
          p = zero
          p = -third * (sig1 + sig2 + sig3)
C Find deviatoric stresses
          s1 = sig1 + p
          s2 = sig2 + p
          s3 = sig3 + p
          s4 = sig4
          s5 = sig5
          s6 = sig6
C Find Mises equivalent stress
          q_trial = zero
          q_trial = s1**2 + s2**2 + s3**2 + two*s4**2
     1            + two*s5**2 + two*s6**2
          q_trial = SQRT( threeHalfs*q_trial )
```

```
C Set up old equivalent plastic strain
            IF ((stepTime + totalTime) .eq. zero) THEN
                  e_pl_old = zero
            ELSE
                  e_pl_old = stateOld(i,1)
            ENDIF
C Set up temperature effects
            T = zero
            T = tempOld(i)
            IF (T .lt. T_trans) THEN
                  T_star = zero
            ELSEIF ( T_trans .ge. T .le. T_melt ) THEN
                  T_star = (T - T_trans)/(T_melt - T_trans)
            ELSE
                  T_star = one
            END IF
C Equivalent total strain increment
            deeq = twoThirds*( del**2 + de2**2 + de3**2 +
     1            two*de4**2 + two*de5**2 + two*de6**2 )
            deeq = SQRT(deeq)
            dedot = deeq/dt
C Equivalent total strain rate
            IF (dedot.ge.one) THEN
                  dedot_log = LOG (dedot)
            ELSE
                  dedot_log = zero
            END IF
            yield = (A + B*(e_pl_old**xn))*(one - T_star**xm)
     1                  *(one + C*dedot_log)
C Check for yielding
            IF (q_trial .le. yield) THEN
                  stressNew(i,1) = sig1
                  stressNew(i,2) = sig2
                  stressNew(i,3) = sig3
                  stressNew(i,4) = sig4
                  stressNew(i,5) = sig5
                  stressNew(i,6) = sig6
                  depl = zero
                  dtheta = zero
                  count = 0
            ELSE
C Scaling factor
                  IF (q_trial.gt.zero) THEN
                        beta = yield / q_trial
                  ELSE
                        beta = zero
                  END IF
C New adjusted deviatoric stresses
                  s1_new = beta*s1
                  s2_new = beta*s2
                  s3_new = beta*s3
                  s4_new = beta*s4
                  s5_new = beta*s5
                  s6_new = beta*s6
C Find old deviatoric stresses
                  sig_mean = zero
                  sig_mean = -third * (stressOld(i,1) +
     1                        stressOld(i,2) + stressOld(i,3))
                  s_old1 = stressOld(i,1) + sig_mean
                  s_old2 = stressOld(i,2) + sig_mean
                  s_old3 = stressOld(i,3) + sig_mean
```

```
                s_old4 = stressOld(i,4)
                s_old5 = stressOld(i,5)
                s_old6 = stressOld(i,6)
C Deviatoric elastic strain increments
                deel1 = (s1_new - s_old1)/(two*G)
                deel2 = (s2_new - s_old2)/(two*G)
                deel3 = (s3_new - s_old3)/(two*G)
                deel4 = (s4_new - s_old4)/(two*G)
                deel5 = (s5_new - s_old5)/(two*G)
                deel6 = (s6_new - s_old6)/(two*G)
C Deviatoric plastic strain increments
                deepl1 = de1 - deel1
                deepl2 = de2 - deel2
                deepl3 = de3 - deel3
                deepl4 = de4 - deel4
                deepl5 = de5 - deel5
                deepl6 = de6 - deel6
C Equivalent deviatoric plastic strain increment
                depl = twoThirds*( deepl1**2 + deepl2**2 + deepl3**2 +
     1                 two*deepl4**2 + two*deepl5**2 + two*deepl6**2 )
                depl = SQRT(depl)
C Mises equivalent stress
                q_depl = (A + B*((depl + e_pl_old)**xn))*(one-T_star**xm)
     1                        *(one + C*dedot_log)
C Find equivalent deviatoric strains
                ehat1 = s_old1/(two*G) + de1
                ehat2 = s_old2/(two*G) + de2
                ehat3 = s_old3/(two*G) + de3
                ehat4 = s_old4/(two*G) + de4
                ehat5 = s_old5/(two*G) + de5
                ehat6 = s_old6/(two*G) + de6
C The deviatoric stresses are therefore:
                denom = one + three*G*depl/q_depl
                s1 = (two*G / denom) * ehat1
                s2 = (two*G / denom) * ehat2
                s3 = (two*G / denom) * ehat3
                s4 = (two*G / denom) * ehat4
                s5 = (two*G / denom) * ehat5
                s6 = (two*G / denom) * ehat6
C Find Mises yield function to check - this is optional
                q_mises = zero
                q_mises = s1**2 + s2**2 + s3**2 + two*s4**2
     1               + two*s5**2 + two*s6**2
                q_mises = SQRT( threeHalfs * q_mises )
                f_depl = q_mises - q_depl
C Find normals (in direction of deviatoric stresses)
                xnormal1 = threeHalfs*s1/q_depl
                xnormal2 = threeHalfs*s2/q_depl
                xnormal3 = threeHalfs*s3/q_depl
                xnormal4 = threeHalfs*s4/q_depl
                xnormal5 = threeHalfs*s5/q_depl
                xnormal6 = threeHalfs*s6/q_depl
C Find deviatoric plastic strain increment components
C               depl1 = depl*xnormal1
C               depl2 = depl*xnormal2
C               depl3 = depl*xnormal3
C               depl4 = depl*xnormal4
C               depl5 = depl*xnormal5
C               depl6 = depl*xnormal6
C Find new stresses
                s_trial1 = s1 - p
```

```
                  s_trial2 = s2 - p
                  s_trial3 = s3 - p
                  s_trial4 = s4
                  s_trial5 = s5
                  s_trial6 = s6
C Find increase in temperature
                  term = (xneta*depl)/(two*density(i)*s_heat)
                  dtheta = term*( xnormal1*(stressOld(i,1)+s_trial1) +
     1                            xnormal2*(stressOld(i,2)+s_trial2) +
     2                            xnormal3*(stressOld(i,3)+s_trial3) +
     3                            two*xnormal4*(stressOld(i,4)+s_trial4) +
     4                            two*xnormal5*(stressOld(i,5)+s_trial5) +
     5                            two*xnormal6*(stressOld(i,6)+s_trial6) )
C Find new stresses
                  stressNew(i,1) = s_trial1
                  stressNew(i,2) = s_trial2
                  stressNew(i,3) = s_trial3
                  stressNew(i,4) = s_trial4
                  stressNew(i,5) = s_trial5
                  stressNew(i,6) = s_trial6
               END IF
C Update the state variables
         stateNew(i,1) = stateOld(i,1) + depl
         tempNew(i) = tempOld(i) + dtheta
C
C Update the specific internal energy -
         stressPower = zero
         stressPower = half * (  ( stressOld(i,1) + stressNew(i,1) )
     1       *strainInc(i,1) +     ( stressOld(i,2) + stressNew(i,2) )
     2       *strainInc(i,2) +     ( stressOld(i,3) + stressNew(i,3) )
     3       *strainInc(i,3) + two*( stressOld(i,4) + stressNew(i,4) )
     4       *strainInc(i,4) + two*( stressOld(i,5) + stressNew(i,5) )
     5       *strainInc(i,5) + two*( stressOld(i,6) + stressNew(i,6) )
     6       *strainInc(i,6) )
C
         enerInternNew(i) = enerInternOld(i) + stressPower/density(i)
C
C Update the dissipated inelastic specific energy -
         smean = zero
         smean = third * ( stressNew(i,1) + stressNew(i,2)
     1                   + stressNew(i,3) )
         equivStress = threeHalfs *
     1     ( (stressNew(i,1)-smean)**2
     2     + (stressNew(i,2)-smean)**2
     3     + (stressNew(i,3)-smean)**2
     4     + two * stressNew(i,4)**2
     5     + two * stressNew(i,5)**2
     6     + two * stressNew(i,6)**2 )
         IF (equivStress .ne. zero) THEN
           equivStress = SQRT( equivStress )
         ENDIF
         plasticWorkInc = equivStress * depl
         enerInelasNew(i) = enerInelasOld(i) + plasticWorkInc/density(i)
         END IF
C       WRITE (unit=110, fmt=100) i,beta,f_depl,depl
       END DO
  100 FORMAT (I4,f16.10,f16.4,f16.10)
C
       RETURN
       END
```

B.6 Example Input Deck for Combined Loading - Single Element

```
*Heading
** Job name: Combined Model name: JC-complete-combined
**
** PARTS
**
*Part, name=Cube
*End Part
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Cube-1, part=Cube
*Node
      1,          0.01,           0.,          0.01
      2,            0.,           0.,          0.01
      3,          0.01,           0.,            0.
      4,            0.,           0.,            0.
      5,          0.01,          0.01,          0.01
      6,            0.,          0.01,          0.01
      7,          0.01,          0.01,            0.
      8,            0.,          0.01,            0.
*Element, type=C3D8R
1, 5, 6, 8, 7, 1, 2, 4, 3
** Region: (Solid:Picked)
*Elset, elset=_I1, internal
 1,
** Section: Solid
*Solid Section, elset=_I1, material=JC-complete-copper
1.,
*End Instance
*Nset, nset=_PickedSet8, internal, instance=Cube-1, generate
 1,  8,  1
*Elset, elset=_PickedSet8, internal, instance=Cube-1
 1,
*Nset, nset=_PickedSet13, internal, instance=Cube-1
 1, 2, 5, 6
*Elset, elset=_PickedSet13, internal, instance=Cube-1
 1,
*Nset, nset=_PickedSet14, internal, instance=Cube-1
 3, 4, 7, 8
*Elset, elset=_PickedSet14, internal, instance=Cube-1
 1,
*Nset, nset=_PickedSet31, internal, instance=Cube-1
 1, 2, 5, 6
*Elset, elset=_PickedSet31, internal, instance=Cube-1
 1,
*Nset, nset=_PickedSet32, internal, instance=Cube-1
 3, 4, 7, 8
*Elset, elset=_PickedSet32, internal, instance=Cube-1
 1,
*End Assembly
**
** MATERIALS
**
*Material, name=JC-complete-copper
*Density
8960.,
```

```
*Depvar
      2,
*User Material, constants=11
 7e+10,       0.3,     9e+07, 2.92e+08,      0.31,      1.09,      300.,
1356.
      0.9,      383.,     0.025
*Material, name=JC-complete-iron
*Density
7890.,
*Depvar
      2,
*User Material, constants=11
 2e+11,       0.3, 1.75e+08,  3.8e+08,      0.32,      0.55,      300.,
1811.
      0.9,      452.,      0.06
**
** FIELDS
**
** Name: Temperature   Type: Temperature
*Initial Conditions, type=TEMPERATURE
_PickedSet8, 300.
** ----------------------------------------------------------------
**
** STEP: Load
**
*Step, name=Load
*Dynamic, Explicit
, 0.001
*Bulk Viscosity
0.06, 1.2
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet13, 3, 3, 10.
** Name: BC-2 Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet14, 3, 3, -10.
** Name: BC-3 Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet31, 2, 2, 10.
** Name: BC-4 Type: Velocity/Angular velocity
*Boundary, type=VELOCITY
_PickedSet32, 2, 2, -10.
**
** FIELDS
**
** Name: Temperature   Type: Temperature
*Temperature
_PickedSet8, 300.
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=1, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, number intervals=200
*Node Output
U, V, A
```

```
*Element Output
S, NE, LE, ELEN, TEMP, SDV
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

B.7 EXAMPLE INPUT DECK FOR TAYLOR TEST

```
*Heading
** Job name: Impact Model name: Cynlinder-user
**
** PARTS
**
*Part, name=Cylinder
*End Part
*Part, name=Rigid
*End Part
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Rigid-1, part=Rigid
** Region: (Rigid:Picked)
*Element, type=MASS, elset=_I1_MASS_
1, 1
*Mass, elset=_I1_MASS_
10.,
*Node
      1,           0.,           0.,           0.
*Nset, nset=Rigid-1-RefPt_, internal
1,
*Surface, type=REVOLUTION, name=Rigid
START,           0.,           0.
 LINE,         0.012,           0.
*Rigid Body, ref node=Rigid-1-RefPt_, analytical surface=Rigid
*End Instance
**
*Instance, name=Cylinder-1, part=Cylinder
          0.,         0.001,           0.
          0.,         0.001,           0.,          -1.,         0.001,
0.,           90.
*Node
      1,           0.,      0.00381,     0.006048
      2,           0.,      0.00381,       0.0126
      .
      .     Details removed due to space constraints
      .
   1949,  0.002250362, 0.0008357069,     0.005544
   1950,  0.002318396, 0.0004192016,     0.005544
*Element, type=C3D8R
      1,   91,  412, 1146,  402,    1,   22,  282,   55
      2,  412,  413, 1147, 1146,   22,   23,  283,  282
      .
      .
      .
   1499,  644,  633,  133,  134, 1949, 1950,  890,  889
   1500,  633,  177,   12,  133, 1950, 1120,  248,  890
** Region: (Cylinder:Picked)
*Elset, elset=_I1, internal, generate
    1,  1500,     1
** Section: Cylinder
*Solid Section, elset=_I1, material=JC-complete-iron
1.,
*End Instance
*Nset, nset=_PickedSet8, internal, instance=Cylinder-1, generate
```

```
    1,  1950,     1
*Elset, elset=_PickedSet8, internal, instance=Cylinder-1, generate
    1,  1500,     1
*Nset, nset=_PickedSet9, internal, instance=Cylinder-1, generate
    1,  1950,     1
*Elset, elset=_PickedSet9, internal, instance=Cylinder-1, generate
    1,  1500,     1
*Nset, nset=_PickedSet10, internal, instance=Cylinder-1
    9,   10,   11,   12,   13,   15,   19,   20,   21,   99,  100,  101,
102,  103,  104,  105
      .
      .
      .
 1111, 1112, 1113, 1114, 1115, 1116, 1117, 1118, 1119, 1120
*Elset, elset=_PickedSet10, internal, instance=Cylinder-1
  162,  168,  174,  180,  186,  192,  198,  204,  210,  216,  222,  228,
234,  240,  246,  252
      .
      .
      .
 1458, 1464, 1470, 1476, 1482, 1488, 1494, 1500
*Nset, nset=_PickedSet11, internal, instance=Cylinder-1
    1,    2,    5,    6,   17,   18,   19,   20,   21,   22,   23,   24,
25,   26,   27,   28
      .
      .
      .
 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065
*Elset, elset=_PickedSet11, internal, instance=Cylinder-1
    1,    2,    3,    4,    5,    6,    7,    8,    9,   10,   11,   12,
13,   79,   80,   81
      .
      .
      .
 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140
*Nset, nset=_PickedSet12, internal, instance=Rigid-1
 1,
*Elset, elset=_Cylinder-bot_S4, internal, instance=Cylinder-1, generate
 324,  456,   12
*Elset, elset=_Cylinder-bot_S3, internal, instance=Cylinder-1
  457,  458,  459,  460,  461,  462,  529,  530,  531,  532,  533,  534,
1069, 1070, 1071, 1072
 1073, 1074, 1141, 1142, 1143, 1144, 1145, 1146, 1213, 1214, 1215, 1216,
1217, 1218, 1285, 1286
 1287, 1288, 1289, 1290, 1357, 1358, 1359, 1360, 1361, 1362, 1429, 1430,
1431, 1432, 1433, 1434
*Surface, type=ELEMENT, name=Cylinder-bot
_Cylinder-bot_S4, S4
_Cylinder-bot_S3, S3
*End Assembly
**
** MATERIALS
**
*Material, name=JC-complete-copper
*Density
8960.,
*Depvar
      2,
*User Material, constants=11
 1.24e+11,     0.34,    9e+07, 2.92e+08,     0.31,     1.09,     300.,
1356.
```

```
      0.9,      383.,     0.025
*Material, name=JC-complete-iron
*Density
7890.,
*Depvar
      2,
*User Material, constants=11
 2.07e+11,      0.29, 1.75e+08,  3.8e+08,      0.32,      0.03,      300.,
1811.
      0.9,      452.,      0.06
**
** INTERACTION PROPERTIES
**
*Surface Interaction, name=IntProp-1
*Friction
0.,
*Surface Behavior, pressure-overclosure=HARD
**
** BOUNDARY CONDITIONS
**
** Name: Rigid_fix Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PickedSet12, ENCASTRE
** Name: Sym1 Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PickedSet11, XSYMM
** Name: Sym3 Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PickedSet10, ZSYMM
**
** FIELDS
**
** Name: Temperature   Type: Temperature
*Initial Conditions, type=TEMPERATURE
_PickedSet9, 300.
** Name: Velocity   Type: Velocity
*Initial Conditions, type=VELOCITY
_PickedSet8, 1, 0.
_PickedSet8, 2, -279.
_PickedSet8, 3, 0.
** ----------------------------------------------------------------
**
** STEP: Impact
**
*Step, name=Impact
*Dynamic, Explicit, element by element
, 0.0001
*Bulk Viscosity
0.06, 1.2
**
** FIELDS
**
** Name: Temperature   Type: Temperature
*Temperature
_PickedSet9, 300.
**
** INTERACTIONS
**
** Interaction: Int-1
*Contact Pair, interaction=IntProp-1, mechanical constraint=KINEMATIC
Cylinder-bot, Rigid-1.Rigid
```

```
**
** OUTPUT REQUESTS
**
*Restart, write, number interval=10, time marks=NO
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, number intervals=200
*Node Output
U, V, A, RF, COORD
*Element Output
S, LE, ELEN, TEMP, SDV
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

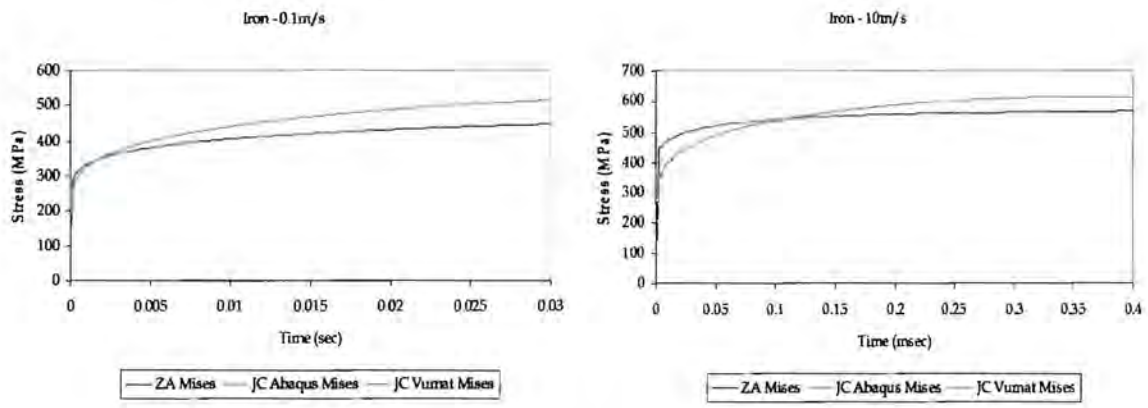
APPENDIX C

ELEMENT TEST RESULTS

C.1 ARMCO-IRON

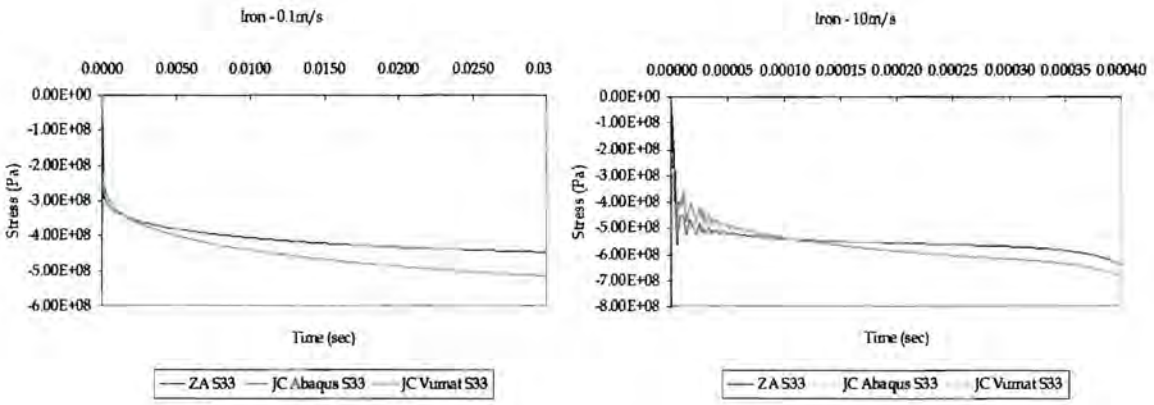
C.1.1 SINGLE ELEMENT

C.1.1.1 COMPRESSION



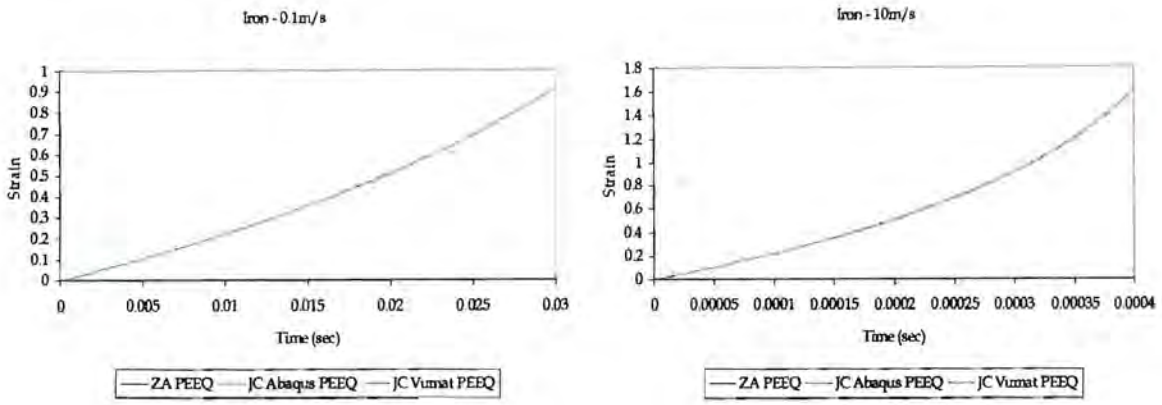
a) b)

Figure C.1: Compression – Mises stress component for Armco-Iron
a) 0.1 m/s, b) 10 m/s.



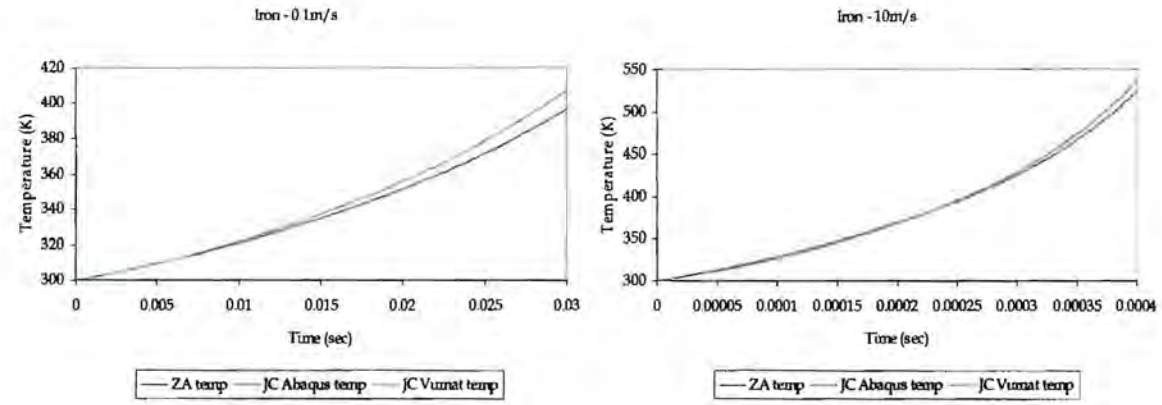
a) b)

Figure C.2: Compression – S33 stress component for Armco-Iron
a) 0.1 m/s, b) 10 m/s.



a) b)

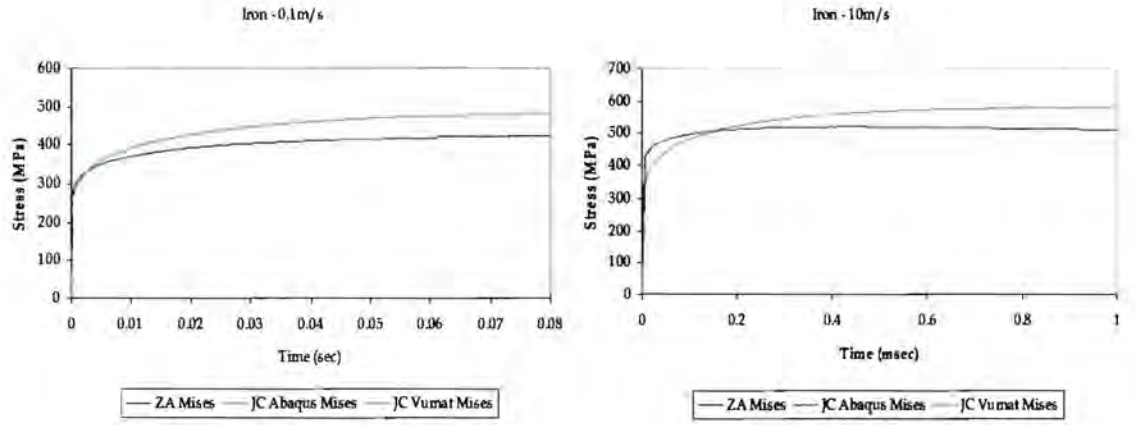
Figure C.3: Compression - PEEQ for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

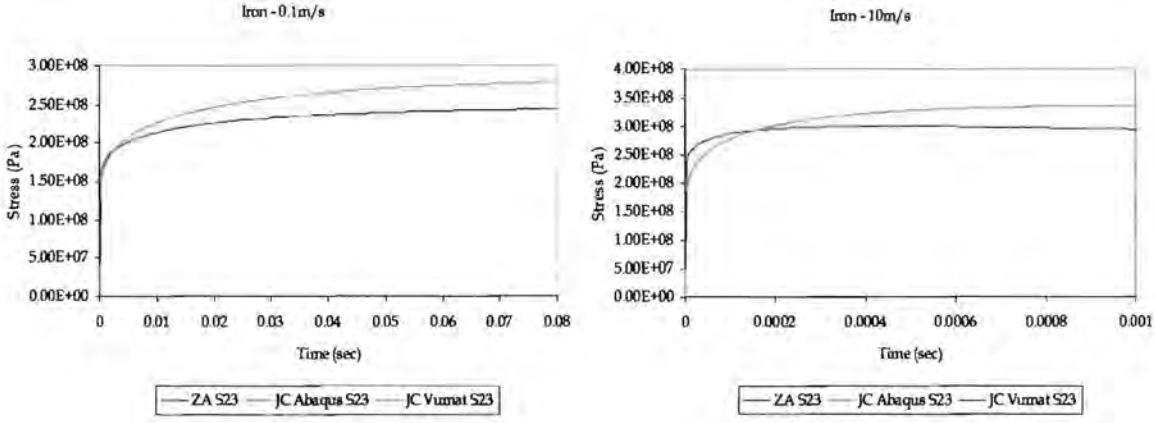
Figure C.4: Compression - temperature for Armco-Iron a) 0.1 m/s, b) 10 m/s.

C.1.2 SHEAR



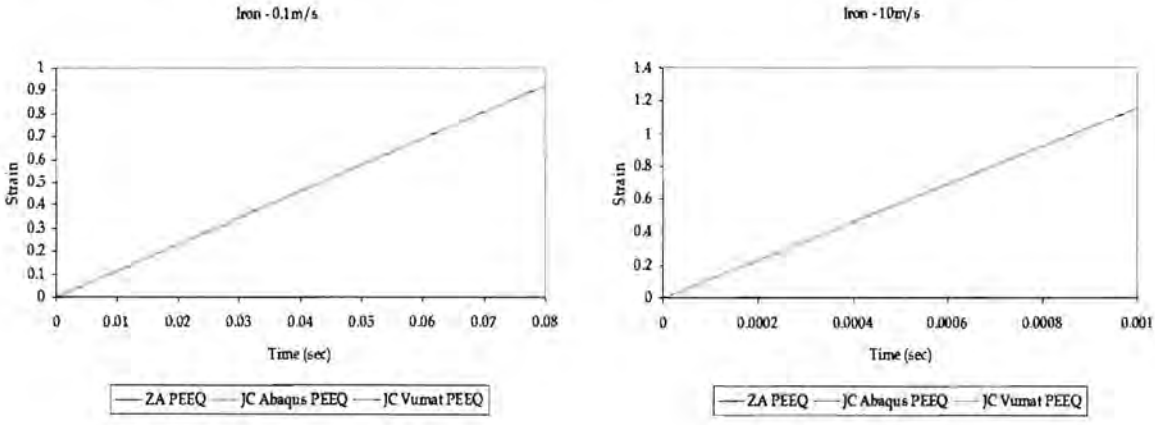
a) b)

Figure C.5: Shear - Mises stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



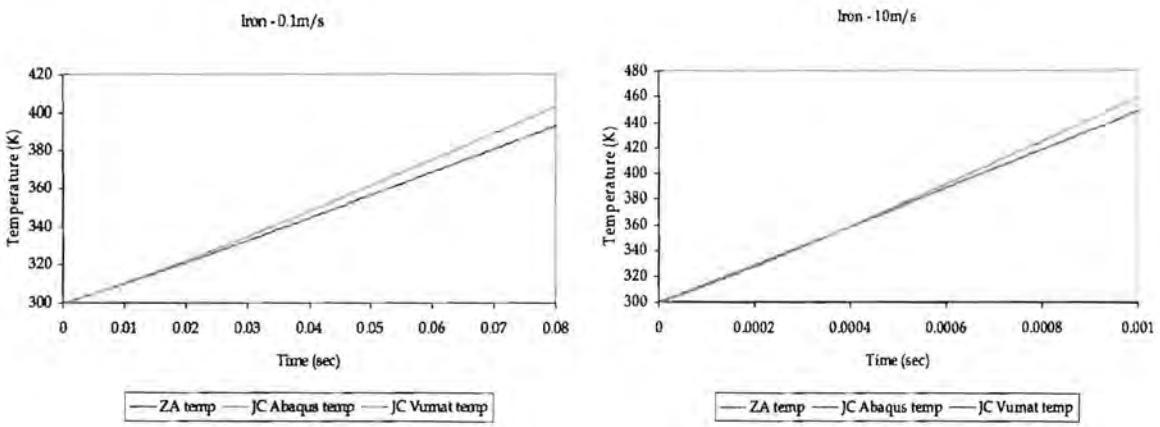
a) b)

Figure C.6: Shear - S23 stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

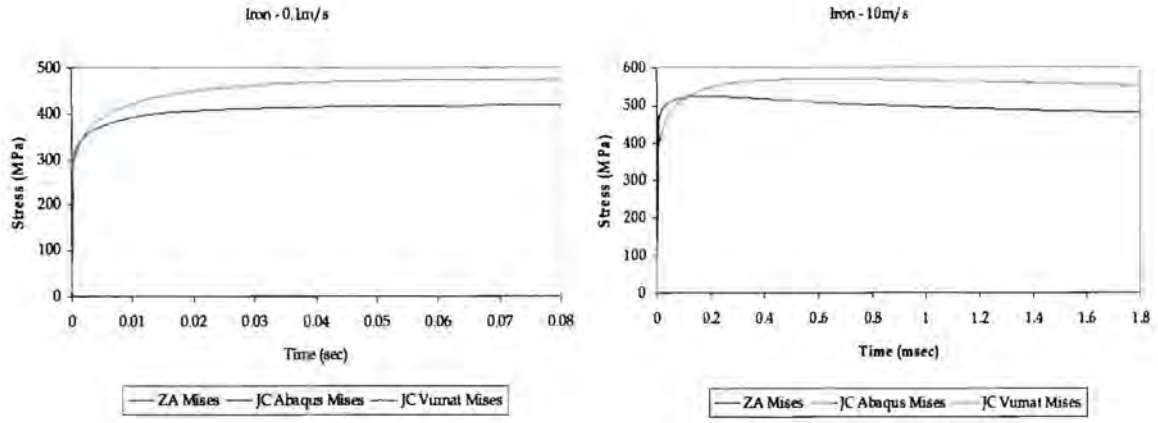
Figure C.7: Shear - PEEQ for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

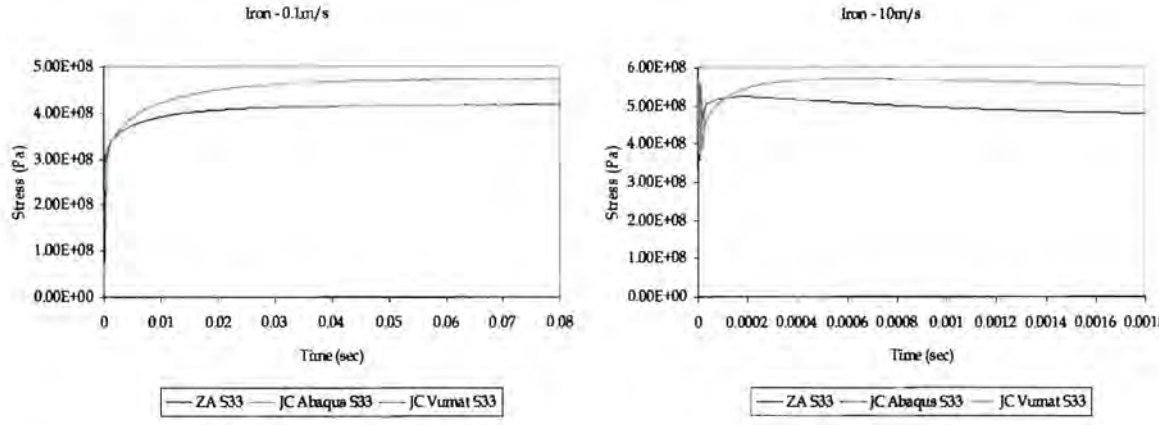
Figure C.8: Shear – temperature for Armco-Iron a) 0.1 m/s, b) 10 m/s.

C.1.1.3 TENSION



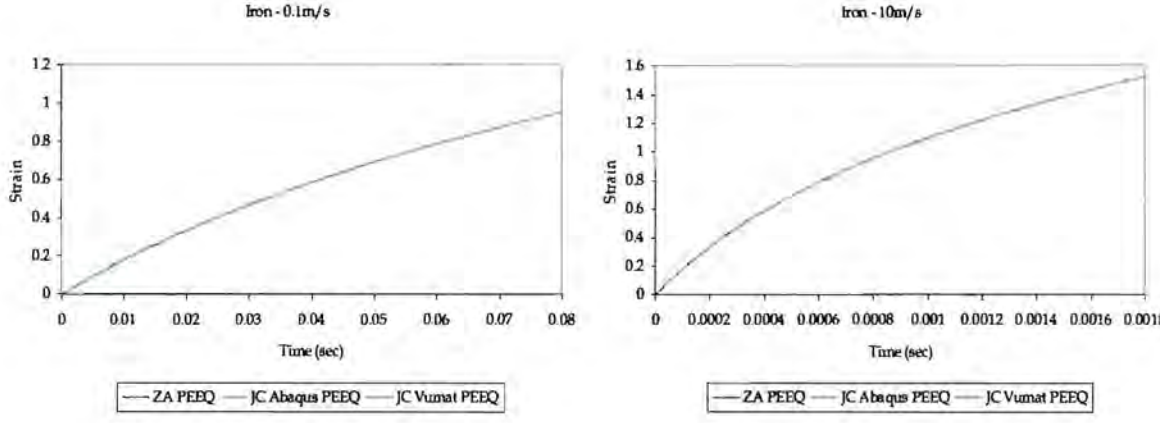
a) b)

Figure C.9: Tension – Mises stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



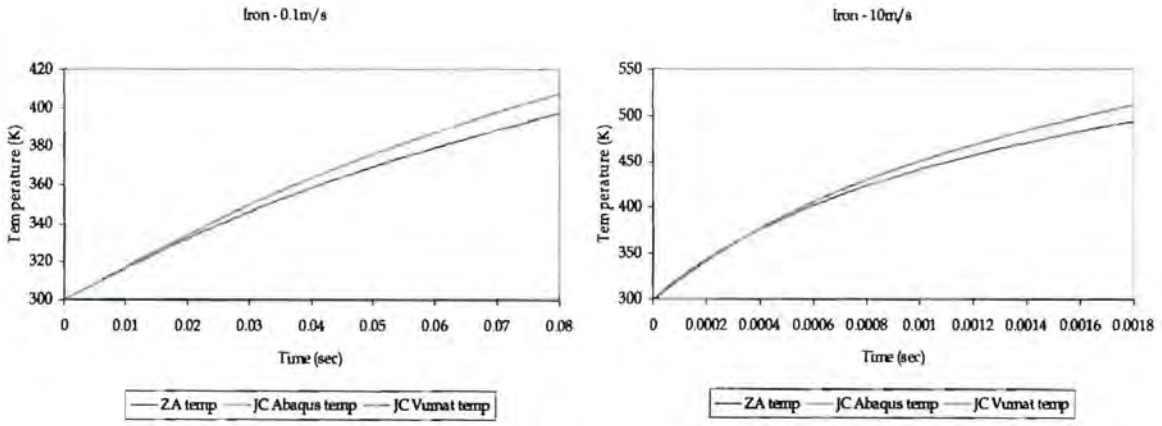
a) b)

Figure C.10: Tension – S33 stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

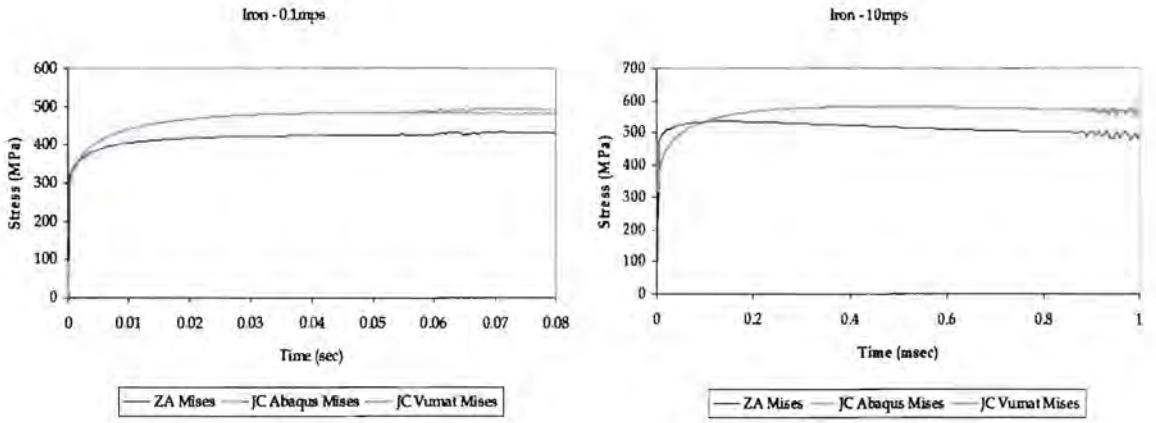
Figure C.11: Tension – PEEQ for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

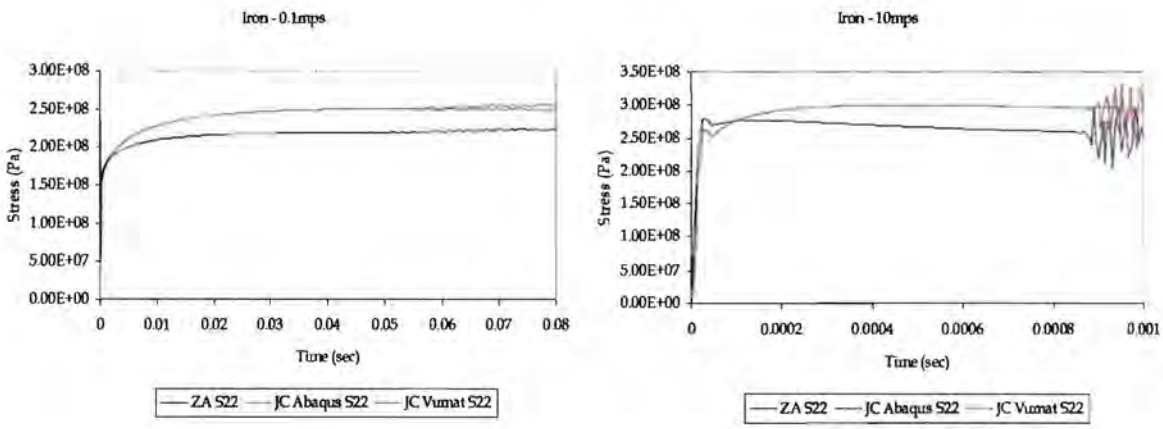
Figure C.12: Tension – temperature for Armco-Iron a) 0.1 m/s, b) 10 m/s.

C.1.1.4 COMBINED



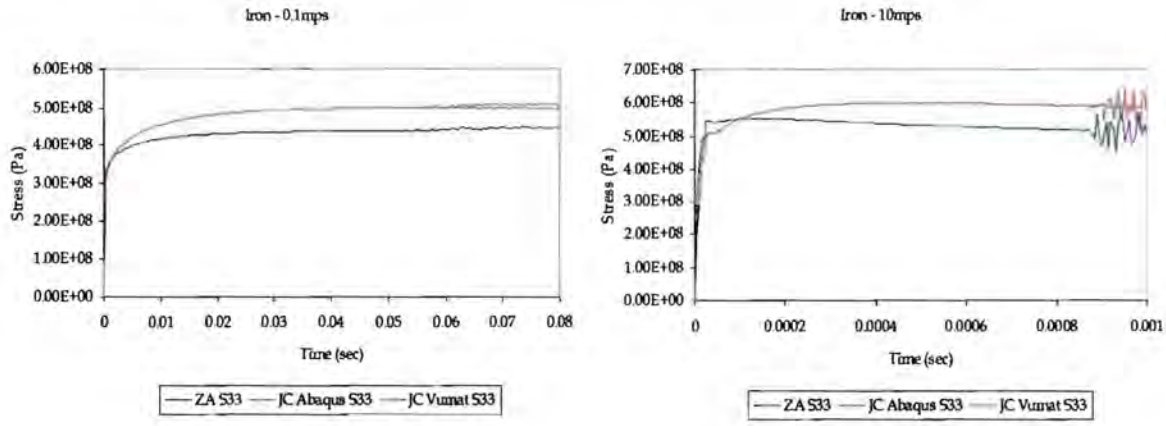
a) b)

Figure C.13: Combined - Mises stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



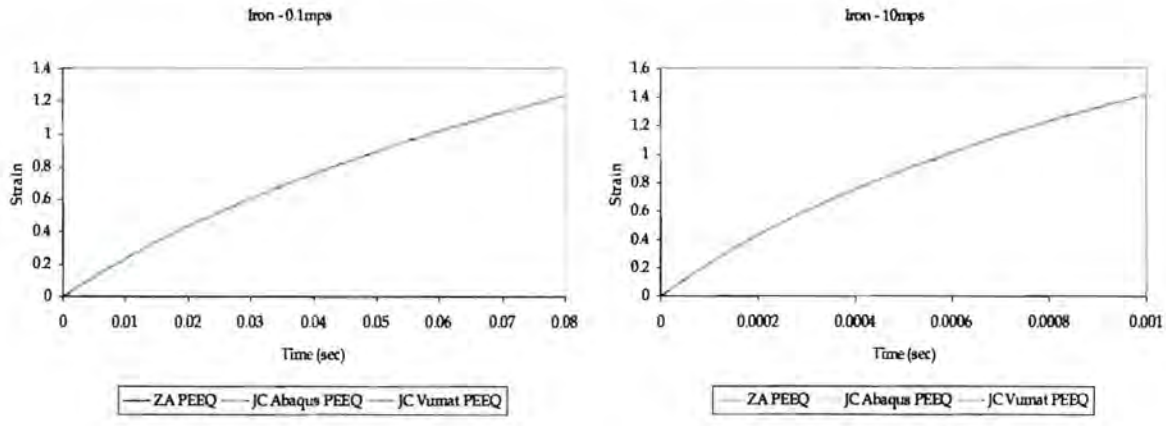
a) b)

Figure C.14: Combined - S22 stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



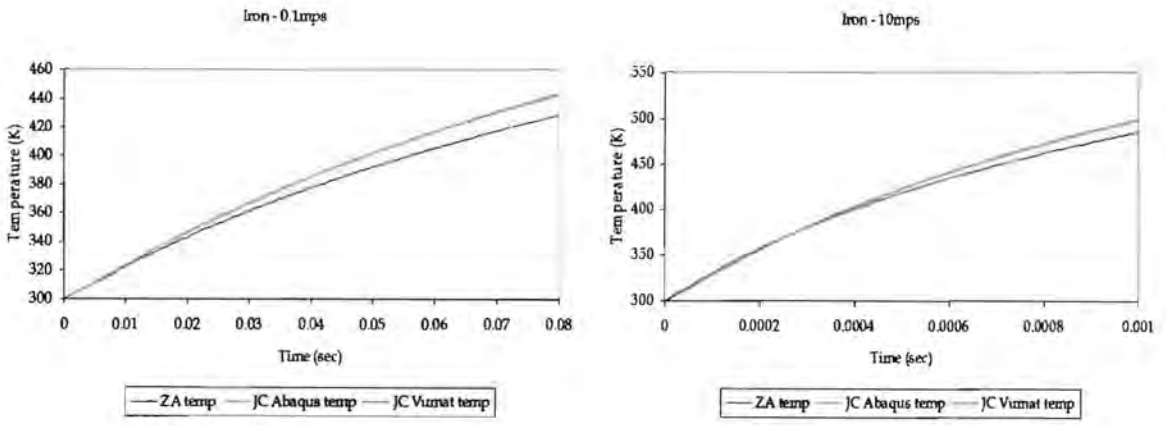
a) b)

Figure C.15: Combined – S33 stress component for Armco-Iron a) 0.1 m/s, b) 10 m/s.



a) b)

Figure C.16: Combined – PEEQ for Armco-Iron a) 0.1 m/s, b) 10 m/s.

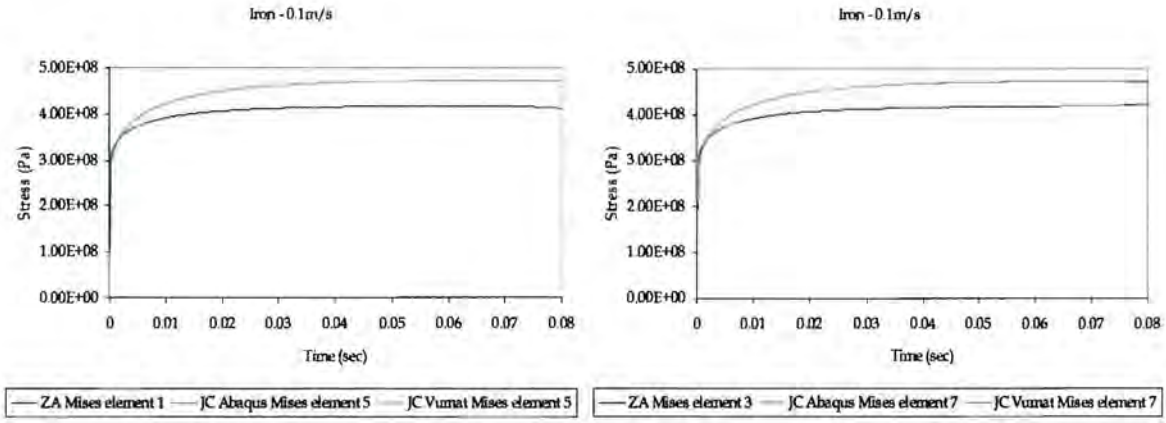


a) b)

Figure C.17: Combined – temperature for Armco-Iron a) 0.1 m/s, b) 10 m/s.

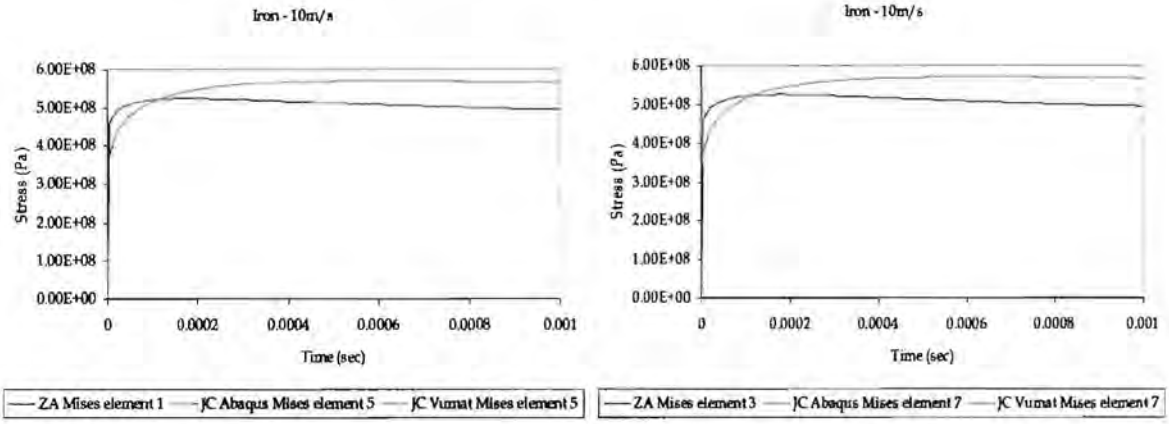
C.1.2 8 (2x2x2) ELEMENTS

C.1.2.1 TENSION



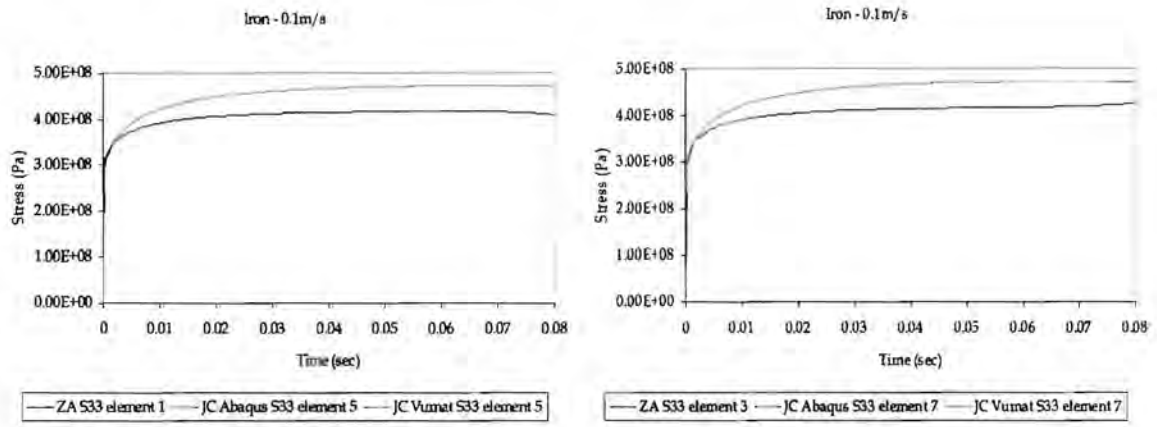
a) b)

Figure C.18: Tension – Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



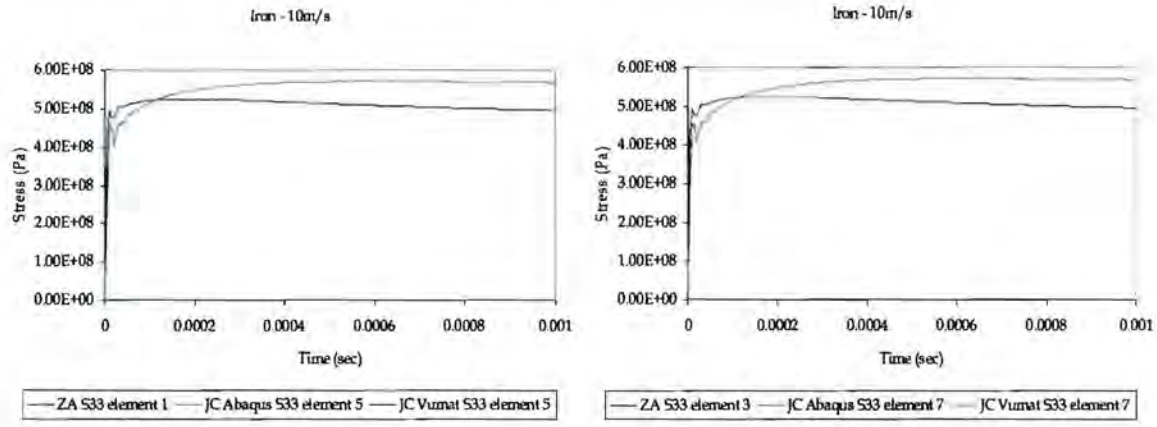
a) b)

Figure C.19: Tension – Mises stress for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



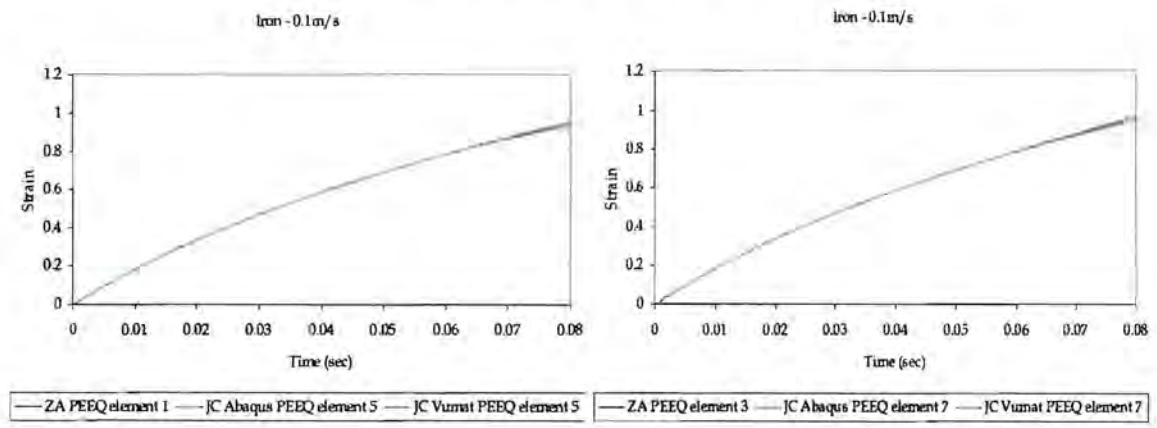
a) b)

Figure C.20: Tension - S33 stress component for Armco-Iron at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



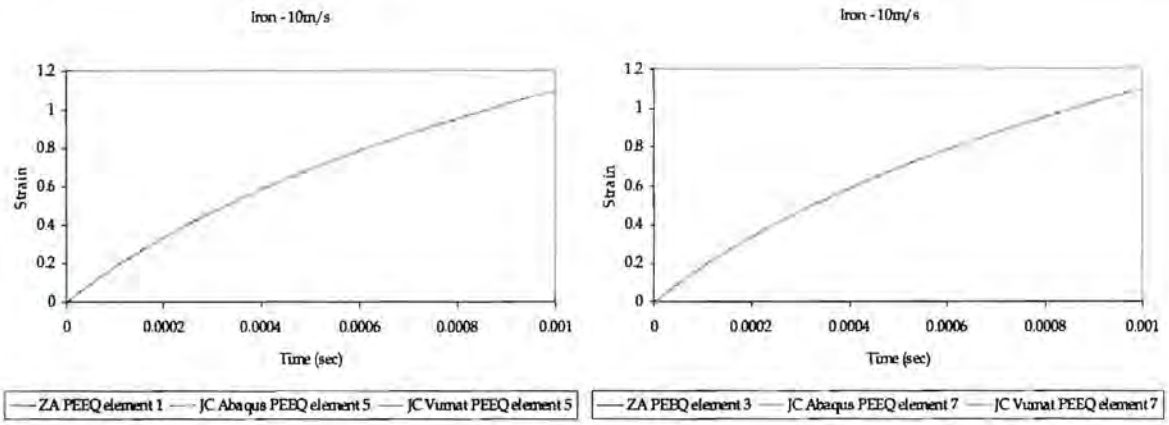
a) b)

Figure C.21: Tension - S33 stress component for Armco-Iron at 10 m/s a) element 1 & 5 and b) element 3 & 7.



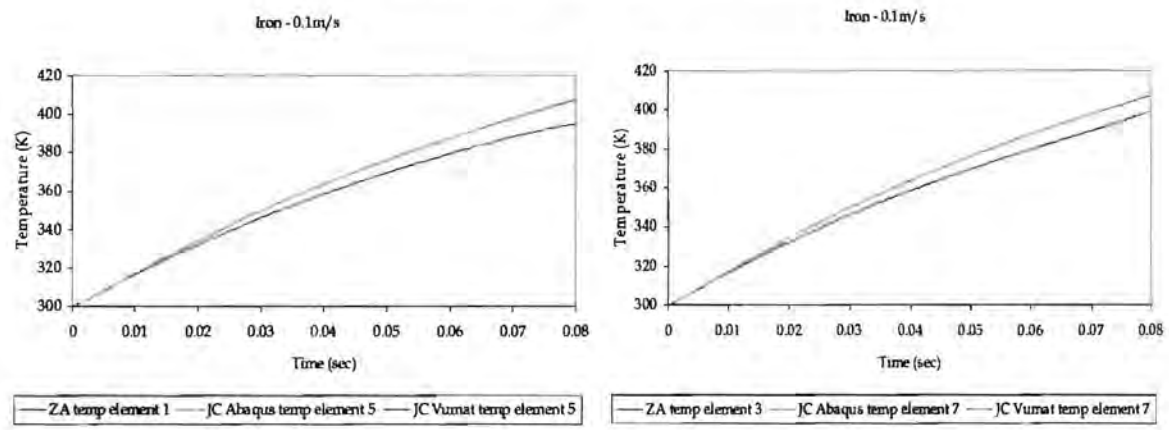
a) b)

Figure C.22: Tension - PEEQ for Armco-Iron at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



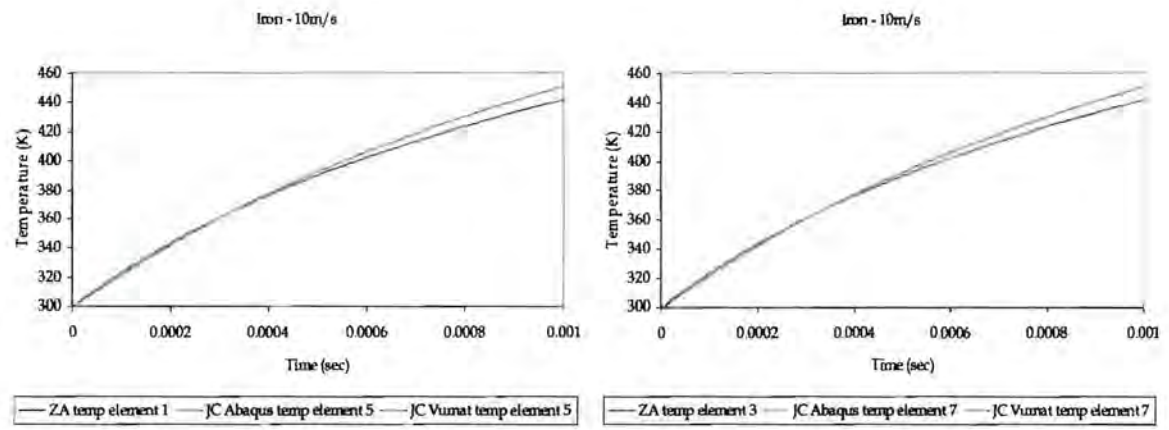
a) b)

Figure C.23: Tension - PEEQ for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



a) b)

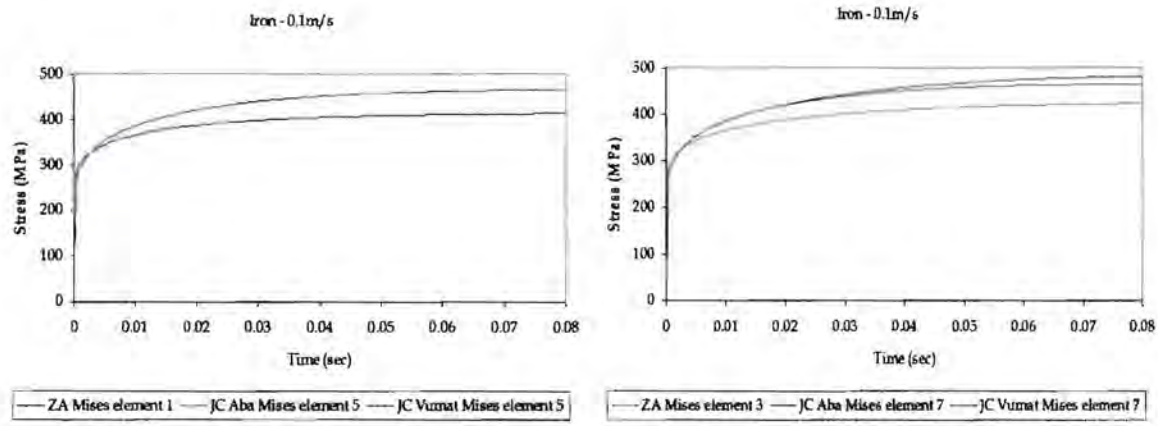
Figure C.24: Tension - temperature for Armco-Iron at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7



a) b)

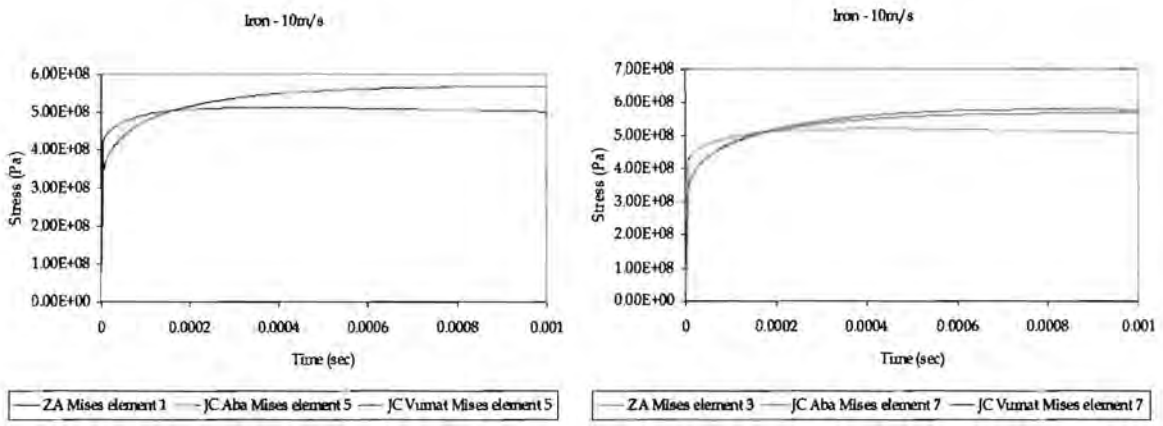
Figure C.25: Tension - temperature for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.

C.1.2.2 SHEAR



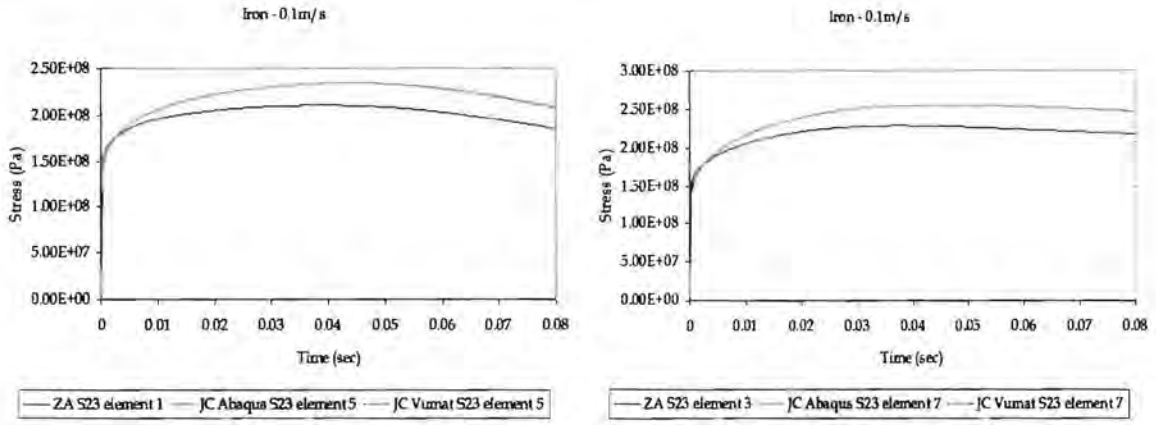
a) b)

Figure C.26: Shear - Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



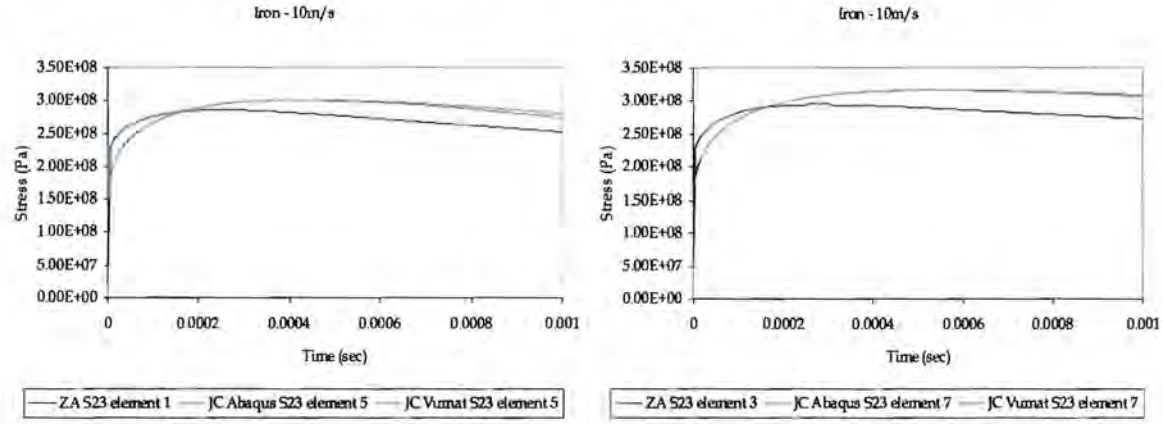
a) b)

Figure C.27: Shear - Mises stress for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



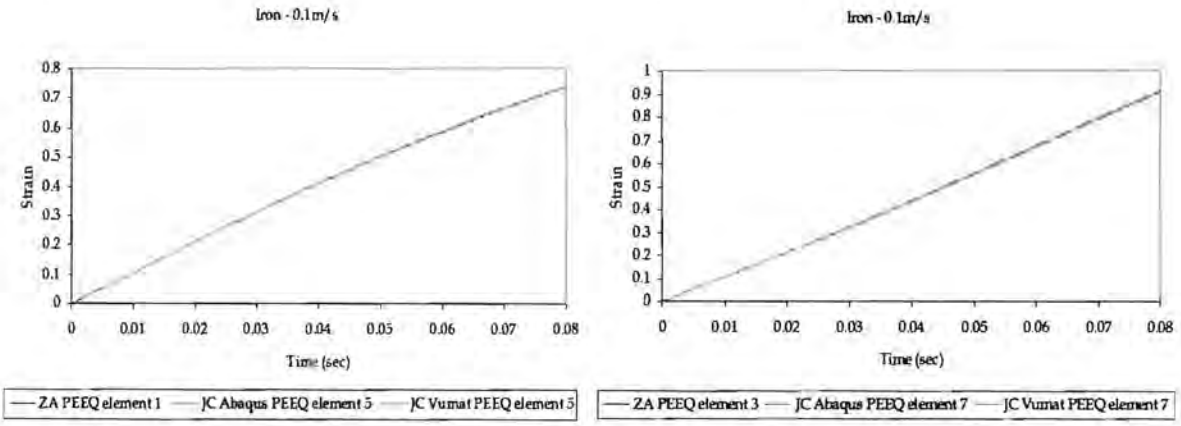
a) b)

Figure C.28: Shear – S23 stress component for Armco-Iron at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



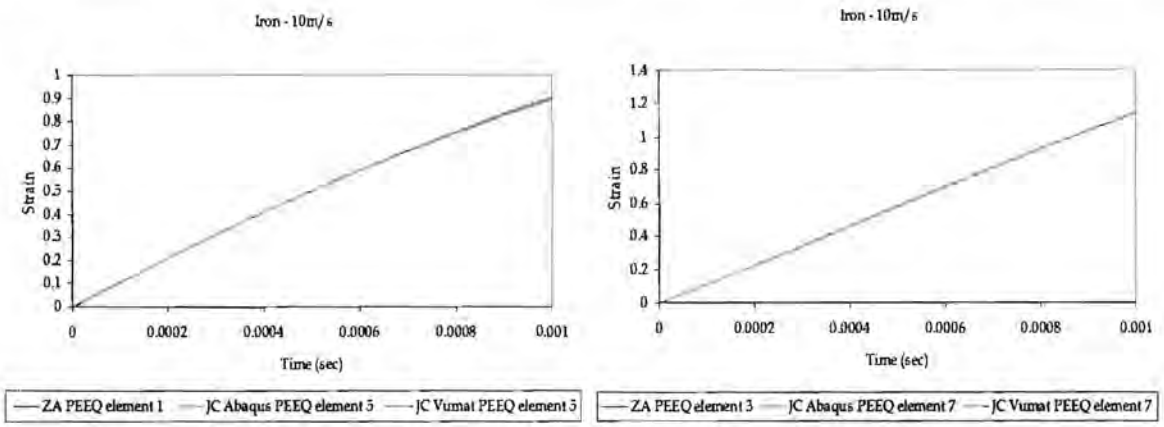
a) b)

Figure C.29: Shear – S23 stress component for Armco-Iron at 10 m/s a) element 1 & 5 and b) element 3 & 7.



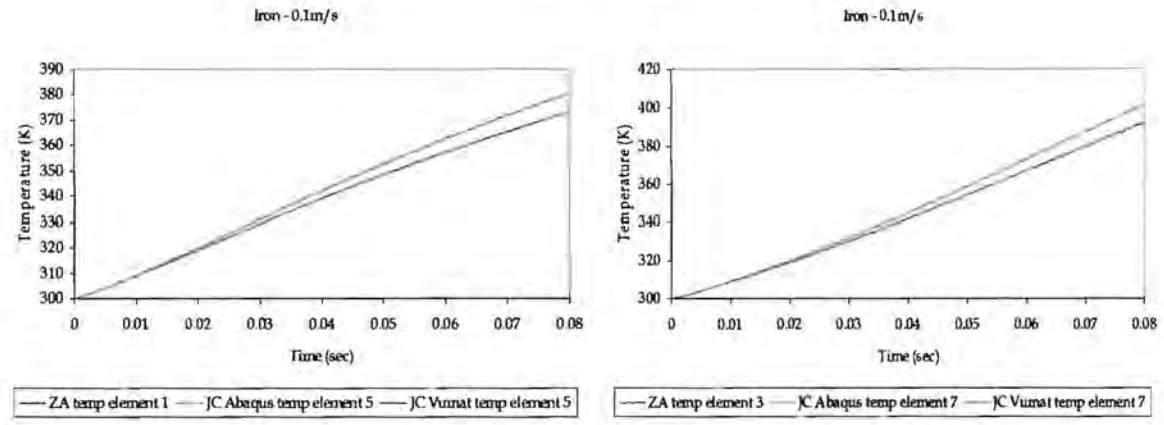
a) b)

Figure C.30: Shear – PEEQ for Armco-Iron at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



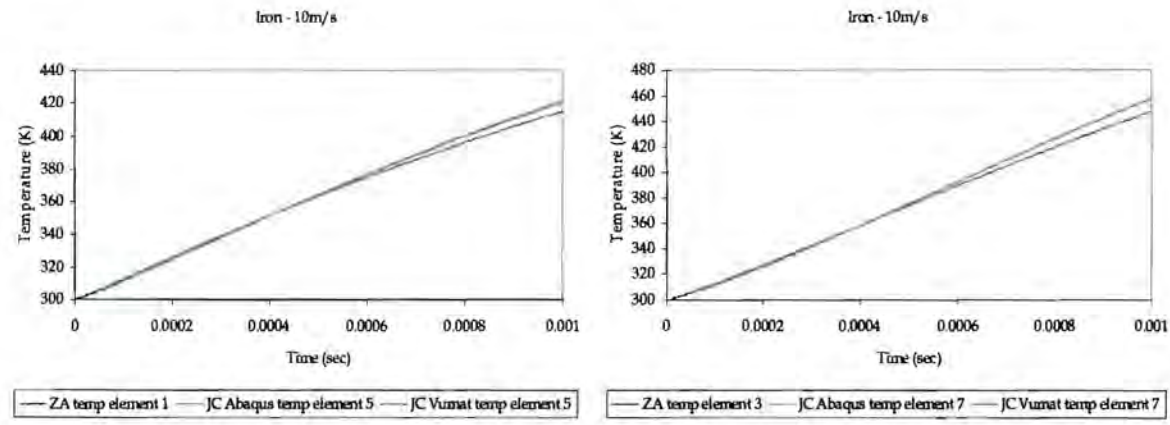
a) b)

Figure C.31: Shear - PEEQ for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



a) b)

Figure C.32: Shear - temperature for Armco-Iron at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.

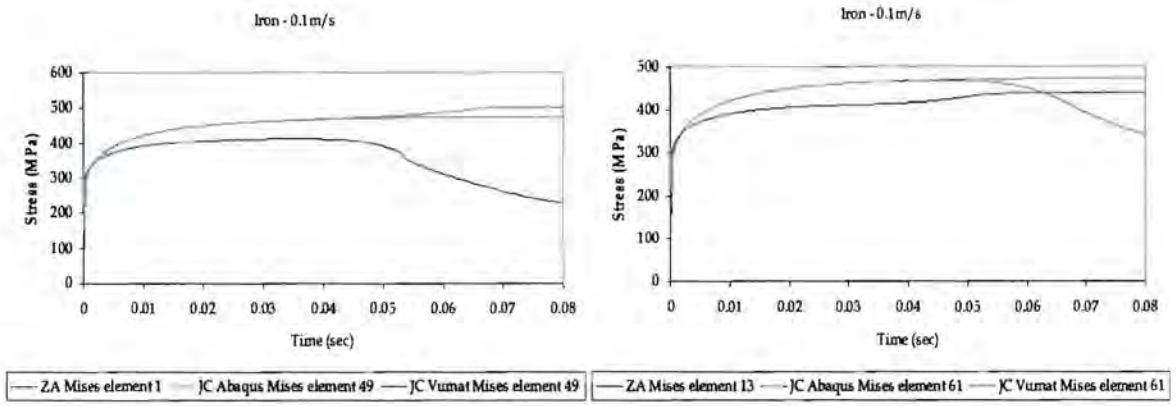


a) b)

Figure C.33: Shear - temperature for Armco-Iron at 10 m/s
a) element 1 & 5 and b) element 3 & 7.

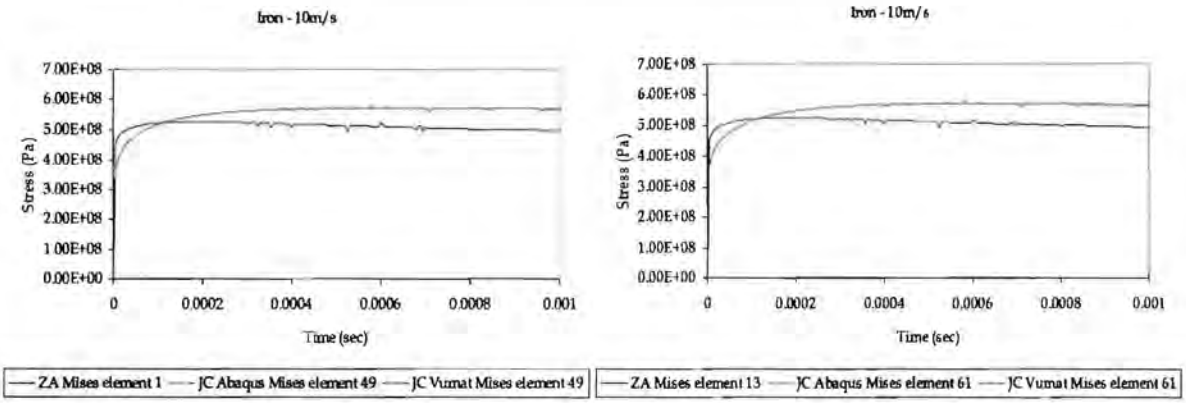
C.1.3 64 (4x4x4) ELEMENTS

C.1.3.1 TENSION



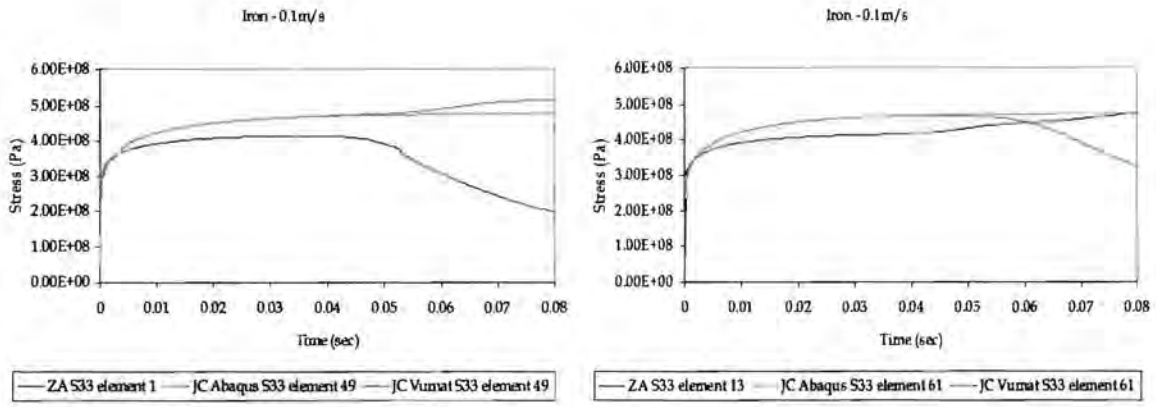
a) b)

Figure C.34: Tension – Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



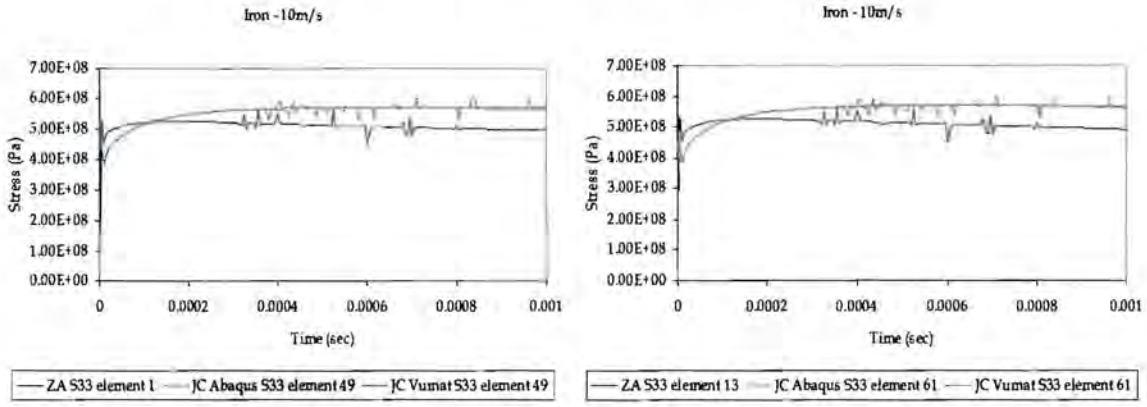
a) b)

Figure C.35: Tension – Mises stress for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



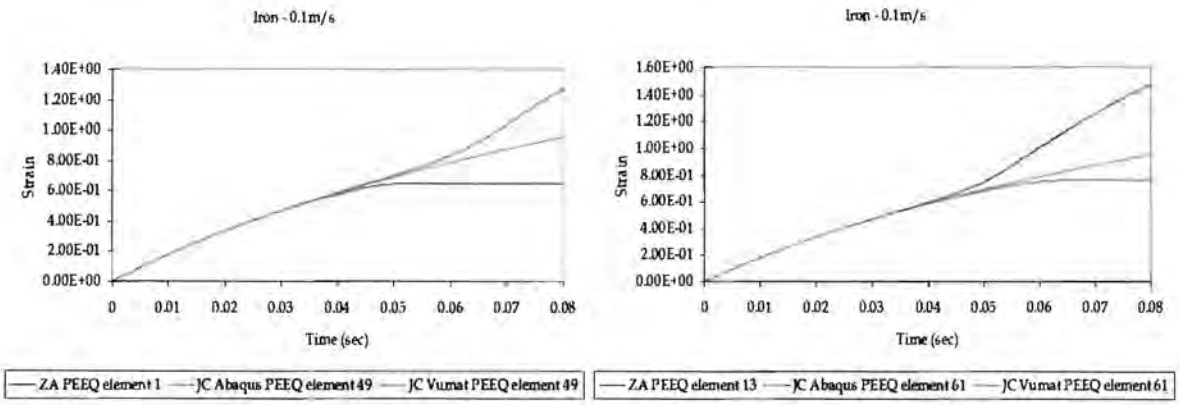
a) b)

Figure C.36: Tension – S33 stress component for Armco-Iron at 0.1 m/s a) element 1 & 49 and b) element 13 & 61.



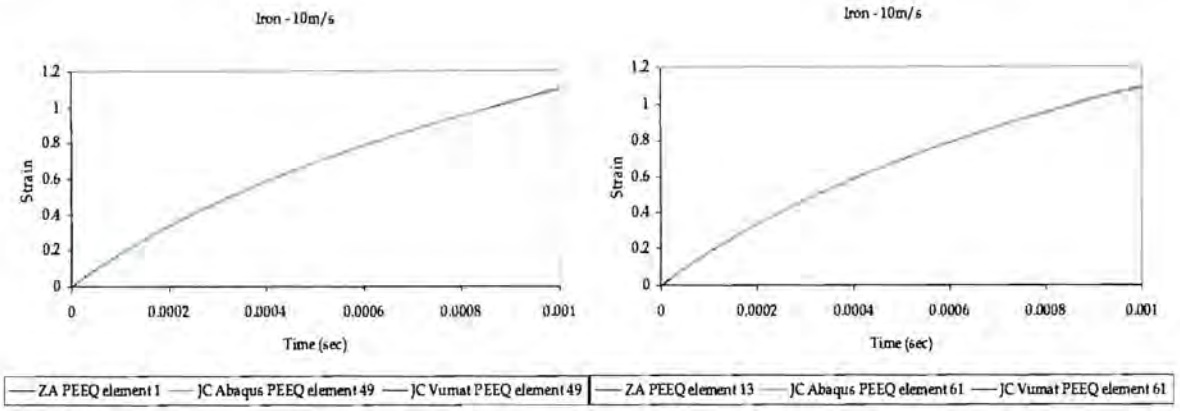
a) b)

Figure C.37: Tension – S33 stress component for Armco-Iron at 10 m/s a) element 1 & 49 and b) element 13 & 61.



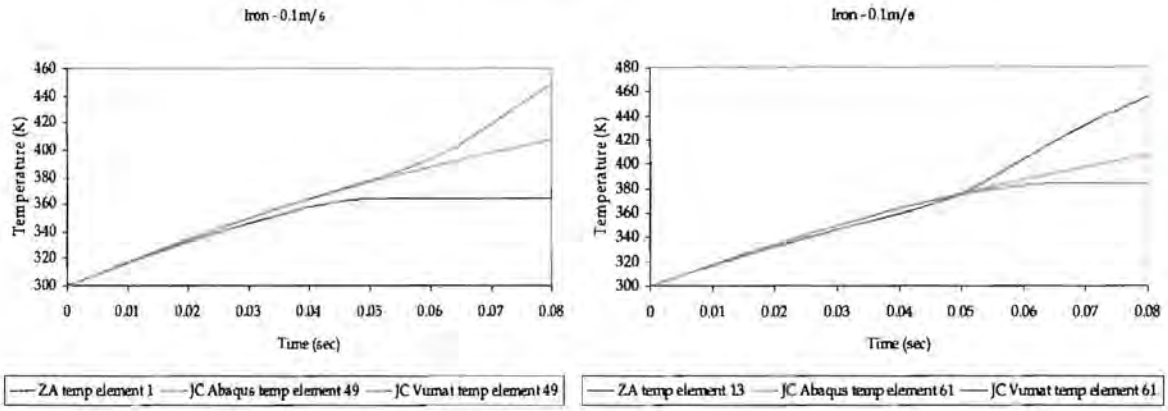
a) b)

Figure C.38: Tension – PEEQ for Armco-Iron at 0.1 m/s a) element 1 & 49 and b) element 13 & 61.



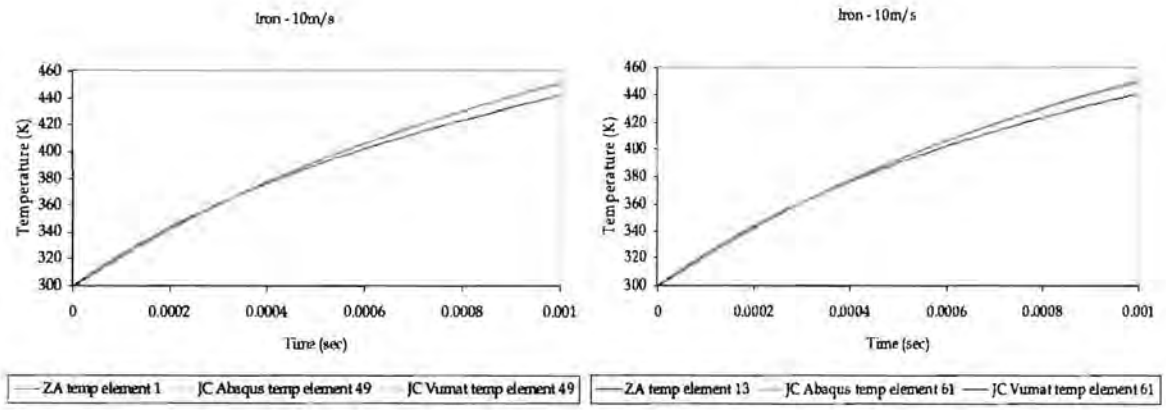
a) b)

Figure C.39: Tension - PEEQ for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

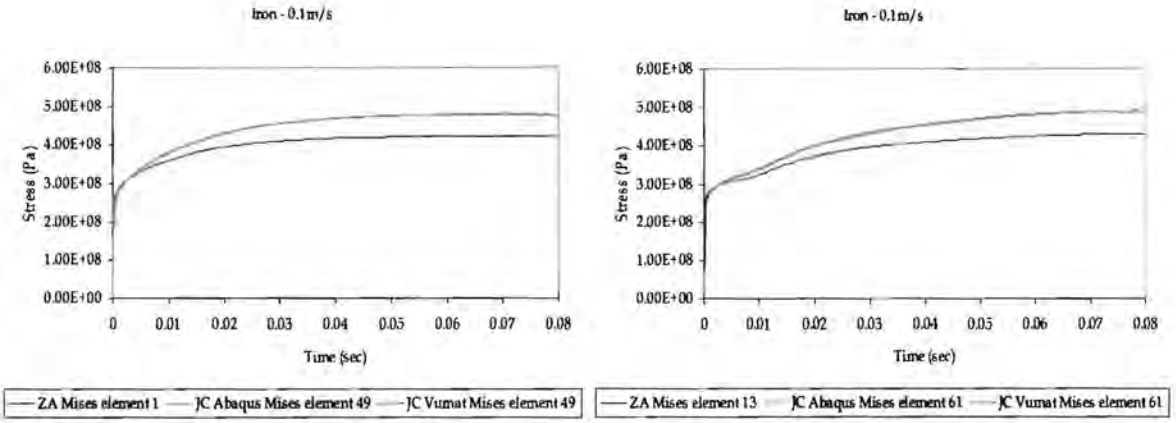
Figure C.40: Tension - temperature for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

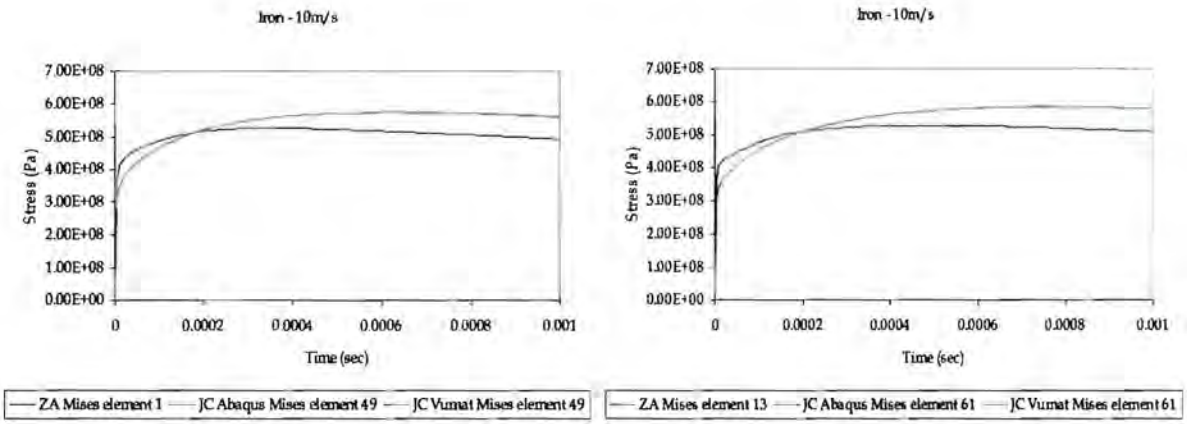
Figure C.41: Tension - temperature for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.

C.1.3.2 SHEAR



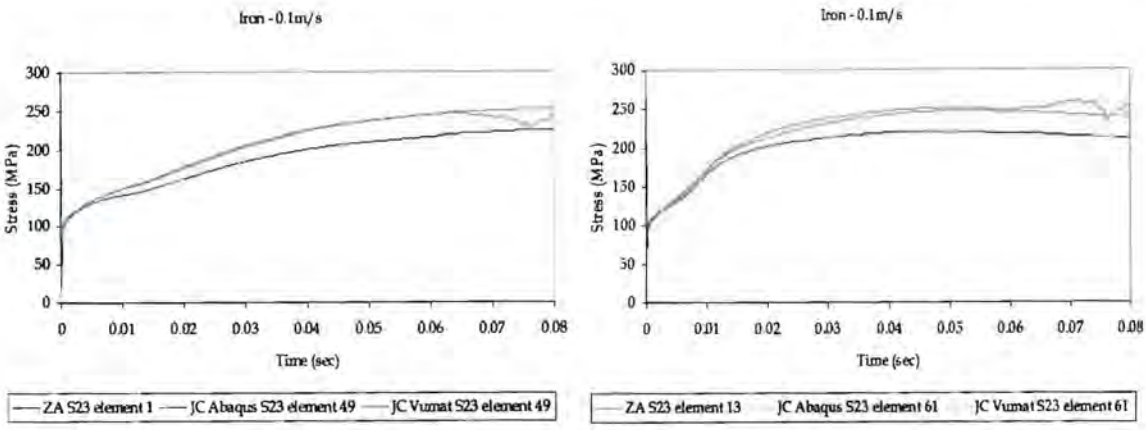
a) b)

Figure C.42: Shear - Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



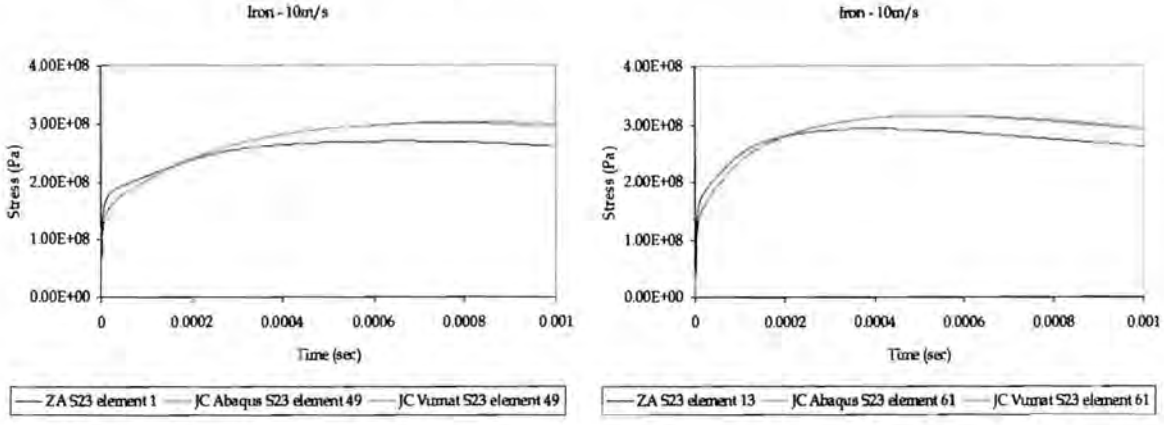
a) b)

Figure C.43: Shear - Mises stress for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61



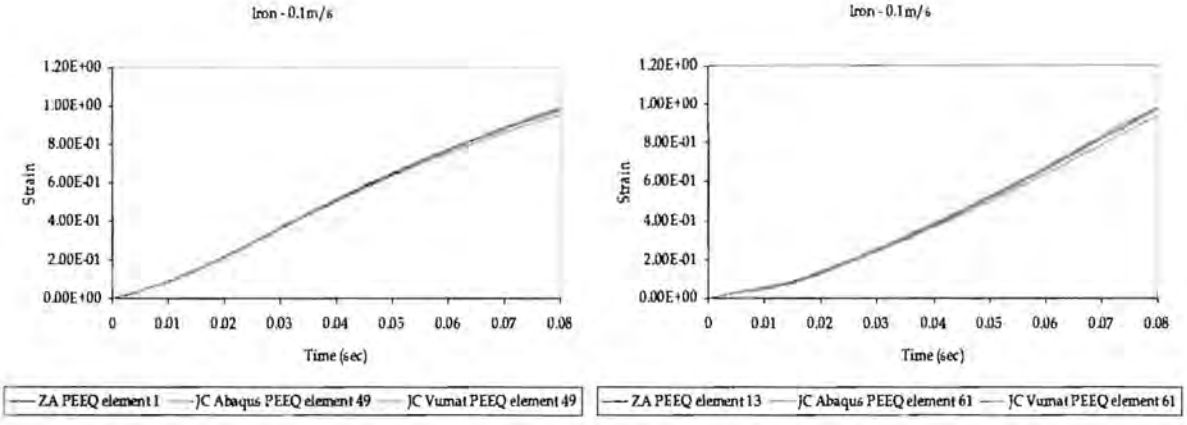
a) b)

Figure C.44: Shear – S23 stress component for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



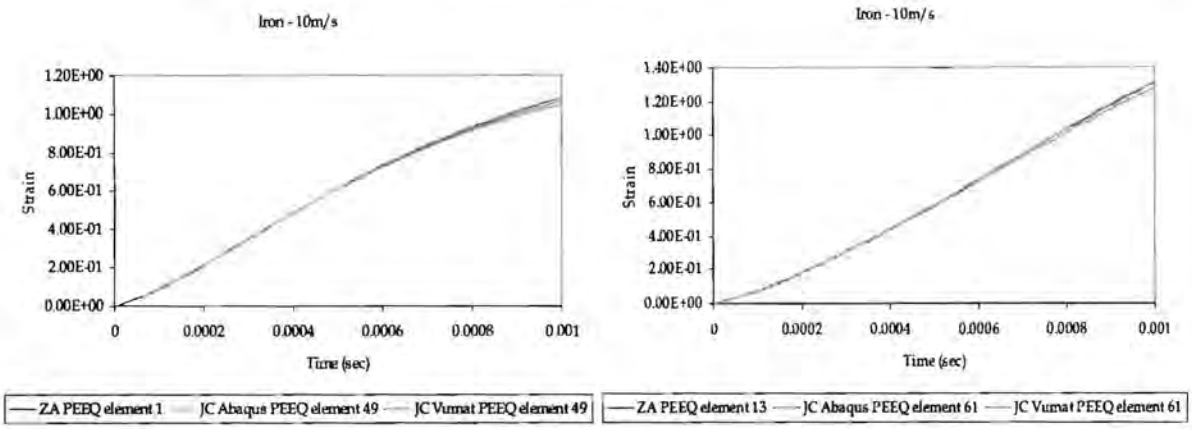
a) b)

Figure C.45: Shear – S23 stress component for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



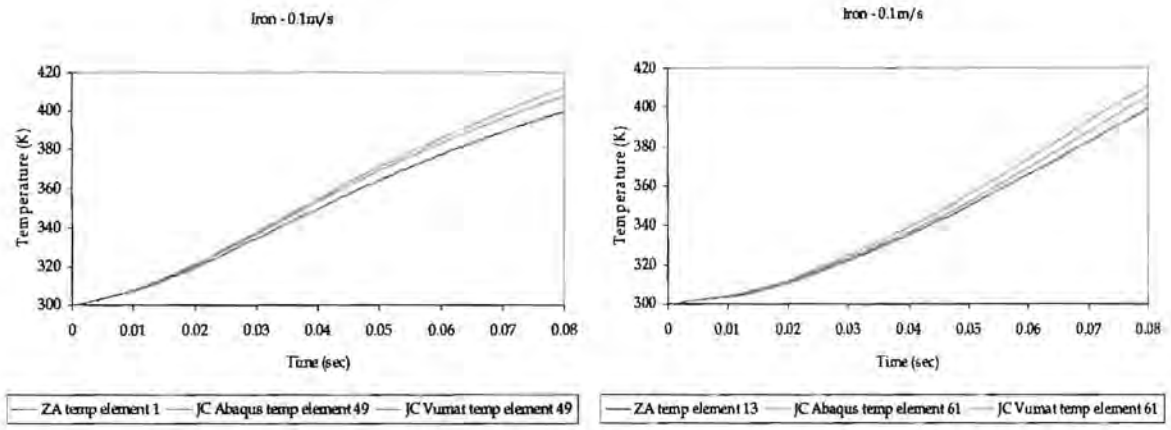
a) b)

Figure C.46: Shear – PEEQ for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



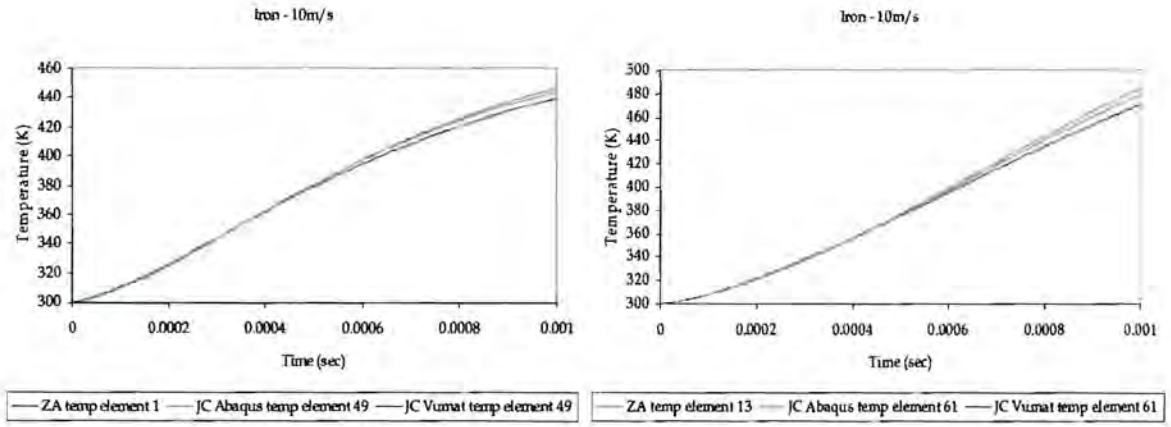
a) b)

Figure C.47: Shear – PEEQ for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

Figure C.48: Shear – temperature for Armco-Iron at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.

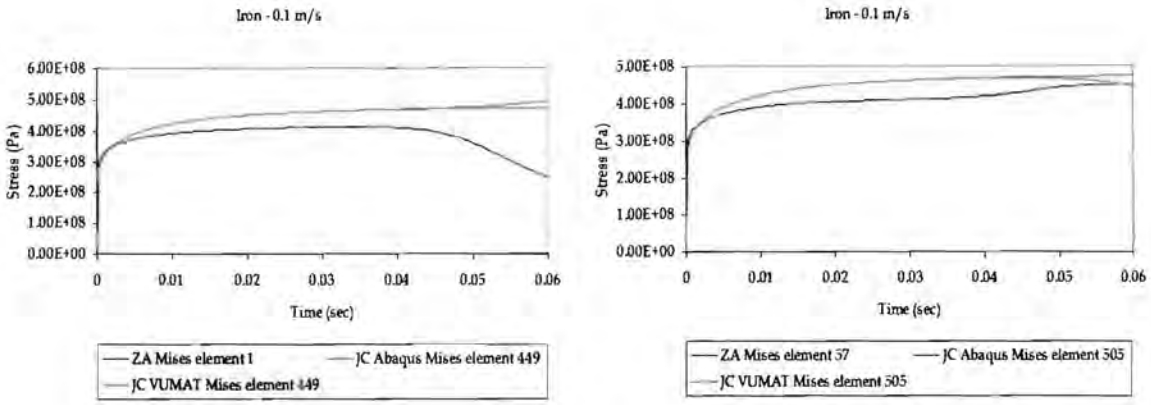


a) b)

Figure C.49: Shear – temperature for Armco-Iron at 10 m/s
a) element 1 & 49 and b) element 13 & 61.

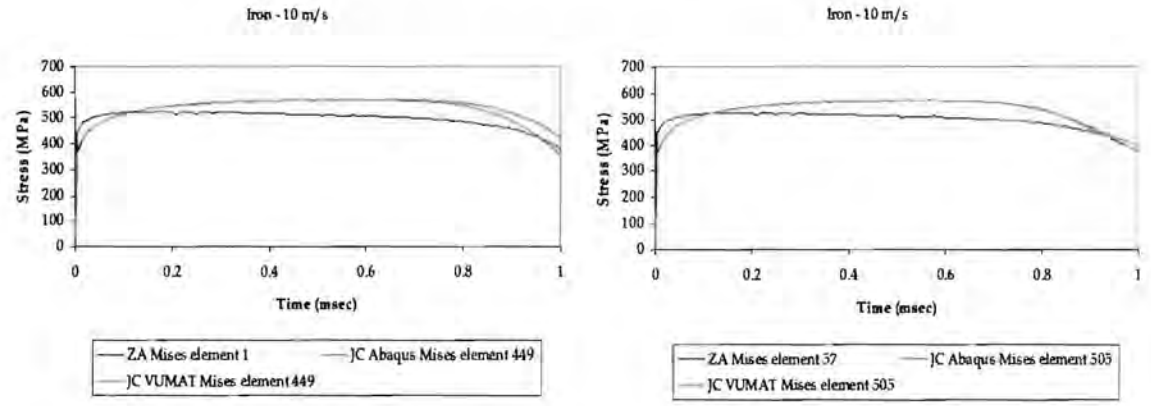
C.1.4 512 (8x8x8) ELEMENTS

C.1.4.1 TENSION



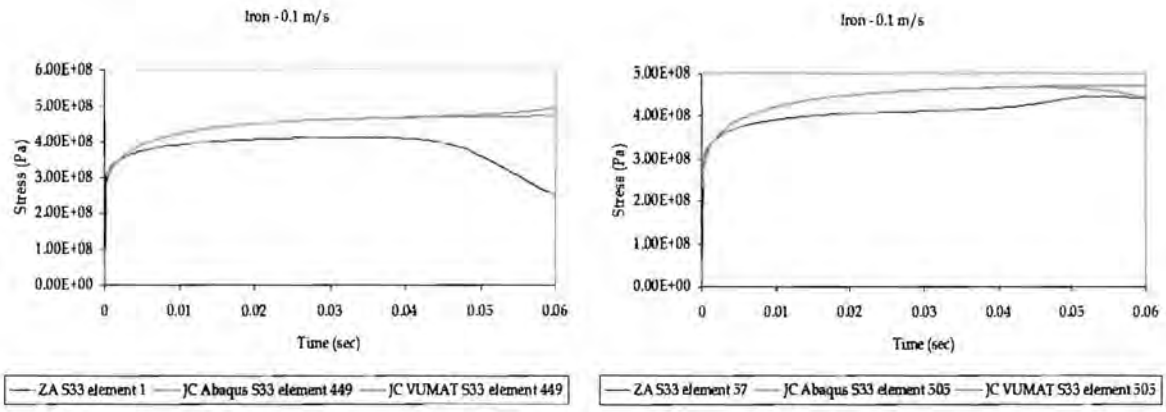
a) b)

Figure C.50: Tension - Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



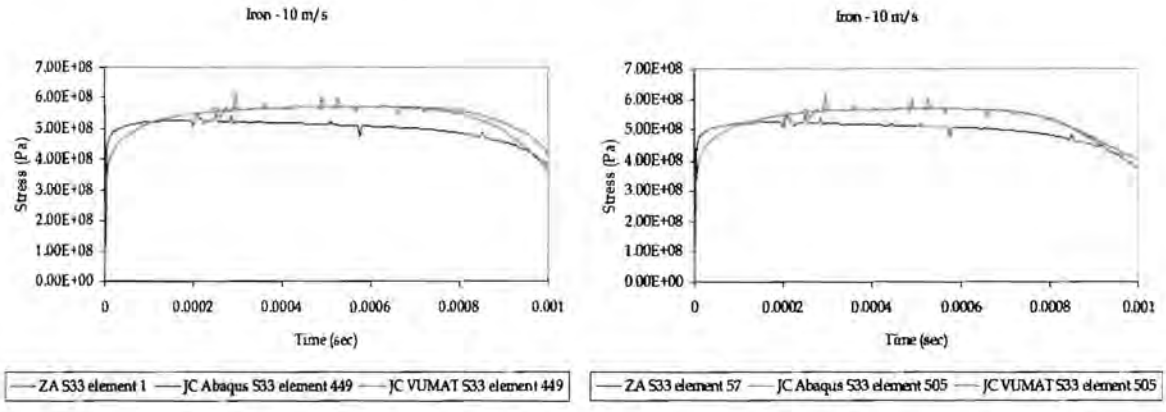
a) b)

Figure C.51: Tension - Mises stress for Armco-Iron at 10 m/s
b) element 1 & 449 and b) element 57 & 505.



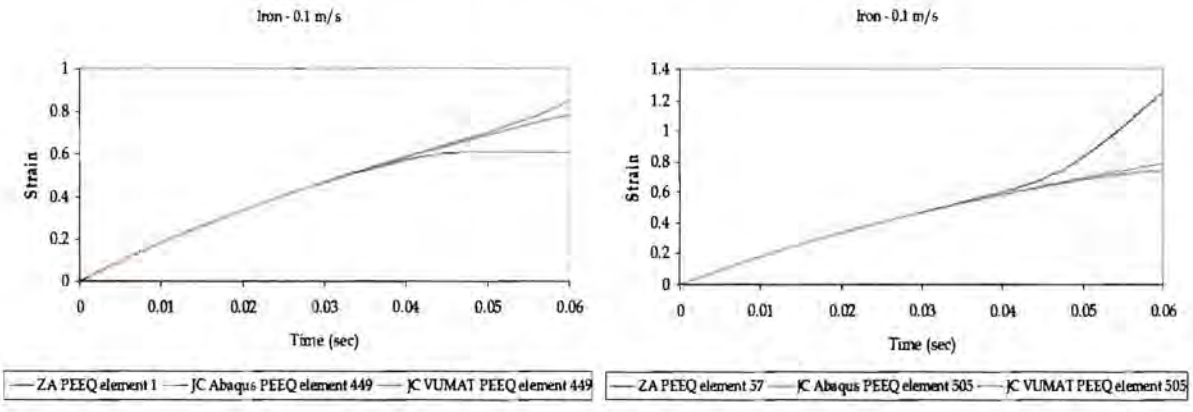
a) b)

Figure C.52: Tension - S33 stress component for Armco-Iron at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



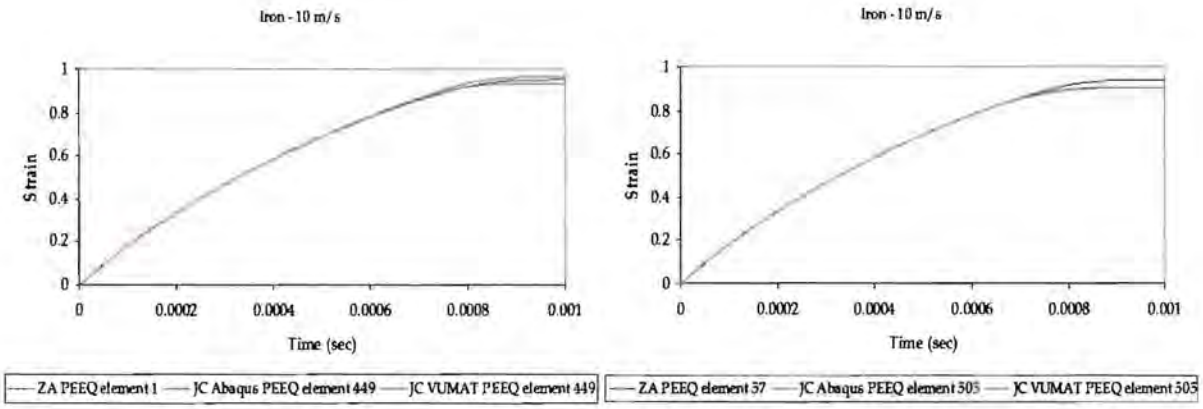
a) b)

Figure C.53: Tension - S33 stress component for Armco-Iron at 10 m/s a) element 1 & 449 and b) element 57 & 505.



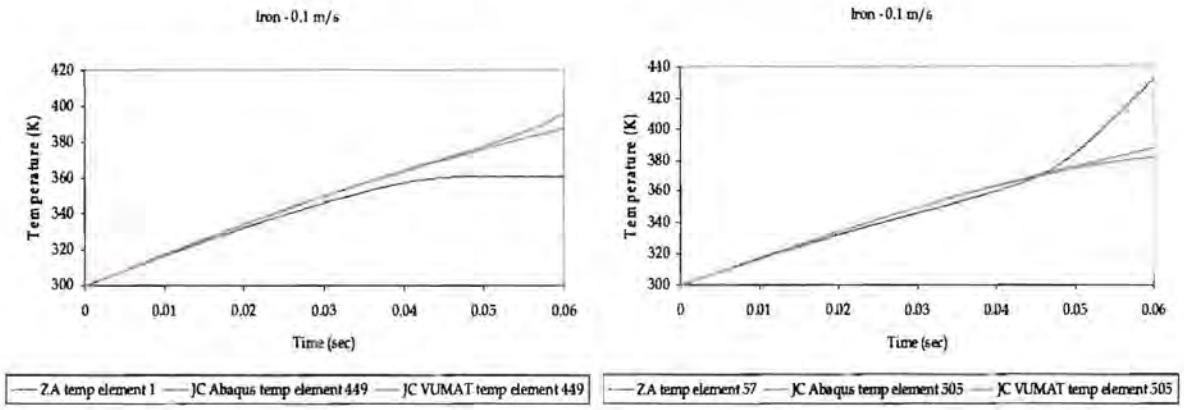
a) b)

Figure C.54: Tension - PEEQ for Armco-Iron at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



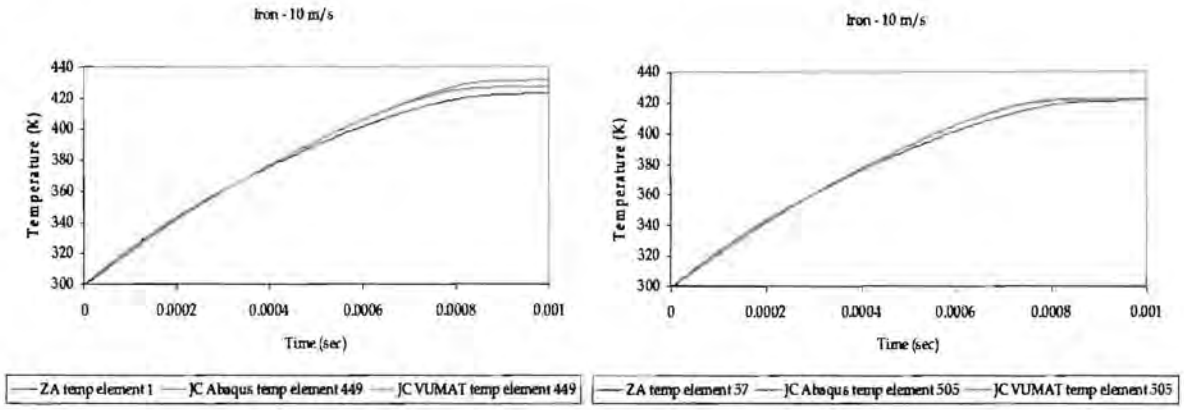
a) b)

Figure C.55: Tension – PEEQ for Armco-Iron at 10 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

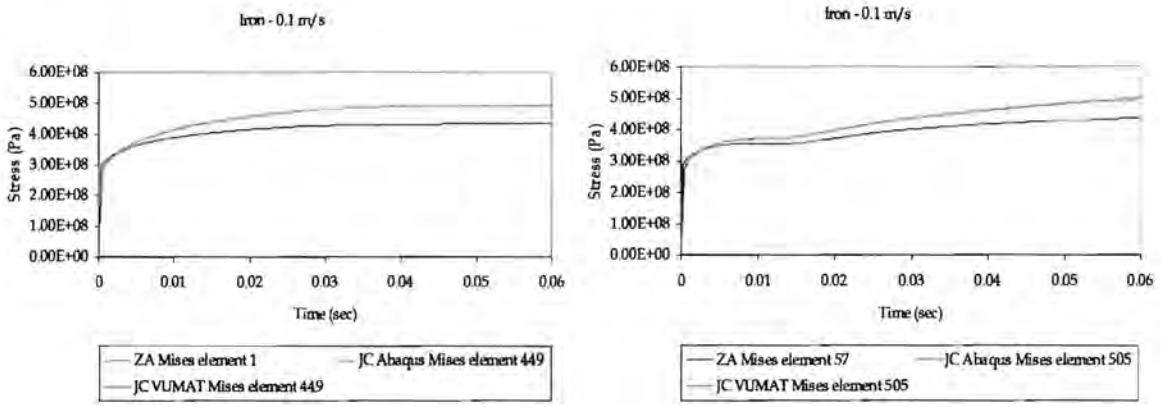
Figure C.56: Tension – temperature for Armco-Iron at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

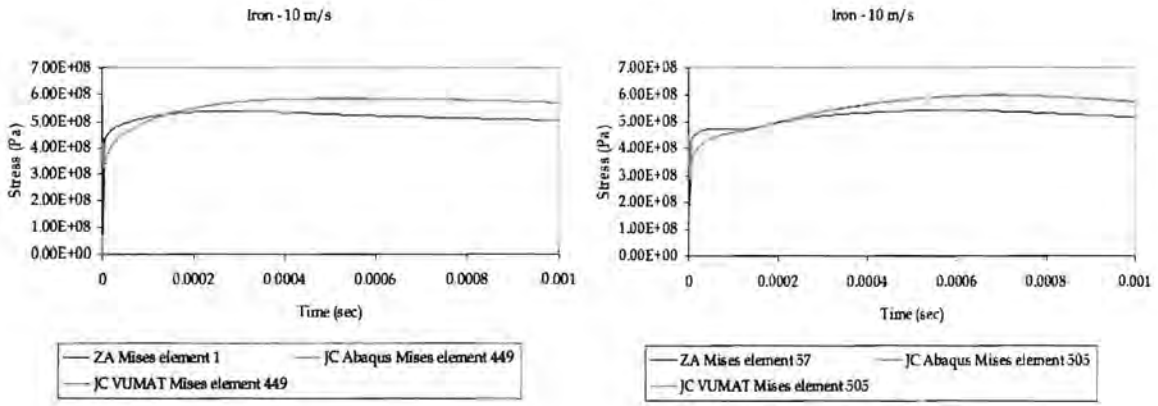
Figure C.57: Tension – temperature for Armco-Iron at 10 m/s
a) element 1 & 449 and b) element 57 & 505.

C.1.4.2 SHEAR



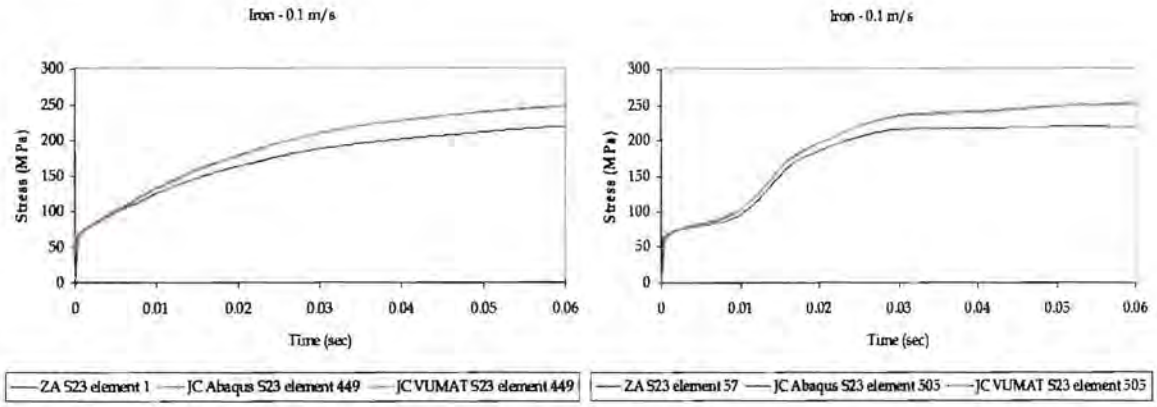
a) b)

Figure C.58: Shear – Mises stress for Armco-Iron at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



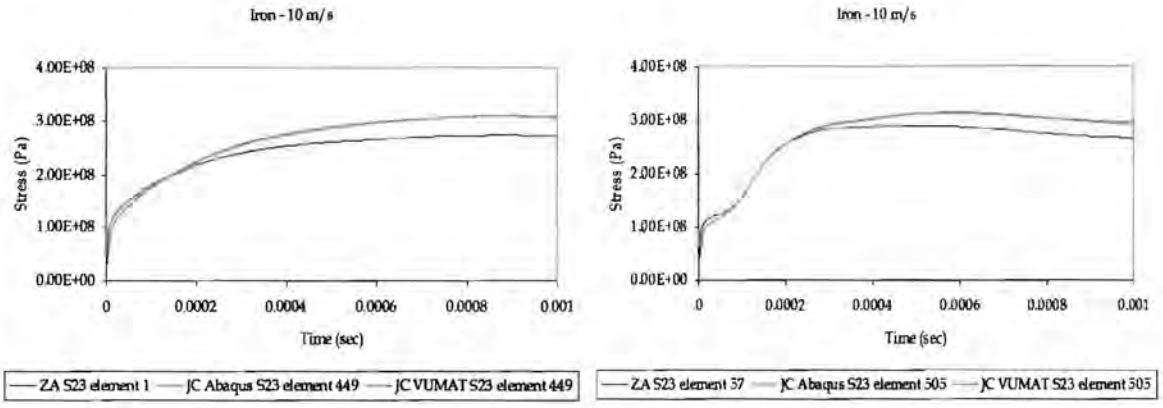
a) b)

Figure C.59: Shear – Mises stress for Armco-Iron at 10 m/s
a) element 1 & 449 and b) element 57 & 505.



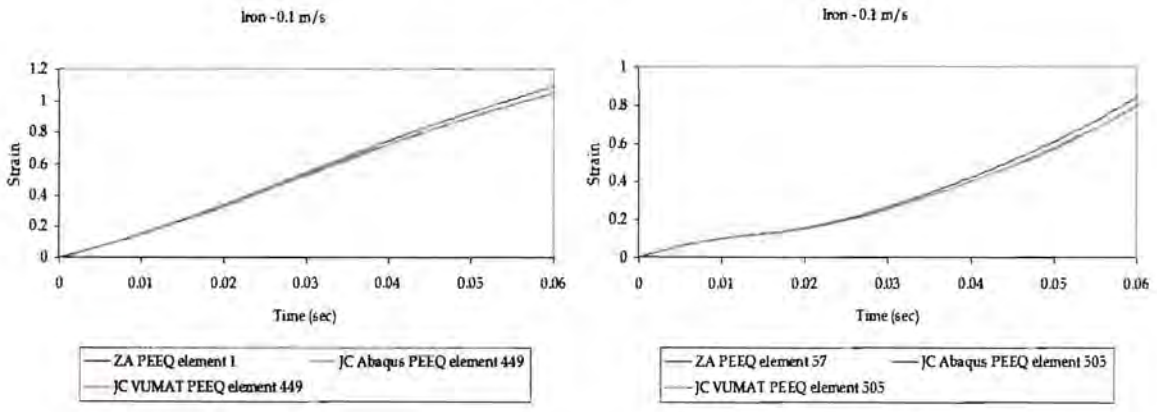
a) b)

Figure C.60: Shear – S23 stress component for Armco-Iron at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



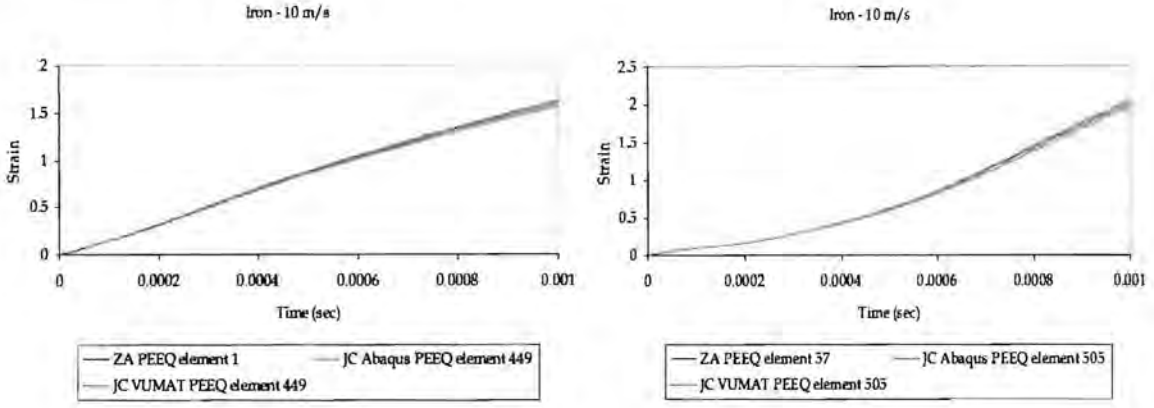
a) b)

Figure C.61: Shear – S23 stress component for Armco-Iron at 10 m/s a) element 1 & 449 and b) element 57 & 505.



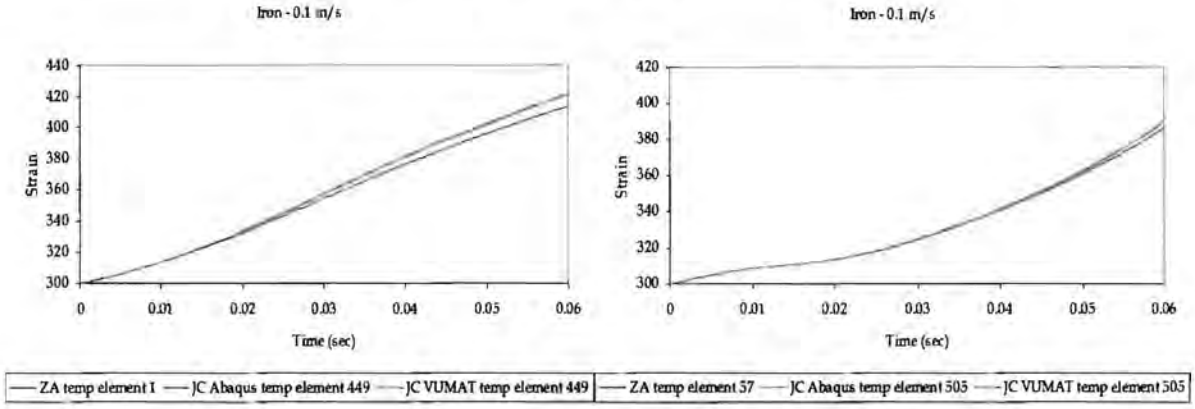
a) b)

Figure C.62: Shear – PEEQ for Armco-Iron at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



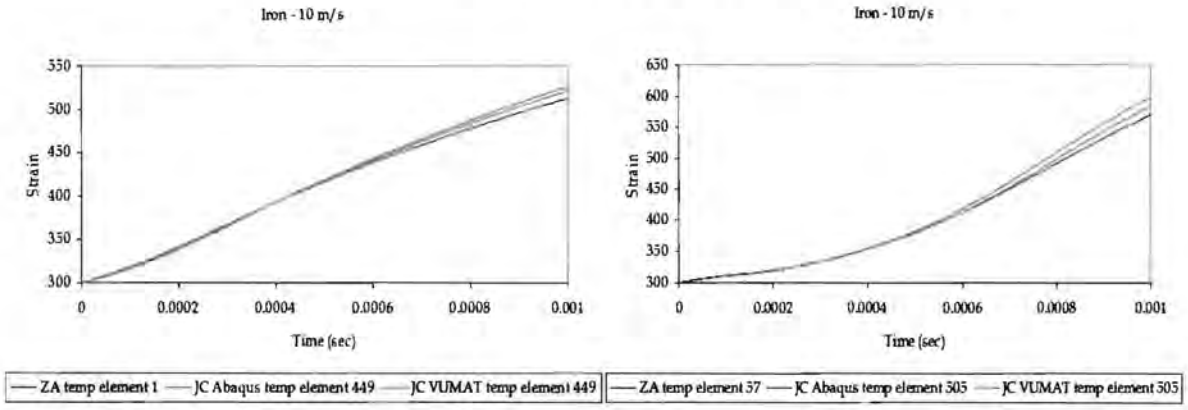
a) b)

Figure C.63: Shear - PEEQ for Armco-Iron at 10 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

Figure C.64: Shear - temperature for Armco-Iron at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



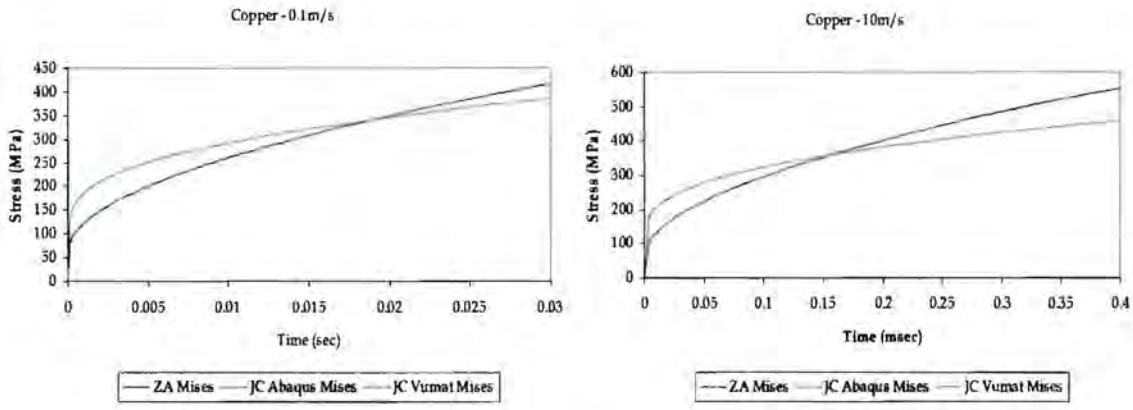
a) b)

Figure C.65: Shear - temperature for Armco-Iron at 10 m/s
a) element 1 & 449 and b) element 57 & 505.

C.2 OFHC COPPER

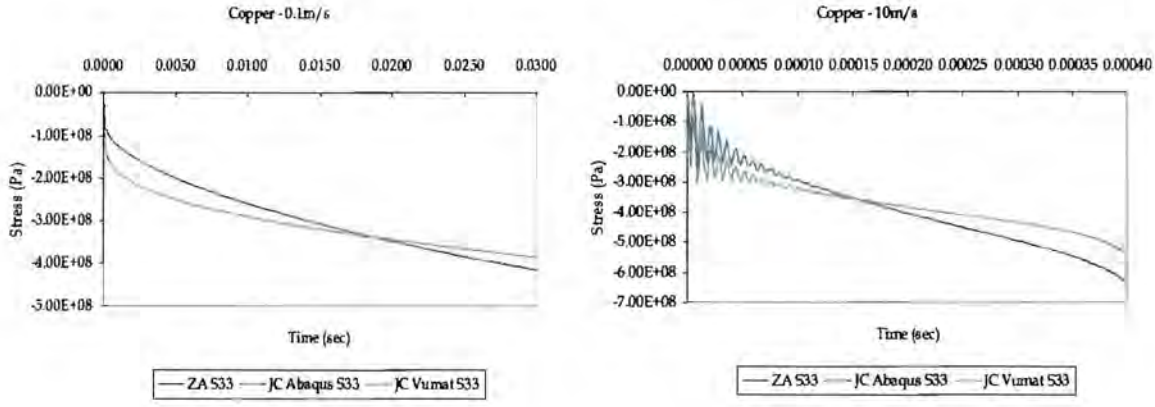
C.2.1 SINGLE ELEMENT

C.2.1.1 COMPRESSION



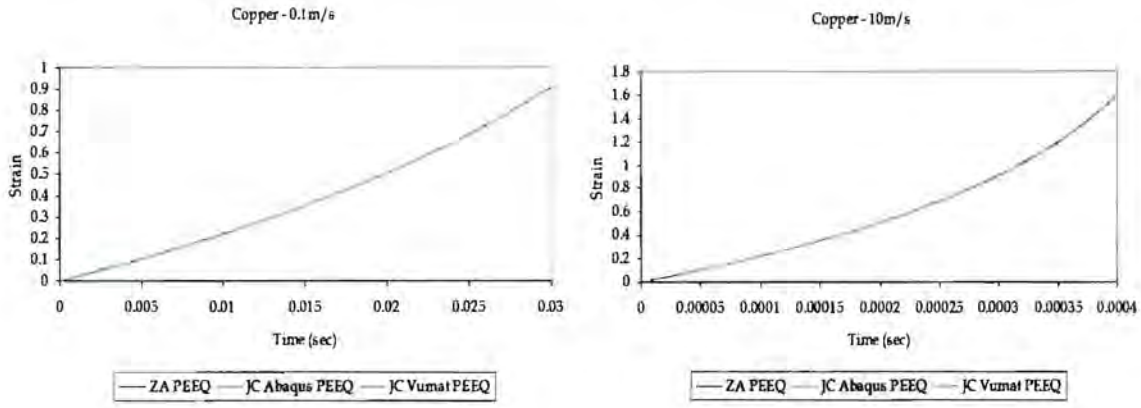
a) b)

Figure C.66: Compression - Mises stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



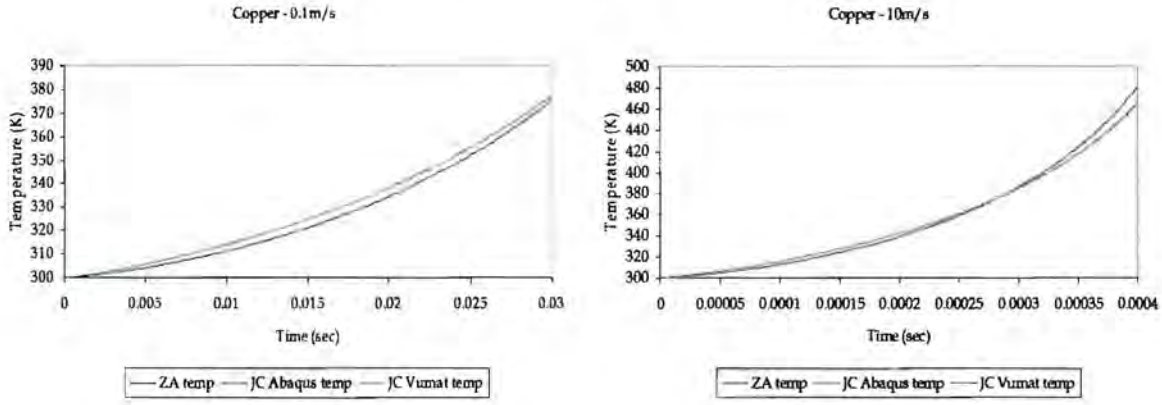
a) b)

Figure C.67: Compression - S33 stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



a) b)

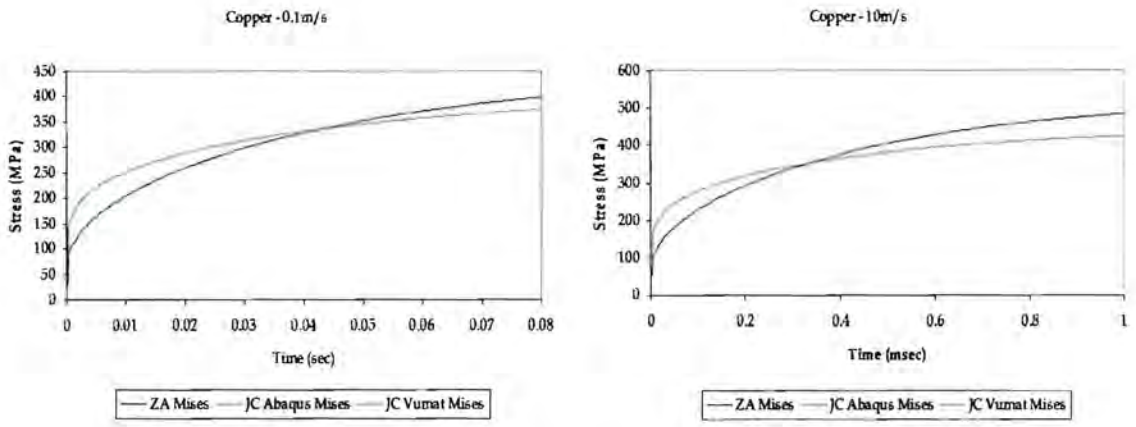
Figure C.68: Compression – PEEQ for OFHC Copper
a) 0.1 m/s, b) 10 m/s.



a) b)

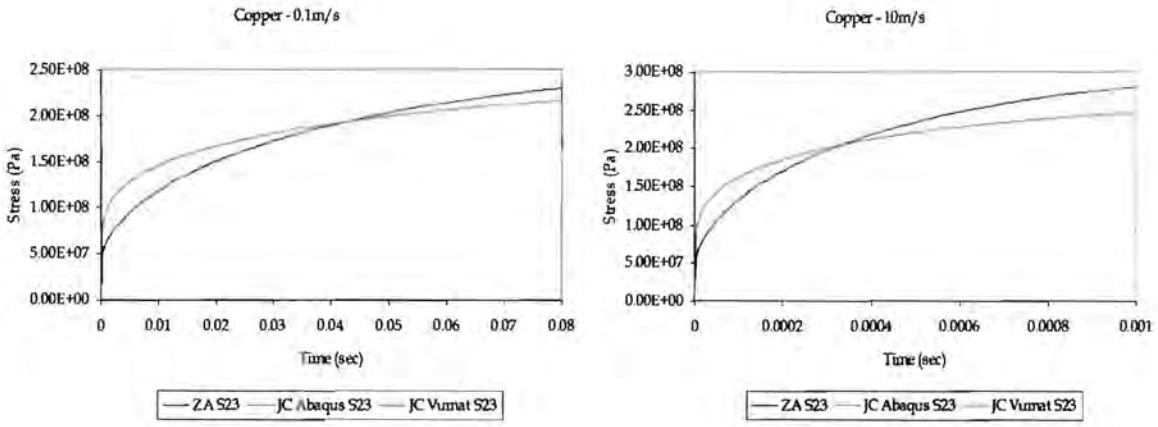
Figure C.69: Compression – temperature for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

C.2.1.2 SHEAR



a) b)

Figure C.70: Shear – Mises stress component for OFHC Copper
a) 0.1 m/s, b) 10 m/s.



a) b)

Figure C.71: Shear – S23 stress component for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

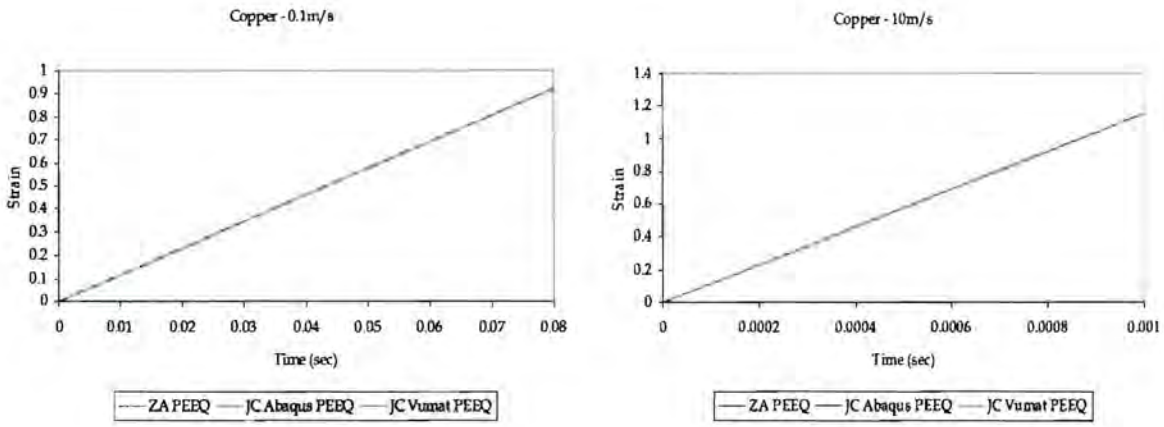
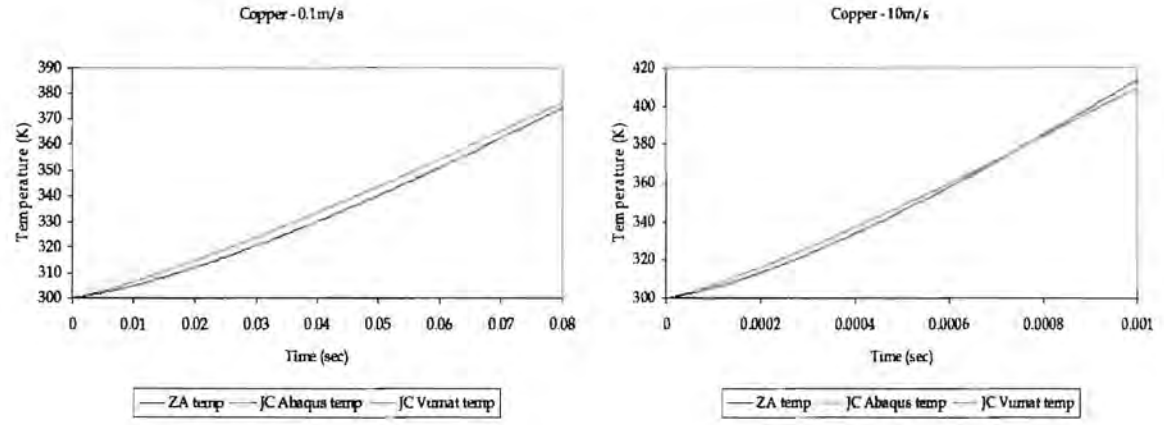


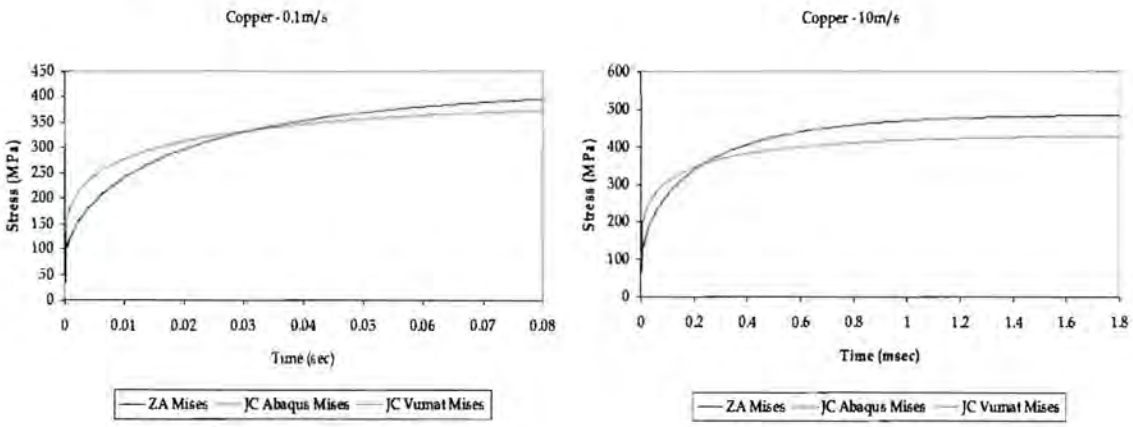
Figure C.72: Shear – PEEQ for OFHC Copper
a) 0.1 m/s, b) 10 m/s.



a) b)

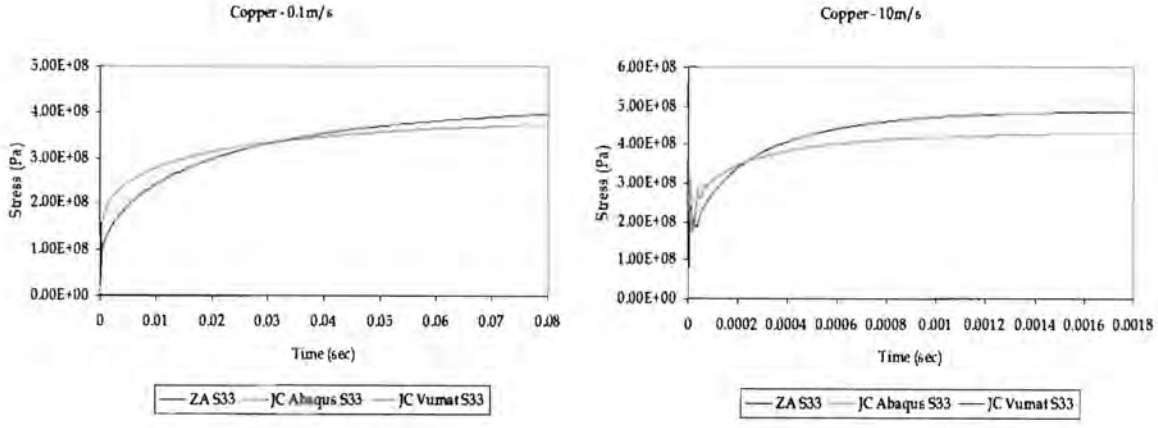
Figure C.73: Shear – temperature for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

C.2.1.3 TENSION



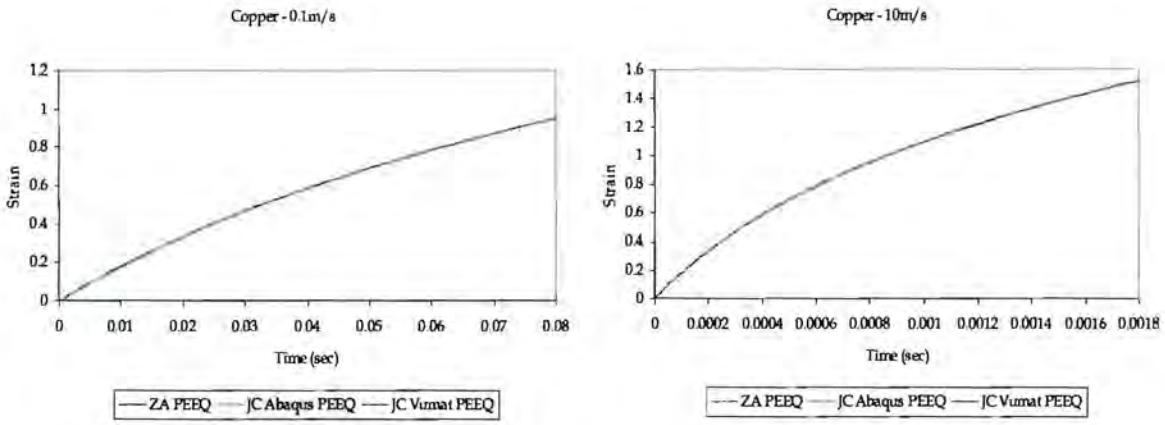
a) b)

Figure C.74: Tension – Mises stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



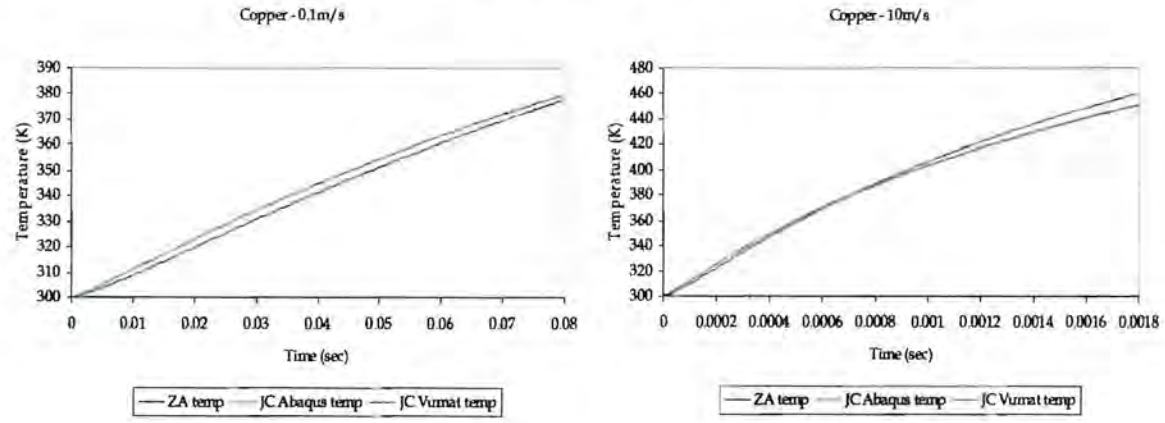
a) b)

Figure C.75: Tension – S33 stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



a) b)

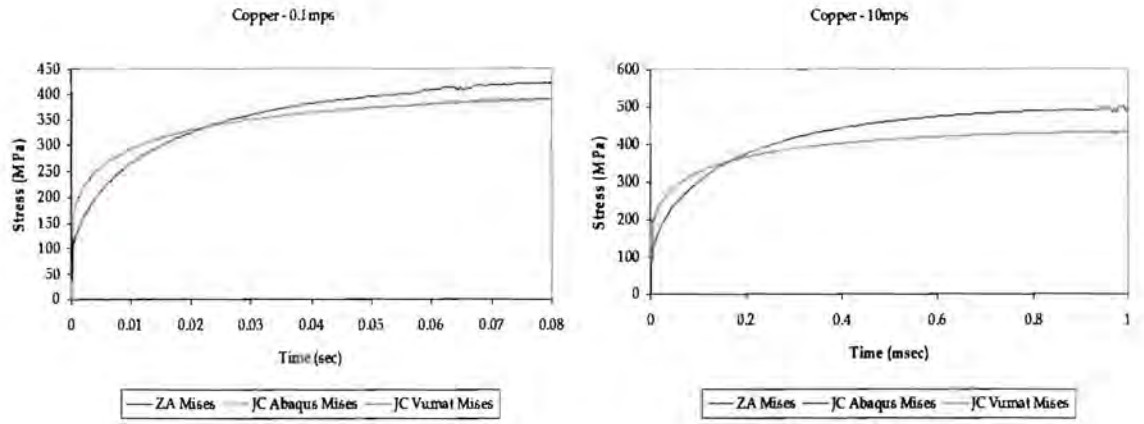
Figure C.76: Tension – PEEQ for OFHC Copper
a) 0.1 m/s, b) 10 m/s.



a) b)

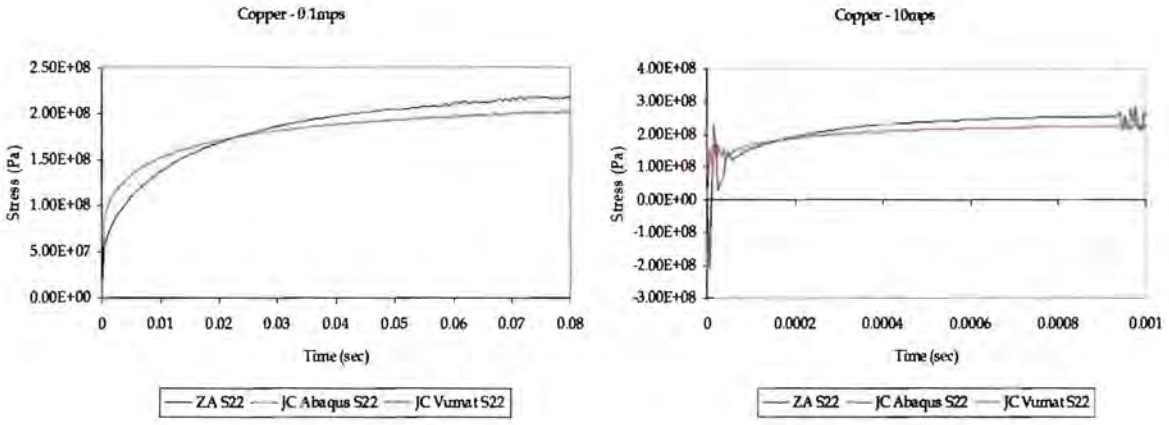
Figure C.77: Tension – temperature for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

C.2.1.4 COMBINED



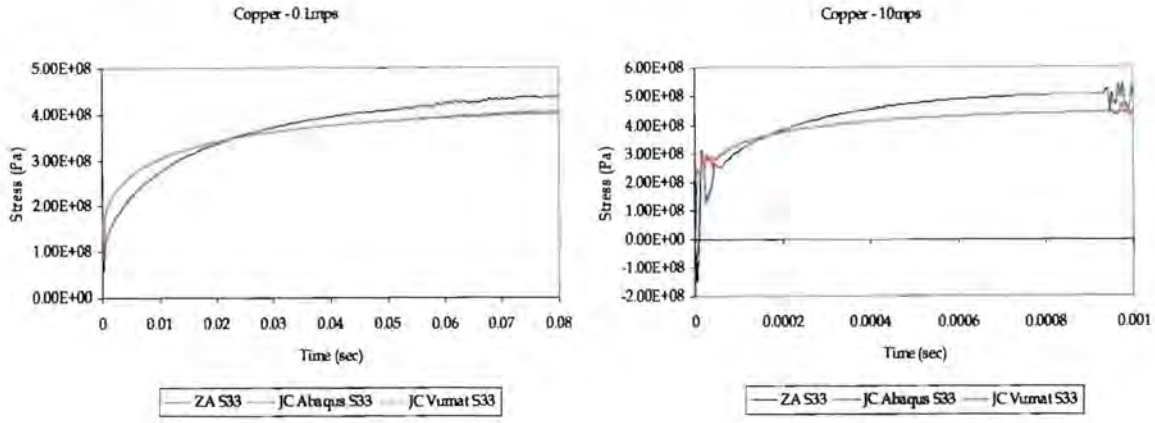
a) b)

Figure C.78: Combined – Mises stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



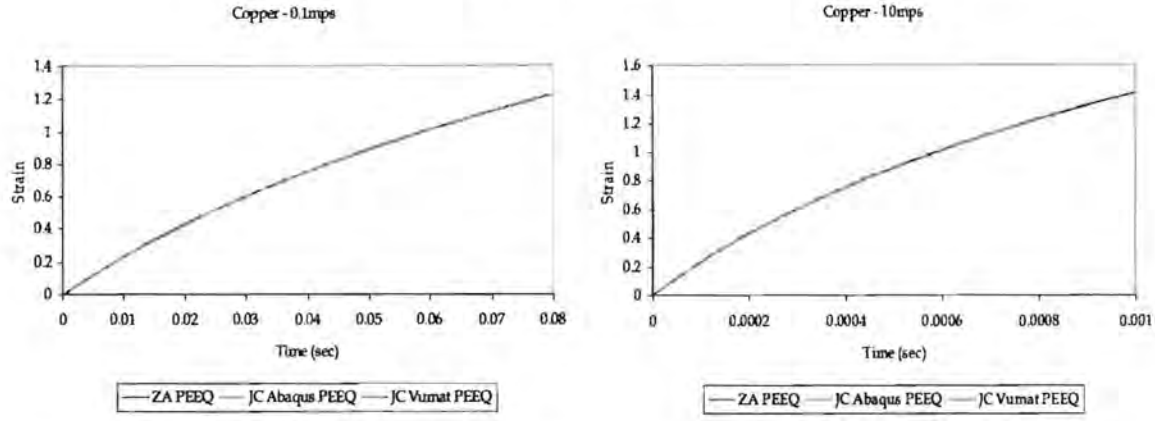
a) b)

Figure C.79: Combined – S22 stress component for OFHC Copper a) 0.1 m/s, b) 10 m/s.



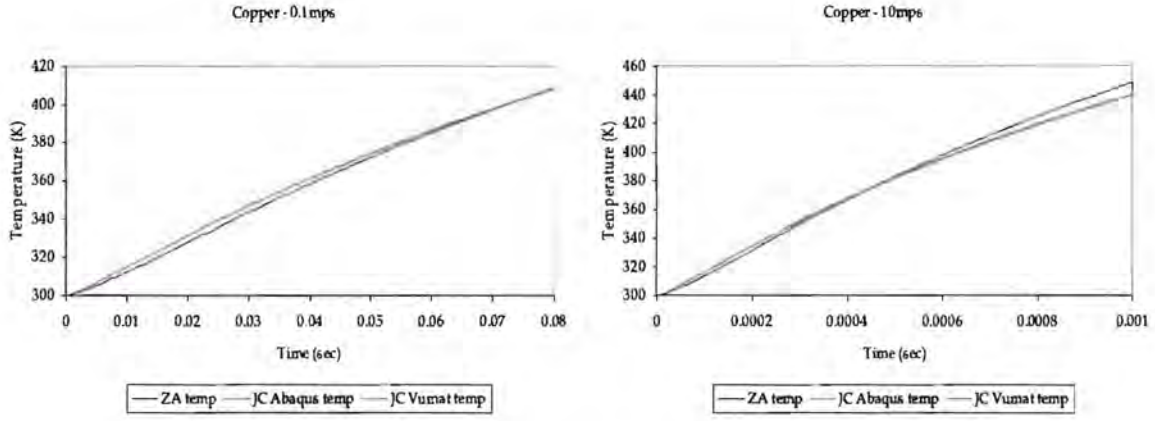
a) b)

Figure C.80: Combined – S33 stress component for OFHC Copper
a) 0.1 m/s, b) 10 m/s.



a) b)

Figure C.81: Combined – PEEQ for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

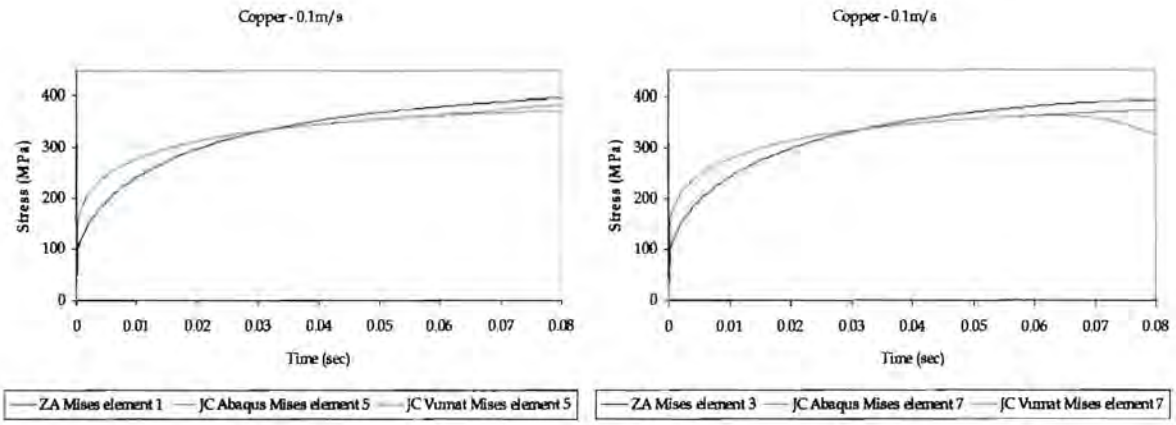


a) b)

Figure C.82: Combined – temperature for OFHC Copper
a) 0.1 m/s, b) 10 m/s.

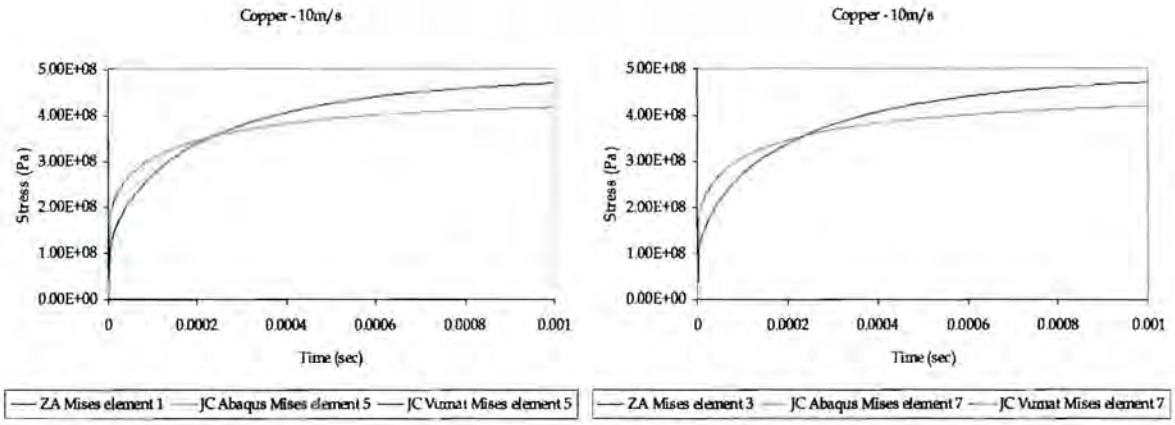
C.2.2 8 (2x2x2) ELEMENTS

C.2.2.1 TENSION



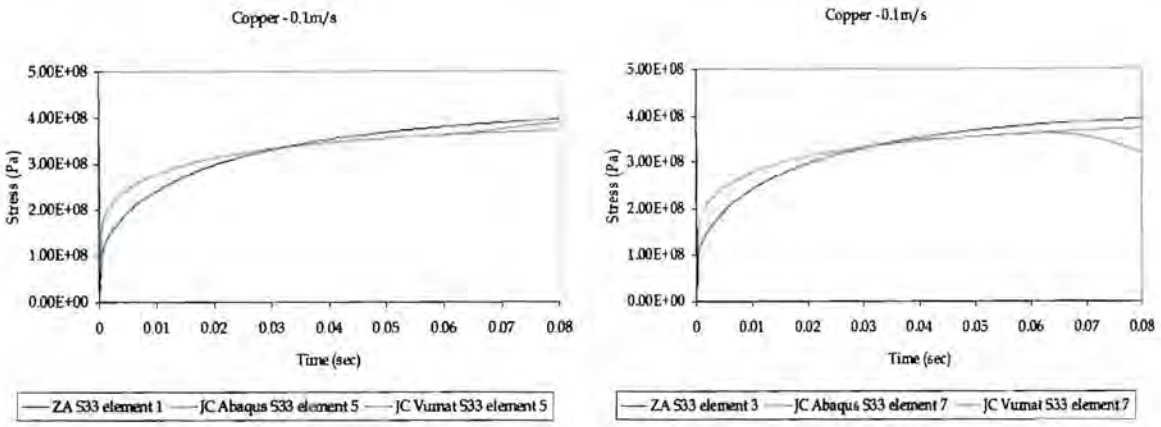
a) b)

Figure C.83: Tension – Mises stress for OFHC Copper at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



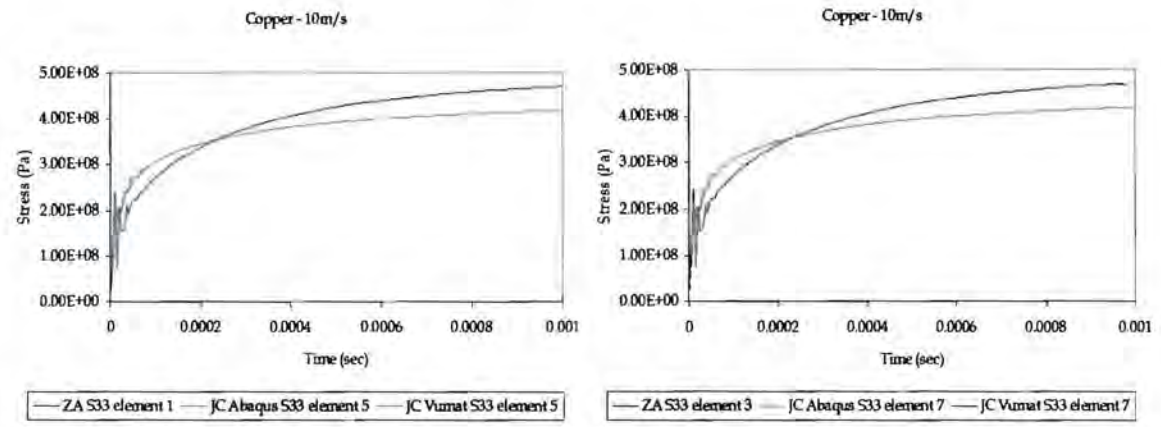
a) b)

Figure C.84: Tension – Mises stress for OFHC Copper at 10 m/s a) element 1 & 5 and b) element 3 & 7.



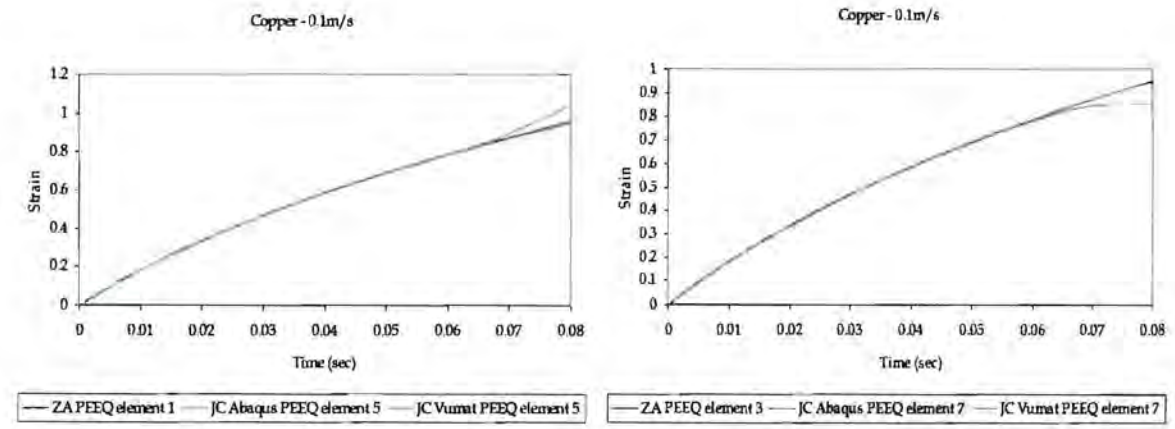
a) b)

Figure C.85: Tension – S33 stress component for OFHC Copper at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



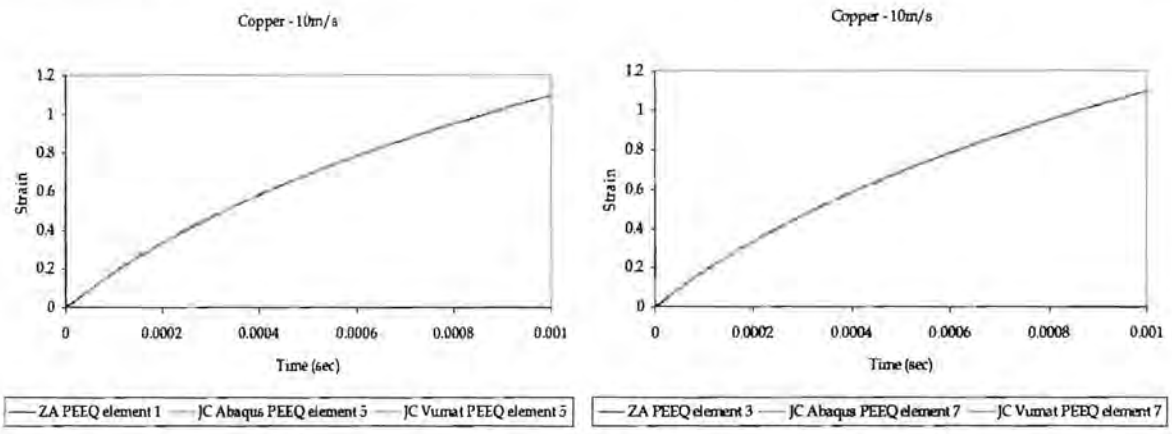
a) b)

Figure C.86: Tension – S33 stress component for OFHC Copper at 10 m/s a) element 1 & 5 and b) element 3 & 7.



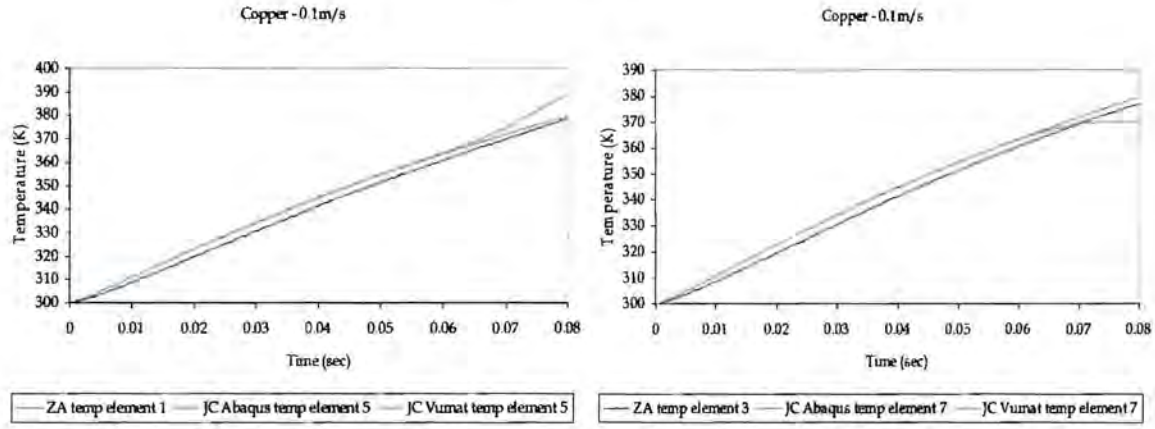
a) b)

Figure C.87: Tension – PEEQ for OFHC Copper at 0.1 m/s a) element 1 & 5 and b) element 3 & 7.



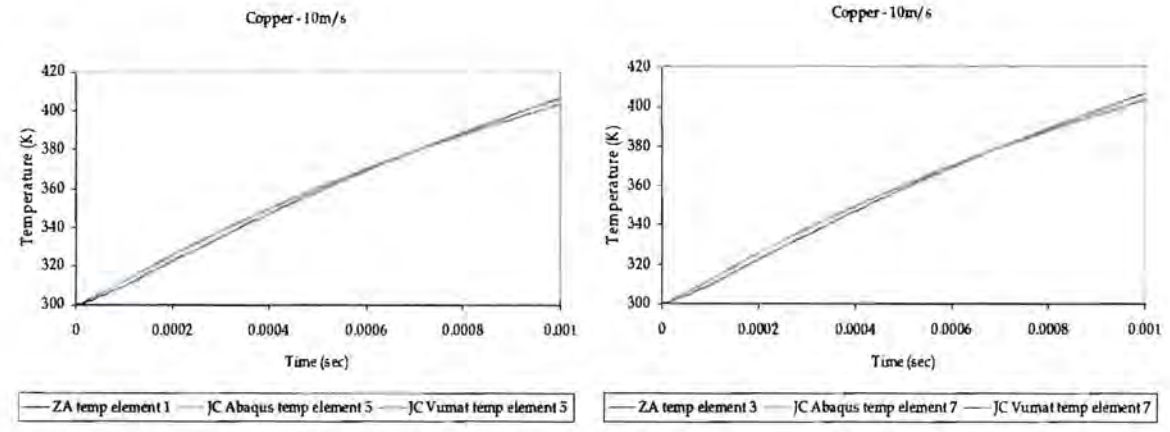
a) b)

Figure C.88: Tension – PEEQ for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



a) b)

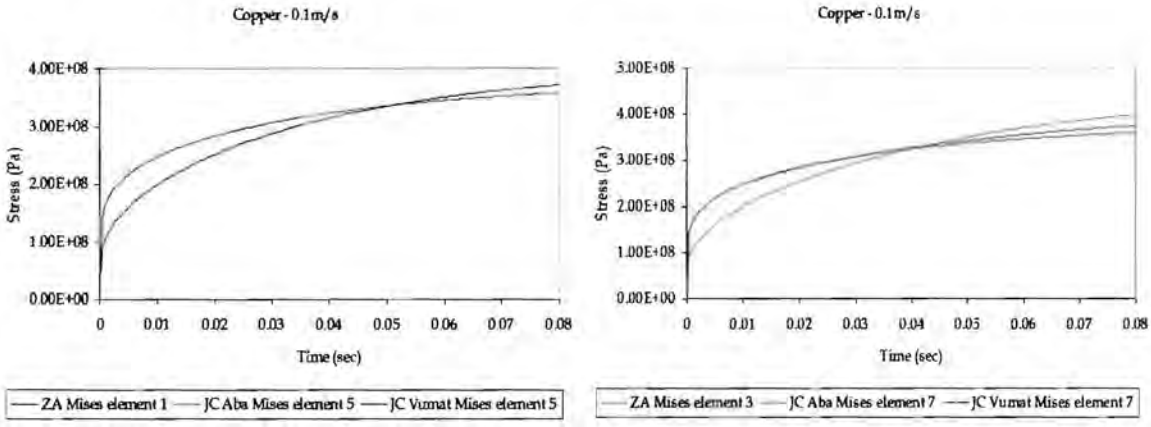
Figure C.89: Tension – temperature for OFHC Copper at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



a) b)

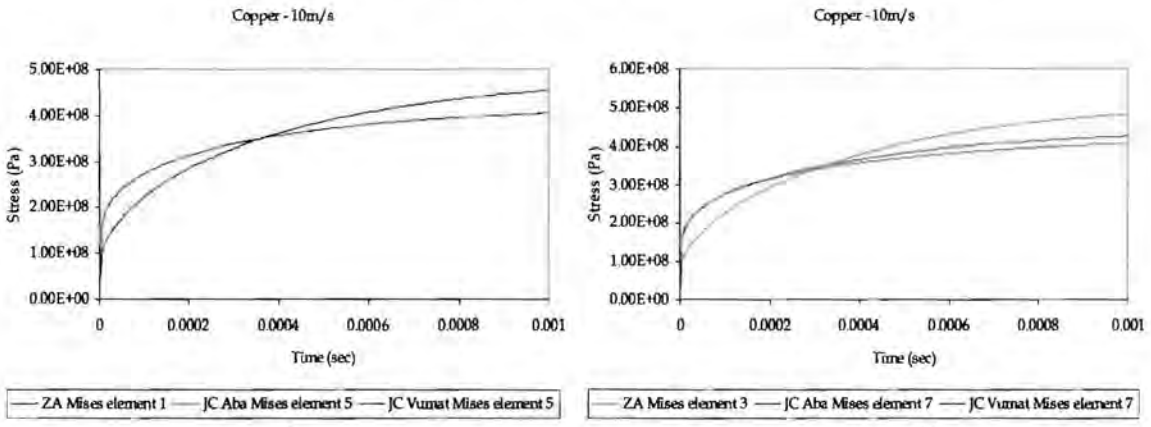
Figure C.90: Tension – temperature for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.

C.2.2.2 SHEAR



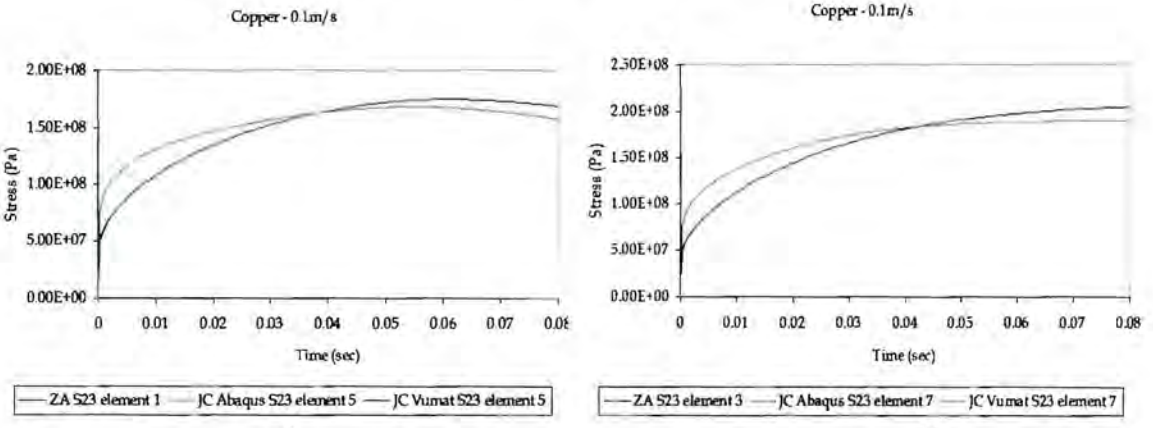
a) b)

Figure C.91: Shear - Mises stress for OFHC Copper at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



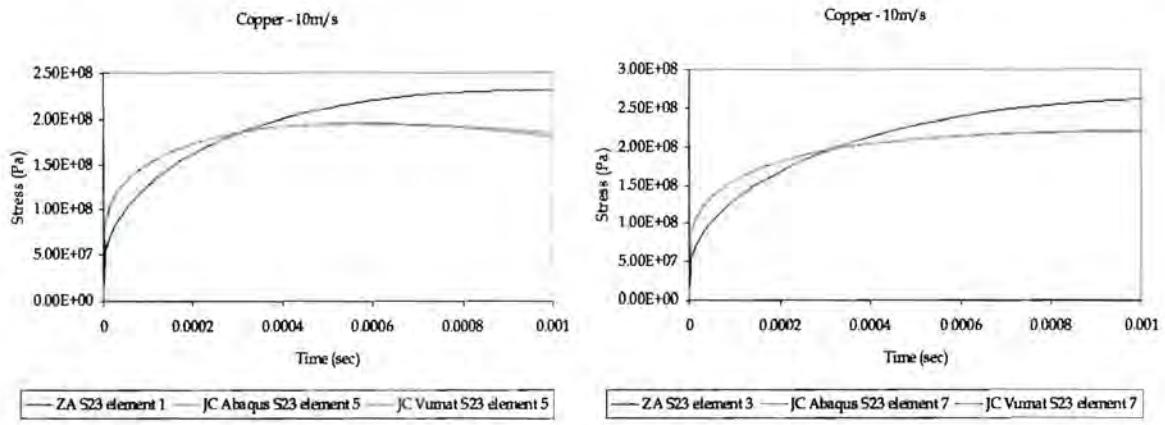
a) b)

Figure C.92: Shear - Mises stress for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



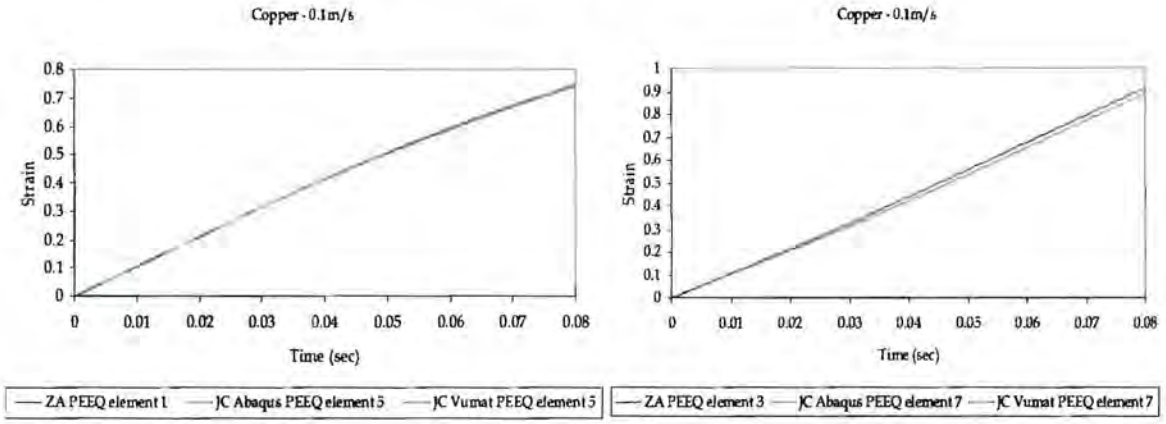
a) b)

Figure C.93: Shear - S23 stress component for OFHC Copper at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



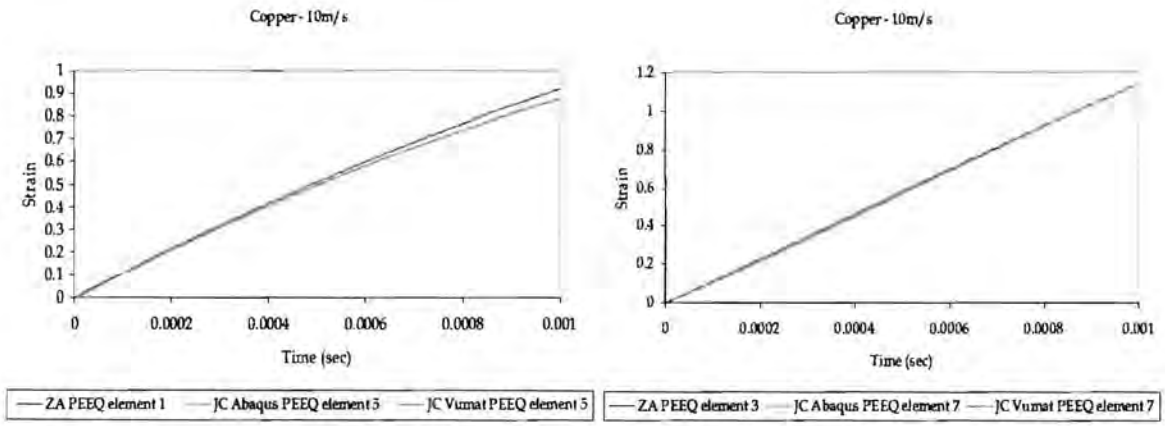
a) b)

Figure C.94: Shear – S23 stress component for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



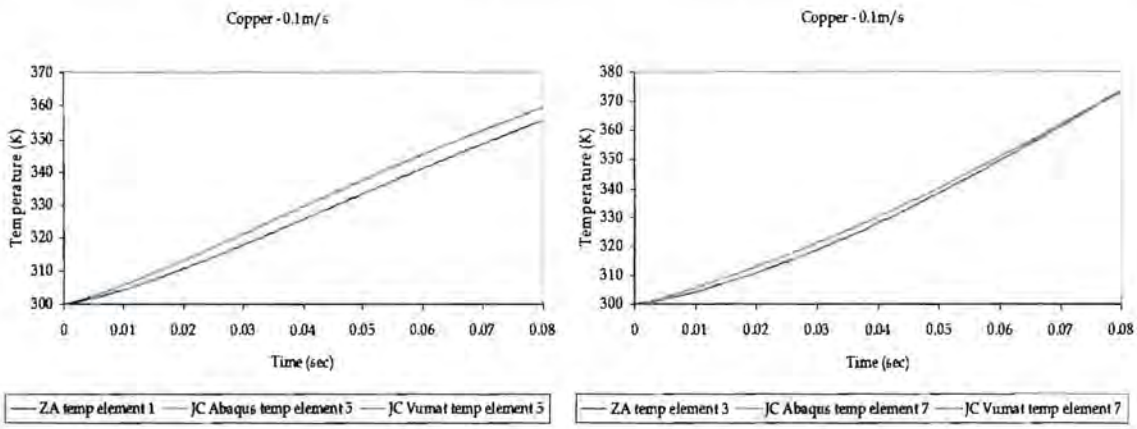
a) b)

Figure C.95: Shear – PEEQ for OFHC Copper at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.



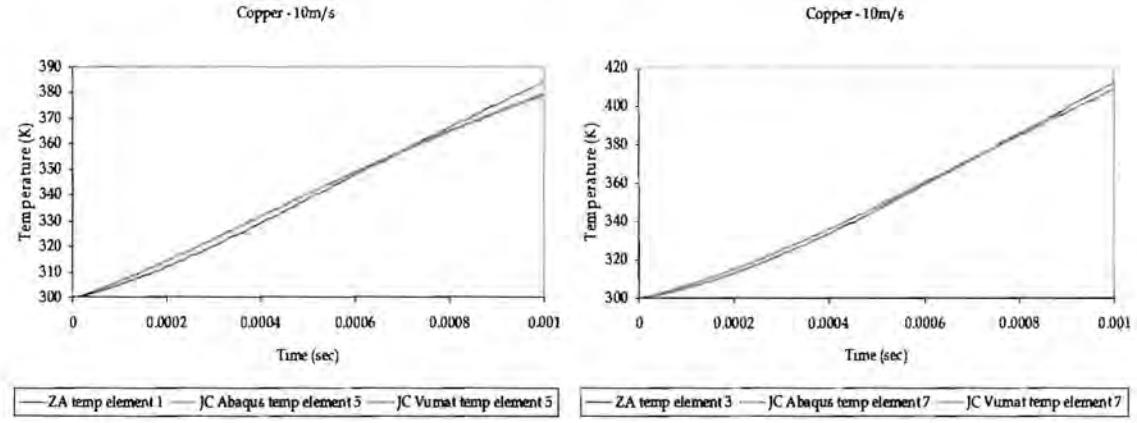
a) b)

Figure C.96: Shear – PEEQ for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.



a) b)

Figure C.97: Shear - temperature for OFHC Copper at 0.1 m/s
a) element 1 & 5 and b) element 3 & 7.

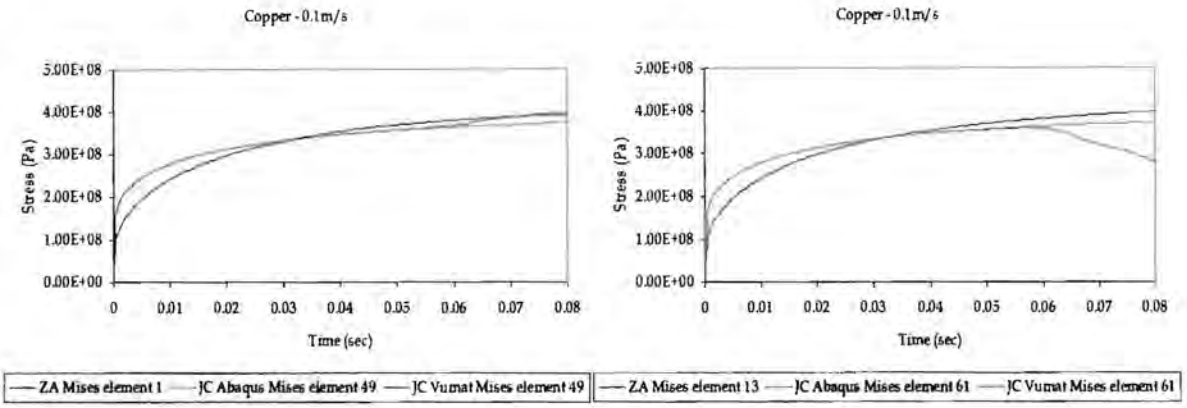


a) b)

Figure C.98: Shear - temperature for OFHC Copper at 10 m/s
a) element 1 & 5 and b) element 3 & 7.

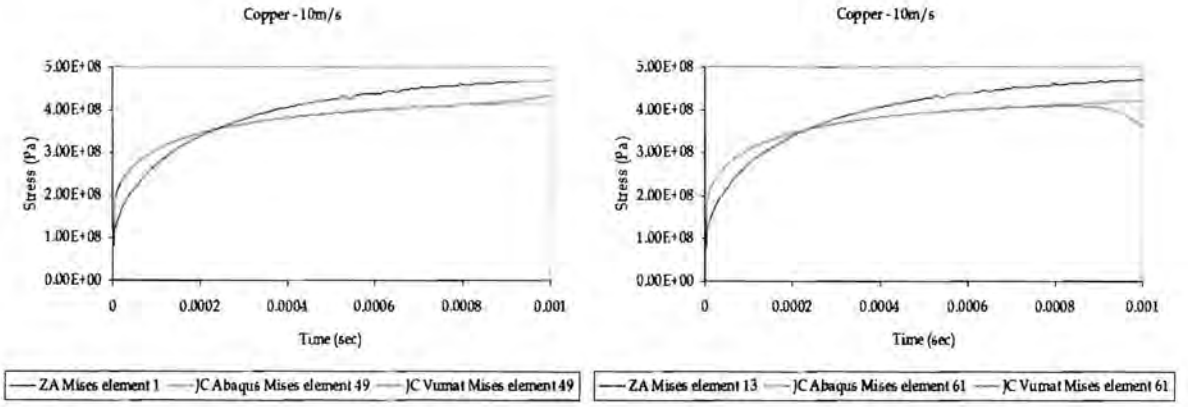
C.2.3 64 (4x4x4) ELEMENTS

C.2.3.1 TENSION



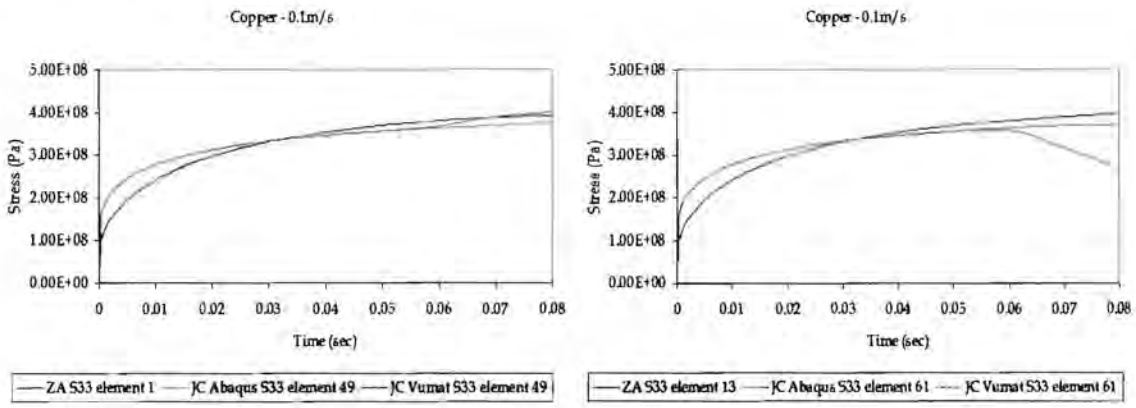
a) b)

Figure C.99: Tension - Mises stress for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



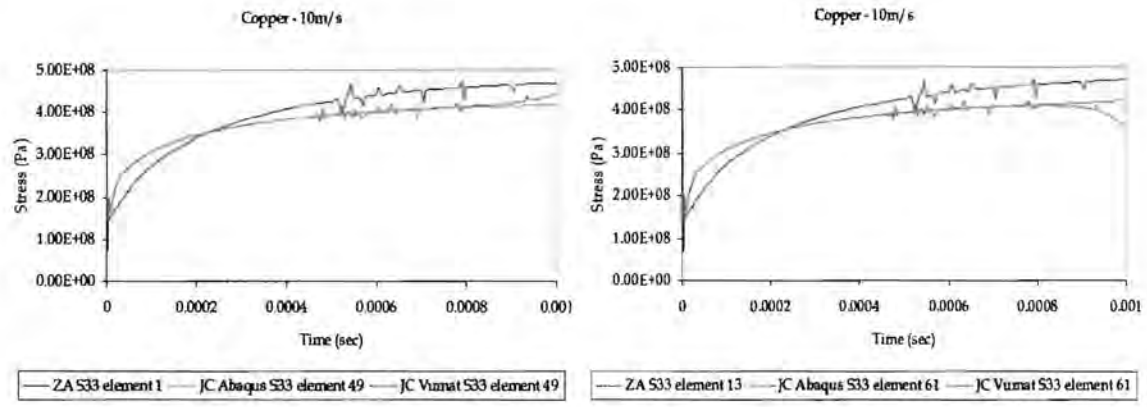
a) b)

Figure C.100: Tension - Mises stress for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



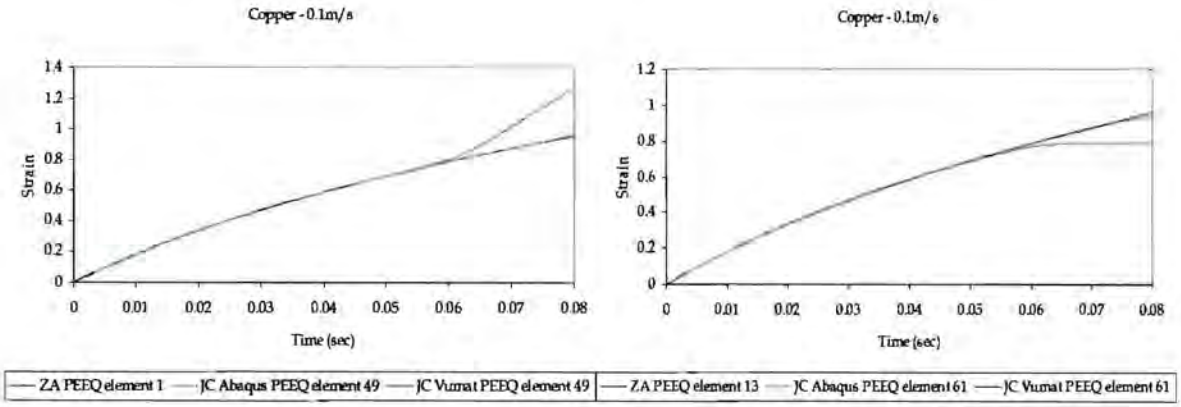
a) b)

Figure C.101: Tension - S33 stress component for OFHC Copper at 0.1 m/s a) element 1 & 49 and b) element 13 & 61.



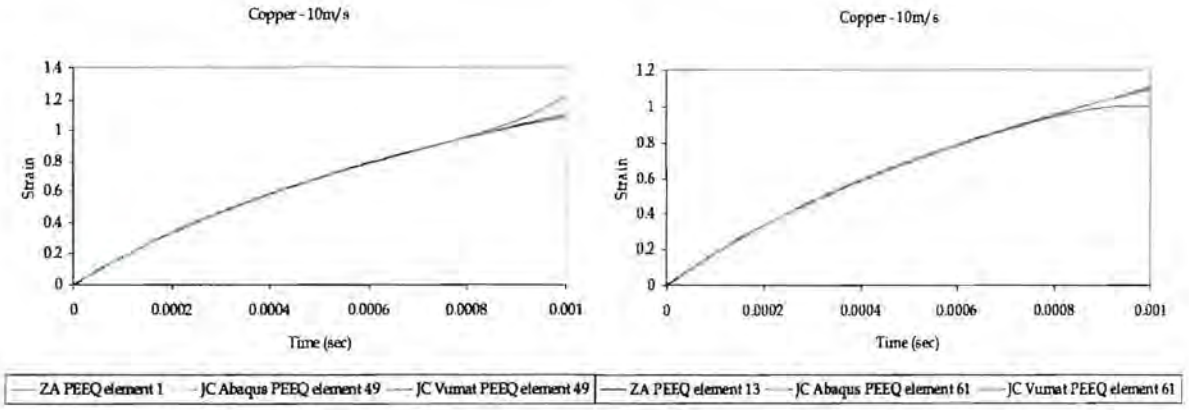
a) b)

Figure C.102: Tension - S33 stress component for OFHC Copper at 10 m/s a) element 1 & 49 and b) element 13 & 61.



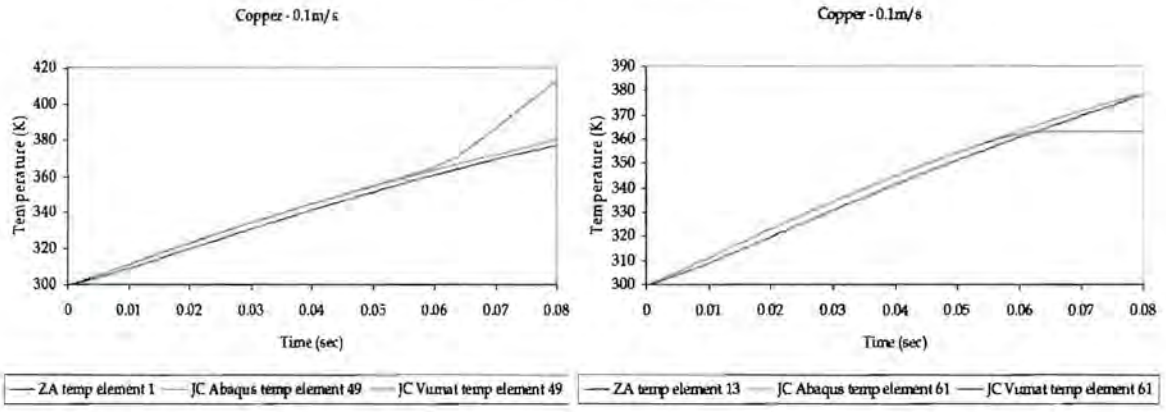
a) b)

Figure C.103: Tension - PEEQ for OFHC Copper at 0.1 m/s a) element 1 & 49 and b) element 13 & 61.



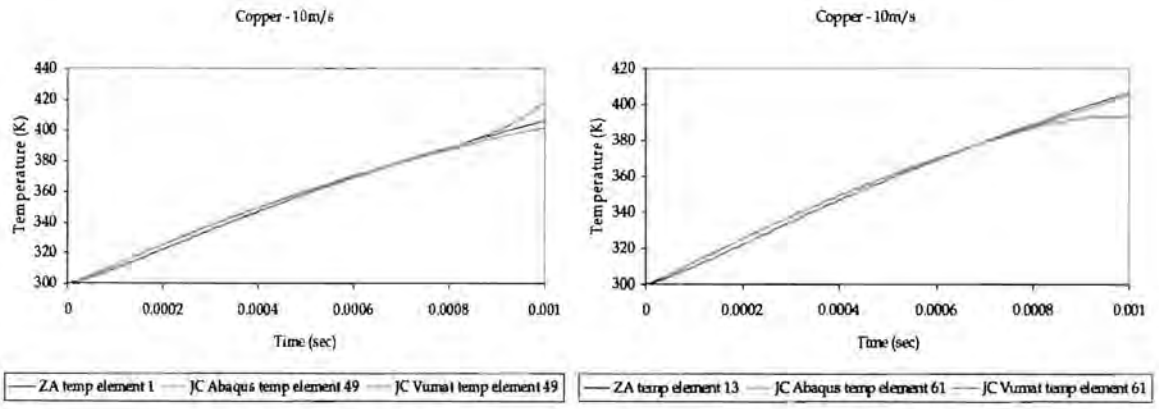
a) b)

Figure C.104: Tension - PEEQ for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

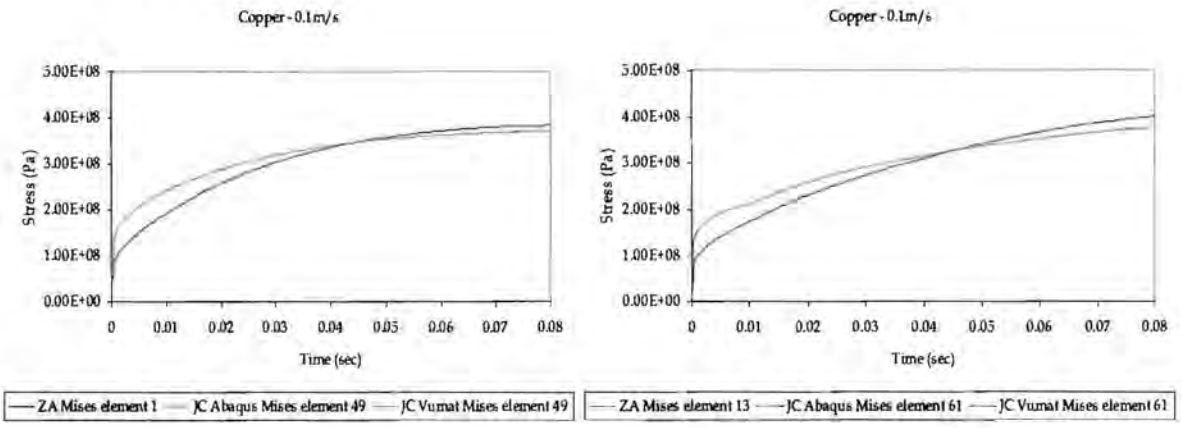
Figure C.105: Tension - temperature for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

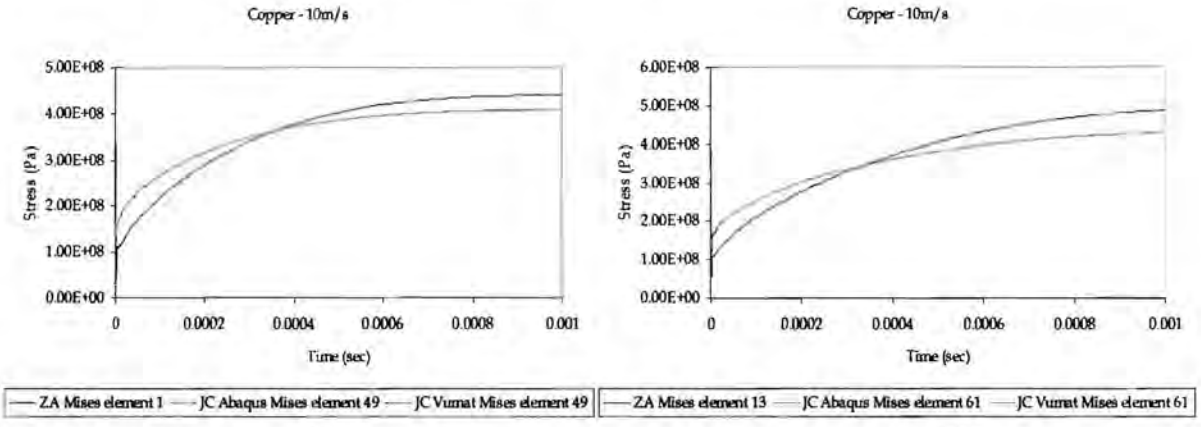
Figure C.106: Tension - temperature for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.

C.2.3.2 SHEAR



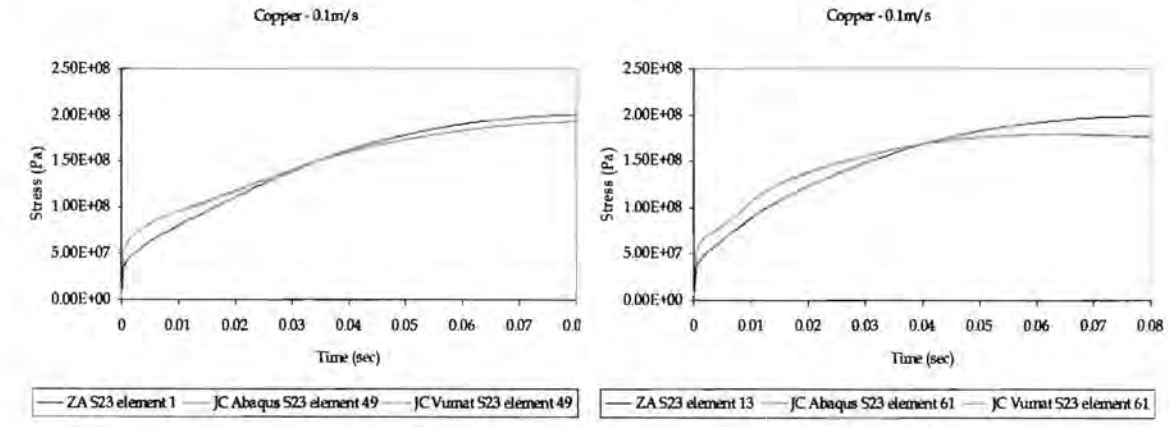
a) b)

Figure C.107: Shear - Mises stress for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61



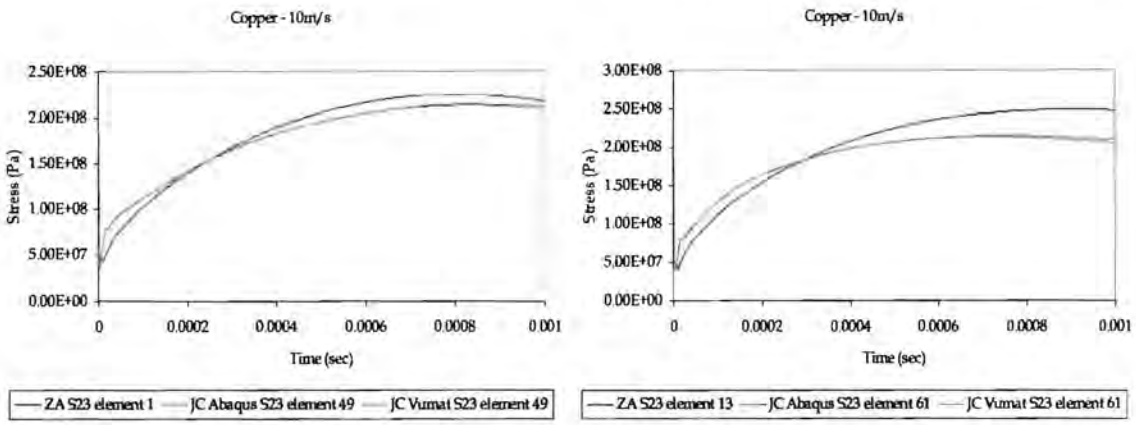
a) b)

Figure C.108: Shear - Mises stress for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



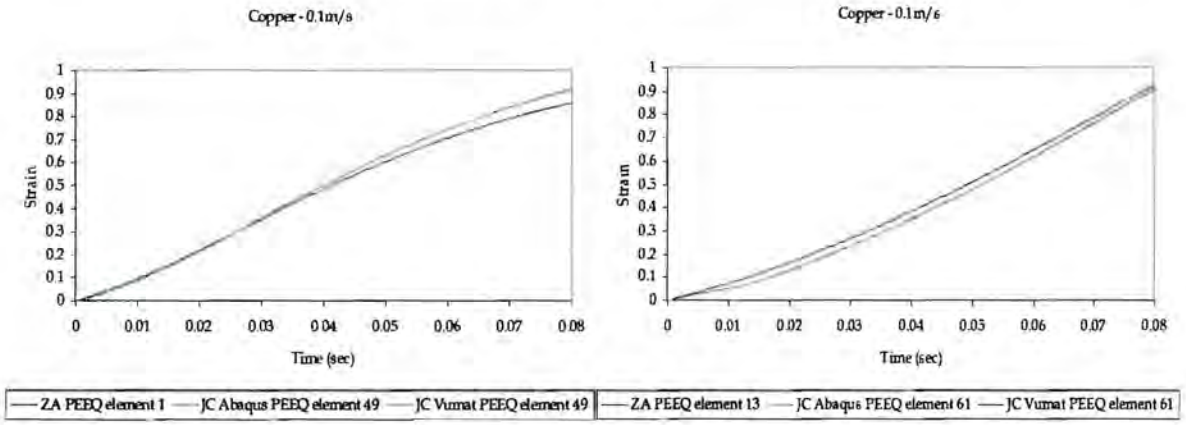
a) b)

Figure C.109: Shear - S23 stress component for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



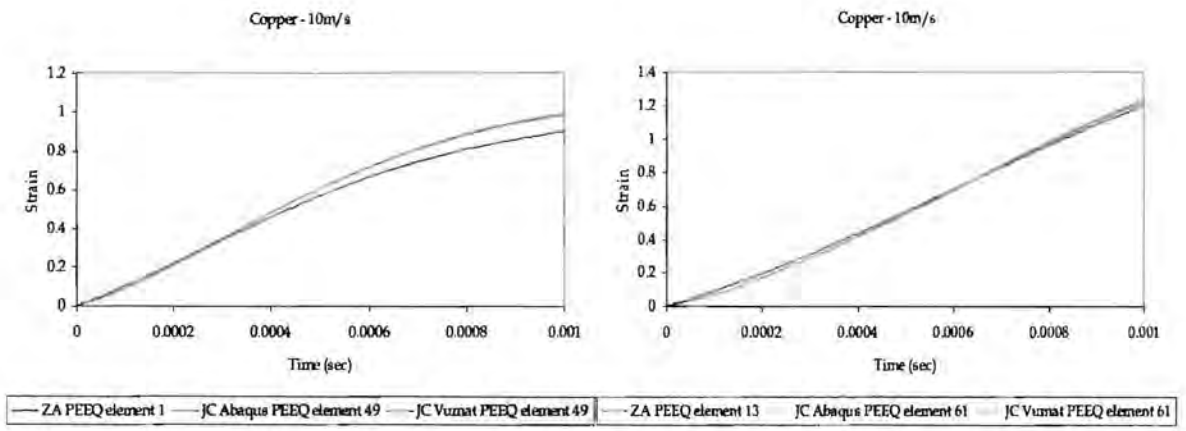
a) b)

Figure C.110: Shear – S23 stress component for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



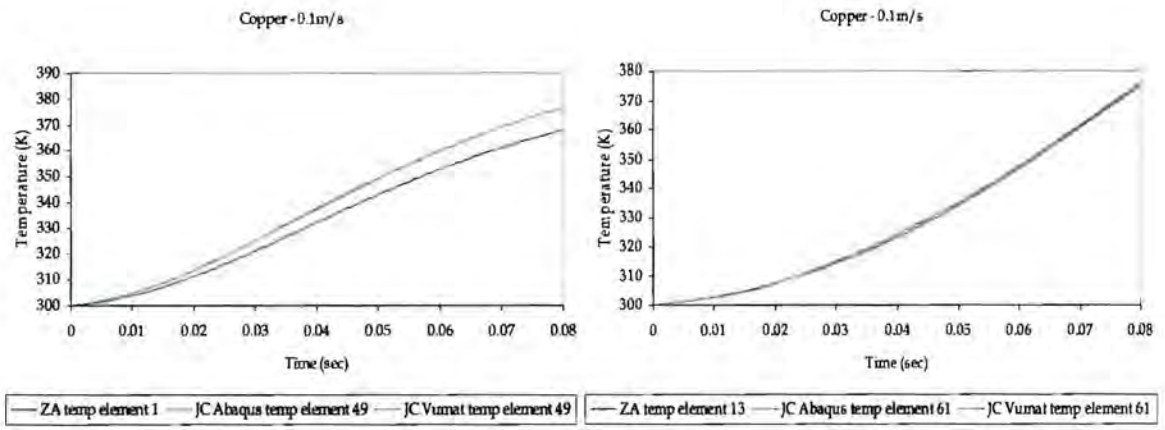
a) b)

Figure C.111: Shear – PEEQ for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.



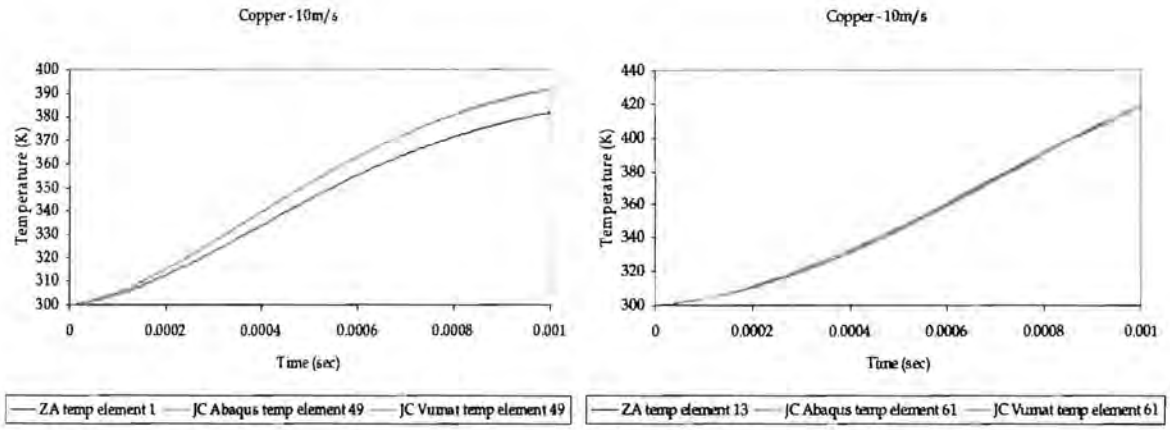
a) b)

Figure C.112: Shear – PEEQ for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.



a) b)

Figure C.113: Shear - temperature for OFHC Copper at 0.1 m/s
a) element 1 & 49 and b) element 13 & 61.

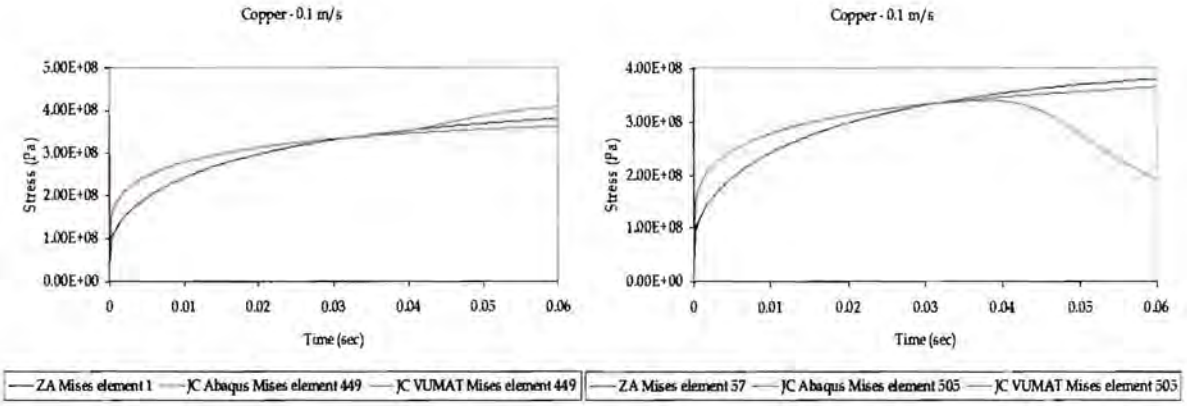


a) b)

Figure C.114: Shear - temperature for OFHC Copper at 10 m/s
a) element 1 & 49 and b) element 13 & 61.

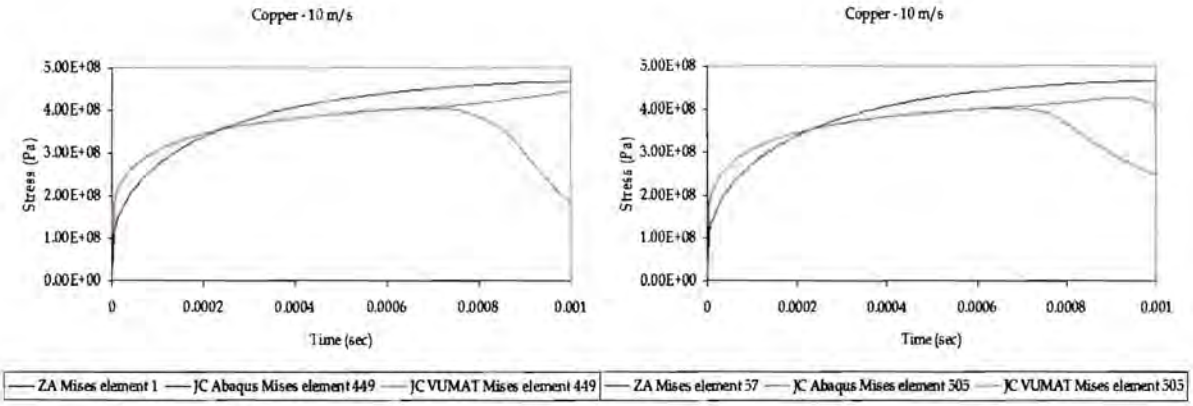
C.2.4 512 (8x8x8) ELEMENTS

C.2.4.1 TENSION



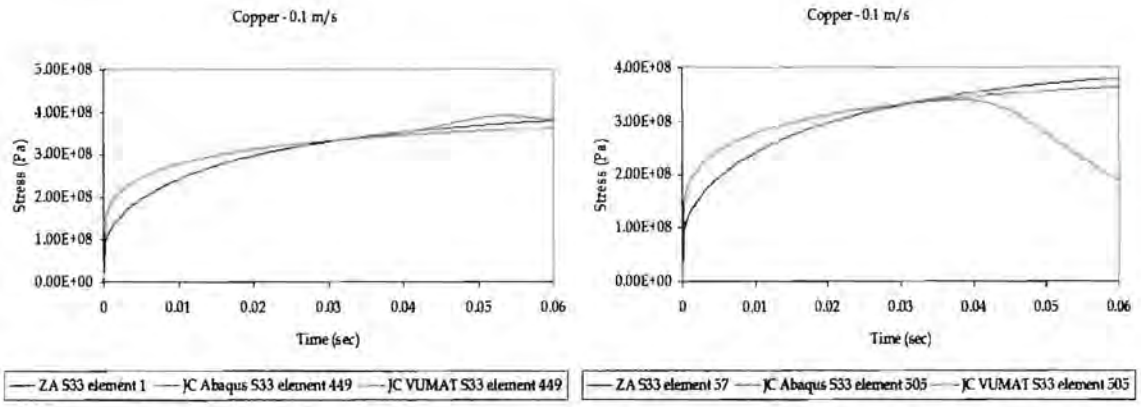
a) b)

Figure C.115: Tension - Mises stress for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



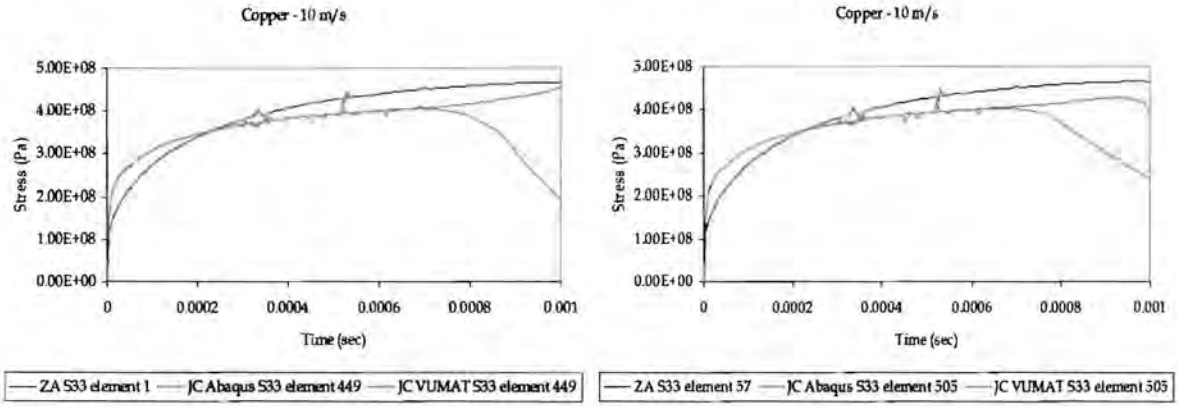
a) b)

Figure C.116: Tension - Mises stress for OFHC Copper at 10 m/s a) element 1 & 449 and b) element 57 & 505.



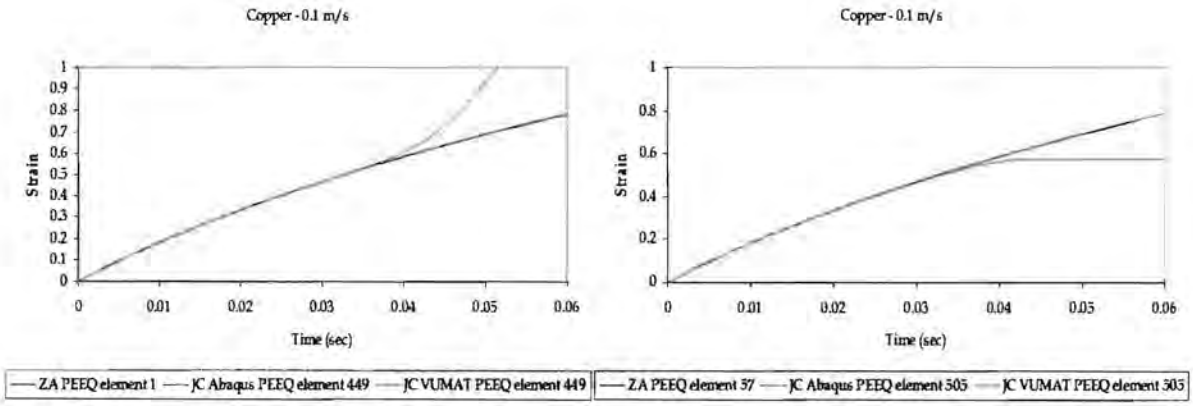
a) b)

Figure C.117: Tension – S33 stress component for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



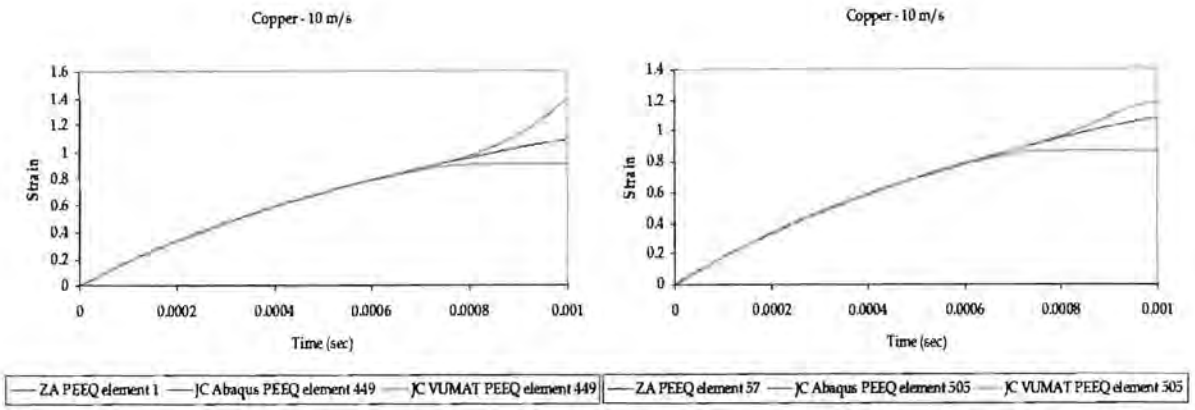
a) b)

Figure C.118: Tension – S33 stress component for OFHC Copper at 10 m/s a) element 1 & 449 and b) element 57 & 505.



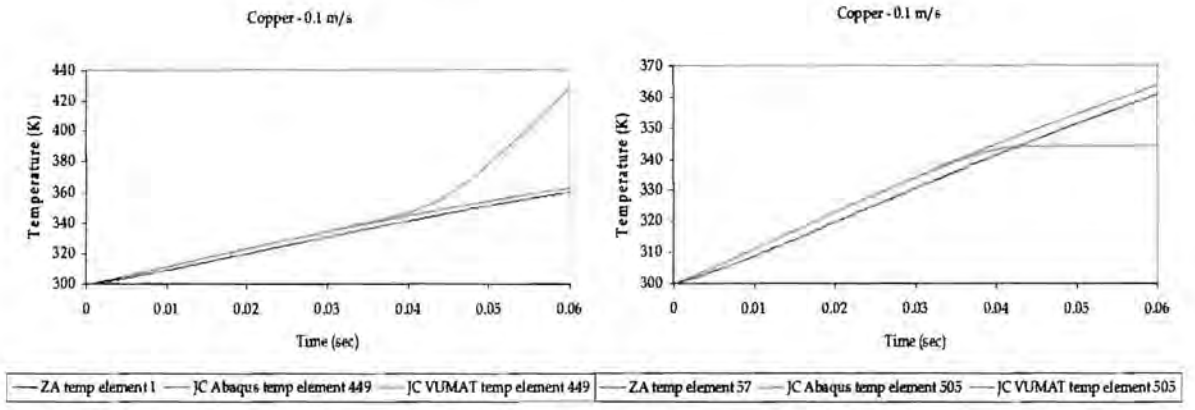
a) b)

Figure C.119: Tension – PEEQ for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



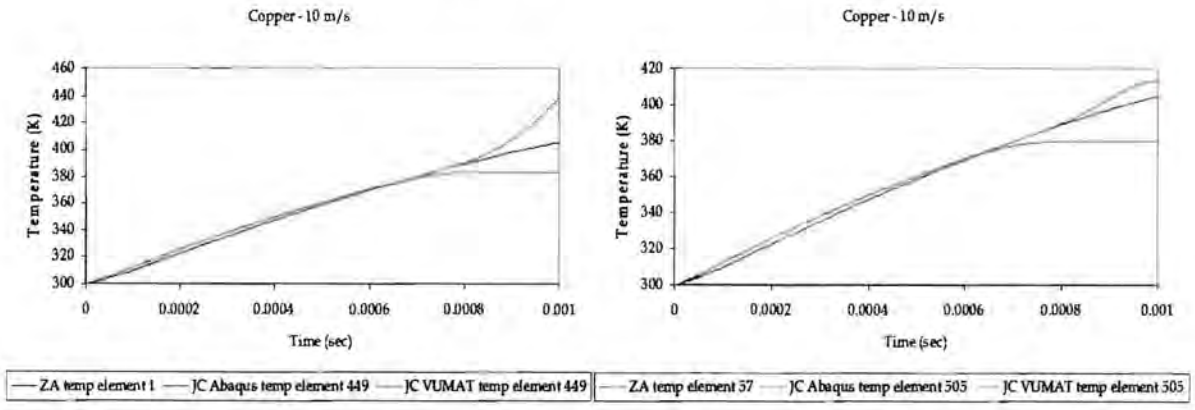
a) b)

Figure C.120: Tension - PEEQ for OFHC Copper at 10 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

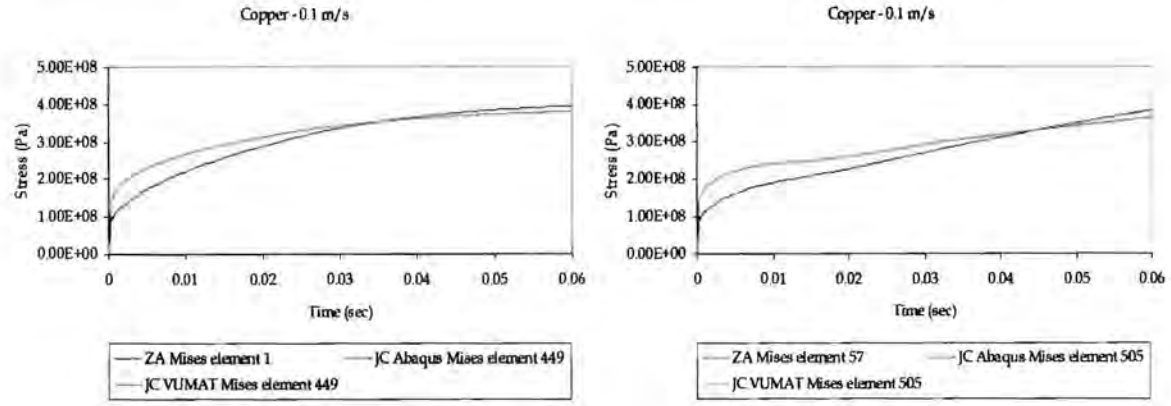
Figure C.121: Tension - temperature for OFHC Copper at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

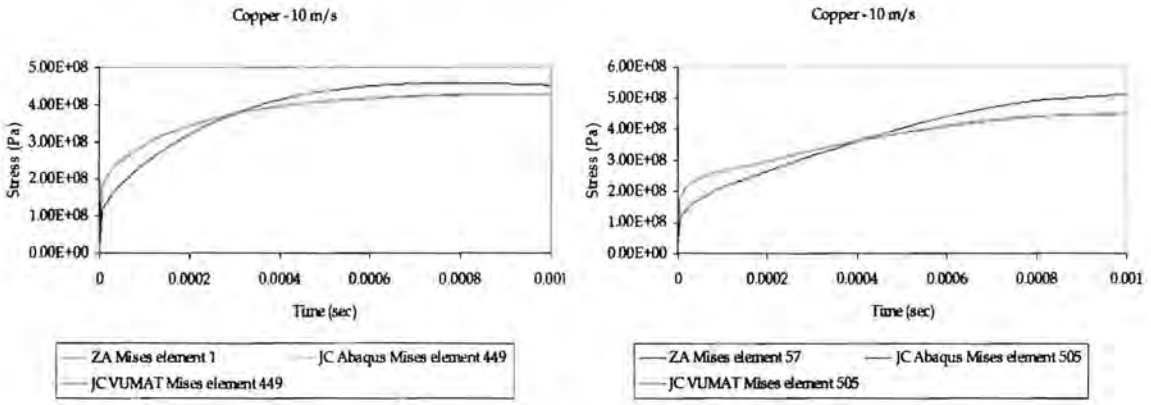
Figure C.122: Tension - temperature for OFHC Copper at 10 m/s
a) element 1 & 449 and b) element 57 & 505.

C.2.4.2 SHEAR



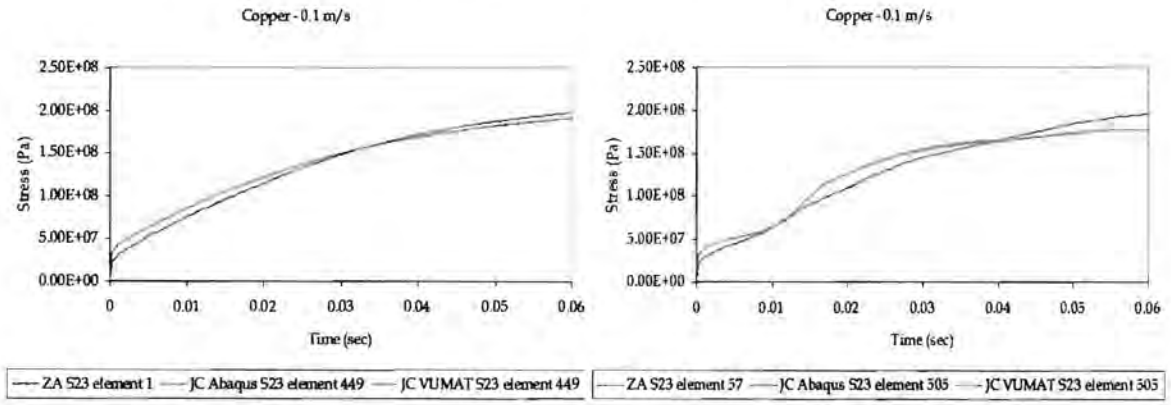
a) b)

Figure C.123: Shear - Mises stress for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



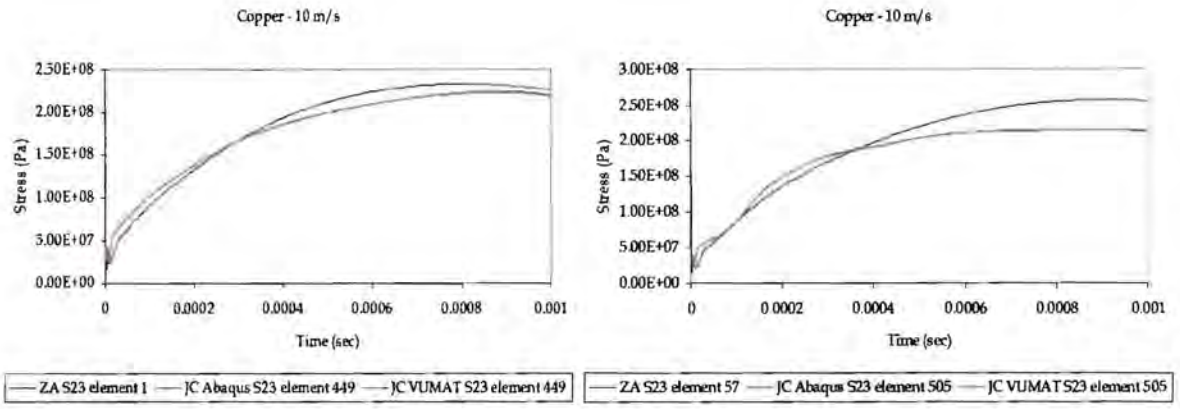
a) b)

Figure C.124: Shear - Mises stress for OFHC Copper at 10 m/s a) element 1 & 449 and b) element 57 & 505.



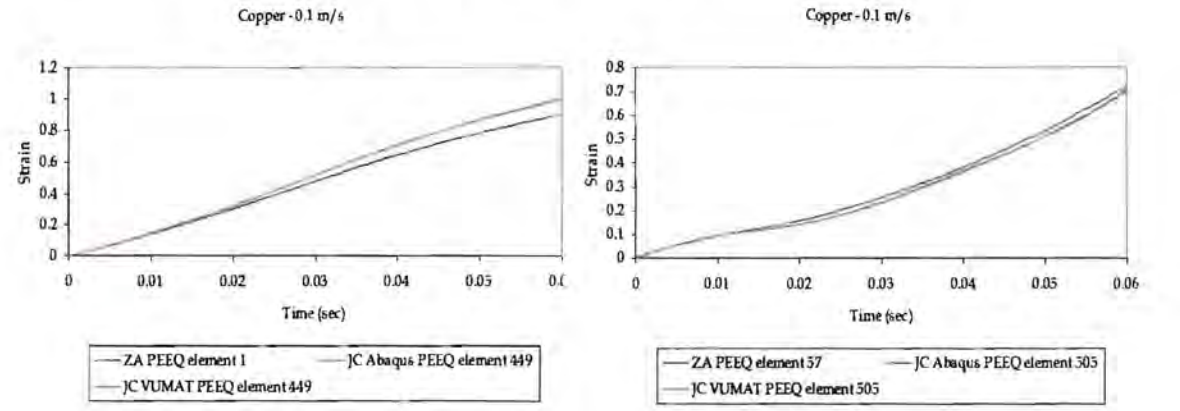
a) b)

Figure C.125: Shear - S23 stress component for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



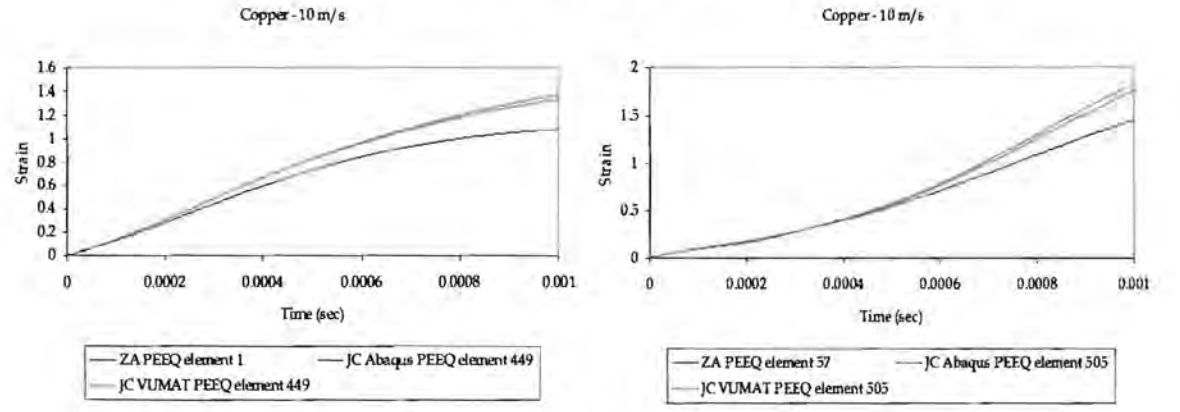
a) b)

Figure C.126: Shear - S23 stress component for OFHC Copper at 10 m/s a) element 1 & 449 and b) element 57 & 505.



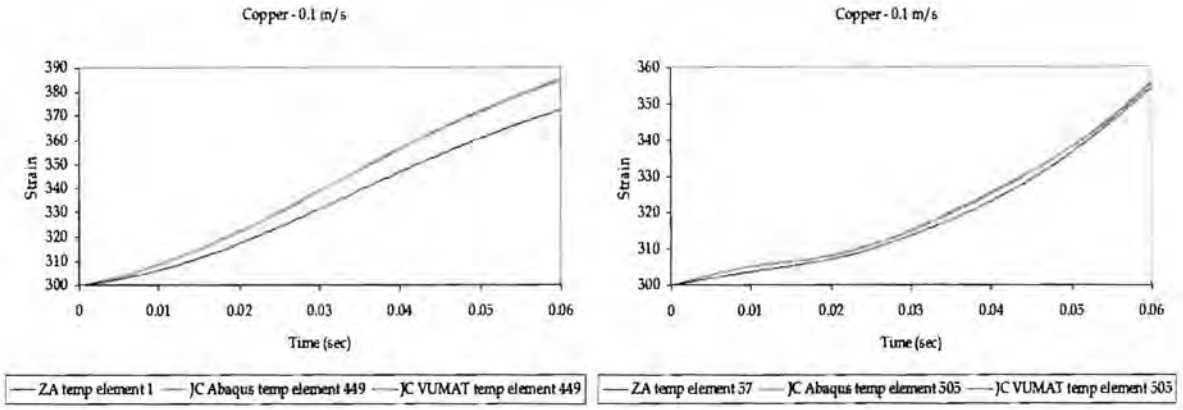
a) b)

Figure C.127: Shear - PEEQ for OFHC Copper at 0.1 m/s a) element 1 & 449 and b) element 57 & 505.



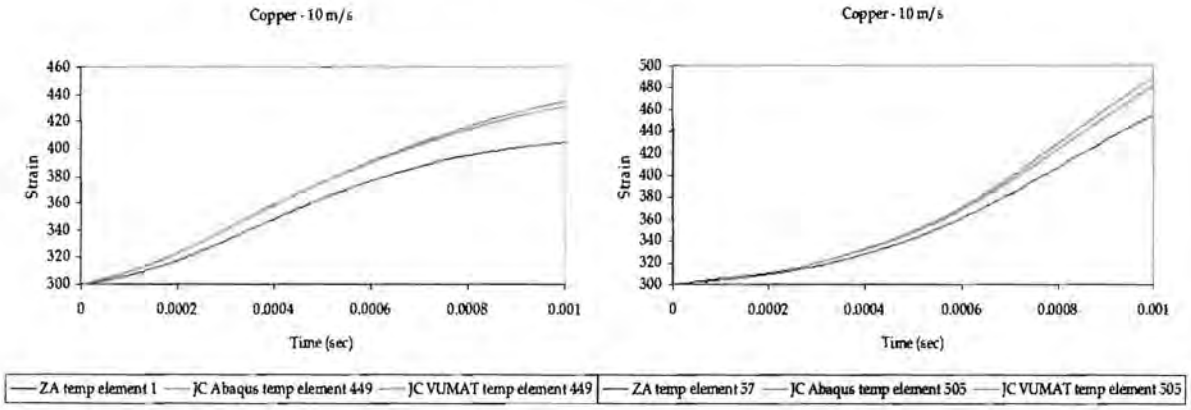
a) b)

Figure C.128: Shear - PEEQ for OFHC Copper at 10 m/s a) element 1 & 449 and b) element 57 & 505.



a) b)

Figure C.129: Shear - temperature for OFHC Copper at 0.1 m/s
a) element 1 & 449 and b) element 57 & 505.



a) b)

Figure C.130: Shear - temperature for OFHC Copper at 10 m/s
a) element 1 & 449 and b) element 57 & 505.

APPENDIX D

DEFORMATION RATIOS

In all the graphs presented the Abaqus and VUMAT Johnson-Cook results are indistinguishable.

D.1 ARMCO-IRON

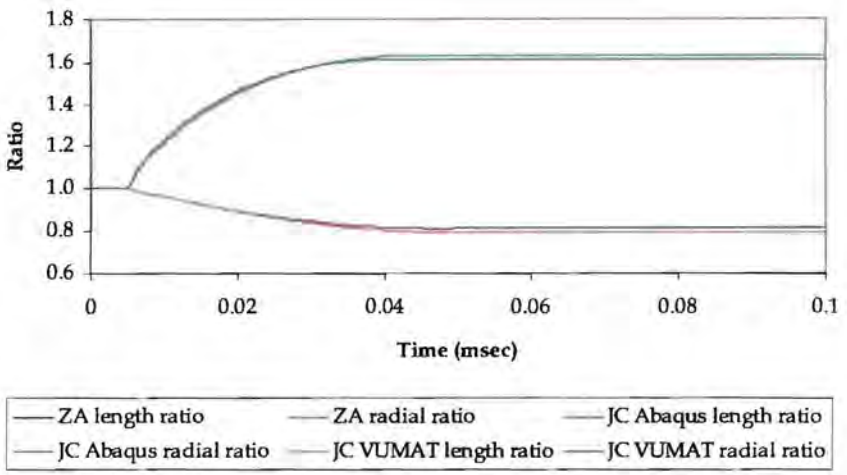


Figure D1: Deformation ratios for Armco-Iron at 197 m/s.

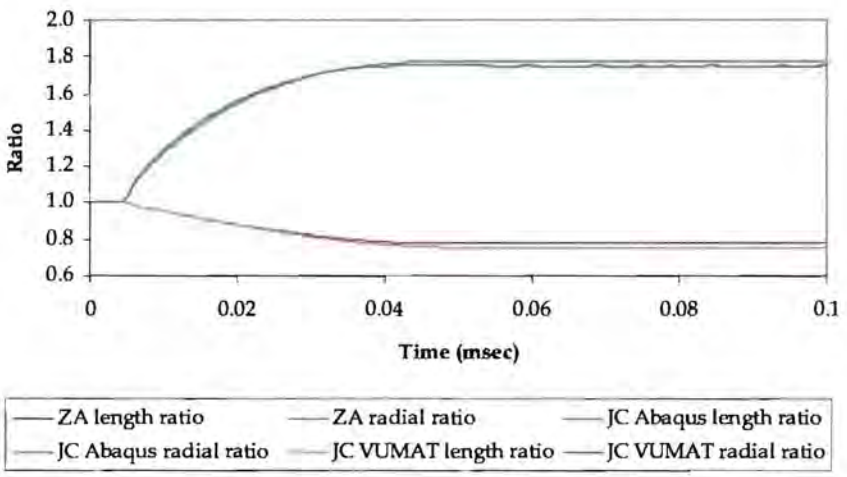


Figure D2: Deformation ratios for Armco-Iron at 221 m/s.

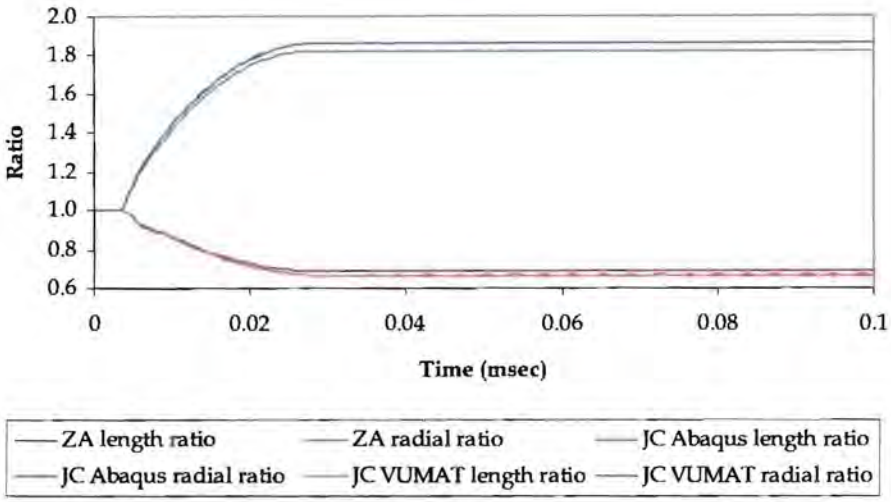


Figure D3: Deformation ratios for Armco-Iron at 279 m/s.

D.2 OFHC Copper

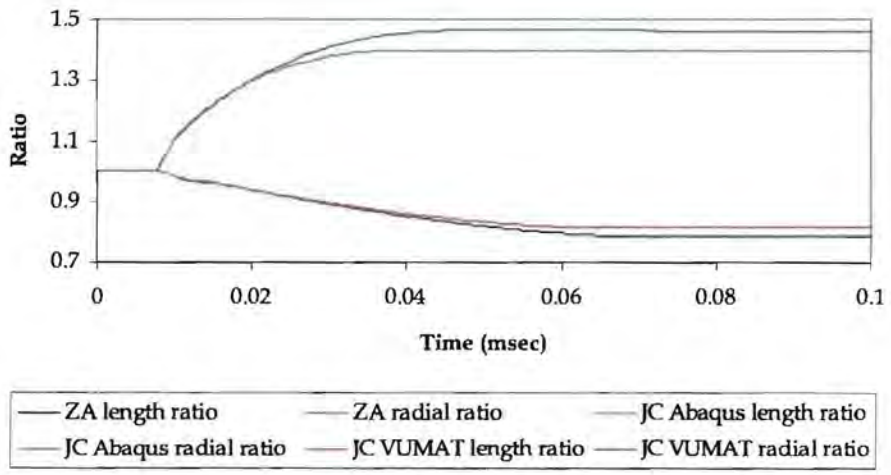


Figure D4: Deformation ratios for OFHC Copper at 130 m/s.

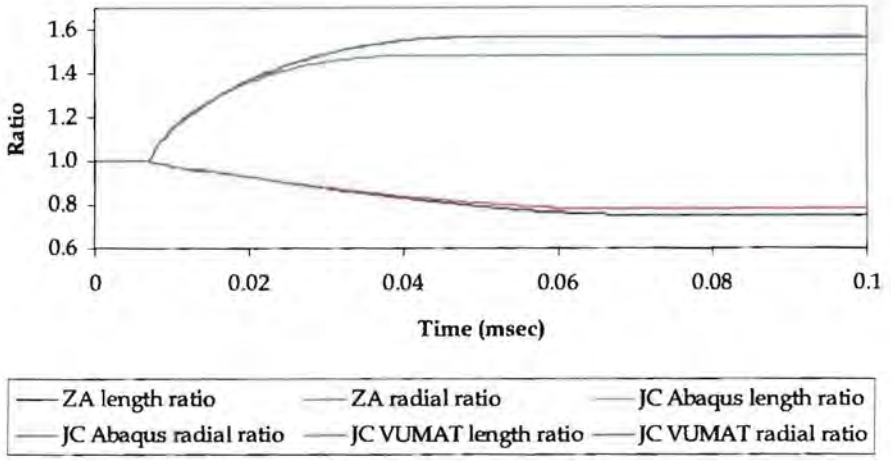


Figure D5: Deformation ratios for OFHC Copper at 146 m/s.

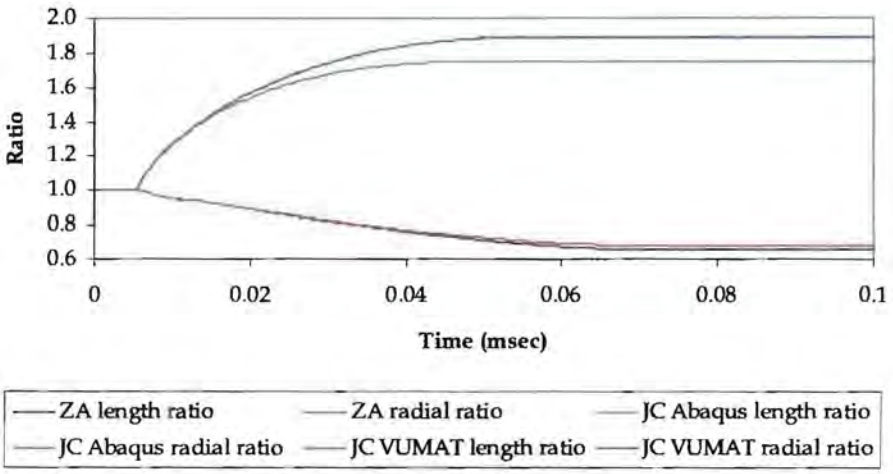


Figure D6: Deformation ratios for OFHC Copper at 190 m/s.

APPENDIX E

THEORETICAL INSTABILITY ANALYSIS

In Chapter 4 a localisation or instability phenomena is observed to occur in the tension tests as the number of elements is increased. As the number of elements is increased the finite element solution should asymptotically converge to a more accurate solution. The finite element model will also begin to capture the physics of the problem it is approximating such as localisation (as in this case). A simple theoretical analysis* is presented which defines the earliest position on the true stress-log strain curve where the localisation can occur in the tension test.

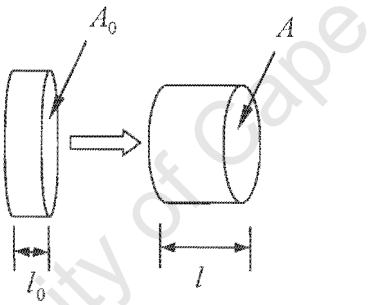


Figure E.1: Stretching of circular specimen in tension.

Figure E.1 shows a circular specimen with initial length and area, l_0 and A_0, and final length and area, l and A. An increase in length corresponds to a decrease in cross-sectional area if the volume is conserved. Beginning with the assumption of constant volume:

$$A_0 l_0 = Al$$

* Many of the ideas used in this Appendix were obtained from discussions with Mr Cloete. References to any original contributions towards these ideas are not known but the author does not claim credit for them.

If the final length is expressed as the initial length plus a length increment, and similarly for the area:

$$A_0 l_0 = (A_0 + \Delta A)(l_0 + \Delta l)$$

$$A_0 l_0 = A_0 l_0 + A_0 \Delta l + \Delta A l_0 + \Delta A \Delta l$$

The product of two incremental values is very small in comparison with the other terms and therefore is set to zero in all equations. After simplifying and re-arranging:

$$\Delta A = -\frac{A_0 \Delta l}{l_0} \text{ but } \Delta\varepsilon = \frac{\Delta l}{l_0}$$

$$\therefore \Delta A = -A_0 \Delta\varepsilon \tag{E.1}$$

Using the chain rule an increment in stress is written as:

$$\Delta\sigma = \frac{d\sigma}{d\varepsilon}\Delta\varepsilon \tag{E.2}$$

An increment in force is the difference between the final and initial forces:

$$\Delta F = \sigma A - \sigma_0 A_0$$

$$= (\sigma_0 + \Delta\sigma)(A_0 + \Delta A) - \sigma_0 A_0$$

$$= \sigma_0 A_0 + \sigma_0 \Delta A + \Delta\sigma A_0 + \Delta\sigma\Delta A - \sigma_0 A_0$$

Substitute the area and stress increments defined in equations E.1 and E.2:

$$\Delta F = \sigma_0(-A_0\Delta\varepsilon) + \left(\frac{d\sigma}{d\varepsilon}\Delta\varepsilon\right)A_0$$

The stability limit is reached when a positive strain increment causes the force increment to be zero or become negative. This corresponds to the peak of the force-displacement diagram as shown in Figure E.2.

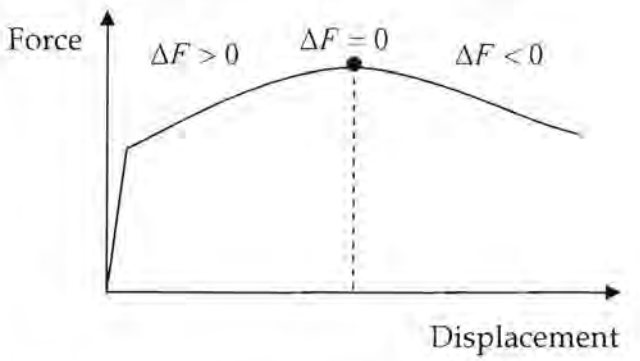


Figure E.2: Stability limit on force-displacement diagram.

Setting the force increment to zero defines the stability limit:

$$\Delta F = 0 = -A_0\sigma_0\Delta\varepsilon + A_0\frac{d\sigma}{d\varepsilon}\Delta\varepsilon$$

$$\therefore \frac{d\sigma}{d\varepsilon} = \sigma_0$$

Therefore for the true stress-log strain curve the stability limit is reached when the slope of the true stress-log strain curve at a point equals the stress value at the same point (Figure E.3 a)). This point is the earliest theoretical point where instability can occur and is before the peak of the true stress-log strain curve as shown in Figure E.3 a).

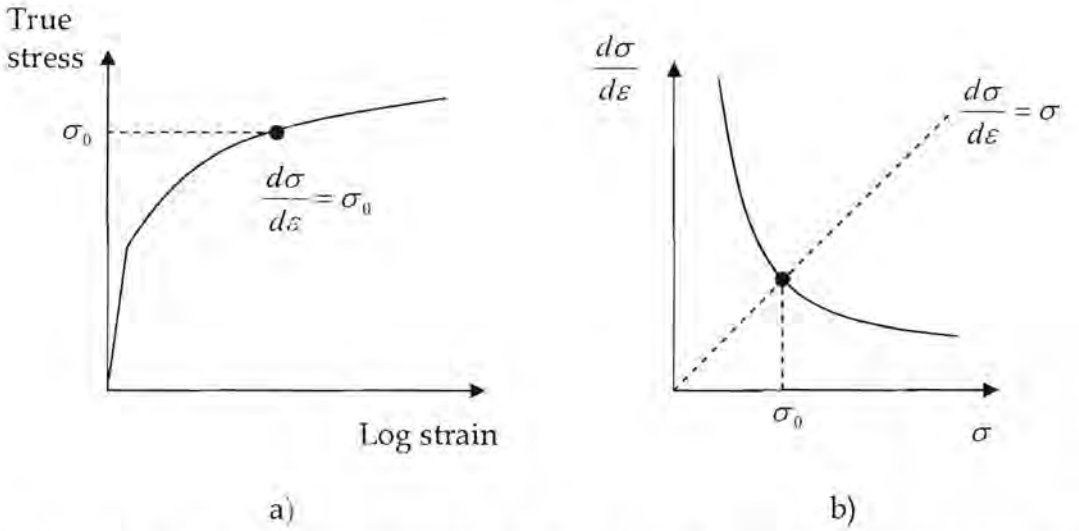


Figure E.3: a) True stress-log strain curve b) slope of true stress vs true stress curve.

The simulations of the tension test in Chapter 4 are based on a perfectly uniform geometry and material homogeneity. The position of the instability in the simulation results occurs much later along the true stress-log strain curve than the theory predicts (Table E.1). In an experiment the material and geometry would have imperfections which would "induce" the instability. The instability would therefore occur closer to the theoretically predicted position on the true stress-log strain curve. No experimental results are however available to substantiate this. The instability in the tension test simulations must therefore be induced by numerical imperfections (non-uniform stress distribution) due to any tolerances used as well as round-off errors as the simulations progress.

No. of elements	Theoretical stress (MPa)	Simulation stress (MPa)
8	312	No instability
64	310	417
512	314	403

Table E.1: Stress values when theoretical and simulation instability occurs for Copper at 10 m/s using JC VUMAT.

The theoretical stress in Table E.1 is found by plotting the simulation results of the slope of the stress-strain curve versus the stress values and finding where they are equal. In Figure E.4 the curves are for Copper (using JC VUMAT) at 0.1 m/s using 64 elements and the instability is predicted to occur at about 280 MPa in this case.

As shown in Table E.1, as the number of elements is increased the stress where instability occurs (simulation stress) tends toward the theoretical value (theoretical stress).

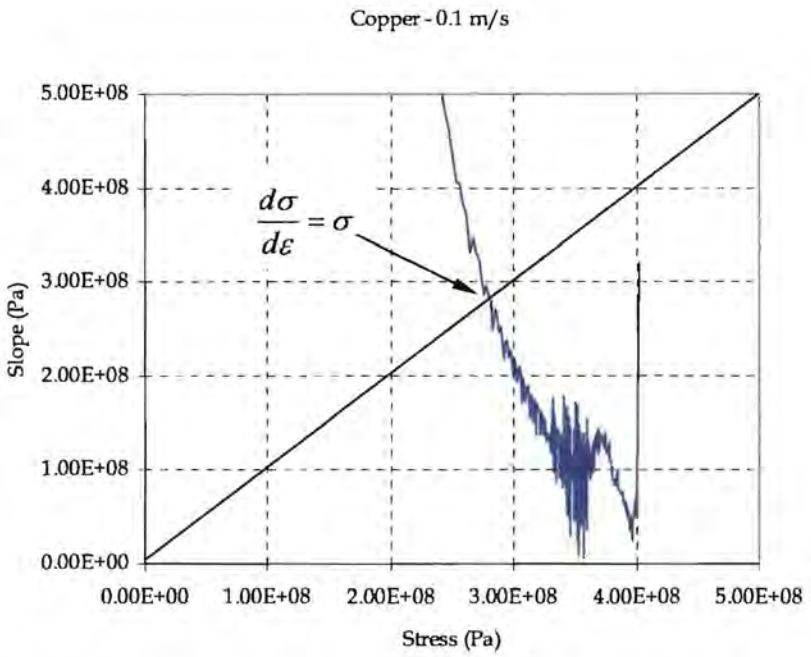


Figure E.4: Example of how to find theoretical instability stress value.