



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ  
ΕΡΓΑΣΤΗΡΙΟ ΜΗΧΑΝΙΚΗΣ ΚΑΙ ΑΝΤΟΧΗΣ ΤΩΝ ΥΛΙΚΩΝ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΜΕ ΘΕΜΑ :

*Αριθμητικές Μέθοδοι για την Μελέτη της Ανισοτροπίας σε Πορώδη Μέταλλα  
λόγω Πεπερασμένων Πλαστικών Παραμορφώσεων*

Φοιτητής : Δανάς Κωνσταντίνος

**Εξεταστική Επιτροπή**

Καθ. Αράβας Νικόλαος (Επιβλέπων)

Αν. Καθ. Γιαννακόπουλος Αντώνης

Επ. Καθ. Καραμάνος Σπύρος



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΥΠΗΡΕΣΙΑ ΒΙΒΛΙΟΘΗΚΗΣ & ΠΛΗΡΟΦΟΡΗΣΗΣ  
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»

Αριθ. Εισ.: 747/1  
Ημερ. Εισ.: 02-03-2004  
Δωρεά:  
Ταξιθετικός Κωδικός: ΠΤ ΜΜΒ  
2002  
ΔΑΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ



004000070525

## Σημείωμα του συγγραφέα

Θα ήθελα πρώτα από όλα να ευχαριστήσω τον καθηγητή και επιβλέποντα της διπλωματικής μου εργασίας κ. Α. Αράβα για την εμπιστοσύνη που μου έδειξε στο συγκεκριμένο θέμα έρευνας αλλά και τη σημαντική βοήθεια που μου προσέφερε καθ' όλη τη διάρκεια της συνεργασίας μας. Για τις συμβουλές, τον τρόπο σκέψης που μου μετέδωσε, για την γνώση που έλαβα. Επίσης, θα ήθελα να ευχαριστήσω και τους κυρίους Αν. Καθ. Α. Γιαννακόπουλο και Επ. Καθ. Σ. Καραμάνο για τη μελέτη και αξιολόγηση της εργασίας μου.

Ακόμη θα ήθελα να ευχαριστήσω όλα τα μέλη της ακαδημαϊκής κοινότητας του τμήματός μας για τις πληροφορίες και τη γνώση που μου προσέφεραν σε ακαδημαϊκά θέματα και όχι μόνο. Ιδιαίτερες ευχαριστίες στους φίλους μου και συμφοιτητές μου.

Τέλος, ένα μεγάλο ευχαριστώ σε όλη την οικογένειά μου για την συμπαράστασή τους καθ' όλη τη διάρκεια των σπουδών μου αλλά και για την ατέλειωτη αγάπη τους.

# ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. Εισαγωγή</b>	
Περίληψη.....	1
<b>2. Ανισότροπο Μοντέλο και Μοντέλο του Gurson</b>	
2.1 Θεωρία των μεγάλων παραμορφώσεων.....	3
2.2 Ελαστο – πλαστικές καταστατικές εξισώσεις.....	4
2.2.1 Ελαστικότητα.....	5
2.2.2 Πλαστικότητα.....	5
2.3 Μοντέλο του <i>Gurson</i> για πορώδη υλικά.....	7
2.4 Ανισότροπο μοντέλο για πορώδη υλικά.....	9
2.4.1 Ελαστικές καταστατικές εξισώσεις.....	10
2.4.2 Συνθήκη διαρροής και κανόνας πλαστικής διαρροής.....	11
2.4.3 Εξέλιξη της μικροδομής.....	12
2.4.4 Ρυθμός μεταβολής των ελαστο – πλαστικών εξισώσεων.....	17
<b>3. Αριθμητική ολοκλήρωση των καταστατικών εξισώσεων</b>	
3.1 Υπολογισμός του τανυστή της κλίσης της κίνησης $\mathbf{F}$ .....	20
3.2 Αριθμητική ολοκλήρωση μοντέλου <i>Gurson</i> .....	21
3.3 Αριθμητική ολοκλήρωση του ανισότροπου μοντέλου.....	25
3.3.1 Forward Euler στο ανισότροπο μοντέλο.....	25
3.3.2 Backward Euler στο ανισότροπο μοντέλο.....	28
3.3.3 Σύγκριση της backward και forward μεθόδου.....	31
3.4 Υπολογισμός των τανυστών του Eshelby και του τανυστή $\mathbf{Q}$ .....	33
<b>4. Εφαρμογές</b>	
4.1 Όλκιμη θραύση στο πρόβλημα της ρωγμής.....	41
4.2 Μονοαξονικός εφελκυσμός.....	54
<b>Παράρτημα Α</b>	
Αρχείο δημιουργίας καννάβου για το πρόβλημα της ρωγμής	
<b>Παράρτημα Β</b>	
Αρχείο δημιουργίας καννάβου για το πρόβλημα του μονοαξονικού εφελκυσμού	
<b>Παράρτημα Γ</b>	
Η umat υπορουτίνα mahesh_back.f	

## Εισαγωγή

Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με την αριθμητική επίλυση προβλημάτων που αφορούν πορώδη μέταλλα. Συνολικά θα αναπτύξουμε δύο μοντέλα:

- 1) το μοντέλο του **Gurson** για ισότροπα πορώδη υλικά
- 2) ένα ανισότροπο μοντέλο

Στο δεύτερο κεφάλαιο θα ασχοληθούμε με την περιγραφή του μοντέλου του **Gurson** και του ανισότροπου μοντέλου. Πιο συγκεκριμένα θα αναπτύξουμε τις καταστατικές εξισώσεις και τη συνθήκη πλαστικής φόρτισης που διέπουν τα δύο προβλήματα.

Κάνοντας μία σύντομη εισαγωγή για τα δύο μοντέλα μπορούμε να πούμε ότι και τα δύο αφορούν μεταλλικά υλικά με πόρους και τέτοια υλικά στην πραγματικότητα είναι αυτά που προέρχονται από κονιομεταλλουργία αλλά και υλικά που περιέχουν εγκλείσματα (όπως άνθρακα, νικέλιο, τιτάνιο κ.α.) και κάτω από συνθήκες φόρτισης δημιουργούνται πόροι από την αποκόλληση ή θραύση των εγκλεισμάτων. Ειδικότερα, ο **Gurson** [9] θεώρησε το 1977 ένα υλικό το οποίο είναι πορώδες και αρχικά οι πόροι έχουν σχήμα σφαιρικό. Υπέθεσε, έπειτα, ότι όταν το υλικό φορτίζεται μεταβάλλεται το μέγεθος των πόρων – συνεπώς και το πορώδες του υλικού (το κλάσμα των πόρων στο συνεχές μέσο) – αλλά το σχήμα τους παραμένει σφαιρικό.

Αντιθέτως, στο ανισότροπο μοντέλο γίνεται η υπόθεση ότι οι πόροι αρχικά έχουν σφαιρικό σχήμα, δηλαδή το υλικό είναι ισότροπο, αλλά κατά την φόρτιση οι πόροι αλλάζουν όγκο, σχήμα (από σφαιρικό →ελλειπτικό) και προσανατολισμό. Συνεπώς το πρόβλημα γίνεται πιο δύσκολο αφού το υλικό παύει να είναι ισότροπο και απαιτούνται περισσότερες καταστατικές μεταβλητές για να επιλυθεί.

Στο τρίτο κεφάλαιο θα παρουσιαστούν οι αριθμητικές μέθοδοι ολοκλήρωσης των δύο παραπάνω μοντέλων. Η επίλυσή τους γίνεται από υπορουτίνα σε γλώσσα προγραμματισμού Fortran σε συνδυασμό με το πρόγραμμα πεπερασμένων στοιχείων Abaqus.

Στο τέταρτο κεφάλαιο θα παρουσιαστεί η εφαρμογή των δύο αυτών μοντέλων σε συγκεκριμένα προβλήματα και θα γίνει σύγκριση των αποτελεσμάτων τους.

Σε ολόκληρη την έκταση της εργασίας χρησιμοποιούνται τυποποιημένοι συμβολισμοί. Κρίνεται επομένως σκόπιμο να γίνει μία σύντομη αναφορά σε αυτούς. Συγκεκριμένα, κεφαλαία και μαυρισμένα σύμβολα δηλώνουν τανυστές των οποίων η τάξη υποδεικνύεται από το κείμενο κάθε φορά. Οι συνιστώσες ενός τανυστή αναφέρονται σε ένα σταθερό καρτεσιανό σύστημα συντεταγμένων και η σύμβαση της αθροίσεως χρησιμοποιείται για επαναλαμβανόμενους δείκτες, εκτός αν αναφέρεται διαφορετικά μέσα στο κείμενο. Το γράμμα  $T$  υποδεικνύει την αναστροφή ενός τανυστή δευτέρας τάξεως και η υπερθετημένη τελεία την υλική παράγωγο. Έστω  $\mathbf{a}$  και  $\mathbf{b}$  διανύσματα,  $\mathbf{A}$  και  $\mathbf{B}$  τανυστές δευτέρας τάξεως και  $\mathbf{C}$  τανυστής τετάρτης τάξεως. Για τους παραπάνω τανυστές ορίζονται τα εξής γινόμενα :

$$(\mathbf{a}\mathbf{b})_{ij} = a_i b_j,$$

$$(\mathbf{A}\cdot\mathbf{a})_{ij} = A_{ij} a_j, \quad (\mathbf{a}\cdot\mathbf{A})_{ij} = a_j A_{ji},$$

$$(\mathbf{A}\cdot\mathbf{B})_{ij} = A_{ik} B_{kj},$$

$$\mathbf{a}\cdot\mathbf{A}\cdot\mathbf{b} = a_i A_{ij} b_j,$$

$$(\mathbf{A}:\mathbf{C})_{ij} = A_{kl} C_{kl ij}, \quad (\mathbf{C}:\mathbf{A})_{ij} = C_{ijkl} A_{kl},$$

$$\mathbf{A}:\mathbf{B} = A_{ij} B_{ij},$$

$$(\mathbf{A}\mathbf{B})_{ijkl} = A_{ij} B_{kl},$$

$$\mathbf{A}:\mathbf{C}:\mathbf{B} = A_{ij} C_{ijkl} B_{kl}.$$

## 2. Ανισότροπο μοντέλο και μοντέλο Gurson

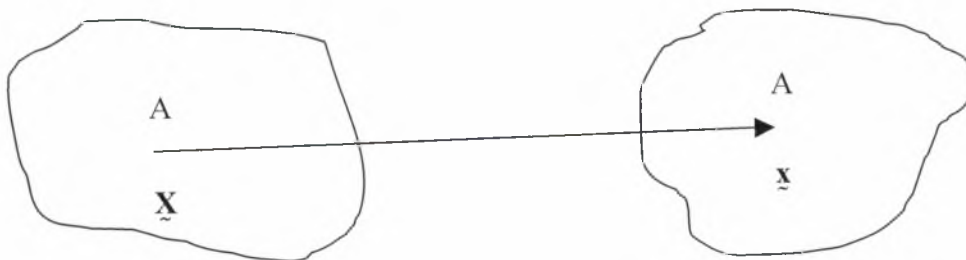
Αρχικά, θα αναπτύξουμε τη θεωρία των πεπερασμένων παραμορφώσεων οι οποίες απαιτούνται για την ανάπτυξη των δύο μοντέλων που θα μας απασχολήσουν στην παρούσα διπλωματική εργασία. Έπειτα, θα παρουσιάσουμε την γενική μορφή των ελαστο – πλαστικών καταστατικών εξισώσεων για ανεξάρτητα της ταχύτητας παραμορφώσεως υλικά που υπόκεινται σε πεπερασμένες παραμορφώσεις. Κατόπιν θα τις χρησιμοποιήσουμε για να περιγράψουμε το ανισότροπο μοντέλο και το μοντέλο του *Gurson*.

### 2.1 Θεωρία των πεπερασμένων παραμορφώσεων

Αρχικά θα γίνει μία σύντομη παρουσίαση της θεωρίας των πεπερασμένων παραμορφώσεων που θα αναπτυχθούν περισσότερο σε επίπεδο αριθμητικής ολοκλήρωσης στο κεφάλαιο 3.

Έστω ότι τη χρονική στιγμή  $t_0$ , ένα συνεχές μέσο καταλαμβάνει μία περιοχή  $B_0$  στο χώρο. Η θέση ενός υλικού σημείου  $A$  του συνεχούς μέσου, στη χρονική στιγμή  $t_0$  περιγράφεται από το διάνυσμα θέσης  $\underline{X}$ . Έστω ότι τη χρονική στιγμή  $\Delta t$  το υλικό σημείο  $A$  έχει μετατοπιστεί και ορίζεται από ένα νέο διάνυσμα θέσης  $\underline{x}$  και το συνεχές μέσο καταλαμβάνει μία νέα περιοχή  $B$  στο χώρο. Η εξίσωση που περιγράφει την κίνηση του συνεχούς μέσου είναι της μορφής :

$$\underline{x} = \underline{x}(\underline{X}, t) \quad (2.1.1)$$

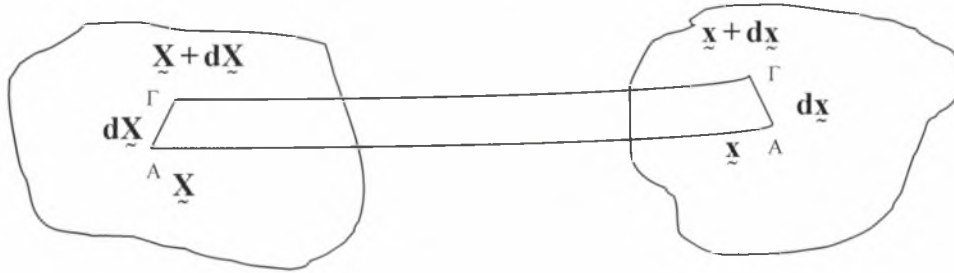


Ο τανυστής της κλίσης της κίνησης ορίζεται ως :

$$\mathbf{F}(\underline{X}, \underline{x}) = \frac{\partial \underline{x}}{\partial \underline{X}} \quad (2.1.2)$$

και περιέχει όλη την πληροφορία σχετικά με την παραμόρφωση.

Η θέση ενός δεύτερου υλικού σημείου  $\Gamma$  θα μπορούσε να καθοριστεί στην απαραμόρφωτη και στην παραμορφωμένη κατάσταση από τα διανύσματα θέσης  $\underline{\underline{X}} + d\underline{\underline{X}}$  και  $\underline{\underline{x}} + d\underline{\underline{x}}$  αντίστοιχα, όπου  $d\underline{\underline{x}} = \mathbf{F} \cdot d\underline{\underline{X}}$ .



Παίρνοντας τη χρονική μεταβολή του διανύσματος  $d\underline{\underline{x}}$

$$\frac{\partial(d\underline{\underline{x}})}{\partial t} = \frac{\partial(\mathbf{F} \cdot d\underline{\underline{X}})}{\partial t} = \dot{\mathbf{F}} \cdot d\underline{\underline{X}} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} \cdot d\underline{\underline{x}} = \mathbf{L} \cdot d\underline{\underline{x}} \quad (2.1.3)$$

προκύπτει ο τανυστής  $\mathbf{L}$  ο οποίος εκφράζει την κλίση της ταχύτητας στο χώρο B,

$$\mathbf{L}(\underline{\underline{x}}, t) = \frac{\partial \mathbf{v}(\underline{\underline{x}}, t)}{\partial \underline{\underline{x}}} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1} \quad (2.1.4)$$

Κάθε τανυστής μπορεί να γραφεί ως το άθροισμα ενός συμμετρικού και ενός αντισυμμετρικού μέρους. Ορίζουμε

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T) \quad (2.1.5)$$

τον τανυστή ρυθμού παραμορφώσεως ως το συμμετρικό μέρος του  $\mathbf{L}$  και

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T) \quad (2.1.6)$$

τον τανυστή στροβιλότητας (spin tensor) ως το αντισυμμετρικό μέρος του  $\mathbf{L}$ .

## 2.2 Ελαστο – πλαστικές καταστατικές εξισώσεις

Ο τανυστής των παραμορφώσεων μπορεί να γραφεί ως το άθροισμα ενός ελαστικού και ενός πλαστικού μέρους :

$$\mathbf{D} = \mathbf{D}^e + \mathbf{D}^p \quad (2.2.1)$$



### 2.2.1 Ελαστικότητα

Η βασική εξίσωση της υπερελαστικότητας για μεγάλες παραμορφώσεις δίνεται από την παρακάτω σχέση :

$$\overset{\nabla}{\sigma} = \mathcal{C}^e : \mathbf{D}^e \quad \text{ή} \quad \mathbf{D}^e = \mathcal{C}^{e-1} : \overset{\nabla}{\sigma} \quad (2.2.2)$$

όπου  $\overset{\nabla}{\sigma}$  είναι η παράγωγος *Jaumann* η οποία αναλύεται στα επόμενα κεφάλαια,  $\mathcal{C}^e$  είναι ο ελαστικός τανυστής τετάρτης τάξεως και  $\mathbf{D}^e$  το ελαστικό μέρος του ρυθμού παραμορφώσεως.

### 2.2.2 Πλαστικότητα

Η γενική μορφή της συνθήκης διαρροής είναι :

$$\Phi(\sigma, s^a) = 0 \quad (2.2.3)$$

όπου  $s^a (a=1, 2, \dots, m)$  είναι ένα σύνολο μεταβλητών που ονομάζονται «καταστατικές μεταβλητές». Για παράδειγμα στο μοντέλο του *von Mises* απαιτείται ο προσδιορισμός μίας μόνον καταστατικής μεταβλητής ( $\bar{\varepsilon}^p \rightarrow$  ισοδύναμη πλαστική παραμόρφωση), για το μοντέλο του *Gurson* για πορώδη υλικά δύο καταστατικών μεταβλητών ( $\bar{\varepsilon}^p, f$ ) και για το ανισότροπο μοντέλο επτά καταστατικών μεταβλητών ( $\bar{\varepsilon}^p, f, w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)} = \mathbf{n}^{(1)} \times \mathbf{n}^{(2)}$ ). Οι επιμέρους καταστατικές μεταβλητές ορίζονται όπου παρουσιάζονται και τα μοντέλα.

Το πλαστικό πρόβλημα καθορίζεται από τις παρακάτω εξισώσεις :

➤ Κριτήριο διαρροής

$$\Phi(\sigma, s^a) \leq 0 \quad (2.2.4)$$

➤ Νόμος πλαστικής διαρροής

$$\mathbf{D}^p = \dot{\lambda} \mathbf{N}(\sigma, s^a) = \dot{\lambda} \frac{\partial \Phi}{\partial \sigma} \quad \text{με} \quad \dot{\lambda} \geq 0 \quad (2.2.5)$$

➤ Εξισώσεις εξελίξεως καταστατικών μεταβλητών

$$\dot{s}^a = G^a(\sigma, s^b, \dot{\varepsilon}^p) \quad (2.2.6)$$

Για ανεξάρτητα της ταχύτητας παραμορφώσεως υλικά, οι καταστατικές συναρτήσεις  $G^a$  είναι ομογενείς συναρτήσεις πρώτου βαθμού ως προς το ρυθμό πλαστικής παραμόρφωσης  $\dot{\epsilon}^P$  και επομένως η εξίσωση (2.2.6) μπορεί να γραφεί ως εξής:

$$\dot{s}^a = \dot{\lambda} G^a(\sigma, s^\beta, \mathbf{N}) \quad (2.2.7)$$

➤ *Συνθήκη συνέχειας*

Η συνθήκη της συνέχειας στην περίπτωση των πεπερασμένων παραμορφώσεων μπορεί να γραφεί ως εξής ( Dafalias [6] ) :

$$\begin{aligned} \dot{\Phi} &= \frac{\partial \Phi}{\partial \sigma} : \dot{\sigma} + \sum_a \frac{\partial \Phi}{\partial s^a} \dot{s}^a = \frac{\partial \Phi}{\partial \sigma} : \dot{\sigma} + \dot{\lambda} \cdot \sum_a \frac{\partial \Phi}{\partial s^a} g^a = \\ &= \frac{\partial \Phi}{\partial \sigma} : \overset{\nabla}{\dot{\sigma}} + \dot{\lambda} \cdot \sum_a \frac{\partial \Phi}{\partial s^a} g^a = \frac{\partial \Phi}{\partial \sigma} : \overset{\circ}{\dot{\sigma}} + \dot{\lambda} \cdot \sum_a \frac{\partial \Phi}{\partial s^a} g^a = 0 \end{aligned} \quad (2.2.8)$$

Από τη συνθήκη της συνέχειας προκύπτει μία σχέση η οποία μας επιτρέπει τον υπολογισμό του  $\dot{\lambda}$  :

$$\dot{\Phi} = 0 \Rightarrow \dot{\lambda} = \frac{1}{H} \frac{\partial \Phi}{\partial \sigma} : \dot{\sigma} = \frac{1}{L} \frac{\partial \Phi}{\partial \sigma} : \mathcal{C}^e : \mathbf{D} \quad (2.2.9)$$

όπου

$$H = - \sum_a \frac{\partial \Phi}{\partial s^a} g^a \quad \text{και} \quad L = H + \frac{\partial \Phi}{\partial \sigma} : \mathcal{C}^e : \mathbf{N} \quad (2.2.10)$$

Η παράμετρος  $H$  καθορίζει την κατάσταση του υλικού. Δηλαδή αν:

- ◇  $H > 0$  τότε έχουμε σκλήρυνση ( hardening )
- ◇  $H < 0$  τότε έχουμε πλάυνση ( softening )
- ◇  $H = 0$  τότε έχουμε τέλεια πλαστικότητα ( perfect plasticity )

Παρατηρούμε ότι η πρώτη έκφραση στη σχέση (2.2.9) δεν ορίζεται στην περίπτωση της τέλει πλαστικότητας ( $H = 0$ ). Αντιθέτως η δεύτερη έκφραση ορίζεται και στις τρεις περιπτώσεις του  $H$  και προκύπτει πάντα θετική ( $L > 0$ ). Το  $\dot{\lambda}$  πρέπει να είναι πάντα θετικό κατά την πλαστική διαρροή και να μηδενίζεται όταν έχουμε ελαστική αποφόρτιση ή ουδέτερη φόρτιση. Συνεπώς ο πλήρης ορισμός του  $\dot{\lambda}$ , ο οποίος καθορίζει και τις συνθήκες για συνεχή πλαστική διαρροή είναι ο ακόλουθος :

$$\dot{\lambda} = \frac{a}{H} \frac{\partial \Phi}{\partial \sigma} : \dot{\sigma} = \frac{a}{L} \frac{\partial \Phi}{\partial \sigma} : \mathcal{C}^e : \mathbf{D}, \quad a = \begin{cases} 1 & \text{αν } \Phi = 0 \text{ και } \frac{\partial \Phi}{\partial \sigma} : \mathcal{C}^e : \mathbf{D} > 0 \\ 0 & \text{διαφορετικά} \end{cases} \quad (2.2.11)$$

### 2.3 Μοντέλο του Gurson για πορώδη υλικά

Το μοντέλο του **Gurson** [9] για υλικά με οπές εξαρτάται από τον αποκλίνοντα τανυστή των τάσεων  $\sigma'$  και από την υδροστατική πίεση  $p$ . Για την επίλυση και την περιγραφή του παραπάνω μοντέλου απαιτείται ο προσδιορισμός δύο καταστατικών μεταβλητών.

- i. η ισοδύναμη μικροσκοπική πλαστική παραμόρφωση  $\varepsilon_m^p$
- ii. το πορώδες  $f$ , το οποίο ορίζεται ως το κλάσμα του όγκου των οπών στο υλικό

Το κριτήριο διαρροής παίρνει την μορφή :

$$\Phi(\sigma, \varepsilon_m^p, f) = \left[ \frac{q}{\sigma_m(\varepsilon_m^p)} \right]^2 + 2f \cosh \left[ \frac{3p}{2\sigma_m(\varepsilon_m^p)} \right] - (1 + f^2) = 0 \quad (2.3.1)$$

όπου  $q$  είναι η ισοδύναμη τάση κατά von Mises,  $p = -\sigma_{kk}/3$  (υδροστατική πίεση) και  $\sigma_m$  η τάση διαρροής του συμπαγούς μετάλλου.

Ο κανόνας πλαστικής διαρροής παίρνει τη μορφή:

$$\mathbf{D}^p = \dot{\lambda} \mathbf{N} = \dot{\lambda} \frac{\partial \Phi}{\partial \sigma} = \frac{\dot{\lambda}}{\sigma_m} \left[ \frac{3\sigma'}{\sigma_m} - f \sinh \left( \frac{3p}{2\sigma_m} \right) \mathbf{I} \right] \quad (2.3.2)$$

Εξισώσεις εξελίξεως των καταστατικών μεταβλητών

$$a. \dot{s}^1 = \dot{\varepsilon}_m^p$$

Η εξίσωση για το  $\dot{\varepsilon}_m^p$  προκύπτει εάν εξισωθεί η μεταβολή του μακροσκοπικού πλαστικού έργου  $\sigma : \mathbf{D}^p$  με τη μεταβολή του μικροσκοπικού πλαστικού έργου  $(1-f)\sigma_m \dot{\varepsilon}_m^p$  οπότε έχουμε :

$$\dot{s}^1 = \dot{\varepsilon}_m^p = \frac{\sigma : \mathbf{D}^p}{(1-f)\sigma_m} \equiv G^1(\sigma, \varepsilon_m^p, f, \mathbf{D}^p) \quad (2.3.3)$$

Συνεπώς η τελική έκφραση για τη μικροσκοπική πλαστική παραμόρφωση είναι :

$$\dot{s}^1 = \dot{\varepsilon}_m^p = \frac{\dot{\lambda}}{(1-f)\sigma_m^2} \left[ \frac{3q^2}{\sigma_m} + 3f p \sinh\left(\frac{3p}{2\sigma_m}\right) \right] = \dot{\lambda} g^1(\sigma, \varepsilon_m^p, f) \quad (2.3.4)$$

b.  $\dot{s}^2 = \dot{f}$

Υποθέτοντας ότι το υλικό είναι πλαστικά ασυμπίεστο και αγνοώντας την ελαστική συνεισφορά στη μεταβολή του πορώδους  $f$  προκύπτει ότι :

$$\dot{s}^2 = \dot{f} = (1-f) \mathbf{D}_{kk}^p \equiv G^2(f, \mathbf{D}^p) \quad (2.3.5)$$

οπότε η τελική έκφραση για το ρυθμό μεταβολής του πορώδους είναι :

$$\dot{s}^2 = \dot{f} = -\dot{\lambda} \frac{3(1-f)f}{\sigma_m} \sinh\left(\frac{3p}{2\sigma_m}\right) = \dot{\lambda} g^2(\sigma, \varepsilon_m^p, f) \quad (2.3.6)$$

Επίσης η παράμετρος σκλήρυνσης ορίζεται με βάση τη σχέση ως εξής :

$$H = -\left( \frac{\partial \Phi}{\partial \varepsilon_m^p} g^1 + \frac{\partial \Phi}{\partial f} g^2 \right) \quad (2.3.7)$$

Αν υποθεθεί γραμμική ισότροπη ελαστικότητα το  $L$  της σχέσης (2.2.10) παίρνει τη μορφή :

$$L = H + \frac{3}{\sigma_m^2} \left[ \frac{4Gq^2}{\sigma_m^2} + 3K f^2 \sinh^2\left(\frac{3p}{2\sigma_m}\right) \right] \quad (2.3.8)$$

Γενικά το  $\dot{\lambda} \geq 0$  , οπότε προκύπτει, στο μοντέλο του **Gurson**, ότι η σχέση η οποία ορίζει την πλαστική φόρτιση όταν  $\Phi = 0$  πρέπει να ικανοποιεί την παρακάτω ανισότητα :

$$\frac{\partial \Phi}{\partial \sigma} : \mathcal{C}^e : \mathbf{D} > 0 \Rightarrow \left[ \frac{6G\sigma'}{\sigma_m^2} - 3K f \sinh\left(\frac{3p}{2\sigma_m}\right) \mathbf{I} \right] > 0 \quad (2.3.9)$$

Παρατήρηση: Αν θέσουμε  $f = 0$  τότε από η σχέση (2.3.1) παίρνει τη μορφή :

$$\Phi = q^2 - \sigma_m^2 = 0 \Rightarrow \Phi = \sigma_e^2 - \sigma_y^2 = 0, \quad (q = \sigma_e \ \& \ \sigma_m = \sigma_y)$$

που είναι το κριτήριο διαρροής κατά **von Mises**.

## 2.4 Ανισότροπο μοντέλο για πορώδη υλικά

Παρακάτω περιγράφεται το ανισότροπο ελαστο – πλαστικό καταστατικό μοντέλο για πορώδη μέταλλα. Αρχικά υποθέτουμε ότι οι πόροι είναι σφαιρικοί και ομοιόμορφα κατανεμημένοι στο ισότροπο σώμα και συνεπώς το υλικό είναι αρχικά ισότροπο. Όταν το σώμα υφίσταται πλαστικές παραμορφώσεις, υποθέτουμε ότι οι πόροι αλλάζουν όγκο και παίρνουν ελλειπτικό σχήμα σε αντίθεση με το μοντέλο του **Gurson** στο οποίο οι πόροι παραμένουν σφαιρικοί σε όλη τη διάρκεια της παραμόρφωσης. Σε κάθε σημείο του ομογενοποιημένου πορώδους υλικού καθορίζεται ένα αντιπροσωπευτικό τοπικό ελλειψοειδές. Έστω  $(\mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)} = \mathbf{n}^{(1)} \times \mathbf{n}^{(2)})$  τα μοναδιαία διανύσματα στις κατευθύνσεις των κυρίων αξόνων του τοπικού ελλειψοειδούς,  $(2a_1, 2a_2, 2a_3)$  τα αντίστοιχα μήκη των κύριων αξόνων του ελλειψοειδούς και επίσης ορίζουμε και τους λόγους των κυρίων αξόνων,  $w_1 = a_3/a_1, w_2 = a_3/a_2$ . Ακόμη, υποθέτουμε ότι, καθώς το σώμα παραμορφώνεται, τόσο οι πόροι όσο και η κατανομή τους εξελίσσονται με παρόμοια σχήματα και κατευθύνσεις. Αυτό σημαίνει ότι το πορώδες υλικό αναπτύσσει και διατηρεί *τοπική ορθοτροπική συμμετρία*. Οι τοπικοί άξονες της ορθοτροπίας ταυτίζονται με τους άξονες του τοπικού ελλειψοειδούς (Ponte Castañeda και Willis[16], Willis[18, 19], Ponte Castañeda και Zaidman [17]). Οι εσωτερικές μεταβλητές που χαρακτηρίζουν την τοπική κατάσταση του ομογενοποιημένου πορώδους υλικού είναι οι  $s = \{\bar{\epsilon}^P, f, w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)}\}$ , όπου  $\bar{\epsilon}^P$  είναι η τοπική ισοδύναμη πλαστική παραμόρφωση στο σώμα και  $f$  είναι το τοπικό κλάσμα των πόρων στο υλικό ή διαφορετικά το πορώδες του υλικού.

Η ελαστική και πλαστική απόκριση των πορώδων μετάλλων αναλύονται ξεχωριστά και έπειτα συνθέτονται για να καταλήξουμε στην ολοκληρωμένη ελαστο – πλαστική απόκριση. Ο τανυστής του ρυθμού παραμόρφωσης  $\mathbf{D}$  του πορώδους υλικού σε κάθε σημείο μπορεί να γραφεί ως το άθροισμα ενός τανυστή ελαστικού ρυθμού παραμόρφωσης  $\mathbf{D}^e$  και ενός τανυστή πλαστικού ρυθμού παραμόρφωσης  $\mathbf{D}^p$ :

$$\mathbf{D} = \mathbf{D}^e + \mathbf{D}^p \quad (2.4.1)$$

Το καταστατικό μοντέλο παρουσιάζεται σε τέσσερα μέρη. Στο πρώτο μέρος 2.4.1 παρουσιάζεται η ελαστική απόκριση του πορώδους μετάλλου. Το δεύτερο μέρος ασχολείται με την καταγραφή της συνθήκης διαρροής και τον κανόνα πλαστικής

διαρροής. Στο τρίτο μέρος παρουσιάζονται οι κανόνες εξέλιξης των εσωτερικών καταστατικών μεταβλητών. Τέλος, οι ελαστικές και πλαστικές καταστατικές εξισώσεις συνδυάζονται με σκοπό να προκύψουν οι τελικές ελαστο – πλαστικές εξισώσεις οι οποίες συνδέουν το ρυθμό παραμόρφωσης  $\mathbf{D}$  με την παράγωγο  $\nabla$  *Jauimman*  $\overset{\circ}{\boldsymbol{\sigma}}$  του τανυστή των τάσεων του *Cauchy*.

### 2.4.1 Ελαστικές καταστατικές εξισώσεις

Υποθέτουμε μια υποελαστική μορφή του ελαστικού μέρους του τανυστή που εκφράζει το ρυθμό παραμόρφωσης :

$$\mathbf{D}^e = \mathcal{M}^e : \overset{\circ}{\boldsymbol{\sigma}} \quad (2.4.2)$$

όπου  $\mathcal{M}^e$  είναι ο ελαστικός τανυστής ενδοτικότητας του σύνθετου υλικού και  $\overset{\circ}{\boldsymbol{\sigma}}$  είναι ο ρυθμός μεταβολής της τάσεως *Cauchy* η οποία είναι συστροφική με το spin των πόρων :

$$\overset{\circ}{\boldsymbol{\sigma}} = \dot{\boldsymbol{\sigma}} - \boldsymbol{\omega} \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \boldsymbol{\omega} \quad (2.4.3)$$

όπου  $\boldsymbol{\omega}$  είναι το spin των πόρων ως προς ένα σταθερό σύστημα συντεταγμένων, π.χ.  $\dot{\mathbf{n}}^{(i)} = \boldsymbol{\omega} \cdot \mathbf{n}^{(i)}$ ,  $i = 1, 2, 3$ . Ο αντισυμμετρικός τανυστής  $\boldsymbol{\omega}$  ονομάζεται το spin της μικροδομής και υπολογίζεται στο υποκεφάλαιο 2.4.3 (εξίσωση (2.4.19) ).

Ο ελαστικός τανυστής ενδοτικότητας του σύνθετου υλικού μπορεί να γραφεί ως [11, 12] :

$$\mathcal{M}^e = \mathcal{M} + \frac{f}{1-f} \mathbf{Q}^{-1} \quad (2.4.4)$$

Στην παραπάνω έκφραση, ο  $\mathcal{M}$  είναι ο ελαστικός τανυστής ενδοτικότητας του συμπαγούς φορέα (matrix=φορέας) και είναι ο αντίστροφος του ελαστικού τανυστή ακαμψίας  $\mathcal{L}$  :

$$\mathcal{L} = 2\mu \mathbf{K} + 3\kappa \mathbf{J}, \quad \mathcal{M} = \mathcal{L}^{-1} = \frac{1}{2\mu} \mathbf{K} + \frac{1}{3\kappa} \mathbf{J} = \frac{1}{2\mu} \left( \mathbf{K} + \frac{1-2\nu}{1+\nu} \mathbf{J} \right), \quad (2.4.5)$$

$$\mathbf{J} = \frac{1}{3} \boldsymbol{\delta} \boldsymbol{\delta}, \quad \mathbf{K} = \mathbf{I} - \mathbf{J}, \quad \mathbf{Q} = \mathcal{L} : (\mathbf{I} - \mathbf{S}),$$

όπου  $\mu$  και  $\kappa$  είναι οι ελαστικές σταθερές διάτμησης και διογκώσεως του φορέα,  $\nu$  είναι ο λόγος του *Poisson* του φορέα,  $\boldsymbol{\delta}$  και  $\mathbf{I}$  οι μοναδιαίοι τανυστές δευτέρας και

τετάρτης τάξεως αντίστοιχα,  $f$  το πορώδες και  $\mathbf{S}$  ο γνωστός τανυστής τετάρτης τάξεως του *Eshelby*. Ο τανυστής του *Eshelby* [7,8]  $\mathbf{S}$  είναι συνάρτηση των  $(\nu, w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)})$  και ο τανυστής  $\mathbf{Q}$  [18] είναι συνάρτηση των  $(\mu, w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)})$ .

## 2.4.2 Συνθήκη διαρροής και νόμος πλαστικής διαρροής

Στην περίπτωση του ανισότροπου μοντέλου η συνάρτηση διαρροής προκύπτει να είναι της μορφής [11, 12]:

$$\Phi(\sigma, s) = \frac{1}{1-f} \sigma : \mathbf{m}(s) : \sigma - \sigma_y^2 (\bar{\epsilon}^p) \quad (2.4.6)$$

όπου  $\sigma_y$  είναι το όριο διαρροής του υλικού σε εφελκυσμό. Το  $\sigma_y$  μπορεί να είναι είτε μία σταθερά οπότε έχουμε τέλεια πλαστικότητα ή να είναι μία συνάρτηση της ισοδύναμης πλαστικής παραμόρφωσης  $\bar{\epsilon}^p$  και συνεπώς σε αυτή την περίπτωση υποθέτουμε ότι ο φορέας κρατύνεται. Ο  $\mathbf{m}(s)$  είναι ένας τανυστής τετάρτης τάξεως και υπολογίζεται σύμφωνα με την παρακάτω σχέση :

$$\mathbf{m}(s) = 3\mu \mathcal{M}^e \Big|_{\nu=1/2} = \frac{3}{2} \mathbf{K} + \frac{3f}{1-f} \mu \mathbf{Q}^{-1} \Big|_{\nu=1/2} \quad (2.4.7)$$

Η έκφραση για το  $\mathcal{M}^e$  είναι ίδια με αυτή της εξίσωσης (2.4.4) μόνο που εξαιτίας της υπόθεσης ότι έχουμε πλαστική ασυμπίεστότητα, ο υπολογισμός του  $\mathcal{M}^e$  πρέπει να γίνει καθώς το  $\nu \rightarrow 1/2$ . Ο τανυστής  $\mathbf{Q}$  είναι ανάλογος του  $\mu$  και εξαρτάται από τα  $w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)}$ . Συνεπώς η συνάρτηση διαρροής είναι και αυτή μία συνάρτηση της τάσεως  $\sigma$  και των καταστατικών μεταβλητών  $s^a$ . Ο πλαστικός τανυστής ρυθμού παραμόρφωσης δίνεται τελικά από τον κανόνα πλαστικής διαρροής:

$$\mathbf{D}^p = \dot{\lambda} \mathbf{N}, \quad \mathbf{N} = \frac{\partial \Phi}{\partial \sigma} = \frac{2}{1-f} \mathbf{m} : \sigma \quad (2.4.8)$$

όπου το  $\dot{\lambda} \geq 0$  είναι ο πλαστικός πολλαπλασιαστής και εξαρτάται από τη σκλήρυνση και την εξέλιξη της μικροδομής. Ο τρόπος υπολογισμού του  $\dot{\lambda}$  έχει προαναφερθεί στην εξίσωση (2.2.10) και (2.2.11) αλλά για το συγκεκριμένο μοντέλο θα παρουσιαστεί ξανά στο υποκεφάλαιο 2.4.4.

### 2.4.3 Εξέλιξη της μικροδομής

Όταν ένα πορώδες μέταλλο παραμορφώνεται πλαστικά, οι καταστατικές μεταβλητές εξελίσσονται με αποτέλεσμα να επηρεάζουν την απόκριση του υλικού. Στην παρούσα εφαρμογή υποθέτουμε ότι όλες οι αλλαγές στην μικροδομή συμβαίνουν μόνο εξαιτίας της πλαστικής παραμόρφωσης η οποία προκαλεί με τη σειρά της αλλαγές στον όγκο, στο σχήμα και στον προσανατολισμό των πόρων. Αυτή η υπόθεση φαίνεται λογική αφού οι ελαστικές παραμορφώσεις είναι πολύ μικρές σε σχέση με τις πλαστικές.

*Εξέλιξη της ισοδύναμης πλαστικής παραμόρφωσης  $\bar{\epsilon}^p$  και του πορώδους  $f$*

Η εξέλιξη της  $\bar{\epsilon}^p$  καθορίζεται από τη συνθήκη ότι το τοπικό μακροσκοπικό πλαστικό έργο  $\sigma : D^p = \dot{\lambda} \sigma : N$  είναι ίσο με το τοπικό μικροσκοπικό έργο  $(1-f)\sigma_y \bar{\epsilon}^p$  οπότε:

$$\bar{\epsilon}^p = \dot{\lambda} \frac{\sigma : N}{(1-f)\sigma_y(\bar{\epsilon}^p)} \equiv \dot{\lambda} g_1(\sigma, s) \quad (2.4.9)$$

Γενικά η παρουσία των πόρων σε ένα μέταλλο μπορεί να θεωρηθεί ως μία μορφή αστοχίας του υλικού και επιπλέον η μεταβολή στο πορώδες λόγω ελαστικών παραμορφώσεων είναι μικρή. Για αυτό το λόγο γίνεται η υπόθεση ότι οι αλλαγές στο πορώδες του υλικού συμβαίνουν κυρίως λόγω του ογκομετρικού πλαστικού ρυθμού παραμόρφωσης  $D_{\kappa\kappa}^p$ . Με βάση την παραπάνω υπόθεση αλλά και την εξίσωση της συνέχειας προκύπτει η ακόλουθη έκφραση για τον υπολογισμό του  $f$ :

$$\dot{f} = (1-f) D_{\kappa\kappa}^p = \dot{\lambda} (1-f) N_{\kappa\kappa} \equiv \dot{\lambda} g_2(\sigma, s) \quad (2.4.10)$$

*Εξέλιξη των  $w_1$  και  $w_2$  και των τοπικών αξόνων της ορθοτροπίας*



Εδώ θα χρειαστεί να ορίσουμε δύο καινούριους τανυστές από τους οποίους ο ένας εκφράζει το μέσο ρυθμό παραμόρφωσης,  $\mathbf{D}^\nu$ , και ο άλλος το μέσο spin,  $\mathbf{W}^\nu$ , του τοπικού ελλειψοειδούς. Οι τελευταίοι συνδέονται με την πλαστική μακροσκοπική παραμόρφωση,  $\mathbf{D}^p$ , και το μακροσκοπικό spin του συνεχούς μέσου,  $\mathbf{W}$ , με τις παρακάτω σχέσεις (Ponte Castañeda και Zaidman [17], M. Kailasam, and Ponte Castañeda [12]):

$$\mathbf{D}^\nu = \mathbf{A} : \mathbf{D}^p \quad (2.4.11)$$

όπου  $\mathbf{A}$  είναι ένας τετάρτης τάξεως τανυστής «συγκεντρώσεως ρυθμού παραμόρφωσης» ο οποίος ορίζεται ως :

$$\mathbf{A} = \left[ \mathbf{I} - (1-f) \mathbf{S} \Big|_{v=1/2} \right]^{-1} \quad (2.4.12)$$

και

$$\mathbf{W}^\nu = \mathbf{W} - \mathbf{C} : \mathbf{D}^p \quad (2.4.13)$$

όπου  $\mathbf{C}$  είναι ένας τετάρτης τάξεως τανυστής «συγκεντρώσεως – spin» και ορίζεται ως:

$$\mathbf{C} = -(1-f) \mathbf{\Pi} : \mathbf{A} \quad (2.4.14)$$

Ο  $\mathbf{\Pi}$  είναι ο τετάρτης τάξεως τανυστής περιστροφής του *Eshelby* [7, 8] ο οποίος χρησιμοποιείται για να καθορίσει το spin ενός απομονωμένου ελλειψοειδούς πόρου στο γραμμικά ελαστικό μέσο και γενικά εξαρτάται από τα  $\{w_1, w_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)}\}$ .

Η εξέλιξη των  $w_1, w_2$  υπολογίζεται ξεκινώντας από τον ορισμό  $w_1 = a_3 / a_1$  και τη σχέση :

$$\dot{w}_1 = w_1 \left( \frac{\dot{a}_3}{a_3} - \frac{\dot{a}_1}{a_1} \right) = w_1 \left( \mathbf{n}^{(3)} \cdot \mathbf{D}^\nu \cdot \mathbf{n}^{(3)} - \mathbf{n}^{(1)} \cdot \mathbf{D}^\nu \cdot \mathbf{n}^{(1)} \right) = w_1 \left( \mathbf{n}^{(3)} \mathbf{n}^{(3)} - \mathbf{n}^{(1)} \mathbf{n}^{(1)} \right) : \mathbf{D}^\nu \quad (2.4.15)$$

όπου  $2a_i$  είναι το μήκος του  $i$ -κύριου άξονα του τοπικού αντιπροσωπευτικού ελλειψοειδούς. Λαμβάνοντας υπόψη τις σχέσεις (2.4.11) και (2.4.8) η τελευταία εξίσωση μπορεί να γραφεί ως :

$$\dot{w}_1 = \dot{\lambda} w_1 \left( \mathbf{n}^{(3)} \mathbf{n}^{(3)} - \mathbf{n}^{(1)} \mathbf{n}^{(1)} \right) : \mathbf{A} : \mathbf{N} \equiv \dot{\lambda} g_3(\sigma, s) \quad (2.4.16)$$

και όμοια

$$\dot{w}_2 = \dot{\lambda} w_2 \left( \mathbf{n}^{(3)} \mathbf{n}^{(3)} - \mathbf{n}^{(2)} \mathbf{n}^{(2)} \right); \mathbf{A} : \mathbf{N} \equiv \dot{\lambda} g_4(\sigma, s) \quad (2.4.17)$$

Στη συνέχεια ορίζουμε τις εξισώσεις εξελίξεως των διανυσμάτων προσανατολισμού  $\mathbf{n}^{(i)}$ . Δεδομένου ότι τα  $\mathbf{n}^{(i)}$  είναι μοναδιαία διανύσματα η χρονική τους παράγωγος γράφεται με την μορφή :

$$\dot{\mathbf{n}}^{(i)} = \boldsymbol{\omega} \cdot \mathbf{n}^{(i)} \quad (2.4.18)$$

όπου  $\boldsymbol{\omega}$  είναι ένας αντισυμμετρικός τανυστής και συμπίπτει με το spin των αξόνων Euler της μέσης παραμόρφωσης του τοπικού ελλειψοειδούς και καθορίζεται από τη γνωστή κινηματική σχέση (Biot [3], Hill [10], Odgen [14]):

$$\omega'_{ij} = W'_{ij} - \frac{\lambda_i^2 + \lambda_j^2}{\lambda_i^2 - \lambda_j^2} D_{kl}^{p'j}, \quad i \neq j, \quad \lambda_i \neq \lambda_j \quad (2.4.19)$$

όπου  $\lambda_i = a_i / a_i^0$  είναι κλάσμα επιμήκυνσης των κυρίων αξόνων και  $a_1^0 = a_2^0 = a_3^0$  είναι οι ακτίνες του τοπικού αντιπροσωπευτικού ελλειψοειδούς που αρχικά έχει σχήμα σφαιρικό. Στην τελευταία σχέση αλλά και στο υπόλοιπο αυτού του κεφαλαίου, οι ποσότητες με τόνο είναι συνιστώσες τανυστών που αναφέρονται σε ένα σύστημα συντεταγμένων το οποίο στιγμιαία συμπίπτει με τους κύριους άξονες του τοπικού αντιπροσωπευτικού ελλειψοειδούς και καθορίζονται από τα διανύσματα  $\mathbf{n}^{(i)}$  ( $i = 1, \dots, 3$ ). Για παράδειγμα,  $\mathbf{D}^p = D_{ij}^{p'j} \mathbf{n}^{(i)} \mathbf{n}^{(j)}$  κ.λ.π.

Αν χρησιμοποιείται η μέθοδος των πεπερασμένων στοιχείων τότε είναι βολικό να αναφέρονται όλοι οι τανυστές στο σταθερό καρτεσιανό κεντρικό σύστημα συντεταγμένων. Συνεπώς η εξίσωση (2.4.19) γράφεται με τη μορφή :

$$\mathbf{n}^{(i)} \cdot \boldsymbol{\omega} \cdot \mathbf{n}^{(j)} = \mathbf{n}^{(i)} \cdot \mathbf{W}^v \cdot \mathbf{n}^{(j)} - \frac{\lambda_i^2 + \lambda_j^2}{\lambda_i^2 - \lambda_j^2} \mathbf{n}^{(i)} \cdot \mathbf{D}^v \cdot \mathbf{n}^{(j)}, \quad i \neq j, \quad \lambda_i \neq \lambda_j \quad (2.4.20)$$

Παρατηρούμε όμως ότι,

$$\lambda_i = \frac{a_i}{a_i^0} = \frac{a_3 / a_i^0}{a_3 / a_i} = \frac{\lambda_3}{w_i}, \quad i = 1, 2, 3 \quad \text{με} \quad w_3 = 1 \quad (2.4.21)$$

Επίσης παίρνοντας υπόψη ότι το  $\mathbf{D}^v$  είναι συμμετρικό μπορούμε να γράψουμε :

$$\mathbf{n}^{(i)} \cdot \mathbf{D}^v \cdot \mathbf{n}^{(j)} = \left( \mathbf{n}^{(i)} \mathbf{n}^{(j)} \right) : \mathbf{D}^v = \frac{1}{2} \left( \mathbf{n}^{(i)} \mathbf{n}^{(j)} + \mathbf{n}^{(j)} \mathbf{n}^{(i)} \right) : \mathbf{D}^v \quad (2.4.22)$$

Επομένως μπορούμε να γράψουμε το spin της μικροδομής  $\omega = (\mathbf{n}^{(i)} \cdot \omega \cdot \mathbf{n}^{(j)}) \mathbf{n}^{(i)} \mathbf{n}^{(j)}$

στην παρακάτω μορφή :

$$\omega = W^v + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j \\ w_i \neq w_j}}^3 \frac{w_i^2 + w_j^2}{w_i^2 - w_j^2} \left[ (\mathbf{n}^{(i)} \mathbf{n}^{(j)} + \mathbf{n}^{(j)} \mathbf{n}^{(i)}) : \mathbf{D}^v \right] \mathbf{n}^{(i)} \mathbf{n}^{(j)}, \quad (w_3 = 1) \quad (2.4.23)$$

Με βάση τις σχέσεις (2.4.11),(2.4.13) και (2.4.8) μπορεί η τελευταία εξίσωση να γραφεί για  $(w_3 = 1)$ , ως :

$$\omega = W - \dot{\lambda} \left[ \mathbf{C} : \mathbf{N} - \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j \\ w_i \neq w_j}}^3 \frac{w_i^2 + w_j^2}{w_i^2 - w_j^2} \left[ (\mathbf{n}^{(i)} \mathbf{n}^{(j)} + \mathbf{n}^{(j)} \mathbf{n}^{(i)}) : \mathbf{A} : \mathbf{N} \right] \mathbf{n}^{(i)} \mathbf{n}^{(j)} \right] \quad (2.4.24)$$

Τελικά είναι βολικό να εισάγουμε το γνωστό πλαστικό spin  $W^p$  το οποίο καθορίζεται ως το spin του συνεχούς μέσου σχετικά με το spin της μικρο - δομής . Χρησιμοποιώντας την τελευταία εξίσωση καταλήγουμε στη σχέση :

$$W^p = \dot{\lambda} \Omega^p \quad (2.4.25)$$

όπου για  $(w_3 = 1)$

$$\Omega^p = \mathbf{C} : \mathbf{N} - \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j \\ w_i \neq w_j}}^3 \frac{w_i^2 + w_j^2}{w_i^2 - w_j^2} \left[ (\mathbf{n}^{(i)} \mathbf{n}^{(j)} + \mathbf{n}^{(j)} \mathbf{n}^{(i)}) : \mathbf{A} : \mathbf{N} \right] \mathbf{n}^{(i)} \mathbf{n}^{(j)} \quad (2.4.26)$$

Στην περίπτωση που το σύστημα συντεταγμένων συμπίπτει με τους κύριους άξονες του τοπικού αντιπροσωπευτικού ελλειψοειδούς, τα στοιχεία της τελευταίας εξίσωσης γίνονται:

$$\Omega'_{ij}{}^p = \left( C'_{ijkl} - \frac{w_i^2 + w_j^2}{w_i^2 - w_j^2} A'_{ijkl} \right) N'_{kl}, \quad i \neq j, \quad w_i \neq w_j, \quad w_3 = 1 \quad (2.4.27)$$

Επίσης θα πρέπει να αναφερθεί ότι όταν  $w_1 = w_2$  τότε το υλικό γίνεται τοπικά εγκαρσίως ισότροπο ( transversely isotropic ) σε σχέση με την  $\mathbf{n}^{(3)}$  - κατεύθυνση, και τα στοιχεία  $C'_{12kl}$  μηδενίζονται, και η εξίσωση (2.4.26) συνεπάγεται ότι (Aravas [2] ):

$$W'_{12}{}^p = 0 \quad (2.4.28)$$

Ακόμη όταν  $w_1 = w_2 = w_3$  το υλικό γίνεται τοπικά ισότροπο και ο τανυστής  $\mathbf{C}$  συγκεντρώσεως – spin μηδενίζεται και από την εξίσωση (2.4.26) βγαίνει το συμπέρασμα ότι (Dafalias [6]):

$$\mathbf{W}^p = \mathbf{0} \quad (2.4.29)$$

Στην ειδική περίπτωση του διδιάστατου προβλήματος, στο οποίο η κίνηση λαμβάνει χώρα στο  $x_1 - x_2$  επίπεδο, το  $\Omega^p$  είναι της μορφής :

$$\Omega^p = \omega^p (-\mathbf{e}_1 \mathbf{e}_2 + \mathbf{e}_2 \mathbf{e}_1) = \omega^p \left( -\mathbf{n}^{(1)} \mathbf{n}^{(2)} + \mathbf{n}^{(2)} \mathbf{n}^{(1)} \right) \quad (2.4.30)$$

όπου  $(\mathbf{e}_1, \mathbf{e}_2)$  είναι τα μοναδιαία διανύσματα των  $x_1 - x_2$  αξόνων, και η ποσότητα  $\omega^p = -\mathbf{e}_1 \cdot \Omega^p \cdot \mathbf{e}_2 = -\mathbf{n}^{(1)} \cdot \Omega^p \cdot \mathbf{n}^{(2)}$  σύμφωνα με την εξίσωση (2.4.26) γράφεται για  $w_1 \neq w_2$  ως :

$$\omega^p = -\mathbf{e}_1 \cdot (\mathbf{C} : \mathbf{N}) \cdot \mathbf{e}_2 + \frac{1}{2} \frac{w_1^2 + w_2^2}{w_1^2 - w_2^2} \left( \mathbf{n}^{(1)} \mathbf{n}^{(2)} + \mathbf{n}^{(2)} \mathbf{n}^{(1)} \right) : \mathbf{A} : \mathbf{N} \quad (2.4.31)$$

και

$$\omega^p = 0 \quad \text{όταν} \quad w_1 = w_2 \quad (2.4.32)$$

Πρέπει, επίσης, να αναφερθεί ότι οι καταστατικές συναρτήσεις  $\Phi, \mathbf{N}, g_1, g_2, g_3, g_4$  και  $\Omega^p$  είναι ισότροπες συναρτήσεις ως προς τις μεταβλητές τους. Για παράδειγμα :

$$\Phi(\mathbf{R} \cdot \boldsymbol{\sigma} \cdot \mathbf{R}^T, f, w_1, w_2, \mathbf{R} \cdot \mathbf{n}^{(i)}) = \Phi(\boldsymbol{\sigma}, f, w_1, w_2, \mathbf{n}^{(i)}), \quad (2.4.33)$$

$$\mathbf{N}(\mathbf{R} \cdot \boldsymbol{\sigma} \cdot \mathbf{R}^T, f, w_1, w_2, \mathbf{R} \cdot \mathbf{n}^{(i)}) = \mathbf{R} \cdot \mathbf{N}(\boldsymbol{\sigma}, f, w_1, w_2, \mathbf{n}^{(i)}) \cdot \mathbf{R}^T \quad (2.4.34)$$

οι οποίες ισχύουν για όλους τους ορθογώνιους τανυστές  $\mathbf{R}$ . Θα πρέπει όμως να δοθεί έμφαση στο γεγονός ότι το υλικό είναι ανισότροπο και αυτό οφείλεται στον τανυστικό χαρακτήρα των  $\mathbf{n}^{(i)}$ . Ακόμη πρέπει να αναφερθεί ότι η εξίσωση (2.4.18) μπορεί να γραφεί με τη μορφή :

$$\overset{\circ}{\mathbf{n}}^{(i)} = \mathbf{0} \quad (2.4.35)$$

όπου  $\overset{\circ}{\mathbf{n}}^{(i)}$  είναι ο ρυθμός των  $\mathbf{n}^{(i)}$  που είναι συστροφικός με το spin των πόρων, δηλαδή,  $\overset{\circ}{\mathbf{n}}^{(i)} = \dot{\mathbf{n}}^{(i)} - \boldsymbol{\omega} \cdot \mathbf{n}^{(i)}$ .

Λαμβάνοντας υπόψη ότι  $\mathbf{W} = \boldsymbol{\omega} + \mathbf{W}^P$ , συμπεραίνουμε ότι η παράγωγος *Jaumann*  $\overset{\nabla}{\mathbf{n}}^{(i)} = \dot{\mathbf{n}}^{(i)} - \mathbf{W} \cdot \mathbf{n}^{(i)}$  μπορεί να γραφεί ως  $\overset{\nabla}{\mathbf{n}}^{(i)} = \overset{\circ}{\mathbf{n}}^{(i)} - \mathbf{W}^P \cdot \mathbf{n}^{(i)}$  και τελικά από την (2.4.35) να προκύψει η παρακάτω έκφραση :

$$\overset{\nabla}{\mathbf{n}}^{(i)} = -\mathbf{W}^P \cdot \mathbf{n}^{(i)} = -\dot{\lambda} \boldsymbol{\Omega}^P \cdot \mathbf{n}^{(i)} \quad (2.4.36)$$

Ακολουθώντας οι εξισώσεις εξελίξεως για όλες τις μεταβλητές  $s$  της μικροδομής, δίνονται από τις εξισώσεις (2.4.9),(2.4.10),(2.4.16),(2.4.17) και (2.4.36) μπορούν να γραφούν συνοπτικά στη μορφή :

$$\overset{\nabla}{s} = \dot{\lambda} \mathbf{G}(\boldsymbol{\sigma}, s) \quad (2.4.37)$$

όπου  $\overset{\nabla}{s} = \left\{ \dot{\boldsymbol{\varepsilon}}^P, \dot{f}, \dot{w}_1, \dot{w}_2, \overset{\nabla}{\mathbf{n}}^{(1)}, \overset{\nabla}{\mathbf{n}}^{(2)}, \overset{\nabla}{\mathbf{n}}^{(3)} \right\}$  και  $\mathbf{G}$  είναι μία συλλογή κατάλληλων

ισότροπων συναρτήσεων. Ο πλαστικός πολλαπλασιαστής  $\dot{\lambda}$  μπορεί να υπολογιστεί από τη συνθήκη της συνέχειας η οποία έχει παρουσιαστεί σε γενική μορφή στην εξίσωση (2.2.8) και θα παρουσιαστεί ξεχωριστά για το ανισότροπο μοντέλο στο επόμενο υποκεφάλαιο.

Συνοπτικά, οι καταστατικοί νόμοι αναπτύχθηκαν παραπάνω για να περιγραφεί η συμπεριφορά του ελαστο – πλαστικού πορώδους υλικού. Στην ελαστική περιοχή η συμπεριφορά χαρακτηρίζεται από τις εξισώσεις (2.4.2) - (2.4.5) και στην πλαστική περιοχή από τις εξισώσεις (2.4.6) - (2.4.8). Η εξέλιξη των καταστατικών μεταβλητών  $s$  καθορίζεται από τις εξισώσεις (2.4.9),(2.4.10),(2.4.16),(2.4.17) και (2.4.36).

#### 2.4.4 Ρυθμός μεταβολής των ελαστο – πλαστικών εξισώσεων

Οι καταστατικές εξισώσεις συνδυάζονται τώρα με σκοπό την εμφάνιση μιας εξίσωσης που συνδέει την παράγωγο *Jaumann* του τανυστή των τάσεων  $\overset{\nabla}{\boldsymbol{\sigma}}$  με τον συνολικό ρυθμό παραμόρφωσης  $\mathbf{D}$ . Η διαδικασία είναι η παρακάτω:

Με αντικατάσταση της  $\mathbf{D}^e = \mathbf{D} - \mathbf{D}^P = \mathbf{D} - \dot{\lambda} \mathbf{N}$  στην εξίσωση (2.4.2) έχουμε:

$$\overset{\circ}{\boldsymbol{\sigma}} = \mathcal{L}^e : \mathbf{D} - \dot{\lambda} \mathcal{L}^e : \mathbf{N} \quad (2.4.38)$$

όπου  $\mathcal{L}^e = \mathcal{M}^{e^{-1}}$ . Εφ' όσον η  $\Phi$  είναι ισότροπη συνάρτηση η συνθήκη της συνέχειας μπορεί να γραφεί στη μορφή ( Dafalias [6] ) :

$$\overset{\circ}{\Phi} = \frac{\partial \Phi}{\partial \sigma} : \overset{\circ}{\sigma} + \frac{\partial \Phi}{\partial s} \cdot \overset{\circ}{s} = 0 \quad (2.4.39)$$

όπου  $\overset{\circ}{s} = \left\{ \dot{\bar{\varepsilon}}^p, \dot{f}, \dot{w}_1, \dot{w}_2, \mathbf{n}^{(1)}, \mathbf{n}^{(2)}, \mathbf{n}^{(3)} \right\}$ . Από την εξίσωση (2.4.35) η προηγούμενη

σχέση γράφεται ως εξής :

$$\mathbf{N} : \overset{\circ}{\sigma} + \frac{\partial \Phi}{\partial \bar{\varepsilon}^p} \dot{\bar{\varepsilon}}^p + \frac{\partial \Phi}{\partial f} \dot{f} + \frac{\partial \Phi}{\partial w_1} \dot{w}_1 + \frac{\partial \Phi}{\partial w_2} \dot{w}_2 = 0 \quad (2.4.40)$$

Με αντικατάσταση των  $\dot{\bar{\varepsilon}}^p, \dot{f}, \dot{w}_1$  και  $\dot{w}_2$  από τις σχέσεις (2.4.9),(2.4.10),(2.4.16) και (2.4.17) στην τελευταία εξίσωση προκύπτει :

$$\mathbf{N} : \overset{\circ}{\sigma} - \dot{\lambda} H = 0 \Rightarrow \dot{\lambda} = \frac{1}{H} \mathbf{N} : \overset{\circ}{\sigma} \quad (2.4.41)$$

όπου

$$H = - \left( \frac{\partial \Phi}{\partial \bar{\varepsilon}^p} g_1 + \frac{\partial \Phi}{\partial f} g_2 + \frac{\partial \Phi}{\partial w_1} g_3 + \frac{\partial \Phi}{\partial w_2} g_4 \right) \quad (2.4.42)$$

Η ανάλυση των τριών περιπτώσεων για το  $H$  έχει γίνει στην ενότητα 2.2.2. Επίσης μία εναλλακτική έκφραση για το  $\dot{\lambda}$  προκύπτει από την αντικατάσταση του  $\overset{\circ}{\sigma}$  από τη σχέση (2.4.38) στην σχέση (2.4.41) :

$$\mathbf{N} : \mathcal{L}^e : \mathbf{D} - \dot{\lambda} (\mathbf{N} : \mathcal{L}^e : \mathbf{N} + H) = 0 \Rightarrow \dot{\lambda} = \frac{1}{L} \mathbf{N} : \mathcal{L}^e : \mathbf{D} \quad (2.4.43)$$

όπου  $L = H + \mathbf{N} : \mathcal{L}^e : \mathbf{N}$  το οποίο πρέπει να είναι θετικό στα πραγματικά υλικά. Για μία εντατική κατάσταση πάνω στην επιφάνεια διαρροής τέτοια ώστε  $\Phi(\sigma, s) = 0$  η απαίτηση το  $\dot{\lambda} > 0$  προσδιορίζει τη “ συνθήκη πλαστικής φόρτισης ” :

$$\mathbf{N} : \mathcal{L}^e : \mathbf{D} > 0 \quad (2.4.44)$$

όπου  $\mathbf{N} : \mathcal{L}^e : \mathbf{D} = 0$  αντιστοιχεί σε “ ουδέτερη φόρτιση ” ( $\dot{\lambda} = 0$ ) και  $\mathbf{N} : \mathcal{L}^e : \mathbf{D} < 0$  σε “ ελαστική αποφόρτιση ” ( $\dot{\lambda} = 0$ ). Έπειτα με αντικατάσταση του  $\dot{\lambda}$  από τη σχέση (2.4.43) στην (2.4.38) παίρνουμε :

$$\overset{\circ}{\sigma} = \left( \mathcal{L}^e - \frac{1}{L} \mathcal{L}^e : \mathbf{N} \mathbf{N} : \mathcal{L}^e \right) : \mathbf{D} \quad (2.4.45)$$

Η παράγωγος  $\overset{\nabla}{\sigma}$  σχετίζεται με το  $\overset{\circ}{\sigma}$  με την ακόλουθη σχέση :

$$\begin{aligned} \overset{\nabla}{\sigma} &= \overset{\circ}{\sigma} + \sigma \cdot W^P - W^P \cdot \sigma = \overset{\circ}{\sigma} + \dot{\lambda} (\sigma \cdot \Omega^P - \Omega^P \cdot \sigma) = \\ &= \overset{\circ}{\sigma} + \frac{1}{L} (\sigma \cdot \Omega^P - \Omega^P \cdot \sigma) (\mathbf{N} : \mathcal{L}^e : \mathbf{D}) \end{aligned} \quad (2.4.46)$$

Τελικά με αντικατάσταση του  $\overset{\circ}{\sigma}$  στην παραπάνω εξίσωση από τη (2.4.45) μας δίνει:

$$\overset{\nabla}{\sigma} = \mathcal{L}^{ep} : \mathbf{D}, \quad \mathcal{L}^{ep} = \mathcal{L}^e - \frac{1}{L} (\mathcal{L}^e : \mathbf{N}) (\mathcal{L}^e : \mathbf{N}) + \frac{1}{L} (\sigma \cdot \Omega^P - \Omega^P \cdot \sigma) (\mathcal{L}^e : \mathbf{N}) \quad (2.4.47)$$

με την προϋπόθεση ότι  $\mathbf{N} : \mathcal{L}^e : \mathbf{D} > 0$  (πλαστική φόρτιση). Επίσης πρέπει να αναφερθεί ότι το  $\mathcal{L}^{ep}$  δεν έχει την κύρια (“ διαγώνια ”) συμμετρία, δηλαδή  $\mathcal{L}_{ijkl}^{ep} \neq \mathcal{L}_{klij}^{ep}$  γεγονός που οφείλεται στον τελευταίο όρο της εξίσωσης (2.4.47), ο οποίος αποτελεί τη συνεισφορά του πλαστικού spin στον πλαστικό τανυστή τετάρτης τάξεως  $\mathcal{L}^{ep}$ . Όταν  $\mathbf{N} : \mathcal{L}^e : \mathbf{D} \leq 0$  (ουδέτερη φόρτιση ή ελαστική αποφόρτιση) η εξίσωση για παίρνει τη μορφή :

$$\overset{\nabla}{\sigma} = \mathcal{L}^e : \mathbf{D} \quad (2.4.48)$$

### 3. Αριθμητική ολοκλήρωση των καταστατικών εξισώσεων

Στο παρόν κεφάλαιο θα αναπτυχθούν οι μέθοδοι ολοκλήρωσης των καταστατικών εξισώσεων των δύο μοντέλων. Στο μοντέλο του *Gurson* η ολοκλήρωση γίνεται με μέθοδο forward Euler. Στο ανισότροπο μοντέλο η ολοκλήρωση γίνεται τόσο με μέθοδο forward Euler ( Kailasam, Aravas και Ponte Castañeda [13] ) αλλά και με backward Euler. Και στα δύο μοντέλα χρησιμοποιούμε τη θεωρία των μεγάλων παραμορφώσεων και ο τρόπος υπολογισμού του ταυυστή κλίσης της παραμόρφωσης  $\mathbf{F}$  αλλά και της λογαριθμικής παραμόρφωσης  $\mathbf{E}_{n+1}$  είναι ο ίδιος και παρουσιάζεται στην ενότητα 3.1.

#### 3.1 Υπολογισμός του ταυυστή της κλίσης της κίνησης $\mathbf{F}$

Πιο συγκεκριμένα, σε ένα περιβάλλον πεπερασμένων στοιχείων, η λύση αναπτύσσεται αυξητικά και οι καταστατικές εξισώσεις επιλύονται στα σημεία ολοκλήρωσης Gauss του κάθε στοιχείου. Έστω ότι  $\mathbf{F}$  είναι ο ταυυστής κλίσης της κίνησης. Σε ένα καθορισμένο σημείο Gauss, είναι γνωστά τόσο τα  $\mathbf{F}_n, \sigma_n, s_n$  στο χρόνο  $t_n$  όσο και το  $\mathbf{F}_{n+1}$  και το πρόβλημα είναι να καθοριστούν τα  $\sigma_{n+1}, s_{n+1}$ .

Η χρονική μεταβολή του ταυυστή κλίσης της κίνησης  $\mathbf{F}$  κατά τη διάρκεια του χρονικού βήματος  $[t_n, t_{n+1}]$  μπορεί να γραφεί ως :

$$\mathbf{F}(t) = \Delta\mathbf{F}(t) \cdot \mathbf{F}_n = \mathbf{R}(t) \cdot \mathbf{U}(t) \cdot \mathbf{F}_n, \quad t_n \leq t \leq t_{n+1} \quad (3.1.1)$$

όπου  $\mathbf{R}(t)$  και  $\mathbf{U}(t)$  είναι οι ταυυστές περιστροφής και ορθής επιμήκυνσης αντίστοιχα οι οποίοι συνδέονται με το  $\Delta\mathbf{F}(t)$ . Οι αντίστοιχοι ταυυστές ρυθμού παραμορφώσεως  $\mathbf{D}(t)$  και στροβιλότητας  $\mathbf{W}(t)$  δίνονται από τις σχέσεις :

$$\mathbf{D}(t) = \left[ \dot{\mathbf{F}}(t) \cdot \mathbf{F}^{-1}(t) \right]_s = \left[ \Delta\dot{\mathbf{F}}(t) \cdot \Delta\mathbf{F}^{-1}(t) \right]_s \quad (3.1.2)$$

και

$$\mathbf{W}(t) = \left[ \dot{\mathbf{F}}(t) \cdot \mathbf{F}^{-1}(t) \right]_a = \left[ \Delta\dot{\mathbf{F}}(t) \cdot \Delta\mathbf{F}^{-1}(t) \right]_a \quad (3.1.3)$$

όπου οι δείκτες  $s$  και  $a$  δηλώνουν τα συμμετρικά και αντι-συμμετρικά μέρη ενός ταυυστή αντίστοιχα.



Κάνοντας την υπόθεση ότι οι άξονες του Lagrange που σχετίζονται με το  $\Delta \mathbf{F}(t)$  (δηλαδή τα ιδιοδιανύσματα του  $\mathbf{U}(t)$ ) παραμένουν σταθερά στο χρονικό διάστημα  $[t_n, t_{n+1}]$ , μπορεί εύκολα να αποδειχθεί ότι :

$$\mathbf{D}(t) = \mathbf{R}(t) \cdot \dot{\mathbf{E}}(t) \cdot \mathbf{R}^T(t), \quad \mathbf{W}(t) = \dot{\mathbf{R}}(t) \cdot \mathbf{R}^T(t) \quad (3.1.4)$$

καθώς και

$$\overset{\nabla}{\sigma}(t) = \mathbf{R}(t) \cdot \dot{\hat{\sigma}} \cdot \mathbf{R}^T(t), \quad \overset{\nabla}{\mathbf{n}}^{(i)}(t) = \mathbf{R}(t) \cdot \dot{\hat{\mathbf{n}}}^{(i)}(t) \quad (3.1.5)$$

όπου  $\mathbf{E}(t) = \ln \mathbf{U}(t)$  είναι η λογαριθμική παραμόρφωση σχετικά με την ανάλυση στο χρόνο  $t_n$ ,  $\hat{\sigma}(t) = \mathbf{R}^T(t) \cdot \sigma \cdot \mathbf{R}(t)$  και  $\hat{\mathbf{n}}^{(i)}(t) = \mathbf{R}^T(t) \cdot \mathbf{n}^{(i)}(t)$ .

Επίσης ισχύει ότι στην αρχή του βήματος ( $t = t_n$ ) :

$$\mathbf{F}_n = \mathbf{R}_n = \mathbf{U}_n = \delta, \quad \hat{\sigma}_n = \sigma_n, \quad \hat{\mathbf{n}}_n^{(i)} = \mathbf{n}_n^{(i)} \quad \text{και} \quad \mathbf{E}_n = 0 \quad (3.1.6)$$

ενώ στο τέλος του βήματος ( $t = t_{n+1}$ ) :

$$\Delta \mathbf{F}_{n+1} = \mathbf{F}_{n+1} \cdot \mathbf{F}_n^{-1} = \mathbf{R}_{n+1} \cdot \mathbf{U}_{n+1} = \text{γνωστό}, \quad \text{και} \quad \mathbf{E}_{n+1} = \ln \mathbf{U}_{n+1} = \text{γνωστό} \quad (3.1.7)$$

Οι παραπάνω σχέσεις ισχύουν και για τα δύο μοντέλα. Παρακάτω θα γίνει ξεχωριστά η περιγραφή της ολοκλήρωσης των δύο καταστατικών μοντέλων.

### 3.2 Αριθμητική ολοκλήρωση μοντέλου *Gurson*

Η αριθμητική ολοκλήρωση του μοντέλου *Gurson* γίνεται με τη μέθοδο forward Euler. Η γενική ιδέα της μεθόδου αυτής είναι ότι χρησιμοποιούνται δεδομένα του προηγούμενου βήματος  $n$  για να υπολογιστούν οι άγνωστοι του επόμενου  $n+1$ .

Ορίζουμε τις ακόλουθες βοηθητικές μεταβλητές :

$$GG|_n = N_{kk}|_n, \quad HH|_n = (1 - f_n) N_{kk}|_n \quad \text{και} \quad H3|_n = \frac{\sigma_n : \mathbf{N}_n}{(1 - f_n) \sigma_m|_n} \quad (3.2.1)$$

Ο σκοπός μας είναι να υπολογίσουμε τα  $\sigma_{n+1}, f_{n+1}$  και  $\varepsilon_m^p|_{n+1}$  τα οποία δίνονται από τις σχέσεις :

$$\sigma_{n+1} = \sigma^e - \Delta \lambda \mathcal{C}_n^e : \mathbf{N}_n \quad \text{με} \quad \frac{\partial \sigma}{\partial (\Delta \lambda)} = -\mathcal{C}_n^e : \mathbf{N}_n \quad \text{και} \quad \sigma^e = \sigma_n + \mathcal{C}_n^e : \Delta \mathbf{E}_{n+1} \quad (3.2.2)$$

$$f_{n+1} = f_n + \Delta\lambda \cdot HH|_n \quad \text{με} \quad \frac{\partial f}{\partial(\Delta\lambda)} = HH|_n \quad (3.2.3)$$

$$\varepsilon_m^p|_{n+1} = \varepsilon_m^p|_n + \Delta\lambda \cdot H3|_n \quad \text{με} \quad \frac{\partial \varepsilon_m^p}{\partial(\Delta\lambda)} = H3|_n \quad (3.2.4)$$

Για να υπολογιστούν τα παραπάνω χρειάζεται να υπολογιστεί το  $\Delta\lambda$ . Το τελευταίο πρέπει να είναι παίρνει τέτοια τιμή ώστε να ικανοποιείται η συνάρτηση διαρροής :

$$\Phi(\sigma(\Delta\lambda), f(\Delta\lambda), \varepsilon_m^p(\Delta\lambda)) = 0 \quad (3.2.5)$$

Η επίλυση της παραπάνω εξίσωσης για τον προσδιορισμό του  $\Delta\lambda$  γίνεται με τη μέθοδο *Newton* η οποία παρουσιάζεται παρακάτω σχηματικά :

i. Αρχική εκτίμηση για το  $\Delta\lambda^{(n)} \Rightarrow \Delta\lambda = \frac{1}{H} \frac{\partial \Phi}{\partial \sigma} : \Delta\sigma = \frac{1}{L} \frac{\partial \Phi}{\partial \varepsilon} : \mathcal{C}^e : \Delta\varepsilon$

ii. Έλεγχος για το αν το υπολογισμένο  $\Delta\lambda$  ικανοποιεί την εξίσωση  $\Phi(\sigma(\Delta\lambda), f(\Delta\lambda), \varepsilon_m^p(\Delta\lambda)) = 0$ . Αν δεν ικανοποιείται η εξίσωση συνεχίζουμε παρακάτω:

iii. Υπολογισμός της παραγώγου

$$\frac{\partial \Phi}{\partial(\Delta\lambda)} = \frac{\partial \Phi}{\partial \sigma} : \frac{\partial \sigma}{\partial(\Delta\lambda)} + \frac{\partial \Phi}{\partial f} : \frac{\partial f}{\partial(\Delta\lambda)} + \frac{\partial \Phi}{\partial \varepsilon_m^p} : \frac{\partial \varepsilon_m^p}{\partial(\Delta\lambda)}$$

$$\Rightarrow \frac{\partial \Phi}{\partial(\Delta\lambda)} = -\mathbf{N}_{n+1} : \mathcal{C}_n^e : \mathbf{N}_n + \frac{\partial \Phi}{\partial f} HH|_n + \frac{\partial \Phi}{\partial \varepsilon_m^p} H3|_n$$

iv. Υπολογισμός μιας διόρθωσης  $\Delta\Delta\lambda$  από τον τύπο  $\Delta\Delta\lambda = -\frac{\Phi(\Delta\lambda)}{\frac{\partial \Phi}{\partial(\Delta\lambda)}}$

v. Υπολογισμός του καινούριου  $\Delta\lambda^{(n+1)}$  από τη σχέση:

$$\Delta\lambda^{(n+1)} = \Delta\lambda^{(n)} + \Delta\Delta\lambda$$

LOOP

Αφού υπολογιστεί το  $\Delta\lambda$  συνεχίζουμε με τον προσδιορισμό των  $\sigma_{n+1}, f_{n+1}$  και  $\varepsilon_m^p|_{n+1}$  από τις εξισώσεις (3.2.2), (3.2.3) και (3.2.4) αντίστοιχα και έπειτα προχωρούμε στον υπολογισμό της ιακωβιανής στο τέλος του βήματος  $n+1$  :

$$\mathcal{C}^{ep} = \mathcal{C}_{n+1}^e - \frac{1}{L} \mathcal{C}_{n+1}^e : \mathbf{N}_{n+1} \mathbf{N}_{n+1} : \mathcal{C}_{n+1}^e \quad (3.2.6)$$

με

$$L = H|_{n+1} + \mathbf{N}_{n+1} : \mathcal{C}_{n+1}^e : \mathbf{N}_{n+1} \quad (3.2.7)$$

όπου  $H$  είναι αυτό στη σχέση (2.3.9).

### Παρατήρηση

Κατά την εφαρμογή της μεθόδου ολοκλήρωσης forward Euler απαιτείται ο υπολογισμός ενός «ελαστικού κλάσματος»,  $r$ , όπως φαίνεται στο σχήμα 3.1. Μπορούν να υπάρξουν δύο περιπτώσεις :

- *Περίπτωση I*

Η εντατική κατάσταση του φορέα να βρίσκεται μέσα στην επιφάνεια διαρροής ( $\sigma_n^{(1)}$ ) και ύστερα από φόρτιση να βγαίνει έξω από αυτή ( $\sigma^e$ ). Η επιστροφή προς την επιφάνεια διαρροής θα πρέπει να γίνει κατά την διεύθυνση του  $\mathbf{N}_n^{(2)}$ . Θα πρέπει, συνεπώς να υπολογιστεί η εντατική κατάσταση  $\sigma_n^{(2)}$  από τη σχέση :

$$\sigma_n^{(2)} = \sigma_n^{(1)} + r \Delta \sigma^e \quad (3.2.8)$$

όπου

$$\Delta \sigma^e = \mathcal{C}^e : \Delta \mathbf{E} \quad (3.2.9)$$

- *Περίπτωση II*

Όμοια, όταν η εντατική κατάσταση του φορέα βρίσκεται πάνω στην επιφάνεια διαρροής ( $\sigma_n^{(1)}$ ) και ύστερα από φόρτιση το διάνυσμα που ενώνει το  $\sigma_n^{(1)}$  με το  $\sigma^e$  τέμνει την επιφάνεια διαρροής απαιτείται ο υπολογισμός του  $\sigma_n^{(2)}$  χρησιμοποιώντας την προηγούμενη σχέση με τα αντίστοιχα  $\sigma_n^{(1)}$  και  $\Delta \sigma^e$  της περίπτωσης II. Η επιστροφή της λύσης πρέπει να γίνει, συνεπώς, στη διεύθυνση του διανύσματος  $\mathbf{N}_n^{(2)}$  και όχι στην  $\mathbf{N}_n^{(1)}$ .

Στην περίπτωση του μοντέλου του **Gurson**, ο υπολογισμός του ελαστικού κλάσματος,  $r$ , δεν μπορεί να γίνει αναλυτικά. Αρχικά, αναπτύσσουμε το υπερβολικό συνημίτονο της σχέσης 2.3.1 σε μία σειρά Taylor :

$$\cosh\left(\frac{3}{2} \frac{p}{\sigma_m}\right) = 1 + \frac{1}{2} \frac{9}{4} \frac{p^2}{\sigma_m^2} \quad (3.2.10)$$

και θέτουμε όπου  $p = p_n + r \cdot \Delta p$  και αντικαθιστούμε στη συνάρτηση διαρροής (2.3.1) και έχουμε:

$$A \cdot r^2 + B \cdot r + C = 0 \quad (3.2.11)$$

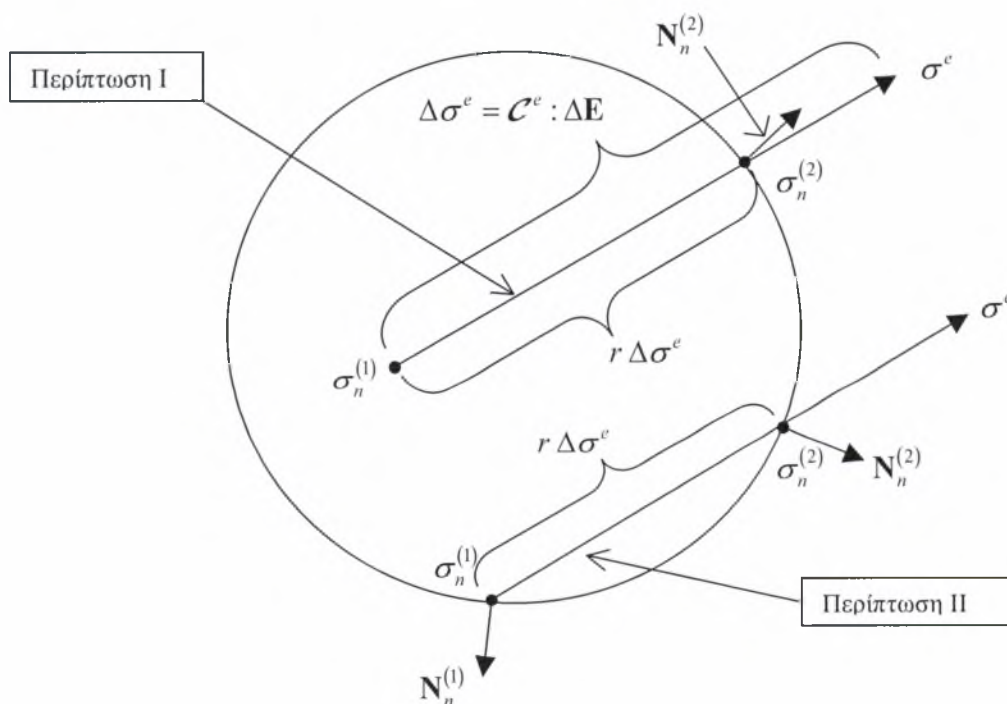
όπου

$$A = \left(\frac{3}{2}b + \frac{9}{4}f_n \Delta p^2\right), \quad B = \left(3a + \frac{9}{2}f_n p_n \Delta p\right), \quad C = q_n^2 + \frac{9}{4}f_n p_n^2 - (1-f_n)^2 \sigma_m^2 \quad (3.2.12)$$

όπου  $p$  η υδροστατική τάση και  $q$  η ισοδύναμη τάση κατά *von Mises* και λύνοντας την παραπάνω εξίσωση :

$$r = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (3.2.13)$$

έπειτα χρησιμοποιούμε μία μέθοδο Newton στην εξίσωση (2.3.1) προσπαθώντας να βρούμε το  $r$  για το οποίο ικανοποιείται ακριβώς η συνάρτηση διαρροής.



Σχήμα 3. 1

### 3.3 Αριθμητική ολοκλήρωση του ανισότροπου μοντέλου

Το ανισότροπο μοντέλο έχει ολοκληρωθεί με δύο διαφορετικές μεθόδους. Στην ενότητα 3.3.1 παρουσιάζεται η μέθοδος ολοκλήρωσης forward Euler η οποία είναι η ίδια με αυτή που χρησιμοποιήθηκε στο μοντέλο του **Gurson**. Στην ενότητα 3.3.2 παρουσιάζεται η μέθοδος backward Euler η οποία χρησιμοποιήθηκε σε μία πιο απλή μορφή και επιλύει την εξίσωση της συνάρτησης διαρροής και του κανόνα διαρροής στο βήμα  $n+1$ . Τέλος στο υποκεφάλαιο 3.4 παρουσιάζεται ο τρόπος υπολογισμού των ταυστών του **Eshelby**.

#### 3.3.1 Forward Euler στο ανισότροπο μοντέλο

Όπως και στο μοντέλο του **Gurson**, ισχύουν και στο ανισότροπο μοντέλο οι σχέσεις της ενότητας 3.1. Παίρνοντας, λοιπόν, υπόψη ότι τα  $\Phi, \mathbf{N}, g_1, g_2, g_3, g_4$  και  $\Omega^p$  είναι ισότροπες συναρτήσεις των μεταβλητών τους οι ελαστοπλαστικές εξισώσεις μπορούν να γραφούν στη μορφή<sup>1</sup> (M. Kailasam, N. Aravas και P. Ponte Castañeda [13]):

$$\dot{\mathbf{E}} = \dot{\mathbf{E}}^e + \dot{\mathbf{E}}^p \quad (3.3.1)$$

$$\dot{\hat{\sigma}} = \hat{\mathcal{L}}^e : \dot{\mathbf{E}}^e + \dot{\lambda} \left[ \hat{\sigma} \cdot \Omega^p(\hat{\sigma}, \hat{s}) - \Omega^p(\hat{\sigma}, \hat{s}) \cdot \hat{\sigma} \right] \quad (3.3.2)$$

$$\Phi(\hat{\sigma}, \hat{s}) = 0 \quad (3.3.3)$$

$$\dot{\mathbf{E}}^p = \dot{\lambda} \mathbf{N}(\hat{\sigma}, \hat{s}) \quad (3.3.4)$$

$$\dot{\varepsilon}^p = \dot{\lambda} g_1(\hat{\sigma}, \hat{s}) \quad (3.3.5)$$

$$\dot{f} = \dot{\lambda} g_2(\hat{\sigma}, \hat{s}) \quad (3.3.6)$$

$$\dot{w}_1 = \dot{\lambda} g_3(\hat{\sigma}, \hat{s}) \quad (3.3.7)$$

$$\dot{w}_2 = \dot{\lambda} g_4(\hat{\sigma}, \hat{s}) \quad (3.3.8)$$

$$\dot{\hat{\mathbf{n}}}^{(i)} = -\dot{\lambda} \Omega^p(\hat{\sigma}, \hat{s}) \cdot \hat{\mathbf{n}}^{(i)} \quad (3.3.9)$$

όπου  $\hat{\mathcal{L}}_{ijkl}^e = R_{mi} R_{nj} R_{pk} R_{ql} \mathcal{L}_{mnpq}^e$  και  $\hat{s} = \left\{ f, w_1, w_2, \hat{\mathbf{n}}^{(1)}, \hat{\mathbf{n}}^{(2)}, \hat{\mathbf{n}}^{(3)} \right\}$ .

<sup>1</sup> Στις παρακάτω εξισώσεις οι συναρτήσεις  $g_1, g_2, g_3, g_4$  χρησιμοποιούνται στις υπορουτίνες mahesh\_forw.f και mahesh\_back.f που θα παρουσιαστούν στο τέλος της εργασίας με τα ονόματα H3, HH, G1, G2 αντίστοιχα.

Η ολοκλήρωση της εξίσωσης (3.3.1) δίνει :

$$\Delta E = \Delta E^e + \Delta E^p \Leftrightarrow \Delta E^e = \Delta E - \Delta E^p \quad (3.3.10)$$

όπου χρησιμοποιείται η έκφραση  $\Delta A = A_{n+1} - A_n$ , και  $\Delta E = E_{n+1} = \text{γνωστό}$ . Η μέθοδος forward Euler χρησιμοποιείται για την αριθμητική ολοκλήρωση των εξισώσεων (3.3.2) και (3.3.4) :

$$\hat{\sigma}_{n+1} = \sigma_n + \mathcal{L}_n^e : \Delta E^e + \Delta \lambda (\sigma_n \cdot \Omega_n^p - \Omega_n^p \cdot \sigma_n) \quad (3.3.11)$$

και

$$\Delta E^p = \Delta \lambda \mathbf{N}_n \quad (3.3.12)$$

όπου χρησιμοποιήθηκε το γεγονός ότι  $\hat{\mathcal{L}}_n^e = \mathcal{L}_n^e$ . Με τη βοήθεια των σχέσεων (3.3.10) και (3.3.12) μπορούμε να γράψουμε την (3.3.11) ως :

$$\hat{\sigma}_{n+1} = \sigma^e - \Delta \lambda (\mathcal{L}_n^e : \mathbf{N}_n - \sigma_n \cdot \Omega_n^p - \Omega_n^p \cdot \sigma_n) \quad (3.3.13)$$

με  $\sigma^e = \sigma_n + \mathcal{L}_n^e : \Delta E = \text{γνωστό}$  που έχει πάρει την ονομασία “ελαστική πρόβλεψη” ή “elastic predictor”.

Οι εξισώσεις εξελίξεως των καταστατικών μεταβλητών  $\bar{\varepsilon}^p$ ,  $f$ ,  $w_1$  και  $w_2$  επίσης ολοκληρώνονται με το σχήμα forward Euler και παίρνουν τη μορφή :

$$\bar{\varepsilon}_{n+1}^p(\Delta \lambda) = \bar{\varepsilon}_n^p + \Delta \lambda g_1(\sigma_n, s_n) \quad (3.3.14)$$

$$f_{n+1}(\Delta \lambda) = f_n + \Delta \lambda g_2(\sigma_n, s_n) \quad (3.3.15)$$

$$w_1|_{n+1}(\Delta \lambda) = w_1|_n + \Delta \lambda g_3(\sigma_n, s_n) \quad (3.3.16)$$

$$w_2|_{n+1}(\Delta \lambda) = w_2|_n + \Delta \lambda g_4(\sigma_n, s_n) \quad (3.3.17)$$

Επίσης η εξίσωση εξελίξεως (3.3.9) για τα  $\hat{\mathbf{n}}^{(i)}$  προσεγγίζεται από τη σχέση :

$$\dot{\hat{\mathbf{n}}}^{(i)} = -\dot{\lambda} \Omega_n^p \cdot \hat{\mathbf{n}}^{(i)} \Rightarrow \frac{d\hat{\mathbf{n}}^{(i)}}{d\lambda} = -\Omega_n^p \cdot \hat{\mathbf{n}}^{(i)} \quad (3.3.18)$$

η οποία με τη σειρά της μπορεί να ολοκληρωθεί και να δώσει :

$$\hat{\mathbf{n}}_{n+1}^{(i)} = \exp(-\Delta \lambda \Omega_n^p) \cdot \mathbf{n}_n^{(i)} \quad (3.3.19)$$

Το εκθετικό ενός αντισυμμετρικού τανυστή δευτέρας τάξεως  $\mathbf{A}$  ( $\mathbf{A}^T = -\mathbf{A}$ ) είναι ένας ορθογώνιος τανυστής και μπορεί να υπολογιστεί από την ακόλουθη έκφραση την οποία απέδειξε ο Gibbs ( Cheng και Gupta [4] ):

$$\exp(\mathbf{A}) = \delta + \frac{\sin a}{s} \mathbf{A} + \frac{1 - \cos a}{a^2} \mathbf{A}^2 \quad (3.3.20)$$

όπου  $a = \sqrt{A_y A_y / 2}$  είναι το μέτρο του αξονικού διανύσματος του  $\mathbf{A}$ .

Στην ειδική περίπτωση ενός διδιάστατου προβλήματος στο οποίο η κίνηση γίνεται στο επίπεδο  $x_1 - x_2$ , τα  $\hat{\mathbf{n}}^{(i)}$  μπορούν να γραφούν ως εξής :

$$\hat{\mathbf{n}}^{(1)} = \cos \theta \mathbf{e}_1 + \sin \theta \mathbf{e}_2, \quad \hat{\mathbf{n}}^{(2)} = -\sin \theta \mathbf{e}_1 + \cos \theta \mathbf{e}_2, \quad \hat{\mathbf{n}}^{(3)} = \hat{\mathbf{n}}^{(1)} \times \hat{\mathbf{n}}^{(2)} \quad (3.3.21)$$

όπου  $(\mathbf{e}_1, \mathbf{e}_2)$  είναι τα μοναδιαία διανύσματα κατά μήκος των αξόνων  $x_1$  και  $x_2$ .

Στην περίπτωση αυτή η εξίσωση (3.3.19) είναι ισοδύναμη με την :

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \Delta \hat{\theta}, \quad \Delta \hat{\theta} = -\Delta \lambda \omega_n^p \quad (3.3.22)$$

όπου  $\omega_n^p$  καθορίζεται από τις εξισώσεις (2.4.31) και (2.4.32).

Οι εξισώσεις (3.3.13) - (3.3.17) και (3.3.19) καθορίζουν τις ποσότητες  $\hat{\sigma}_{n+1}$  και  $\hat{s}_{n+1}$  συναρτήσει του  $\Delta \lambda$ . Η αντικατάσταση των  $\hat{\sigma}_{n+1}(\Delta \lambda)$  και  $\hat{s}_{n+1}(\Delta \lambda)$  στην συνάρτηση διαρροής (3.3.3) δίνει μία αλγεβρική εξίσωση της μορφής :

$$\Phi(\hat{\sigma}_{n+1}(\Delta \lambda), \hat{s}_{n+1}(\Delta \lambda)) = 0 \quad (3.3.23)$$

η οποία λύνεται αριθμητικά με τη μέθοδο Newton που παρουσιάστηκε στην προηγούμενη ενότητα για τον υπολογισμό του  $\Delta \lambda$ . Αφού υπολογιστεί το  $\Delta \lambda$ , οι εξισώσεις (3.3.13) - (3.3.17) και (3.3.19) καθορίζουν τα  $\hat{\sigma}_{n+1}$  και  $\hat{s}_{n+1}$ . Τελικά, τα  $\sigma_{n+1}$  και  $\mathbf{n}_{n+1}^{(i)}$  υπολογίζονται από τις σχέσεις :

$$\sigma_{n+1} = \mathbf{R}_{n+1} \cdot \hat{\sigma}_{n+1} \cdot \mathbf{R}_{n+1}^T \quad \text{και} \quad \mathbf{n}_{n+1}^{(i)} = \mathbf{R}_{n+1} \cdot \hat{\mathbf{n}}_{n+1}^{(i)} \quad (3.3.24)$$

και με αυτό τον τρόπο ολοκληρώνεται η μέθοδος ολοκλήρωσης.

### Παρατήρηση

Όπως και στο μοντέλο του *Gurson* έτσι και στο ανισότροπο μοντέλο απαιτείται ο υπολογισμός ενός « ελαστικού κλάσματος » για κάθε σημείο ολοκλήρωσης σε κάθε

στοιχείο. Στο ανισότροπο μοντέλο είναι δυνατόν να υπολογιστεί αναλυτικά η έκφραση που το υπολογίζει και προκύπτει από τη λύση της εξίσωσης :

$$A \cdot r^2 + 2B \cdot r + C = 0 \quad (3.3.25)$$

όπου

$$A = \Delta \sigma^e : \mathbf{m}(s) : \Delta \sigma^e, \quad B = \sigma_n : \mathbf{m}(s) : \Delta \sigma^e, \quad C = (1 - f_n) \cdot \Phi(\sigma_n, s) \quad (3.3.26)$$

οπότε

$$r = \frac{-B + \sqrt{B^2 - A \cdot C}}{A} \quad (3.3.27)$$

### 3.3.2 Backward Euler στο ανισότροπο μοντέλο

Η μέθοδος αυτή αναπτύχθηκε γιατί γενικά είναι περισσότερο ευσταθής από την forward Euler για μεγάλα βήματα. Πιο συγκεκριμένα δεν ακολουθείται μέθοδος ολοκλήρωσης backward Euler σε όλες τις ελαστο – πλαστικές εξισώσεις αλλά μόνο στις εξισώσεις του κανόνα πλαστικής διαρροής. Οι υπόλοιπες καταστατικές εξισώσεις ολοκληρώνονται με μέθοδο forward Euler. Πιο αναλυτικά, έχουμε τους δύο βασικούς αγνώστους  $\Delta \lambda$  και  $\Delta \mathbf{E}^p$  και τις δύο βασικές εξισώσεις :

$$\Phi(\hat{\sigma}_{n+1}, \bar{\varepsilon}_{n+1}^p, f_{n+1}, w_{n+1}^\alpha, \hat{\mathbf{n}}_{n+1}^{(i)}) = 0 \quad (3.3.28)$$

$$\mathbf{G} = \Delta \mathbf{E}^p - \Delta \lambda \mathbf{N}_{n+1}(\hat{\sigma}_{n+1}, f_{n+1}, w_{n+1}^\alpha, \hat{\mathbf{n}}_{n+1}^{(i)}) = \mathbf{0} \quad (3.3.29)$$

οι οποίες λύνονται την επαναληπτική μέθοδο Newton.

Κατά τη διάρκεια της ολοκλήρωσης, για κάθε ζεύγος  $\Delta \lambda$  και  $\Delta \mathbf{E}^p$ , τα  $\hat{\sigma}_{n+1}, \Delta \mathbf{E}^p, \Delta f, \Delta w^\alpha$  και  $\hat{\mathbf{n}}_{n+1}^{(i)}$  καθορίζονται από τις παρακάτω σχέσεις :

$$\begin{aligned} \hat{\sigma}_{n+1}(\Delta \lambda, \Delta \mathbf{E}^p) &= \hat{\sigma}^e - \mathcal{L}_n^e : \Delta \mathbf{E}^p + \Delta \lambda (\sigma_n : \mathbf{\Omega}_n^p - \mathbf{\Omega}_n^p : \sigma_n) \\ &\equiv \hat{\sigma}^e - \mathcal{L}_n^e : \Delta \mathbf{E}^p + \Delta \lambda \mathbf{R}_1 \end{aligned} \quad (3.3.30)$$

$$\Delta \bar{\varepsilon}^p(\Delta \mathbf{E}^p) = \frac{\hat{\sigma}_n : \Delta \mathbf{E}^p}{(1 - f_n) \sigma_y(\bar{\varepsilon}_n^p)} \equiv \mathbf{R}_2 : \Delta \mathbf{E}^p \quad (3.3.31)$$

$$\Delta f(\Delta \mathbf{E}^p) = (1 - f_n) \Delta E_{kk}^p \quad (3.3.32)$$

$$\Delta w_\alpha(\Delta \mathbf{E}^p) = w_n^\alpha (\mathbf{n}_n^{(3)} \mathbf{n}_n^{(3)} - \mathbf{n}_n^{(\alpha)} \mathbf{n}_n^{(\alpha)}) : \mathbf{A}_n : \Delta \mathbf{E}^p \equiv \mathbf{R}_{3\alpha} : \Delta \mathbf{E}^p \quad (3.3.33)$$



$$\hat{\mathbf{n}}_{n+1}^{(i)}(\Delta\lambda) = \exp(-\Delta\lambda \boldsymbol{\Omega}_n^p) \cdot \mathbf{n}_n^{(i)} \quad (3.3.34)$$

το παραπάνω σύστημα εξισώσεων λύνεται θέτοντας μία πρώτη εκτίμηση για το  $\Delta\lambda$  που δίνεται από τη συνθήκη της συνέχειας :

$$\dot{\Phi} = 0 \Rightarrow \Delta\lambda = \frac{1}{H} \frac{\partial \Phi}{\partial \sigma} : \Delta\sigma = \frac{1}{L} \frac{\partial \Phi}{\partial \sigma} : \mathcal{L}_n^e : \Delta\mathbf{E} \quad (3.3.35)$$

Η ιακωβιανή του παραπάνω συστήματος καθορίζεται από τις παρακάτω εξισώσεις :

$$\frac{\partial \Phi}{\partial \lambda} = \frac{\partial \Phi}{\partial \hat{\boldsymbol{\sigma}}} : \frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \lambda} + \frac{\partial \Phi}{\partial \hat{\mathbf{n}}^{(i)}} \cdot \frac{\partial \hat{\mathbf{n}}^{(i)}}{\partial \lambda} = \mathbf{N} : \mathbf{R}_1 \quad (3.3.36)$$

$$\begin{aligned} \frac{\partial \Phi}{\partial \mathbf{E}^p} &= \frac{\partial \Phi}{\partial \hat{\boldsymbol{\sigma}}} : \frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \mathbf{E}^p} + \frac{\partial \Phi}{\partial \bar{\boldsymbol{\varepsilon}}^p} \frac{\partial \bar{\boldsymbol{\varepsilon}}^p}{\partial \mathbf{E}^p} + \frac{\partial \Phi}{\partial f} \frac{\partial f}{\partial \mathbf{E}^p} + \frac{\partial \Phi}{\partial w_\alpha} \frac{\partial w_\alpha}{\partial \mathbf{E}^p} = \\ &= -\mathbf{N} : \mathcal{L}_n^e + \frac{\partial \Phi}{\partial \bar{\boldsymbol{\varepsilon}}^p} \mathbf{R}_2 + \frac{\partial \Phi}{\partial f} \frac{\partial f}{\partial \mathbf{E}^p} + \frac{\partial \Phi}{\partial w_\alpha} \mathbf{R}_{3\alpha} \end{aligned} \quad (3.3.37)$$

και

$$\frac{\partial \mathbf{G}}{\partial \lambda} = -\mathbf{N} - \Delta\lambda \left( \frac{\partial \mathbf{N}}{\partial \hat{\boldsymbol{\sigma}}} : \frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \lambda} + \frac{\partial \mathbf{N}}{\partial \hat{\mathbf{n}}^{(i)}} \cdot \frac{\partial \hat{\mathbf{n}}^{(i)}}{\partial \lambda} \right) = -\mathbf{N} - \Delta\lambda \frac{2}{1-f} \mathbf{m} : \mathbf{R}_1 \quad (3.3.38)$$

$$\begin{aligned} \frac{\partial \mathbf{G}}{\partial \mathbf{E}^p} &= \mathbf{I} - \Delta\lambda \left( \frac{\partial \mathbf{N}}{\partial \hat{\boldsymbol{\sigma}}} : \frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \mathbf{E}^p} + \frac{\partial \mathbf{N}}{\partial f} \frac{\partial f}{\partial \mathbf{E}^p} + \frac{\partial \mathbf{N}}{\partial w_\alpha} \frac{\partial w_\alpha}{\partial \mathbf{E}^p} \right) = \\ &= \mathbf{I} - \Delta\lambda \left( -\frac{2}{1-f} \mathbf{m} : \mathcal{L}_n^e + \frac{\partial \mathbf{N}}{\partial f} \frac{\partial f}{\partial \mathbf{E}^p} + \frac{\partial \mathbf{N}}{\partial w_\alpha} \mathbf{R}_{3\alpha} \right) \end{aligned} \quad (3.3.39)$$

Πιο συγκεκριμένα οι παραπάνω παράγωγοι δίνονται από τις παρακάτω σχέσεις :

$$\frac{\partial \mathbf{N}}{\partial \hat{\boldsymbol{\sigma}}} = \frac{2}{1-f} \mathbf{m} \quad (3.3.40)$$

$$\begin{aligned} \frac{\partial \mathbf{N}}{\partial f} &= \frac{2}{(1-f)^2} \mathbf{m} : \hat{\boldsymbol{\sigma}} + \frac{2}{1-f} \frac{\partial \mathbf{m}}{\partial f} : \hat{\boldsymbol{\sigma}} = \frac{2}{(1-f)^2} \left[ \mathbf{m} + (1-f) \frac{\partial \mathbf{m}}{\partial f} \right] : \hat{\boldsymbol{\sigma}} = \\ &= \frac{3}{(1-f)^2} \left[ \mathbf{K} + \frac{2(1+f)}{1-f} \mu \mathbf{Q}^{-1} \Big|_{\nu=1/2} \right] : \hat{\boldsymbol{\sigma}} \end{aligned} \quad (3.3.41)$$

Η παράγωγος  $\frac{\partial \mathbf{N}}{\partial w_\alpha}$  θα υπολογιστεί αριθμητικά γιατί είναι πολύ δύσκολο να βρεθεί

αναλυτική έκφραση για το  $\frac{\partial \mathbf{m}}{\partial w_\alpha}$  ενώ οι παράγωγοι  $\frac{\partial \mathbf{N}}{\partial \hat{\mathbf{n}}^{(i)}}$  και  $\frac{\partial \Phi}{\partial \hat{\mathbf{n}}^{(i)}} \cdot \frac{\partial \hat{\mathbf{n}}^{(i)}}{\partial \lambda}$  θα τεθούν

προσεγγιστικά ίσες με μηδέν:

$$\frac{\partial \mathbf{N}}{\partial w_\alpha} = \frac{2}{1-f} \frac{\partial m}{\partial w_\alpha} : \hat{\boldsymbol{\sigma}} \quad (3.3.42)$$

$$\frac{\partial \mathbf{N}}{\partial \hat{\mathbf{n}}^{(i)}} = \frac{2}{1-f} \frac{\partial m}{\partial \hat{\mathbf{n}}^{(i)}} : \hat{\boldsymbol{\sigma}} = \mathbf{0} \quad (3.3.43)$$

και επιπλέον έχουμε ότι

$$\frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \lambda} = \mathbf{R}_1, \quad \frac{\partial \hat{\boldsymbol{\sigma}}}{\partial \mathbf{E}^p} = -\mathcal{L}_n^e, \quad \frac{\partial \hat{\mathbf{n}}^{(i)}}{\partial \lambda} = -\boldsymbol{\Omega}_n^p \cdot \mathbf{n}_n^{(i)} \quad (3.3.44)$$

και

$$\frac{\partial \bar{\mathbf{E}}^p}{\partial \mathbf{E}^p} = \mathbf{R}_2, \quad \frac{\partial f}{\partial \mathbf{E}^p} = (1-f_n)\boldsymbol{\delta}, \quad \frac{\partial w_\alpha}{\partial \mathbf{E}^p} = \mathbf{R}_{3\alpha} \quad (3.3.45)$$

Συγκεκριμένα, το σύστημα που θα λυθεί σε ένα επίπεδο πρόβλημα δίνεται σε μητρωϊκή μορφή από την ακόλουθη έκφραση :

$$\begin{bmatrix} \frac{\partial \Phi}{\partial \lambda} & \frac{\partial \Phi}{\partial E_{11}^p} & \frac{\partial \Phi}{\partial E_{22}^p} & \frac{\partial \Phi}{\partial E_{33}^p} & \frac{\partial \Phi}{\partial E_{12}^p} \\ \frac{\partial G_1}{\partial \lambda} & \frac{\partial G_1}{\partial E_{11}^p} & \frac{\partial G_1}{\partial E_{22}^p} & \frac{\partial G_1}{\partial E_{33}^p} & \frac{\partial G_1}{\partial E_{12}^p} \\ \frac{\partial G_2}{\partial \lambda} & \frac{\partial G_2}{\partial E_{11}^p} & \frac{\partial G_2}{\partial E_{22}^p} & \frac{\partial G_2}{\partial E_{33}^p} & \frac{\partial G_2}{\partial E_{12}^p} \\ \frac{\partial G_3}{\partial \lambda} & \frac{\partial G_3}{\partial E_{11}^p} & \frac{\partial G_3}{\partial E_{22}^p} & \frac{\partial G_3}{\partial E_{33}^p} & \frac{\partial G_3}{\partial E_{12}^p} \\ \frac{\partial G_4}{\partial \lambda} & \frac{\partial G_4}{\partial E_{11}^p} & \frac{\partial G_4}{\partial E_{22}^p} & \frac{\partial G_4}{\partial E_{33}^p} & \frac{\partial G_4}{\partial E_{12}^p} \end{bmatrix} \cdot \begin{Bmatrix} \delta(\Delta\lambda) \\ \delta(\Delta E_{11}^p) \\ \delta(\Delta E_{22}^p) \\ \delta(\Delta E_{33}^p) \\ \delta(\Delta E_{12}^p) \end{Bmatrix} = - \begin{Bmatrix} \Phi \\ G_1 \\ G_2 \\ G_3 \\ G_4 \end{Bmatrix} \quad (3.3.46)$$

Το σύστημα λύνεται ως προς  $\delta(\Delta\lambda), \delta(\Delta E^p)$  και όπου  $G_a = \Delta E_{aa}^p - \Delta\lambda N_{aa}$  με  $a=1,2,3$  (όχι άθροιση στο  $a$ ) και  $G_4 = \Delta E_{12}^p - \Delta\lambda N_{12}$ . Όταν συγκλίνει το σύστημα και για τους πέντε παραπάνω αγνώστους, δηλαδή ικανοποιηθούν και οι πέντε εξισώσεις

$$\Phi(\hat{\boldsymbol{\sigma}}_{n+1}, \bar{\mathbf{E}}_{n+1}^p, f_{n+1}, w_{n+1}^\alpha, \hat{\mathbf{n}}_{n+1}^{(i)}) = 0 \quad (1 \text{ εξίσωση}) \quad (3.3.47)$$

$$\mathbf{G} = \Delta \mathbf{E}^p - \Delta\lambda \mathbf{N}_{n+1}(\hat{\boldsymbol{\sigma}}_{n+1}, f_{n+1}, w_{n+1}^\alpha, \hat{\mathbf{n}}_{n+1}^{(i)}) = \mathbf{0} \quad (4 \text{ εξισώσεις}) \quad (3.3.48)$$

τότε τα  $\boldsymbol{\sigma}_{n+1}$  και  $\mathbf{n}_{n+1}^{(i)}$  υπολογίζονται από τις σχέσεις :

$$\boldsymbol{\sigma}_{n+1} = \mathbf{R}_{n+1} \cdot \hat{\boldsymbol{\sigma}}_{n+1} \cdot \mathbf{R}_{n+1}^T \quad \text{και} \quad \mathbf{n}_{n+1}^{(i)} = \mathbf{R}_{n+1} \cdot \hat{\mathbf{n}}_{n+1}^{(i)} \quad (3.3.49)$$



διαρροής και συνεχίζει έξω από αυτή έως το σημείο B. Το μέγεθος του διανύσματος που ενώνει το  $\sigma_n$  με το  $\sigma^e$  εξαρτάται από το μέγεθος του εκάστοτε βήματος. Έπειτα το σημείο αυτό πρέπει να επιστρέψει προς τα πίσω μέχρι να πέσει πάνω στην, κατά λίγο μεγαλύτερη, καινούρια επιφάνεια διαρροής – η οποία στην περίπτωση του σχήματος 3.2 για λόγους απλότητας είναι η ίδια με αυτή στο προηγούμενο βήμα. Ο τρόπος με τον οποίο θα γίνει η επιστροφή αυτή διαχωρίζει τις δύο μεθόδους.

Στην backward μέθοδο το σημείο B επιστρέφει «ακτινικά» στη διεύθυνση του διανύσματος  $N_{n+1}$  και είναι σίγουρο ότι θα τέμνει την καινούρια επιφάνεια διαρροής. Το σημείο τομής A θα αποτελεί την λύση στο βήμα  $n+1$ . Αντιθέτως στην περίπτωση της forward μεθόδου η επιστροφή γίνεται σε διεύθυνση «παράλληλη» με αυτή του διανύσματος  $N_n$ . Όπως φαίνεται και από το σχήμα 4.22 αν το βήμα είναι μεγάλο τότε το σημείο G δεν θα τέμνει σε καμία περίπτωση την νέα επιφάνεια διαρροής με αποτέλεσμα να μην υπάρχει λύση για το πρόβλημα.

Συνεπώς, προκύπτει ότι η forward μέθοδος είναι περισσότερο ευαίσθητη σε μεγάλα βήματα φόρτισης ενός σώματος συγκριτικά με την backward μέθοδο. Αυτό σημαίνει ότι η backward μέθοδος χρειάζεται λιγότερο χρόνο (Πίνακας 1) από τη forward για να ολοκληρωθεί το συνολικό βήμα φόρτισης (step time).

Ειδικότερα προκύπτει ότι όταν έχουμε μικρή ή καθόλου σκλήρυνση και οι κύριες διευθύνσεις των τάσεων περιστρέφονται κατά τη διάρκεια ενός βήματος τότε υπάρχει πιθανότητα να μην υπάρχει λύση στο πρόβλημά μας, αν το μέγεθος του  $\Delta E$  ξεπεράσει την τιμή της παραμόρφωσης διαρροής ( Ortiz και Popov [15] ).

Πίνακας 1

Χρόνος εκπλήρωσης ενός συνολικού βήματος φόρτισης (sec)		
	backward	forward
Βήμα 1	1218	1347
Βήμα 2	1316	1989
Βήμα 3	1175	1745

### 3.4 Υπολογισμός των τανυστών του Eshelby και του τανυστή $\mathbf{Q}$

Στην ενότητα αυτή θα παρουσιαστεί ο τρόπος υπολογισμού των τανυστών του Eshelby και του τανυστή  $\mathbf{Q}$ . Ο Willis [18] έχει αποδείξει ότι οι ακριβείς σχέσεις υπολογισμού των τανυστών  $\mathbf{\Pi}$ ,  $\mathbf{S}$  και  $\mathbf{Q}$  είναι οι εξής :

$$\mathbf{\Pi} = \frac{1}{4\pi w_1 w_2} \int_{|\xi|=1} \hat{\mathbf{H}}(\xi) : \mathbf{L} \frac{dS(\xi)}{|Z^{-1} \cdot \xi|^3} \quad (3.4.1)$$

$$\mathbf{S} = \frac{1}{4\pi w_1 w_2} \int_{|\xi|=1} \mathbf{H}(\xi) : \mathbf{L} \frac{dS(\xi)}{|Z^{-1} \cdot \xi|^3} \quad (3.4.2)$$

$$\mathbf{Q} = \frac{\mu}{4\pi w_1 w_2} \int_{|\xi|=1} \mathbf{E}(\xi, \nu) \frac{dS(\xi)}{|Z^{-1} \cdot \xi|^3} \quad (3.4.3)$$

όπου  $Z = w_1 \mathbf{n}^{(1)} \mathbf{n}^{(1)} + w_2 \mathbf{n}^{(2)} \mathbf{n}^{(2)} + \mathbf{n}^{(3)} \mathbf{n}^{(3)}$  και

$$\begin{aligned} (\mathbf{H}(\xi) : \mathbf{L})_{ijkl} = & \frac{1}{2|\xi|^2} (\delta_{ik} \xi_j \xi_l + \delta_{jk} \xi_i \xi_l + \delta_{il} \xi_j \xi_k + \delta_{jl} \xi_i \xi_k) - \\ & - \frac{1}{|\xi|^4} \frac{1}{1-\nu} \xi_i \xi_j \xi_k \xi_l + \frac{1}{|\xi|^4} \frac{1}{1-\nu} \xi_i \xi_j \delta_{kl} \end{aligned} \quad (3.4.4)$$

$$(\hat{\mathbf{H}}(\xi) : \mathbf{L})_{ijkl} = \frac{1}{2|\xi|^2} (\delta_{ik} \xi_j \xi_l - \delta_{jk} \xi_i \xi_l + \delta_{il} \xi_j \xi_k - \delta_{jl} \xi_i \xi_k) \quad (3.4.5)$$

και

$$\begin{aligned} E_{ijkl}(\xi, \nu) = & \delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk} - \frac{1}{2|\xi|^2} (\delta_{ik} \xi_j \xi_l + \delta_{jk} \xi_i \xi_l + \delta_{il} \xi_j \xi_k + \delta_{jl} \xi_i \xi_k) + \\ & + \frac{2\nu}{1-\nu} \left[ \delta_{ij} \delta_{kl} - \frac{1}{|\xi|^2} (\delta_{ij} \xi_k \xi_l + \delta_{kl} \xi_i \xi_j) \right] + \frac{2}{|\xi|^4} \frac{1}{1-\nu} \xi_i \xi_j \xi_k \xi_l \end{aligned} \quad (3.4.6)$$

Οι εξισώσεις (3.4.1), (3.4.2) και (3.4.3) συνδυάζονται με τις (3.4.4), (3.4.5) και (3.4.6) για τον αριθμητικό υπολογισμό των  $\mathbf{\Pi}$ ,  $\mathbf{S}$  και  $\mathbf{Q}$ . Και στις τρεις περιπτώσεις χρειάζεται να υπολογιστούν αριθμητικά, ολοκληρώματα της μορφής

$$\int_{|\xi|=1} A(\xi) dS(\xi) = \int_{\phi=0}^{\pi} \int_{\theta=0}^{2\pi} A(\xi(\theta, \phi)) \sin \phi d\theta d\phi \quad (3.4.7)$$

όπου  $\xi = (\sin \phi \cos \theta, \sin \phi \sin \theta, \cos \phi)$ . Μετά από μετασχηματισμούς της μορφής  $\theta(r) = (r+1)\pi$  και  $\phi(s) = (s+1)\pi/2$  το παραπάνω ολοκλήρωμα μετατρέπεται σε

$$\frac{\pi^2}{2} \int_{-1}^1 \int_{-1}^1 A(\xi(\theta(r), \phi(s))) \sin \phi(s) dr ds \quad (3.4.8)$$

και το οποίο υπολογίζεται αριθμητικά με ολοκλήρωση Gauss της μορφής

$$\frac{\pi^2}{2} \sum_{i=1}^{NG} \sum_{j=1}^{NG} W_i W_j A(\xi(\theta_i, \phi_j)) \sin \phi_j \quad (3.4.9)$$

όπου  $\theta_i = \theta(r_i)$  και  $\phi_j = \phi(s_j)$ ,  $r_i$  και  $s_j$  είναι οι σταθμοί ολοκλήρωσης, και  $W_i$  και  $W_j$  είναι οι αντίστοιχοι συντελεστές βαρύτητας.

Παρακάτω παρουσιάζουμε ένα διαφορετικό τρόπο υπολογισμού των τανυστών αυτών τον οποίο πρότεινε ο Eshelby [7, 8]. Μπορούμε να μετατρέψουμε τα διπλά επιφανειακά ολοκληρώματα των προηγούμενων σχέσεων σε σχέσεις που περιέχουν μονά ελλειπτικά ολοκληρώματα. Έτσι καταλήγουμε στις παρακάτω σχέσεις για τον υπολογισμό των στοιχείων των τανυστών του Eshelby και του τανυστή  $\mathbf{Q}$ . Οι παρακάτω σχέσεις ισχύουν με τις προϋποθέσεις ότι:

- $a > b > c$  όπου  $a, b, c$  είναι οι κύριοι άξονες  $a_1, a_2, a_3$  του τοπικού ελλειψοειδούς και πρέπει κάθε φορά να είναι κατανεμημένοι έτσι ώστε να ισχύει η προηγούμενη ανισότητα.
- πρέπει επιπλέον να αναφερθεί ότι οι εξισώσεις για τα στοιχεία των τανυστών του Eshelby και του τανυστή  $\mathbf{Q}$  ισχύουν μόνο στην περίπτωση ενός κεντρικού συστήματος συντεταγμένων του οποίου οι άξονες είναι παράλληλοι με τους κύριους άξονες του τοπικού ελλειψοειδούς.

Για τον τανυστή  $\mathbf{S}$  έχουμε:

$$\begin{aligned} S_{11} &= S_{1111} = P a^2 I_{11} + R I_1 \\ S_{22} &= S_{2222} = P b^2 I_{22} + R I_2 \\ S_{33} &= S_{3333} = P c^2 I_{33} + R I_3 \\ S_{12} &= S_{1122} = P b^2 I_{12} - R I_1 \\ S_{13} &= S_{1133} = P c^2 I_{13} - R I_1 \\ S_{21} &= S_{2211} = P a^2 I_{12} - R I_2 \end{aligned} \quad (3.4.10)$$

$$\begin{aligned}
 S_{23} &= S_{2233} = P c^2 I_{23} - R I_2 \\
 S_{31} &= S_{3311} = P a^2 I_{13} - R I_3 \\
 S_{32} &= S_{3322} = P b^2 I_{23} - R I_3 \\
 S_{44} &= S_{1212} = \frac{P}{2}(a^2 + b^2)I_{12} + \frac{R}{2}(I_1 + I_2) \\
 S_{55} &= S_{1313} = \frac{P}{2}(a^2 + c^2)I_{13} + \frac{R}{2}(I_1 + I_3) \\
 S_{66} &= S_{2323} = \frac{P}{2}(b^2 + c^2)I_{23} + \frac{R}{2}(I_2 + I_3)
 \end{aligned}$$

όπου

$$P = \frac{3}{8\pi(1-\nu)}, \quad R = \frac{1-2\nu}{8\pi(1-\nu)} \quad (3.4.11)$$

Για τον τανυστή  $\Pi$  τα μοναδικά μη - μηδενικά στοιχεία είναι τα  $\Pi_{1212}, \Pi_{2323}, \Pi_{3131}$  και τα στοιχεία αυτά που προκύπτουν από εναλλαγή των δεικτών του πρώτου ζευγαριού και του δεύτερου (π.χ.  $\Pi_{2112}, \Pi_{2121}, \dots$ ). Πρέπει να σημειωθεί ότι ο τανυστής  $\Pi$  είναι αντισυμμετρικός ως προς τους πρώτους δύο δείκτες και συμμετρικός ως προς τους δύο τελευταίους. Άρα :

$$\begin{aligned}
 \Pi_{2121} &= \Pi_{2112} = -\Pi_{1221} = \frac{I_1 - I_2}{8\pi}, \\
 \Pi_{3131} &= \Pi_{3113} = -\Pi_{1331} = \frac{I_1 - I_3}{8\pi}
 \end{aligned} \quad (3.4.12)$$

Τέλος για τον τανυστή  $\mathcal{Q}$  ισχύουν οι παρακάτω σχέσεις :

$$\mathcal{Q}_{11} = \mathcal{Q}_{1111} = \frac{\mu}{2\pi(1-\nu)} \left( 4\pi - \frac{1}{2}I_1 - \frac{3a^2}{2}I_{11} \right) \quad (3.4.13)$$

$$\mathcal{Q}_{22} = \mathcal{Q}_{2222} = \frac{\mu}{2\pi(1-\nu)} \left( 4\pi - \frac{1}{2}I_2 - \frac{3b^2}{2}I_{22} \right) \quad (3.4.14)$$

$$\mathcal{Q}_{33} = \mathcal{Q}_{3333} = \frac{\mu}{2\pi(1-\nu)} \left( 4\pi - \frac{1}{2}I_3 - \frac{3c^2}{2}I_{33} \right) \quad (3.4.15)$$

$$\mathcal{Q}_{12} = \mathcal{Q}_{21} = \mathcal{Q}_{1122} = \frac{\mu}{8\pi(1-\nu)} \left[ 16\pi\nu + (1-4\nu)(I_1 + I_2) - 3(a^2 + b^2)I_{12} \right] \quad (3.4.16)$$

$$\mathcal{Q}_{13} = \mathcal{Q}_{31} = \mathcal{Q}_{1133} = \frac{\mu}{8\pi(1-\nu)} \left[ 16\pi\nu + (1-4\nu)(I_1 + I_3) - 3(a^2 + c^2)I_{13} \right] \quad (3.4.17)$$

$$Q_{23} = Q_{32} = Q_{2233} = \frac{\mu}{8\pi(1-\nu)} \left[ 16\pi\nu + (1-4\nu)(I_2 + I_3) - 3(b^2 + c^2)I_{23} \right] \quad (3.4.18)$$

$$Q_{44} = Q_{1212} = \mu - \frac{\mu}{8\pi(1-\nu)} \left[ (1-2\nu)(I_1 + I_2) + 3(a^2 + b^2)I_{12} \right] \quad (3.4.19)$$

$$Q_{55} = Q_{1313} = \mu - \frac{\mu}{8\pi(1-\nu)} \left[ (1-2\nu)(I_1 + I_3) + 3(a^2 + c^2)I_{13} \right] \quad (3.4.20)$$

$$Q_{66} = Q_{2323} = \mu - \frac{\mu}{8\pi(1-\nu)} \left[ (1-2\nu)(I_2 + I_3) + 3(b^2 + c^2)I_{23} \right] \quad (3.4.21)$$

όπου  $\mu$  τα μέτρο διατμήσεως και  $\nu$  το μέτρο του Poisson. Σε όλες τις παραπάνω σχέσεις χρησιμοποιούνται οι συντελεστές  $I$  οι οποίοι ορίζονται παρακάτω. Αρχικά θα πρέπει να διαχωρίσουμε τρεις βασικές περιπτώσεις για τις οποίες τα  $I$  παίρνουν διαφορετική μορφή και αυτές είναι :

I. για  $a > b > c$  έχουμε:

$$I_1 = \frac{4\pi abc}{(a^2 - b^2)(a^2 - c^2)^{1/2}} (F - E) \quad (3.4.22)$$

$$I_3 = \frac{4\pi abc}{(b^2 - c^2)(a^2 - c^2)^{1/2}} \left[ \frac{b(a^2 - c^2)^{1/2}}{ac} - E \right] \quad (3.4.23)$$

$$I_2 = 4\pi - I_1 - I_3 \quad (3.4.24)$$

$$I_{12} = \frac{I_2 - I_1}{3(a^2 - b^2)}, \quad I_{13} = \frac{I_3 - I_1}{3(a^2 - c^2)}, \quad I_{23} = \frac{I_3 - I_2}{3(b^2 - c^2)} \quad (3.4.25)$$

$$I_{11} = \frac{4\pi}{3a^2} - I_{12} - I_{13}, \quad I_{22} = \frac{4\pi}{3b^2} - I_{12} - I_{23}, \quad I_{33} = \frac{4\pi}{3c^2} - I_{13} - I_{23} \quad (3.4.26)$$

II. για  $a = b > c$  έχουμε

$$I_1 = I_2 = \frac{2\pi a^2 c}{(a^2 - c^2)^{3/2}} \left[ \cos^{-1} \left( \frac{c}{a} \right) - \frac{c}{a} \left( 1 - \frac{c^2}{a^2} \right)^{1/2} \right] \quad (3.4.27)$$

$$I_3 = 4\pi - 2I_1 \quad (3.4.28)$$

$$I_{13} = \frac{I_3 - I_1}{3(a^2 - c^2)}, \quad I_{23} = I_{13}, \quad I_{12} = \frac{\pi}{3a^2} - \frac{I_{13}}{4} \quad (3.4.29)$$



$$I_{11} = 3I_{12}, \quad I_{22} = I_{11}, \quad I_{33} = \frac{4\pi}{3c^2} - I_{13} - I_{23} \quad (3.4.30)$$

III. για  $a > b = c$  έχουμε

$$I_2 = I_3 = \frac{2\pi ac^2}{(a^2 - c^2)^{3/2}} \left[ \frac{a}{c} \left( \frac{a^2}{c^2} - 1 \right)^{1/2} - \cosh^{-1} \left( \frac{a}{c} \right) \right], \quad I_1 = 4\pi - 2I_2 \quad (3.4.31)$$

$$I_{12} = \frac{I_2 - I_1}{3(a^2 - b^2)}, \quad I_{13} = I_{12}, \quad I_{23} = \frac{\pi^2}{3b^2} - \frac{I_{12}}{4} \quad (3.4.32)$$

$$I_{11} = \frac{4\pi}{3a^2} - I_{12} - I_{13}, \quad I_{22} = 3I_{23}, \quad I_{33} = I_{22} \quad (3.4.33)$$

IV. για  $a = b = c$  έχουμε

$$I_1 = I_2 = I_3 = \frac{4\pi}{3}, \quad I_{11} = I_{22} = I_{33} = \frac{4\pi}{5a^2}, \quad I_{12} = I_{13} = I_{23} = \frac{I_{11}}{3} \quad (3.4.34)$$

όπου  $F = F(\theta, k)$  και  $E = E(\theta, k)$  είναι ολοκληρώματα πρώτου και δεύτερου τύπου

$$E(\theta, k) = \int_0^\theta \sqrt{1 - k^2 \sin^2 \phi} d\phi, \quad F(\theta, k) = \int_0^\theta \frac{d\phi}{\sqrt{1 - k^2 \sin^2 \phi}} \quad (3.4.35)$$

με

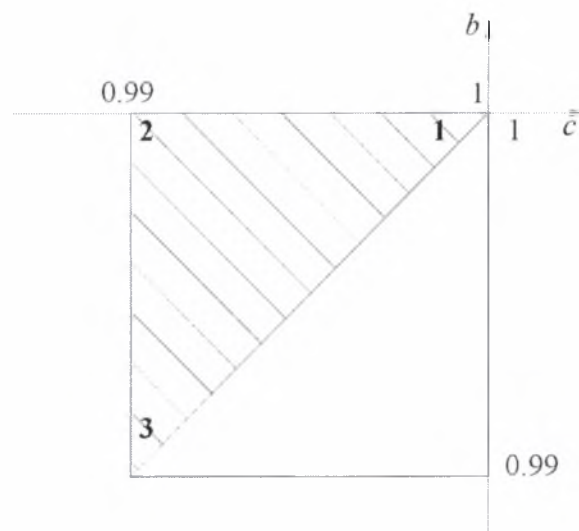
$$\theta = \sin^{-1} \left( 1 - \frac{c^2}{a^2} \right)^{1/2}, \quad k = \frac{(a^2 - b^2)^{1/2}}{(a^2 - c^2)^{1/2}} \quad (3.4.36)$$

Παρατηρήσεις

◇ Πρέπει να σημειωθεί ότι κατά τον υπολογισμό των ολοκληρωμάτων παρατηρήθηκε ότι υπήρχε πρόβλημα αριθμητικού υπολογισμού των τανυστών του Eshelby και του τανυστή  $\mathbf{Q}$  όταν τα  $a, b, c$  έπαιρναν τιμές πολύ κοντά στη μονάδα. Στην περίπτωση αυτή χρειάστηκε να γίνει γραμμική παρεμβολή μεταξύ των τεσσάρων προαναφερθέντων περιπτώσεων.

Πιο συγκεκριμένα η παρεμβολή γίνεται όταν τα  $w_1$  και  $w_2$  βρίσκονται ταυτόχρονα στο διάστημα  $(0.99, 1.01)$ . Στην περίπτωση αυτή απαιτείται ο διαχωρισμός σε τρεις περιπτώσεις όπου οι συναρτήσεις παρεμβολής παίρνουν διαφορετική μορφή:

$$\square \quad a = 1 > b > c > 0.99$$



Σχήμα 3.3

Χρησιμοποιούμε τις παρακάτω συναρτήσεις παρεμβολής

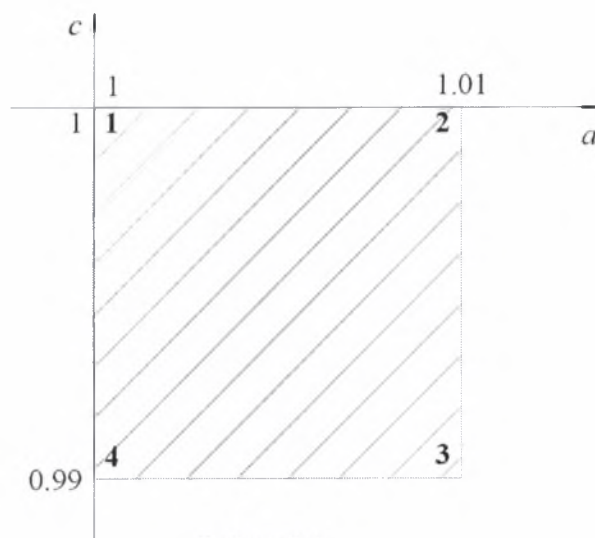
$$N_1(b, c) = \frac{c - 0.99}{0.01}, \quad N_2(b, c) = \frac{b - c}{0.01}, \quad N_3(b, c) = \frac{1 - b}{0.01} \quad (3.4.37)$$

όπου η  $N_1$  μηδενίζεται στα σημεία 2 και 3, η  $N_2$  μηδενίζεται στα σημεία 1 και 3 και η  $N_3$  μηδενίζεται στα σημεία 1 και 2. Τα  $I$  υπολογίζονται ακριβώς στα

σημεία **1,2** και **3** με τη βοήθεια των σχέσεων που έχουν αναφερθεί παραπάνω. Έπειτα γίνεται γραμμική παρεμβολή οπουδήποτε μέσα στο γραμμοσκιασμένου τριγωνικό τμήμα του σχήματος 1. Οι τανυστές (π.χ. ο τανυστής  $Q$ ) έπειτα υπολογίζονται από την σχέση :

$$Q(b,c) = \sum_{i=1}^3 Q^{(i)} \cdot N_i(b,c) \quad (3.4.38)$$

$$\square \quad 1.01 > a > b = 1 > c > 0.99$$



Σχήμα 3.4

Όμοια για την περίπτωση αυτή οι συναρτήσεις παρεμβολής είναι:

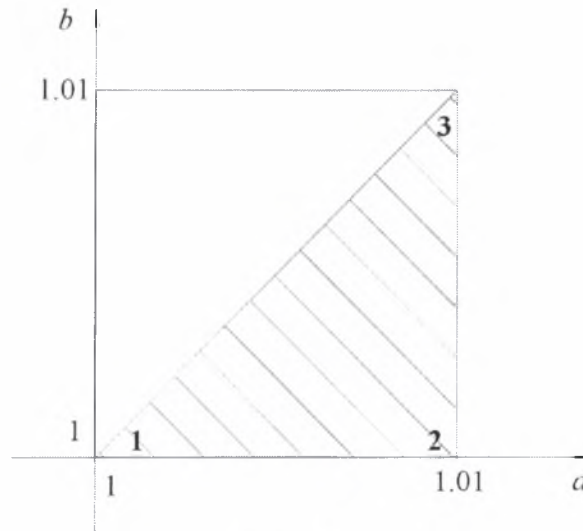
$$N_1(a,c) = \frac{(c-0.99)(1.01-a)}{0.0001}, \quad N_2(a,c) = \frac{(c-0.99)(a-1)}{0.0001}, \quad (3.4.39)$$

$$N_3(a,c) = \frac{(1-c)(a-1)}{0.0001}, \quad N_4(a,c) = \frac{(1-c)(1.01-a)}{0.0001}$$

και

$$Q(a,c) = \sum_{i=1}^4 Q^{(i)} \cdot N_i(a,c) \quad (3.4.40)$$

$$\square \quad 1.01 > a > b > c = 1$$



Σχήμα 3.5

Οι συναρτήσεις παρεμβολής για την περίπτωση αυτή είναι :

$$N_1(a,b) = \frac{1.01-a}{0.01}, \quad N_2(a,b) = \frac{a-b}{0.01}, \quad N_3(a,b) = \frac{b-1}{0.01} \quad (3.4.41)$$

και

$$Q(a,b) = \sum_{i=1}^3 Q^{(i)} \cdot N_i(a,b) \quad (3.4.42)$$

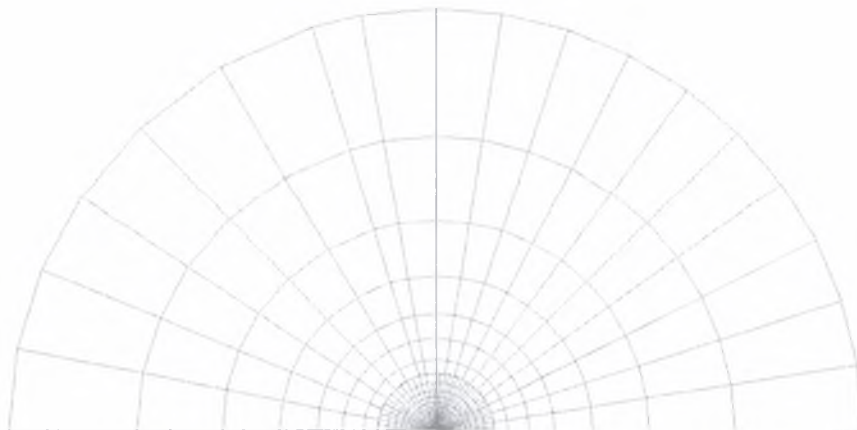
- ◇ Επίσης, είναι αναγκαίος ο υπολογισμός της παραγώγου  $\frac{\partial Q}{\partial w_a}$  όπου  $w_1$  είναι το κλάσμα του μοναδιαίου άξονα του ελλειψοειδούς προς τον μεγαλύτερο, ενώ το  $w_2$  είναι το κλάσμα του μοναδιαίου άξονα του ελλειψοειδούς προς τον μικρότερο, με άξονές του τα  $a, b, c$ . Ο υπολογισμός της παραγώγου γίνεται αριθμητικά.

## 4. Εφαρμογές

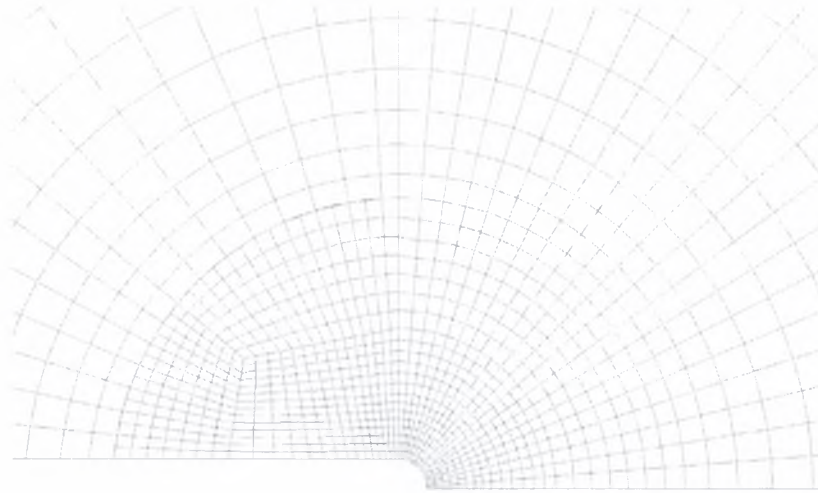
Στο παρόν κεφάλαιο θα παρουσιάσουμε δύο εφαρμογές των μοντέλων που αναλύθηκαν παραπάνω. Στην πρώτη από αυτές, θεωρούμε το πρόβλημα της ρωγμής εφελκυστικού τύπου (mode-I) σε συνθήκες επίπεδης παραμόρφωσης ενός ομογενούς πορώδους ελαστοπλαστικού υλικού. Στην δεύτερη εφαρμογή θεωρούμε το πρόβλημα του μονοαξονικού εφελκυσμού σε συνθήκες επίπεδης παραμόρφωσης ενός ομογενούς πορώδους ελαστοπλαστικού υλικού.

### 4.1 Όλκιμη θραύση εφελκυστικού τύπου

Στη συγκεκριμένη εφαρμογή χρησιμοποιούμε το πρόγραμμα πεπερασμένων στοιχείων ABAQUS. Ο κάνναβος που χρησιμοποιήθηκε φαίνεται στο Σχήμα 4.1 ενώ στο Σχήμα 4.2 παρουσιάζεται μία μεγέθυνση αυτού κοντά στην ακμή της ρωγμής. Μπροστά στην ακμή της ρωγμής αναπτύσσεται πλαστικότητα μικρής κλίμακας και το πεδίο των τάσεων και παραμορφώσεων είναι πολύ δύσκολο να προσδιορισθεί αναλυτικά. Συνεπώς, χρησιμοποιούμε τη μέθοδο των πεπερασμένων στοιχείων σε συνδυασμό με αριθμητικές μεθόδους ολοκλήρωσης οι οποίες όμως κοντά στην ακμή της ρωγμής απαιτούν μεγάλη ακρίβεια και μεγάλο αριθμό κόμβων και στοιχείων.



Σχήμα 4. 1



Σχήμα 4. 2

#### Συνοριακές συνθήκες

Στην άνω επιφάνεια του σχήματος 4.1 εφαρμόζουμε βηματικά την ελαστική λύση του προβλήματος mode-I της ρωγμής. Δηλαδή οι περιφερειακοί κόμβοι μετατοπίζονται σύμφωνα με την παρακάτω έκφραση:

$$\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \frac{K_I}{2\mu} \sqrt{\frac{r}{2\pi}} (3 - 4\nu - \cos\theta) \begin{Bmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2} \end{Bmatrix} \quad (4.1.1)$$

όπου  $u_i$  είναι οι συνιστώσες της μετατόπισης κάθε περιφερειακού κόμβου και  $(r, \theta)$  οι πολικές συντεταγμένες των κόμβων αυτών. Το  $K_I$  ονομάζεται συντελεστής εντάσεως των τάσεων (stress intensity factor) και αυξάνεται σταδιακά.

Στην κάτω δεξιά επιφάνεια έχουμε οριακές συνθήκες συμμετρίας σε σχέση με το άλλο μισό κομμάτι ενώ η κάτω αριστερά επιφάνεια αποτελεί την ελεύθερη επιφάνεια της ρωγμής και δεν εφαρμόζονται οριακές συνθήκες σε αυτή.

#### Υλικό

Ο φορέας είναι ένα ελαστοπλαστικό υλικό με μέτρο ελαστικότητας  $E$ , λόγο Poisson  $\nu = 0.3$  και ακολουθεί τον παρακάτω νόμο σκλήρυνσης :

$$\sigma_y(\bar{\epsilon}^p) = \sigma_0 \left( 1 + \frac{\bar{\epsilon}^p}{\epsilon_0} \right)^{1/n} \quad (4.1.2)$$

όπου  $\sigma_0$  είναι το όριο διαρροής του φορέα,  $\epsilon_0 = \sigma_0 / E$  και  $n$  είναι ο εκθέτης σκληρύνσεως. Στους υπολογισμούς χρησιμοποιούνται οι τιμές  $E / \sigma_0 = 300$  και  $n = 10$  ενώ θεωρούμε αρχικό πορώδες  $f_0 = 0.04$ .

Συγκεκριμένα, η όλκιμη θραύση των μετάλλων συμβαίνει εξαιτίας της πυρήνωσης και συνένωσης των μικροσπών. Η εξίσωση εξελίξεως του πορώδους παίρνει τη μορφή :

$$\dot{f} = \dot{f}_{growth} + \dot{f}_{nucleation} \quad (4.1.3)$$

όπου  $\dot{f}_{growth}$  και  $\dot{f}_{nucleation}$  δίνονται από τη σχέση:

$$\dot{f}_{growth} = (1 - f) D_{kk}^p \quad (4.1.4)$$

$$\dot{f}_{nucleation} = \mathcal{A}(\bar{\epsilon}^p) \dot{\bar{\epsilon}}^p \quad \mu\epsilon \quad \mathcal{A}(\bar{\epsilon}^p) = \frac{f_N}{s_N \sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{\epsilon_m^p - \epsilon_N}{s_N} \right)^2 \right] \quad (4.1.5)$$

όπου η παράμετρος  $\mathcal{A}$  που εκφράζει τον όρο της πυρήνωσης, έχει επιλεγθεί έτσι ώστε η παραμόρφωση πυρήνωσης να ακολουθεί κανονική κατανομή με μέση τιμή  $\epsilon_N$  και τυπική απόκλιση  $s_N$  (Chu και Needleman [5]). Το  $f_N$  αποτελεί το κλάσμα του όγκου των εγκλεισμάτων στο συνεχές μέσο που πυρηνώνονται.

Οι πόροι πυρηνώνονται λόγω της θραύσης που συμβαίνει σε κάποια σημεία του υλικού και μεγαλώνουν εξαιτίας των πλαστικών παραμορφώσεων του υλικού που τους περιβάλλει. Συχνά υπάρχουν δύο οικογένειες μικροσπών που παίζουν σημαντικό ρόλο στη διαδικασία της θραύσης:

α) οι μεγαλύτερες μικροσπές οι οποίες πυρηνώνονται σε σχετικά μικρές παραμορφώσεις και

β) οι μικρότερης κλίμακας μικροσπές που πυρηνώνονται σε αρκετά μεγάλες παραμορφώσεις.

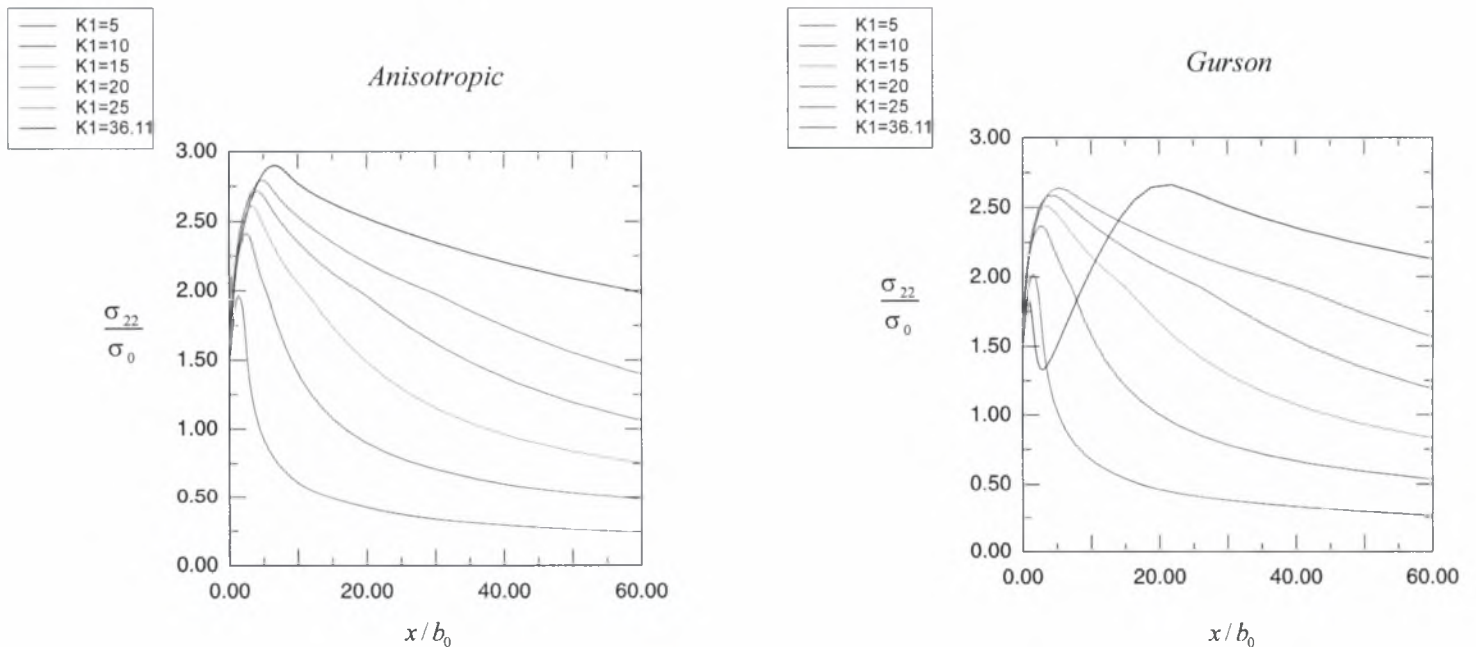
Η θραύση συμβαίνει όταν οι μεγαλύτερες μικροσπές ενώνονται μεταξύ τους ή με την ακμή της ρωγμής.

## Πεπερασμένα στοιχεία

Χρησιμοποιούνται τετρακομβικά στοιχεία επίπεδης παραμόρφωσης με συνολικά  $2 \times 2$  σταθμούς ολοκλήρωσης Gauss. Συνολικά χρησιμοποιούνται 1658 στοιχεία και 1810 κόμβοι. Η εξωτερική ακτίνα του καννάβου είναι  $R = 1.2 \times 10^3 b_0$  όπου  $b_0$  είναι η αρχική ακτίνα της ημικυκλικής ακμής της ρωγμής. Το υλικό θεωρείται αρχικά ισότροπο με  $w_1 = w_2 = 1$ .

## Αποτελέσματα

Παρακάτω παρουσιάζονται συγκριτικά διαγράμματα της εξέλιξης των τάσεων και των καταστατικών μεταβλητών του ανισότροπου μοντέλου και του μοντέλου του **Gurson**. Στα παρακάτω διαγράμματα χρησιμοποιούμε το αδιάστατο μέγεθος  $K_I = K_I / (\sigma_0 \sqrt{b_0})$ . Πρέπει να αναφερθεί ότι το μοντέλο του **Gurson** σε φορτίο  $K_I = 36.11$  έγινε μη ρεαλιστικό ( $f > 0.25$ ) με αποτέλεσμα ο κώδικας να μην μπορεί να συνεχίσει την ανάλυση πέρα από αυτό το σημείο.



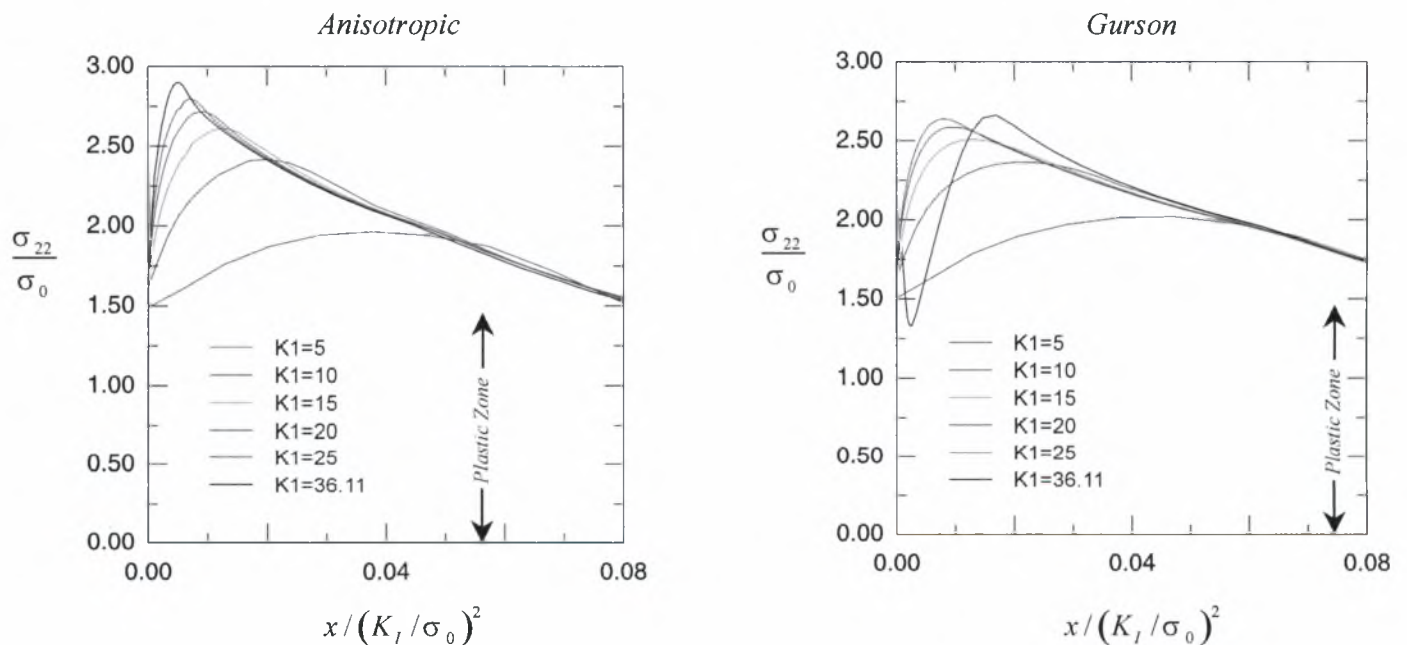
Σχήμα 4.3. Μεταβολή της αδιαστατοποιημένης τάσεως  $\sigma_{22}/\sigma_0$  μπροστά από τη ρωγμή.

Στο σχήμα 4.3 φαίνεται η κατανομή της ορθής τάσης  $\sigma_{22}$  μπροστά από τη ρωγμή. Στο σχήμα αυτό αλλά και σε όσα ακολουθούν,  $x$  είναι η απόσταση ενός υλικού σημείου στην απαραμόρφωτη κατάσταση από τη ρίζα της ημικυκλικής εγκοπής μπροστά από τη ρωγμή. Παρατηρούμε ότι τάσεις σε κάθε υλικό σημείο



μπροστά από τη ρωγμή αρχικά αυξάνονται καθώς αυξάνεται το  $K_I$ . Παρατηρούμε ότι η εφελκυστική τάση  $\sigma_{22}$  μπροστά από τη ρωγμή παίρνει τη μέγιστη τιμή της στο εσωτερικό του σώματος και όχι στη ρίζα της ημικυκλικής εγκοπής. Αυτό οφείλεται στο ότι η επιφάνεια της ημικυκλικής εγκοπής είναι αφόρτιστη, γεγονός που μειώνει τοπικά την υδροστατική τάση. Στην περίπτωση της «μαθηματικώς αιχμηρής» ρωγμής οι τάσεις απειρίζονται στην ακμή της ρωγμής και μειώνονται καθώς μετακινούμεθα προς το εσωτερικό του σώματος.

Η μεταβολή της  $\sigma_{22}$  μπροστά από τη ρωγμή φαίνεται επίσης στο σχήμα 4.4, όπου όμως η θέση του  $x$  των υλικών σημείων αδιαστατοποιείται με το «χαρακτηριστικό μήκος» του προβλήματος της «πλαστικότητας μικρής κλίμακας»  $(K_I/\sigma_0)^2$  (small scale yielding). Παρατηρούμε ότι, όπως προβλέπεται από τη θεωρία της πλαστικότητας μικρής κλίμακας, οι καμπύλες της τάσεως συμπίπτουν, εκτός από την περιοχή κοντά στην ακμή της ρωγμής όπου αναπτύσσονται πεπερασμένες παραμορφώσεις και οι τάσεις επηρεάζονται από την τοπική καμπυλότητα του συνόρου.



Σχήμα 4.4. Μεταβολή της αδιαστατοποιημένης τάσεως  $\sigma_{22}/\sigma_0$  μπροστά από τη ρωγμή.

Παρατηρούμε ότι στο μοντέλο του Gurson, και για τιμές του αδιάστατου φορτίου  $K_I$  μεγαλύτερες του 25, η  $\sigma_{22}$  παρουσιάζει ένα τοπικό ελάχιστο μπροστά από την ακμή της ρωγμής. Όπως θα εξηγηθεί σε λεπτομέρεια στη συνέχεια, αυτό

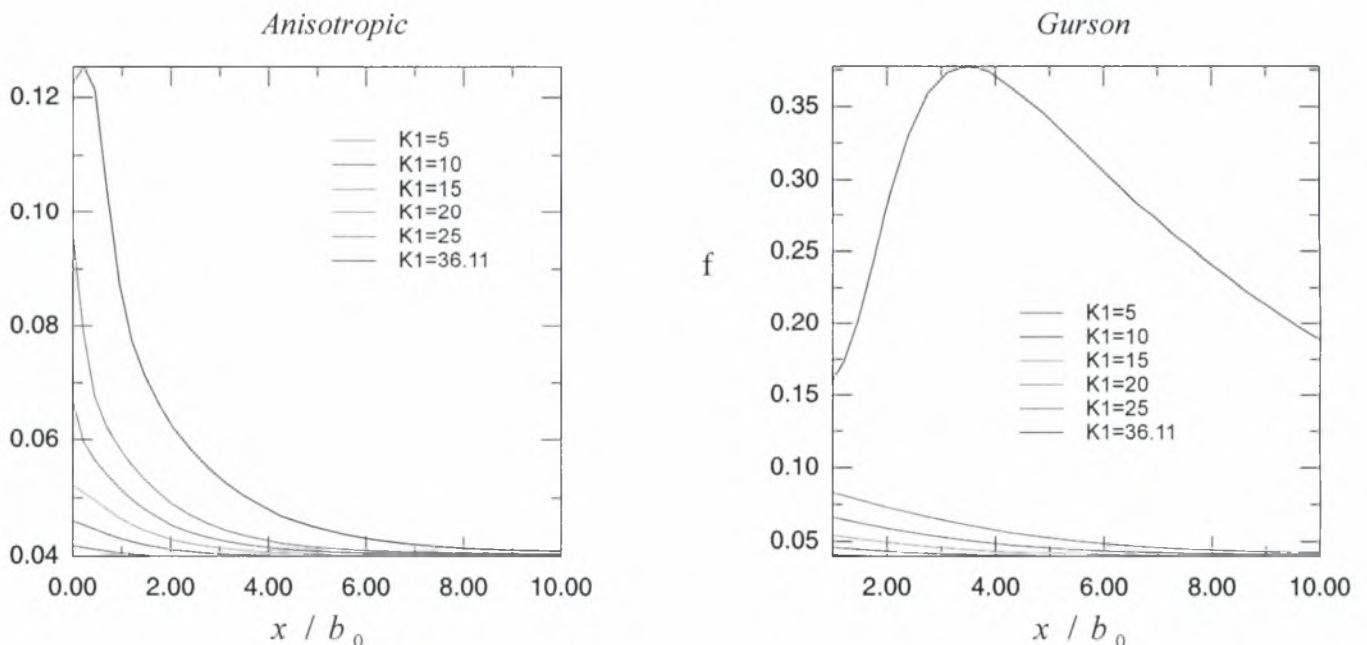
οφείλεται στο γεγονός ότι το πορώδες  $f$  αυξάνει υπερβολικά στη θέση αυτή δημιουργώντας τοπικά μία νησίδα «μειωμένης συνοχής».

Στο σχήμα 4.5 φαίνεται η μεταβολή του πορώδους  $f$  μπροστά από την ακμή της ρωγμής. Παρατηρούμε ότι για τιμές του αδιάστατου φορτίου  $K_1$  μεγαλύτερες του 25, το  $f$  παρουσιάζει ένα τοπικό μέγιστο σε κάποια απόσταση μπροστά από τη ρωγμή. Αυτό οφείλεται στο ότι η μέγιστη υδροστατική πίεση δεν εμφανίζεται στην ρίζα της ρωγμής αλλά στο εσωτερικό του σώματος (σχήμα 4.6) και στην ισχυρή (εκθετική) εξάρτηση του  $\dot{f}$  από την  $p$ . Συγκεκριμένα, η μεταβολή του  $f$  ορίζεται για το μοντέλο του Gurson από τη σχέση  $\dot{f} = (1-f)D_{kk}^p + \mathcal{A}(\bar{\epsilon}^p)\dot{\bar{\epsilon}}^p$  ή

$$\dot{f} = \frac{3\lambda}{\sigma_y} f(1-f) \sinh\left(\frac{3p}{2\sigma_y}\right) + \frac{f_N}{s_N \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\epsilon_m^p - \epsilon_N}{s_N}\right)^2\right] \dot{\bar{\epsilon}}^p.$$

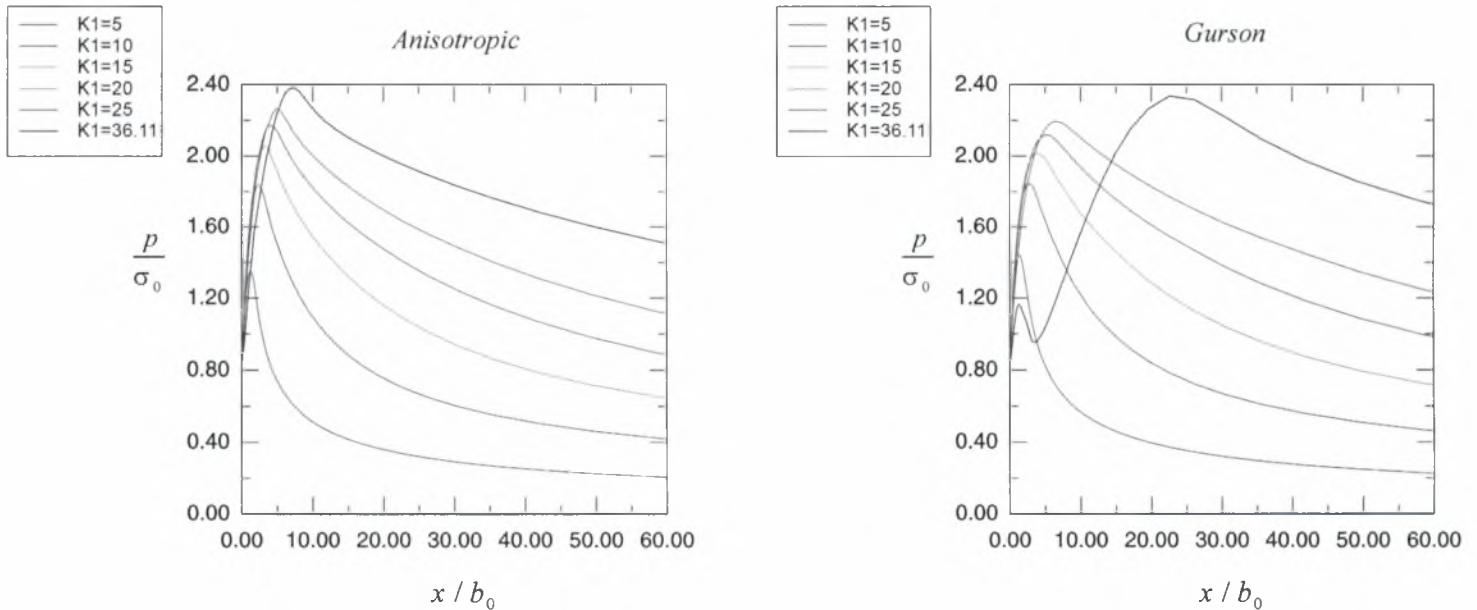
Το μοντέλο Gurson προβλέπει λοιπόν τη δημιουργία μίας μικρορωγμής μπροστά από την ακμή της μακροσκοπικής ρωγμής και όχι στη ρίζα της αρχικά ημικυκλικής εγκοπής. Η πρόβλεψη αυτή κρίνεται ως μη-ρεαλιστική. Αντιθέτως, στο ανισότροπο μοντέλο η εξάρτηση του  $\dot{f}$  από την  $\sigma$  είναι γραμμική και η μεταβολή του  $f$  ορίζεται από τη σχέση :

$$\dot{f} = 2\lambda m(s) : \sigma + \frac{f_N}{s_N \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\bar{\epsilon}^p - \epsilon_N}{s_N}\right)^2\right] \dot{\bar{\epsilon}}^p.$$



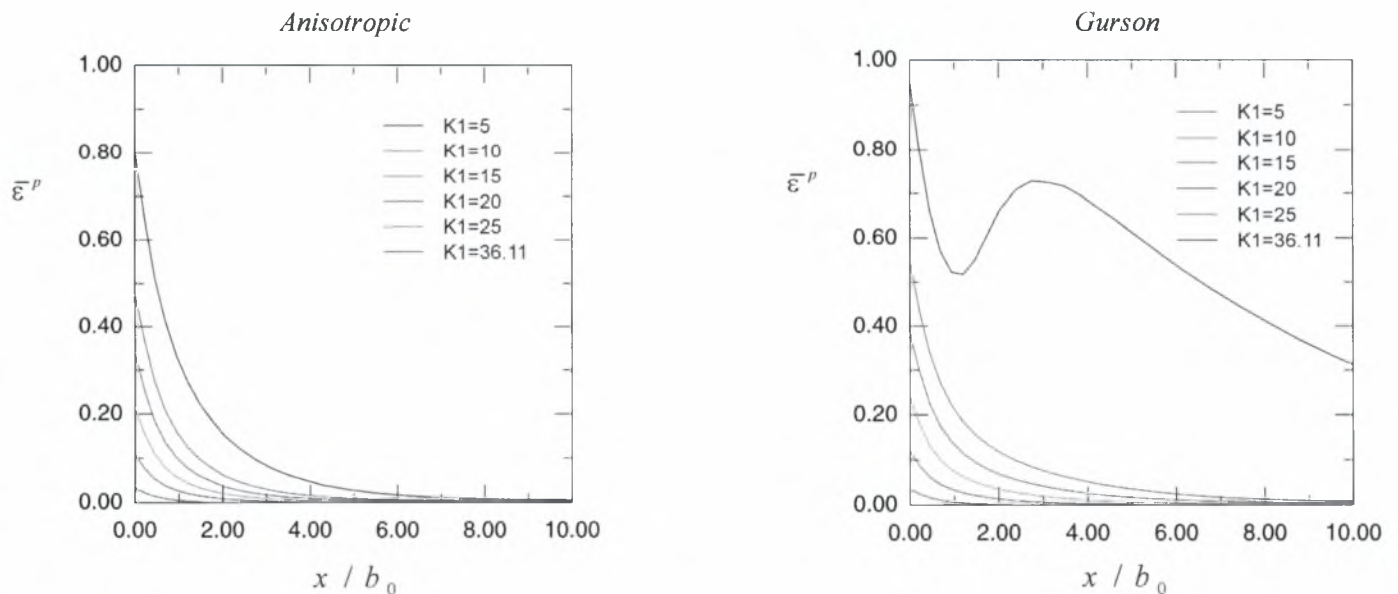
Σχήμα 4. 5. Μεταβολή του πορώδους  $f$  μπροστά από τη ρωγμή.

Αντιθέτως, το ανισότροπο μοντέλο λαμβάνει υπ' όψιν τη μεταβολή του σχήματος των μικροοπών και προβλέπει μία σημαντικά μικρότερη και ρεαλιστικότερη κατανομή των μικροοπών. Επίσης, η «φθορά» (damage) στο υλικό, δηλαδή το πορώδες, παίρνει μέγιστη τιμή στη ρίζα της εγκοπής και οδηγεί στη ρεαλιστική πρόβλεψη ότι η όποια επέκταση της ρωγμής αρχίζει στη ρίζα της εγκοπής.



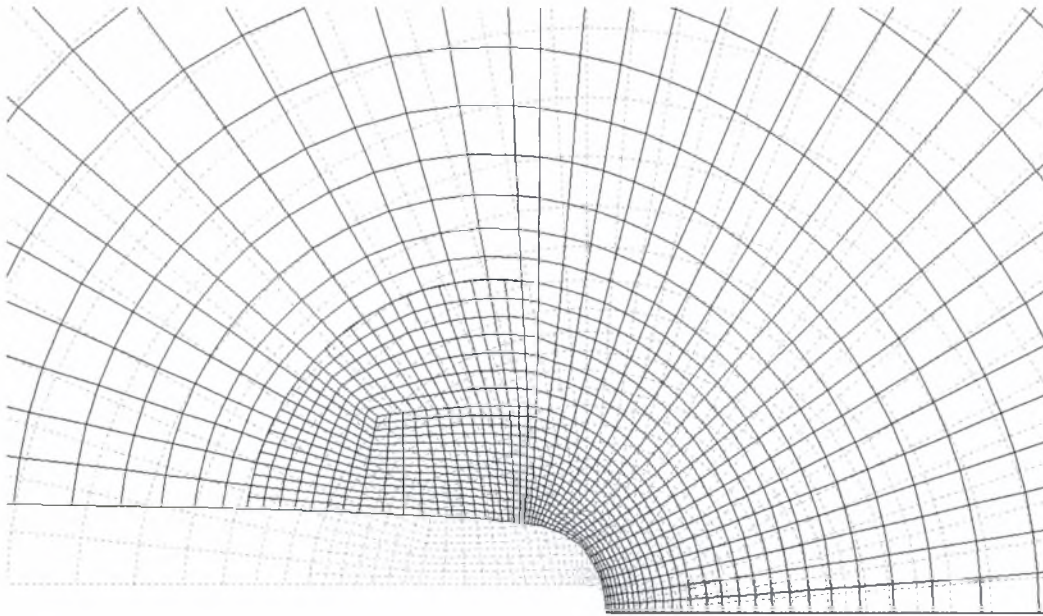
Σχήμα 4. 6. Μεταβολή της υδροστατικής πίεσης  $p = \sigma_{kk} / 3$  μπροστά από τη ρωγμή.

Η κατανομή της ισοδύναμης πλαστικής παραμόρφωσης στο φορέα  $\bar{\epsilon}^P$  φαίνεται στο σχήμα 4.7. Παρατηρούμε και πάλι ότι για τιμές του φορτίου  $K_1$  μεγαλύτερες του 25 η  $\bar{\epsilon}^P$  για το μοντέλο του Gurson παρουσιάζει ένα τοπικό μέγιστο στην υποτιθέμενη «μικρορωγμή» μπροστά από την ακμή της μακροσκοπικής ρωγμής.



Σχήμα 4. 7. Μεταβολή της ισοδύναμης πλαστικής παραμόρφωσης  $\bar{\epsilon}^P$  μπροστά από τη ρωγμή.

Στο σχήμα 4.8 φαίνεται ενδεικτικά η παραμορφωμένη γεωμετρία κοντά στην ακμή της ρωγμής για το μοντέλο του Gurson και για  $K_1 = 36.11$ .



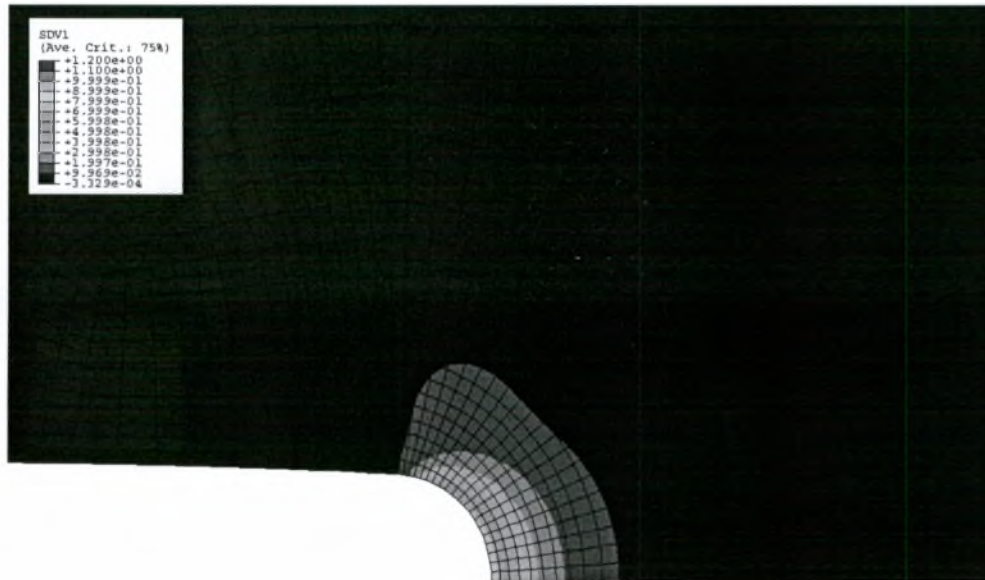
**Σχήμα 4. 8.** Παραμορφωμένα ( μαύρες γραμμές ) και απαραμόρφωτα ( πράσινες γραμμές ) στοιχεία στην περιοχή κοντά στην ακμή της ρωγμής.

Στα σχήματα 4.9 έως 4.12 φαίνονται οι ισοϋψείς για την ισοδύναμη πλαστική παραμόρφωση  $\bar{\epsilon}^p$  και το πορώδες  $f$  κοντά στην ακμή της ρωγμής για τα δύο μοντέλα. Όπως προαναφέρθηκε, το μοντέλο του Gurson προβλέπει τη δημιουργία μίας νησίδας μειωμένης συνοχής μέσα στο σώμα και μπροστά από την ακμή της ρωγμής.

Στα σχήματα 4.13 και 4.14 παρουσιάζεται η κατανομή των ποσοτήτων  $w_1$  και  $w_2$  κοντά στην ακμή της ρωγμής. Παρατηρούμε ότι οι αρχικά σφαιρικές μικροσπές μεταβάλλονται σε ελλειψοειδή με τον μέγιστο κύριο άξονα στην κατεύθυνση του μακροσκοπικού εφελκυστικού φορτίου.

Στα σχήματα 4.15 και 4.16 φαίνονται οι πλαστικές ζώνες που αντιστοιχούν στα δύο μοντέλα για  $K_1 = 36.11$ .

Τέλος, στο σχήμα 4.17 παρουσιάζεται ενδεικτικά η μεταβολή της πλαστικής ζώνης καθώς αυξάνεται το φορτίο. Υπενθυμίζουμε ότι το μέγεθος της πλαστικής ζώνης πρέπει να παραμένει μικρό σε σχέση με την εξωτερική ακτίνα  $R$  του καννάβου, ώστε να ισχύουν οι συνθήκες της «πλαστικότητας μικρής κλίμακας».



Σχήμα 4.9. Ισοδύναμη πλαστική παραμόρφωση  $\bar{\varepsilon}^P$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  για το ανισότροπο μοντέλο.



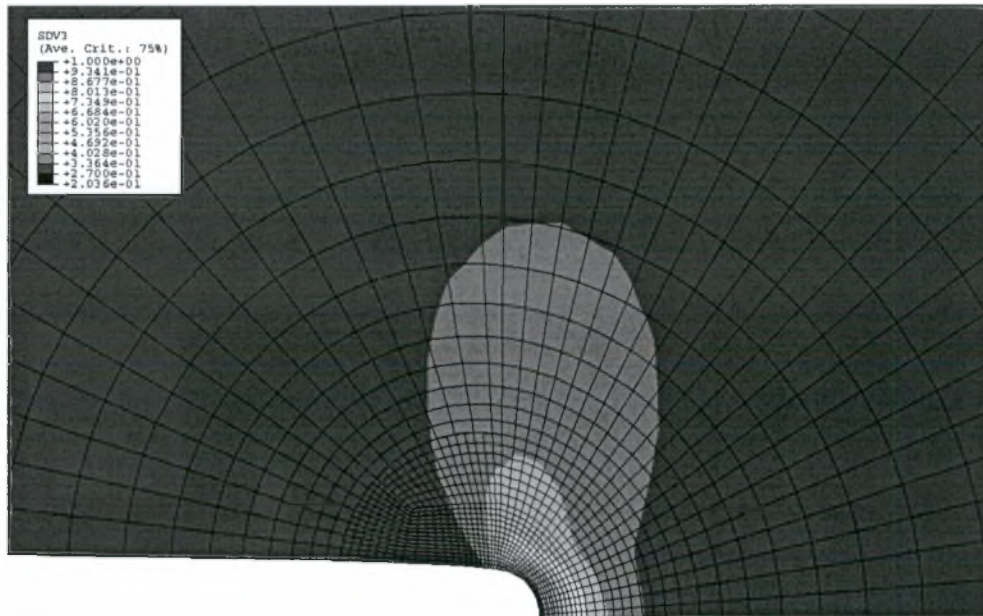
Σχήμα 4.10. Ισοδύναμη πλαστική παραμόρφωση  $\bar{\varepsilon}^P$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  για το μοντέλο του Gurson.



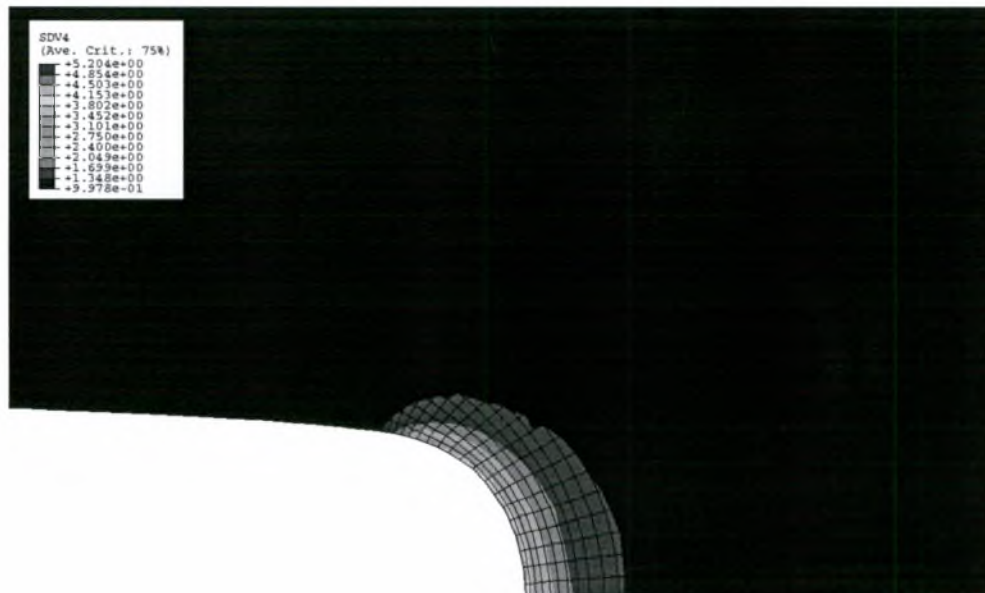
Σχήμα 4.11. Πορώδες  $f$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  για το ανισότροπο μοντέλο.



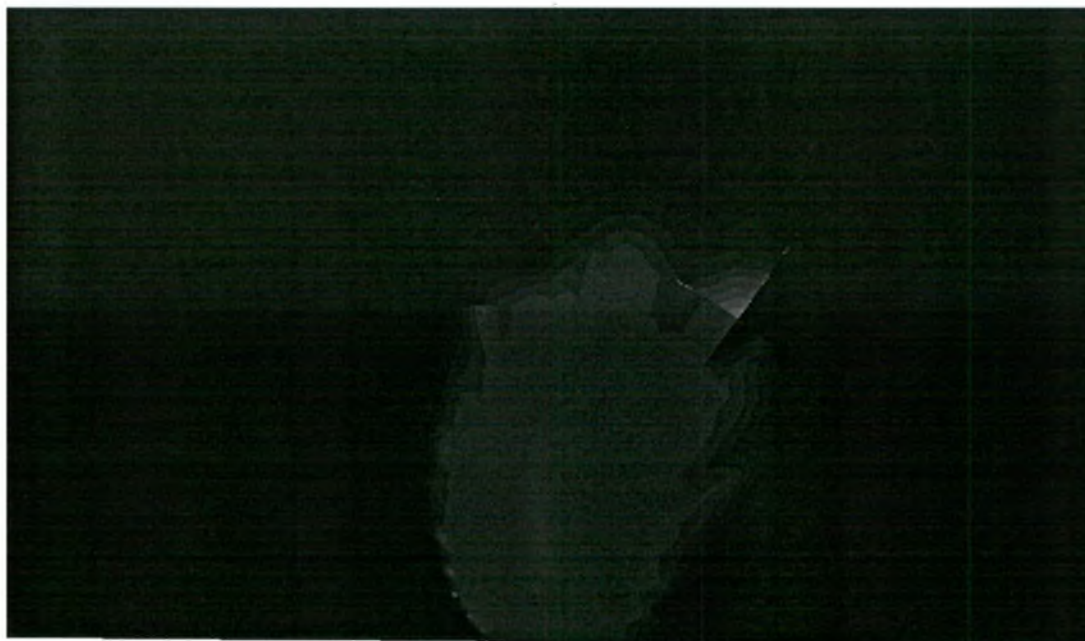
Σχήμα 4.12. Πορώδες  $f$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  για το μοντέλο του Gurson.



Σχήμα 4.13. Το  $w_1$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 60$  για το ανισότροπο μοντέλο.



Σχήμα 4.14. Το  $w_2$  για  $K_I / (\sigma_0 \sqrt{b_0}) = 60$  για το ανισότροπο μοντέλο.

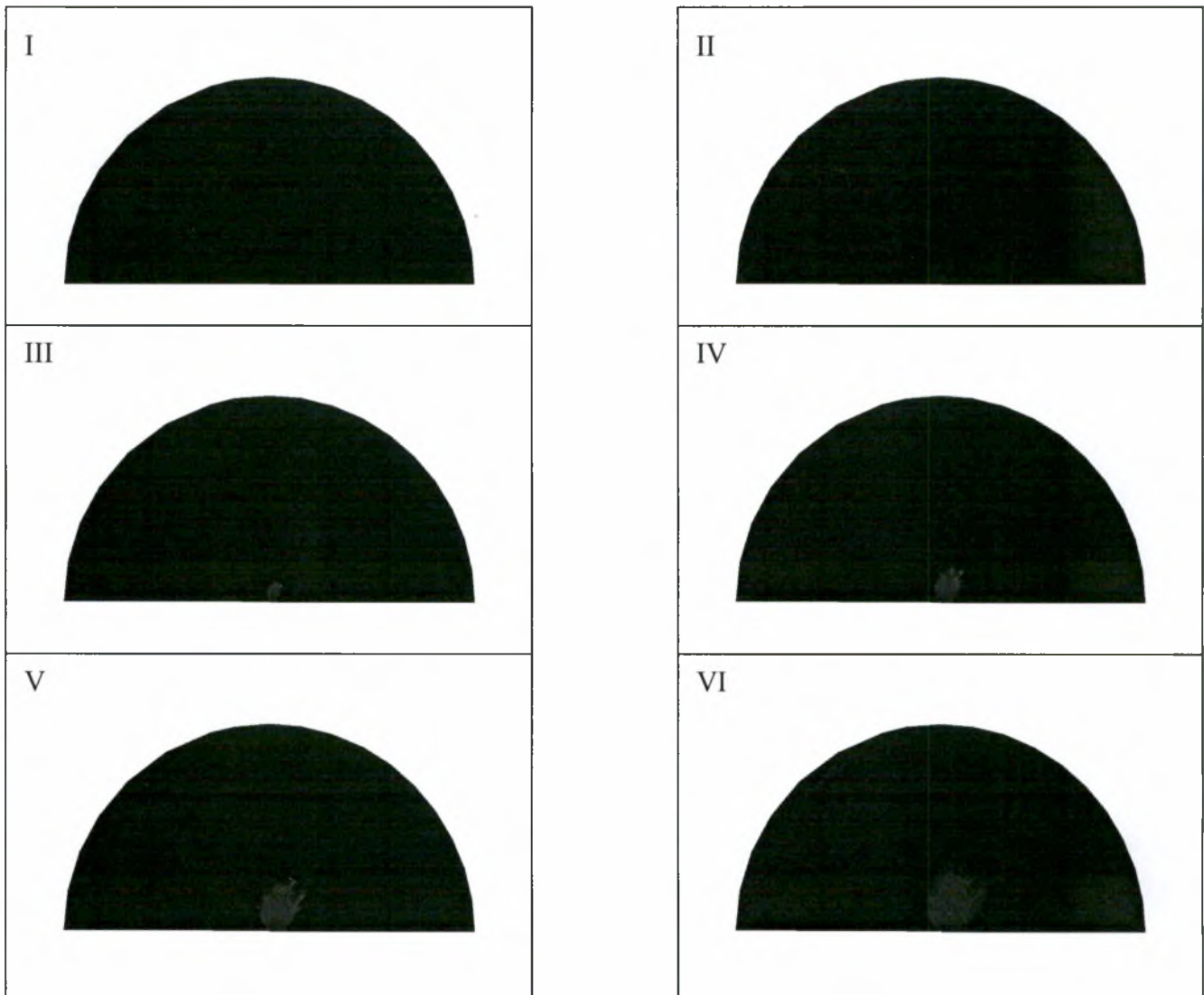


Σχήμα 4. 15. Πλαστική ζώνη για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  στο ανισότροπο μοντέλο.



Σχήμα 4. 16. Πλαστική ζώνη για  $K_I / (\sigma_0 \sqrt{b_0}) = 36.11$  στο μοντέλο του *Gurson*.

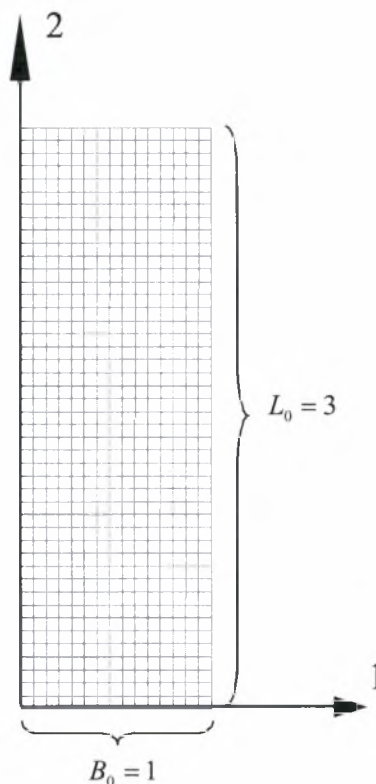




Σχήμα 4. 17. Εξέλιξη της πλαστικής ζώνης έως  $K_I / (\sigma_0 \sqrt{b_0}) = 60$  για το ανισότροπο μοντέλο.

## 4.2 Μονοαξονικός εφελκυσμός

Αναλύουμε το πρόβλημα του μονοαξονικού εφελκυσμού σε συνθήκες επίπεδης παραμόρφωσης. Θεωρούμε ένα ορθογώνιο κάρναβο με λόγο πλευρών  $L_0/B_0 = 3$  όπου  $2L_0$  είναι το μήκος του τεμαχίου και  $2B_0$  είναι το πλάτος του. Εισάγουμε το καρτεσιανό σύστημα συντεταγμένων που φαίνεται στο σχήμα 4.18 και κάθε υλικό σημείο του τεμαχίου καθορίζεται από το διάνυσμα θέσης του,  $\mathbf{X} = (X_1, X_2)$ , στην απαραμόρφωτη κατάσταση. Ενδιαφερόμαστε για συμμετρικές λύσεις και θεωρούμε το ένα τέταρτο του συνολικού τεμαχίου που φαίνεται στο σχήμα 4.18.



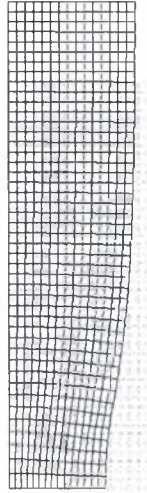
Σχήμα 4. 18

### Συνοριακές συνθήκες

Στην πάνω πλευρά εφαρμόζουμε οριακές συνθήκες μετατόπισης  $\hat{u}$  στην  $X_2$  κατεύθυνση. Πιο συγκεκριμένα, ενώνουμε τους κόμβους της πάνω πλευράς με ένα ανεξάρτητο κόμβο (fictitious node) στον οποίο δίνουμε μία μετατόπιση  $u_2^{fc}$ . Τέλος, εισάγουμε μία συνημιτονοειδή γεωμετρική ατέλεια στην αφόρτιστη παρειά και μελετούμε τη δημιουργία λαιμού. Χρησιμοποιούμε όπως και στη ρωγή μέθοδο

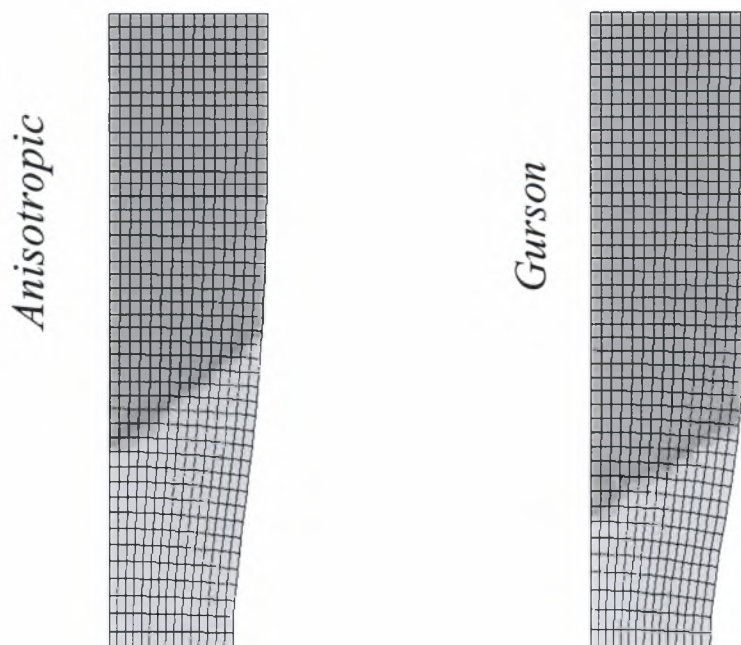
πεπερασμένων στοιχείων, με τετρακομβικά ισοπαραμετρικά στοιχεία με  $2 \times 2$  σταθμούς ολοκλήρωσης Gauss.

Το σχήμα 4.19 δείχνει την παραμορφωμένη κατάσταση του σώματος για μετατόπιση του ανεξάρτητου κόμβου  $u_2^{fc} = 0.60B_0$ , σε σχέση με την απαραμόρφωτη. Παρατηρείτε η δημιουργία λαιμού γεγονός που προκύπτει λόγω της αρχικής ατέλειας που υπάρχει στην δεξιά επιφάνεια του δοκιμίου.



**Σχήμα 4. 19.** Παραμορφωμένα (μαύρες γραμμές) και απαραμόρφωτα (πράσινες γραμμές) στοιχεία.

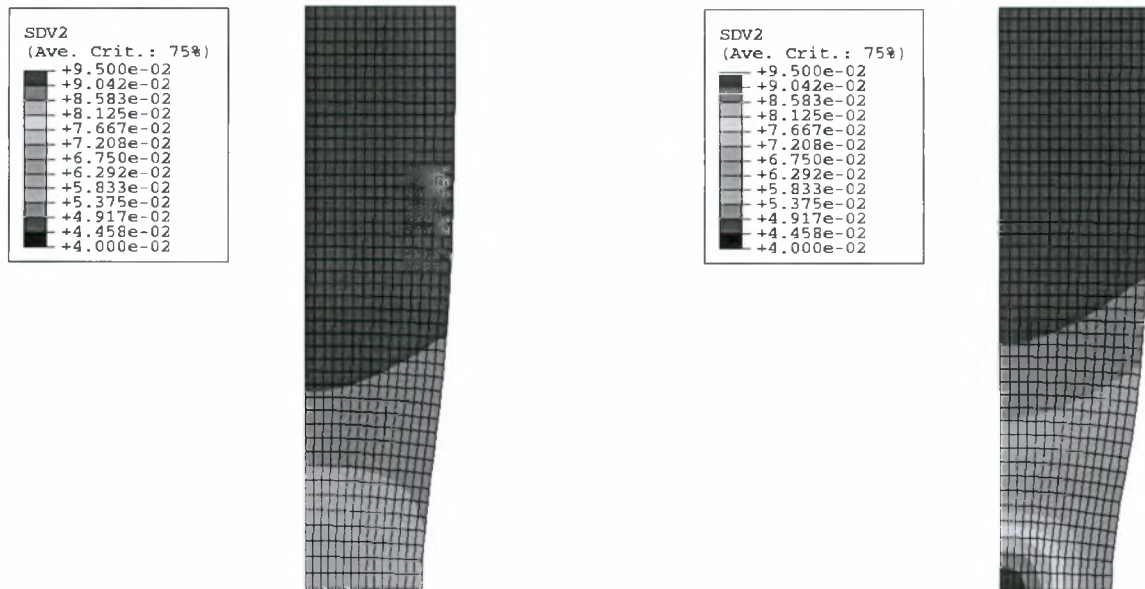
Τα επόμενα δύο σχήματα δείχνουν συγκριτικά γραφήματα του ανισότροπου μοντέλου και του μοντέλου του Gurson σε μετατόπιση του ανεξάρτητου κόμβου  $u_2^{fc} = 0.60B_0$ .



**Σχήμα 4. 20.** Πλαστική ζώνη στα δύο μοντέλα και σχηματισμός λαιμού.

*Anisotropic*

*Gurson*

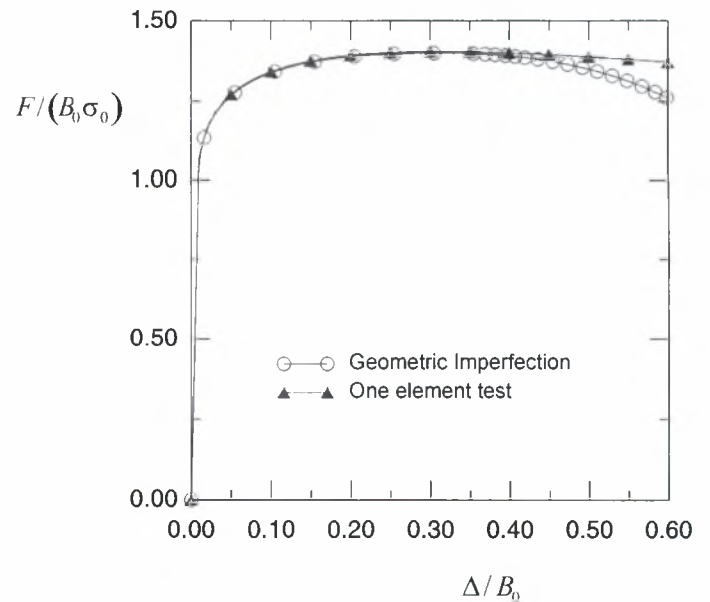
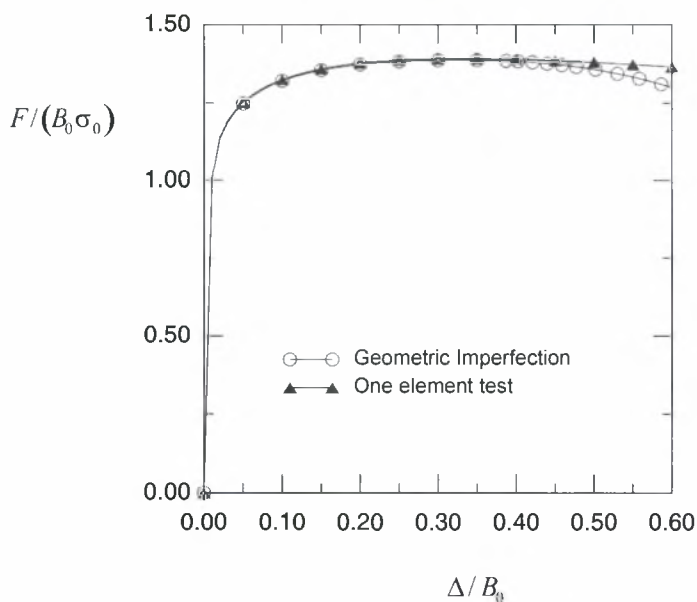


Σχήμα 4. 21. Πορώδες  $f$  στα δύο μοντέλα.

Παρατηρούμε ότι και στα δύο μοντέλα παρατηρείται σημαντική αύξηση του πορώδους στην κάτω αριστερή περιοχή του δοκιμίου. Επειδή όμως το παραπάνω πλέγμα είναι το ένα τέταρτο του συνολικού δοκιμίου η προαναφερθείσα περιοχή αποτελεί το κέντρο του, από την οποία και ξεκινά ο μηχανισμός της θραύσης. Επίσης παρατηρούμε ότι στο ανισότροπο μοντέλο το πορώδες είναι μικρότερο από αυτό στο μοντέλο του Gurson.

*Anisotropic*

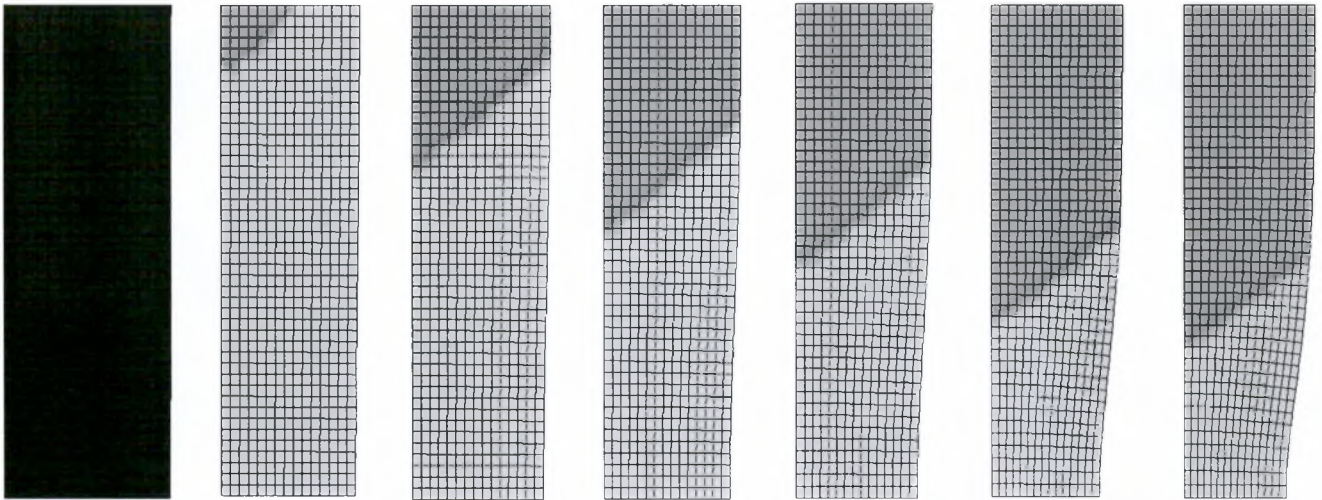
*Gurson*



Σχήμα 4. 22. Γραφική απεικόνιση της καμπύλης δύναμης – μετατόπισης στα δύο μοντέλα.

Τέλος στο σχήμα 4.22 βλέπουμε το σημείο που αρχίζει να σχηματίζεται λαιμός στα δύο μοντέλα και είναι το σημείο που αποχωρίζεται η κόκκινη καμπύλη από την πράσινη. Η πράσινη καμπύλη αφορά ένα υλικό τέλει στη γεωμετρία του.

$$u_2^{fic} = 0 \rightarrow 0.60B_0$$



**Σχήμα 4. 23.** Εξέλιξη της πλαστικής ζώνης (κίτρινο χρώμα) κατά τη διάρκεια της φόρτισης στο ανισότροπο μοντέλο.

## **Παράρτημα Α**

*Αρχείο δημιουργίας καννάβου για το πρόβλημα της ρωγμής*

```

*HEADING
MODE I CRACK
*PREPRINT,ECHO=YES,MODEL=NO
*RESTART,WRITE,FREQ=10
** Job name: Job-1 Model name: Model-1
**
**
**
**
*NODE,INPUT=nodes
**
**ELEMENTS
**
**
*ELEMENT,TYPE=CPE4,ELSET=A,INPUT=elements
**
**MPC
**
*MPC
LINEAR, 31, 3, 30
LINEAR, 29, 30, 28
LINEAR, 27, 28, 26
LINEAR, 25, 26, 24
LINEAR, 23, 24, 22
LINEAR, 21, 22, 20
LINEAR, 19, 20, 2
LINEAR, 84, 2, 83
LINEAR, 82, 83, 81
LINEAR, 80, 81, 79
LINEAR, 78, 79, 77
LINEAR, 76, 77, 75
LINEAR, 74, 75, 73
LINEAR, 72, 73, 71
*NODE
10000, 0., 0.
1000, 10.834, 0.
1001, 11.8348, 0.
1002, 13.0358, 0.
1003, 14.4768, 0.
1004, 16.2058, 0.
1005, 18.2808, 0.
1006, 20.7708, 0.
1007, 23.7588, 0.
1008, 27.3448, 0.
1009, 31.6478, 0.
1010, 36.8118, 0.
1011, 43.0088, 0.
1012, 50.4448, 0.
1013, 59.3678, 0.
1014, 70.0748, 0.
1015, 82.9228, 0.
1016, 98.3404, 0.
1017, 116.8415, 0.
1018, 139.0428, 0.
1019, 165.6844, 0.
1400, 0., 10.834
1401, 0., 11.8348
1402, 0., 13.0358
1403, 0., 14.4768
1404, 0., 16.2058
1405, 0., 18.2808

```



1406, 0., 20.7708  
1407, 0., 23.7588  
1408, 0., 27.3448  
1409, 0., 31.6478  
1410, 0., 36.8118  
1411, 0., 43.0088  
1412, 0., 50.4448  
1413, 0., 59.3678  
1414, 0., 70.0748  
1415, 0., 82.9228  
1416, 0., 98.3404  
1417, 0., 116.8415  
1418, 0., 139.0428  
1419, 0., 165.6844  
1450, -0.636943860, 10.81525718  
1451, -0.695782093, 11.81432580  
1452, -0.766390325, 13.01324807  
1453, -0.851108444, 14.45175514  
1454, -0.952758429, 16.17776397  
1455, -1.074750169, 18.24917422  
1456, -1.221140257, 20.73486652  
1457, -1.396808363, 23.71769728  
1458, -1.607633606, 27.29749350  
1459, -1.860612139, 31.59304931  
1460, -2.164209896, 36.74811559  
1461, -2.528538963, 42.93439478  
1462, -2.965710326, 50.35753050  
1463, -3.490304203, 59.26509371  
1464, -4.119781582, 69.95357060  
1465, -4.875130919, 82.77934356  
1466, -5.781550124, 98.17027111  
1467, -6.869251995, 116.6393642  
1468, -8.174493063, 138.8022560  
1469, -9.740784697, 165.3977660  
1520, -36.10362122, 202.4527064  
1521, -46.62739805, 261.4652661  
1522, -59.25592673, 332.2803180  
1523, -78.19871098, 438.5028466  
1524, -106.6128874, 597.8366396  
1525, -149.2341519, 836.8373290  
1526, -213.1660488, 1195.338363  
1800, -8.366567334, 6.883024378  
1801, -9.139436135, 7.518849632  
1802, -10.06690959, 8.281865349  
1803, -11.17972328, 9.197357146  
1804, -12.51494526, 10.29582024  
1805, -14.11736608, 11.61410301  
1806, -16.04027107, 13.19604234  
1807, -18.34775706, 15.09436954  
1808, -21.11704914, 17.37261630  
1809, -24.4400452, 20.10638535  
1810, -28.42794936, 23.38716234  
1811, -33.21358881, 27.32422179  
1812, -38.95604724, 32.04843900  
1813, -45.84684292, 37.71737259  
1814, -54.11533437, 44.51971171  
1815, -64.03721522, 52.68226453  
1816, -75.94347224, 62.47732791  
1817, -90.23096522, 74.23139126  
1818, -107.3759413, 88.33625457  
1819, -127.9499436, 105.2621160



1820, -158.8109469, 130.6509080  
1821, -205.1024520, 168.7340962  
1822, -260.6522426, 214.4339092  
1823, -343.9768899, 282.9835970  
1824, -468.9638608, 385.8081287  
1825, -656.4443173, 540.0449262  
1826, -937.6650019, 771.4001225  
2150, -10.834, 1.  
2151, -11.8348, 1.  
2152, -13.0358, 1.  
2153, -14.4768, 1.  
2154, -16.2058, 1.  
2155, -18.2808, 1.  
2156, -20.7708, 1.  
2157, -23.7588, 1.  
2158, -27.3448, 1.  
2159, -31.6478, 1.  
2160, -36.8118, 1.  
2161, -43.0088, 1.  
2162, -50.4448, 1.  
2163, -59.3678, 1.  
2164, -70.0748, 1.  
2165, -82.9228, 1.  
2166, -98.3404, 1.  
2167, -116.8415, 1.  
2168, -139.0428, 1.  
2169, -165.6844, 1.  
2170, -205.6468, 1.  
2171, -265.5904, 1.  
2172, -337.5227, 1.  
2173, -445.4211, 1.  
2174, -607.2687, 1.  
2175, -850.0401, 1.  
2176, -1214.1972, 1.  
3020, 205.6468, 0.  
3021, 265.5904, 0.  
3022, 337.5227, 0.  
3023, 445.4211, 0.  
3024, 607.2687, 0.  
3025, 850.0401, 0.  
3026, 1214.1972, 0.  
3420, 0., 205.6468  
3421, 0., 265.5904  
3422, 0., 337.5227  
3423, 0., 445.4211  
3424, 0., 607.2687  
3425, 0., 850.0401  
3426, 0., 1214.1972  
\*NGEN, LINE=C  
1000, 1400, 20, 100000  
1001, 1401, 20, 100000  
1002, 1402, 20, 100000  
1003, 1403, 20, 100000  
1004, 1404, 20, 100000  
1005, 1405, 20, 100000  
1006, 1406, 20, 100000  
1007, 1407, 20, 100000  
1008, 1408, 20, 100000  
1009, 1409, 20, 100000  
1010, 1410, 20, 100000  
1011, 1411, 20, 100000

1012, 1412, 20, 100000  
1013, 1413, 20, 100000  
1014, 1414, 20, 100000  
1015, 1415, 20, 100000  
1016, 1416, 20, 100000  
1017, 1417, 20, 100000  
1018, 1418, 20, 100000  
1019, 1419, 20, 100000  
3020, 3420, 40, 100000  
3021, 3421, 40, 100000  
3022, 3422, 40, 100000  
3023, 3423, 40, 100000  
3024, 3424, 40, 100000  
3025, 3425, 40, 100000  
3026, 3426, 40, 100000  
2150, 1800, -50, 100000  
2151, 1801, -50, 100000  
2152, 1802, -50, 100000  
2153, 1803, -50, 100000  
2154, 1804, -50, 100000  
2155, 1805, -50, 100000  
2156, 1806, -50, 100000  
2157, 1807, -50, 100000  
2158, 1808, -50, 100000  
2159, 1809, -50, 100000  
2160, 1810, -50, 100000  
2161, 1811, -50, 100000  
2162, 1812, -50, 100000  
2163, 1813, -50, 100000  
2164, 1814, -50, 100000  
2165, 1815, -50, 100000  
2166, 1816, -50, 100000  
2167, 1817, -50, 100000  
2168, 1818, -50, 100000  
2169, 1819, -50, 100000  
2170, 1820, -50, 100000  
2171, 1821, -50, 100000  
2172, 1822, -50, 100000  
2173, 1823, -50, 100000  
2174, 1824, -50, 100000  
2175, 1825, -50, 100000  
2176, 1826, -50, 100000  
1800, 1450, -50, 100000  
1801, 1451, -50, 100000  
1802, 1452, -50, 100000  
1803, 1453, -50, 100000  
1804, 1454, -50, 100000  
1805, 1455, -50, 100000  
1806, 1456, -50, 100000  
1807, 1457, -50, 100000  
1808, 1458, -50, 100000  
1809, 1459, -50, 100000  
1810, 1460, -50, 100000  
1811, 1461, -50, 100000  
1812, 1462, -50, 100000  
1813, 1463, -50, 100000  
1814, 1464, -50, 100000  
1815, 1465, -50, 100000  
1816, 1466, -50, 100000  
1817, 1467, -50, 100000  
1818, 1468, -50, 100000

```

1819, 1469, -50, 100000
1820, 1520, -50, 100000
1821, 1521, -50, 100000
1822, 1522, -50, 100000
1823, 1523, -50, 100000
1824, 1524, -50, 100000
1825, 1525, -50, 100000
1826, 1526, -50, 100000
**
**MPC
**
*MPC
LINEAR, 2119, 2169, 2069
LINEAR, 2019, 2069, 1969
LINEAR, 1919, 1969, 1869
LINEAR, 1819, 1869, 1769
LINEAR, 1719, 1769, 1669
LINEAR, 1619, 1669, 1569
LINEAR, 1039, 1059, 1019
LINEAR, 1079, 1099, 1059
LINEAR, 1119, 1139, 1099
LINEAR, 1159, 1179, 1139
LINEAR, 1199, 1219, 1179
LINEAR, 1239, 1259, 1219
LINEAR, 1279, 1299, 1259
LINEAR, 1319, 1339, 1299
LINEAR, 1359, 1379, 1339
LINEAR, 1399, 1419, 1379
LINEAR, 1468, 1518, 1418
**
**ELEMENT
**
*ELEMENT,TYPE=CPE4,ELSET=B1
1000, 1000, 1001, 1021, 1020
2000, 2151, 2150, 2100, 2101
2007, 1800, 1750, 1751, 1801
3000, 2150, 3, 30, 2100
3001, 2100, 30, 28, 2050
3002, 2050, 28, 26, 2000
3003, 2000, 26, 24, 1950
3004, 1950, 24, 22, 1900
3005, 1900, 22, 20, 1850
3006, 1850, 20, 2, 1800
3007, 1800, 2, 83, 1750
3008, 1750, 83, 81, 1700
3009, 1700, 81, 79, 1650
3010, 1650, 79, 77, 1600
3011, 1600, 77, 75, 1550
3012, 1550, 75, 73, 1500
3013, 1500, 73, 71, 1450
3014, 71, 6, 1400, 1450
3015, 6, 154, 1380, 1400
3016, 154, 153, 1360, 1380
3017, 153, 152, 1340, 1360
3018, 152, 151, 1320, 1340
3019, 151, 150, 1300, 1320
3020, 150, 149, 1280, 1300
3021, 149, 148, 1260, 1280
3022, 148, 147, 1240, 1260
3023, 147, 146, 1220, 1240
3024, 146, 145, 1200, 1220

```

3025, 145, 144, 1180, 1200  
3026, 144, 143, 1160, 1180  
3027, 143, 142, 1140, 1160  
3028, 142, 141, 1120, 1140  
3029, 141, 140, 1100, 1120  
3030, 140, 139, 1080, 1100  
3031, 139, 138, 1060, 1080  
3032, 138, 137, 1040, 1060  
3033, 137, 136, 1020, 1040  
3034, 136, 10, 1000, 1020  
3035, 1450, 1400, 1401, 1451  
4000, 2170, 2169, 2069, 2070  
4001, 2070, 2069, 1969, 1970  
4002, 1970, 1969, 1869, 1870  
4003, 1870, 1869, 1769, 1770  
4004, 1770, 1769, 1669, 1670  
4005, 1670, 1669, 1569, 1570  
4006, 2171, 2170, 2070, 2071  
4007, 2071, 2070, 1970, 1971  
4008, 1971, 1970, 1870, 1871  
4009, 1871, 1870, 1770, 1771  
4010, 1771, 1770, 1670, 1671  
4011, 1671, 1670, 1570, 1571  
4012, 2172, 2171, 2071, 2072  
4013, 2072, 2071, 1971, 1972  
4014, 1972, 1971, 1871, 1872  
4015, 1872, 1871, 1771, 1772  
4016, 1772, 1771, 1671, 1672  
4017, 1672, 1671, 1571, 1572  
4018, 2173, 2172, 2072, 2073  
4019, 2073, 2072, 1972, 1973  
4020, 1973, 1972, 1872, 1873  
4021, 1873, 1872, 1772, 1773  
4022, 1773, 1772, 1672, 1673  
4023, 1673, 1672, 1572, 1573  
4024, 2174, 2173, 2073, 2074  
4025, 2074, 2073, 1973, 1974  
4026, 1974, 1973, 1873, 1874  
4027, 1874, 1873, 1773, 1774  
4028, 1774, 1773, 1673, 1674  
4029, 1674, 1673, 1573, 1574  
4030, 2175, 2174, 2074, 2075  
4031, 2075, 2074, 1974, 1975  
4032, 1975, 1974, 1874, 1875  
4033, 1875, 1874, 1774, 1775  
4034, 1775, 1774, 1674, 1675  
4035, 1675, 1674, 1574, 1575  
4036, 2176, 2175, 2075, 2076  
4037, 2076, 2075, 1975, 1976  
4038, 1976, 1975, 1875, 1876  
4039, 1876, 1875, 1775, 1776  
4040, 1776, 1775, 1675, 1676  
4041, 1676, 1675, 1575, 1576  
4042, 1569, 1519, 1520, 1570  
4049, 1519, 1419, 3420, 1520  
4050, 1520, 3420, 3421, 1521  
4051, 1521, 3421, 3422, 1522  
4052, 1522, 3422, 3423, 1523  
4053, 1523, 3423, 3424, 1524  
4054, 1524, 3424, 3425, 1525  
4055, 1525, 3425, 3426, 1526

```

4090, 1419, 1379, 3380, 3420
4091, 1379, 1339, 3340, 3380
4092, 1339, 1299, 3300, 3340
4093, 1299, 1259, 3260, 3300
4094, 1259, 1219, 3220, 3260
4095, 1219, 1179, 3180, 3220
4096, 1179, 1139, 3140, 3180
4097, 1139, 1099, 3100, 3140
4098, 1099, 1059, 3060, 3100
4099, 1059, 1019, 3020, 3060
5000, 3020, 3021, 3061, 3060
*ELGEN,ELSET=C1
1000, 19, 1, 1, 20, 20, 20
2000, 7, -50, 1, 19, 1, 14
2007, 7, -50, 1, 19, 1, 14
3035, 18, 1, 1
4042, 7, 1, 1
5000, 6, 1, 1, 10, 40, 7
*ELEMENT, TYPE=CPE4,ELSET=D1
2265, 1518, 1418, 1419, 1519
*ELSET,ELSET=ALLE
A,B1,C1,D1
**
**
**NSETS
**
**
*NSET, NSET=A11,GENERATE
130, 135, 1
193, 204, 1
1000, 1019, 1
3020, 3025, 1
*NSET, NSET=XAXIS
9, 10, 11, 12
A11
*NSET, NSET=B21
1526
*NSET, NSET=B22, GENERATE
1576, 2176, 100
3066, 3426, 40
*NSET, NSET=BOUNDA
B21, B22
**
**
**
** FIXED BOUNDARY CONDITIONS
**
*BOUNDARY
XAXIS, 2, 2
*BOUNDARY
3026, 1, 2
**
**
**
** BOUNDARY CONDITIONS
**
**
*NODE,NSET=VIC
9999, 0., 0.
*BOUNDARY

```



```

*STEP,INC=1000,NLGEOM
*STATIC
0.2, 20. , , 1.
*CONTROLS,PARAMETERS=FIELD, FIELD=DISPLACEMENT
5.D-3
*CONTROLS,PARAMETERS=TIME INCREMENTATION
8, 10
**CONTROLS,PARAMETERS=LINE SEARCH
** 4
*BOUNDARY
9999, 1, 1, 20.
**
**
**
**
**
**OUTPUT DATABASE
**
**
**
**
*NODE PRINT,FREQ=1000
U1
*EL PRINT,FREQ=1000
S11
*EL FILE,POSITION=CENTROIDAL
COORD, S,SDV
*NODE FILE
COORD, U
**
**
*OUTPUT, FIELD, FREQUENCY=1
*ELEMENT OUTPUT,ELSET=ALLE
S,SDV
*NODE OUTPUT
U
*OUTPUT, HISTORY,FREQ=1
*NODE OUTPUT,NSET=FIC
RF1,U1
*END STEP

```

## **Παράρτημα Β**

*Αρχείο δημιουργίας καννάβου για το πρόβλημα του μονοαξονικού εφελκυσμού*



```

*HEADING
  PLANE STRAIN TENSION TEST --- FINITE STRAINS
*PREPRINT,ECHO=YES,MODEL=YES,HISTORY=NO
*RESTART,WRITE,FREQ=5
**
**
**
*NODE
  1, 0., 0.
  721, 0., 3.
  10000, 0., 0.
*NGEN,NSET=LEFT
  1, 721, 16
*NODE,NSET=RIGHT
** 15 X 45 elements
** 0.5% cosine-type imperfection on right free surface
  16, 0.9950000, 0.0000
  32, 0.9950030, 0.0667
  48, 0.9950122, 0.1333
  64, 0.9950274, 0.2000
  80, 0.9950487, 0.2667
  96, 0.9950760, 0.3333
  112, 0.9951093, 0.4000
  128, 0.9951485, 0.4667
  144, 0.9951937, 0.5333
  160, 0.9952447, 0.6000
  176, 0.9953015, 0.6667
  192, 0.9953641, 0.7333
  208, 0.9954323, 0.8000
  224, 0.9955060, 0.8667
  240, 0.9955853, 0.9333
  256, 0.9956699, 1.0000
  272, 0.9957598, 1.0667
  288, 0.9958548, 1.1333
  304, 0.9959549, 1.2000
  320, 0.9960599, 1.2667
  336, 0.9961698, 1.3333
  352, 0.9962843, 1.4000
  368, 0.9964033, 1.4667
  384, 0.9965267, 1.5333
  400, 0.9966543, 1.6000
  416, 0.9967861, 1.6667
  432, 0.9969217, 1.7333
  448, 0.9970611, 1.8000
  464, 0.9972040, 1.8667
  480, 0.9973504, 1.9333
  496, 0.9975000, 2.0000
  512, 0.9976526, 2.0667
  528, 0.9978081, 2.1333
  544, 0.9979663, 2.2000
  560, 0.9981270, 2.2667
  576, 0.9982899, 2.3333
  592, 0.9984549, 2.4000
  608, 0.9986218, 2.4667
  624, 0.9987904, 2.5333
  640, 0.9989604, 2.6000
  656, 0.9991318, 2.6667
  672, 0.9993041, 2.7333
  688, 0.9994774, 2.8000
  704, 0.9996512, 2.8667
  720, 0.9998255, 2.9333

```

```

736, 1.0000000, 3.0000
*NFILL,NSET=ALLN
LEFT, RIGHT, 15, 1
*NSET,NSET=BOT,GENERATE
1, 16, 1
*NSET,NSET=TOP,GENERATE
721, 736, 1
*NSET,NSET=FIC
10000
**
**
**
*ELEMENT,TYPE=CPE4,ELSET=CEN
1, 1,2,18,17
*ELGEN,ELSET=ALLE
1, 15,1,1, 45,16,15
*ELSET,ELSET=TOP,GENERATE
661, 675, 1
**
**
**
*SOLID SECTION,ELSET=ALLE,MATERIAL=ONE
*MATERIAL,NAME=ONE
*USER MATERIAL,CONSTANTS=7,UNSYMM
300., 0.3, 1., 10., 0.0, 0.4, 0.1
*DEPVAR
6
**
**
**
*BOUNDARY
BOT,2
LEFT,1
*EQUATION
2
TOP, 2 , 1., 10000, 2, -1.
**
**
**
*INITIAL CONDITIONS,TYPE=SOLUTION
ALLE, 0., 0.0400, 1., 1., 0., 0.
**
**
**
*****STEP*****
**
*STEP,INC=10000,NLGEOM,UNSYMM=YES,EXTRAPOLATION=NO
*STATIC
0.010, 0.600, , 0.010
*BOUNDARY
10000, 2, 2, 0.600
*CONTROLS,PARAMETERS=FIELD,FIELD=DISPLACEMENT
1.D-3
**
**
**
**
**
**
*EL PRINT,FREQ=100
SDV

```

```
**
**
*OUTPUT, FIELD, FREQUENCY=1
*ELEMENT OUTPUT, ELSET=ALLE
  S, E, SDV
*NODE OUTPUT
  U
*OUTPUT, HISTORY, FREQUENCY=1
*ELEMENT OUTPUT, ELSET=TOP
  S, E, SDV
*NODE OUTPUT, NSET=TOP
  U
*NODE OUTPUT, NSET=FIC
  RF2, U2
*END STEP
```

## **Παράρτημα Γ**

Η umat υπορουτίνα mahesh\_back.f

```

SUBROUTINE UMAT (STRESS, STATEV, DDSDE, SSE, SPD, SCD,
+RPL, DDSDDT, DRPLDE, DRPLDT,
+STRAN, DSTRAN, TIME, DTIME, TEMP, DTEMP, PREDEF, DPRED, CMNAME,
+NDI, NSHR, NTENS, NSTATV, PROPS, NPROPS, CODS, DROT, PNEWDT,
+CELENT, DFGRD0, DFGRD1, NOEL, NPT, LAYER, KSPT, KSTEP, KINC)
C
C*** 30 March 2003
C
C   Finite strain Ponte-Kailasam model for voided metals with
C   deformation induced anisotropy
C
C   If IUNLOAD.NE.0 elastic stiffness on first iteration of increment
C   If ITESTDE.NE.0 tests the size of the strain increment
C   If ICONSISTENT.NE.0 uses "consistent" tangent of forward-Euler
C
C   INCLUDE 'ABA_PARAM.INC'
c   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
CHARACTER*80 CMNAME
C
C   DIMENSION STRESS (NTENS), STATEV (NSTATV),
+DDSDE (NTENS, NTENS), DDSDDT (NTENS), DRPLDE (NTENS),
+STRAN (NTENS), DSTRAN (NTENS), TIME (2), PREDEF (1), DPRED (1),
+PROPS (NPROPS), COORDS (3), DROT (3, 3), DFGRD0 (3, 3), DFGRD1 (3, 3)
C
C   DIMENSION X (200), W (200),
+ S (4, 4), P (4, 4), Q (4, 4), QINV (4, 4),
+ CEL (4, 4), AMTLD (4, 4),
+ AMTLDT (4, 4), DQDW1 (4, 4), DQDW2 (4, 4),
+ AIMX (4, 4), AJMX (4, 4), AKMX (4, 4), DELTA (6),
+ AMX (4, 4), CMX (4, 4), AN (4), QMX (4, 4),
+ AUXM1 (4, 4), AUXM2 (4, 4), AUXM3 (4, 4),
+ AUXM4 (4, 4), AUXM5 (4, 4), CELT (4, 4), ANT (4),
+ DQDN1 (4, 4, 3), DQDN2 (4, 4, 3), DQDN3 (4, 4, 3)
C
C   DIMENSION AN1 (3), AN2 (3), AN3 (3),
+ AN1T (3), AN2T (3), AN3T (3), SEL (6), AUXM (3, 3), DF (3, 3),
+ R (3, 3), RT (3, 3), U (3, 3), V (3, 3), UMI1 (3, 3), UMI2 (3, 3), UMI3 (3, 3),
+ UMI4 (3, 3), DSEL (6), STRESST (6), OMEGA (3, 3), V1 (6),
+ V2 (6), V3 (6), DETENS (3, 3), DE (6), DEP (6), DROTT (3, 3), OMEGAT (3, 3),
+ ADOTN (6), DEVOID (6), DPHIDN1 (3), DPHIDN2 (3), DPHIDN3 (3),
+ STRSABA (6)
C
C   DIMENSION R1 (6), R2 (6), R31 (6), R32 (6), DUN (7), GV (7), DNDF (6), DNDW1 (6),
+ DNDW2 (6), DFDEP (6), DYDEP (6), DGDL (6)
DIMENSION DGDEP (4, 4), DJAC (5, 5), DINV (5, 5)
C
COMMON /KNICK /AJMX, AKMX
COMMON /KNICK2 /PI, DELTA, AIMX, X, W
C
C*** ARRAYS FROM THE ESHELBY SUBROUTINE
C   DIMENSION X (200), W (200),
C   + S (NTENS, NTENS), P (NTENS, NTENS), Q (NTENS, NTENS), QINV (NTENS, NTENS),
C   + CEL (NTENS, NTENS), AMTLD (NTENS, NTENS),
C   + AMTLDT (NTENS, NTENS), DQDW1 (NTENS, NTENS), DQDW2 (NTENS, NTENS),
C   + AIMX (NTENS, NTENS), AJMX (NTENS, NTENS), AKMX (NTENS, NTENS),
C   + AMX (NTENS, NTENS), CMX (NTENS, NTENS), AN (NTENS), QMX (NTENS, NTENS),
C   + AUXM1 (NTENS, NTENS), AUXM2 (NTENS, NTENS), AUXM3 (NTENS, NTENS),
C   + AUXM4 (NTENS, NTENS), AUXM5 (NTENS, NTENS), CELT (NTENS, NTENS), ANT (NTENS),
C   + DQDN1 (NTENS, NTENS, 3), DQDN2 (NTENS, NTENS, 3), DQDN3 (NTENS, NTENS, 3)
C
C*** STATE VARIABLES

```

```

C
C STATEV(1)=EBAR
C STATEV(2)=F
C STATEV(3)=W1
C STATEV(4)=W2
C STATEV(5)=THETA IN DEGREES
C STATEV(6)=YFLAG (0=ELASTICITY, 1=PLASTICITY)
C
C
C*** IOUT=7 WRITES ON THE .msg FILE
      IOUT=7
C*** SEE USER'S MANUAL SECTION 3.6.1
C
C
C      FLAGS!!!!!!!!!!!!!!!!!!!!
C
C
C      IUNLOAD=01
C      ICONSISTENT=0
C      ITESTDE=0
C
C      NGAUS=32
C
C
C
C      iwr=0
C      if (kinc.eq.20) IWR=01
C      if (kinc.eq.6) write(iout,*) 'noel, npt', noel, npt
C      IF (KINC.EQ.6.AND.NOEL.EQ.649.AND.NPT.EQ.2) IWR=01
C
      IF (IWR.NE.0) THEN
        WRITE(IOUT,*)
        WRITE(IOUT,*) 'KSTEP, KINC'
        WRITE(IOUT,1002) KSTEP,KINC
        WRITE(IOUT,*) 'STEP-TIME, TOTAL-TIME, DTIME'
        WRITE(IOUT,1001) TIME(1),TIME(2),DTIME
        WRITE(IOUT,*) 'NOEL, NPT'
        WRITE(IOUT,1002) NOEL,NPT
      END IF
C
C** TIME(2)=TOTAL-time at the beginning of the current increment
C
C** TIME(1)=STEP-time at the beginning of the current increment
      IF (TIME(1).EQ.0.D0) THEN
        PI=4.D0*DATAN(1.D0)
        CALL KINITIA(Delta,NTENS)
        DO I=1,NDI
          Delta(I)=1.D0
        ENDDO
        CALL KINITIA(AIMX,NTENS*NTENS)
        DO I=1,NDI
          AIMX(I,I)=1.D0
        ENDDO
        DO I=NDI+1,NTENS
          AIMX(I,I)=0.5D0
        ENDDO
        CALL KINITIA(AJMX,NTENS*NTENS)
        DO I=1,NDI
        DO J=1,NDI

```

```

        AJMX (I, J) = 1. D0 / 3. D0
    ENDDO
    ENDDO
    DO I = 1, NTENS
    DO J = 1, NTENS
        AKMX (I, J) = AIMX (I, J) - AJMX (I, J)
    ENDDO
    ENDDO
    AA = 0. D0
    BB = 0. D0
    CALL JACOBI (AA, BB, NGAUS, X, W)
END IF

C
E = PROPS (1)
ANU = PROPS (2)
SIG0 = PROPS (3)
EXPO = PROPS (4)
FN = PROPS (5)
EN = PROPS (6)
SN = PROPS (7)
G = E / (2. D0 * (1. D0 + ANU))
AK = E / (3. D0 * (1. D0 - 2. D0 * ANU))
E0 = SIG0 / E
IF (IWR.NE.0) THEN
    WRITE (IOUT, *) 'E, ANU, SIG0, EXPO'
    WRITE (IOUT, 1001) E, ANU, SIG0, EXPO
    WRITE (IOUT, *) 'G, AK, E0'
    WRITE (IOUT, 1001) G, AK, E0
    IF (FN.EQ.0.D0) WRITE (IOUT, *) 'NO VOID NUCLATION'
    IF (FN.NE.0.D0) THEN
        WRITE (IOUT, *) 'VOID NUCLATION'
        WRITE (IOUT, *) ' FN, EN, SN'
        WRITE (IOUT, 1001) FN, EN, SN
    END IF
END IF

C
EBART = STATEV (1)
FT = STATEV (2)
W1T = STATEV (3)
W2T = STATEV (4)
THETADT = STATEV (5)
THETAT = THETADT * PI / 180. D0
YFLAG = STATEV (6)
ZS = DSIN (THETAT)
ZC = DCOS (THETAT)
AN1T (1) = ZC
AN1T (2) = ZS
AN1T (3) = 0. D0
AN2T (1) = -ZS
AN2T (2) = ZC
AN2T (3) = 0. D0
AN3T (1) = 0. D0
AN3T (2) = 0. D0
AN3T (3) = 1. D0

C
CALL KCOPY (STRESS, STRESST, NTENS)
CALL KCOPY (STRESS, STRSABA, NTENS)
CALL KTRANSPOSE (DROT, DROTT, 3, 3)
CALL KROTSTRS (STRESST, DROTT, QMX, NTENS)

C
IF (IWR.NE.0) THEN

```

```

WRITE(IOUT,*) 'EBART, FT, W1T, W2T, THETAT, YFLAG'
WRITE(IOUT,1001) EBART,FT,W1T,W2T,THETADT,YFLAG
WRITE(IOUT,*) 'AN1T(I), AN2T(I), AN3T(I)'
WRITE(IOUT,1001) (AN1T(I),I=1,3)
WRITE(IOUT,1001) (AN2T(I),I=1,3)
WRITE(IOUT,1001) (AN3T(I),I=1,3)
WRITE(IOUT,*) 'NGAUS'
WRITE(IOUT,1002) NGAUS
WRITE(IOUT,*) 'X(1), X(2), ...'
WRITE(IOUT,1001) X(1),X(2)
WRITE(IOUT,*) 'W(1), W(2), ...'
WRITE(IOUT,1001) W(1),W(2)
WRITE(IOUT,*) 'STRESST BEFORE UN-ROTATION'
WRITE(IOUT,1001) (STRSABA(I),I=1,NTENS)
WRITE(IOUT,*) 'STRESST'
WRITE(IOUT,1001) (STRESST(I),I=1,NTENS)
WRITE(IOUT,*) 'DFGRD0'
DO I=1,3
  WRITE(IOUT,1001) (DFGRD0(I,J),J=1,3)
ENDDO
WRITE(IOUT,*) 'DFGRD1'
DO I=1,3
  WRITE(IOUT,1001) (DFGRD1(I,J),J=1,3)
ENDDO
WRITE(IOUT,*) 'DROT'
DO I=1,3
  WRITE(IOUT,1001) (DROT(I,J),J=1,3)
ENDDO
END IF
C
C*** FORM ELASTIC STIFFNESS CEL
C
CALL KESH_Q(Q,ANU,W1T,W2T,AN1T,AN2T,AN3T,X,W,NGAUS,NTENS,G)
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'Q'
  DO I=1,NTENS
    WRITE(IOUT,1001) (Q(I,J),J=1,NTENS)
  ENDDO
END IF
C
CALL KINV(Q,QINV,NTENS)
DO I=1,NTENS
  DO J=NDI+1,NTENS
    QINV(I,J)=QINV(I,J)/2.D0
  ENDDO
ENDDO
DO I=NDI+1,NTENS
  DO J=1,NTENS
    QINV(I,J)=QINV(I,J)/2.D0
  ENDDO
ENDDO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'QINV'
  DO I=1,NTENS
    WRITE(IOUT,1001) (QINV(I,J),J=1,NTENS)
  ENDDO
END IF
C
Z1=2.D0*G
Z2=3.D0*AK
Z3=FT/(1.D0-FT)

```



```

DO I=1,NTENS
DO J=1,NTENS
  CEL(I,J)=AKMX(I,J)/Z1+AJMX(I,J)/Z2+QINV(I,J)*Z3
ENDDO
ENDDO
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'CEL^(-1)'
  DO I=1,NTENS
    WRITE(IOUT,1001) (CEL(I,J),J=1,NTENS)
  ENDDO
END IF
C
CALL KCOPY(CEL,AUXM1,NTENS*NTENS)
CALL KINV(AUXM1,CEL,NTENS)
C
DO I=1,NTENS
DO J=NDI+1,NTENS
  CEL(I,J)=CEL(I,J)/2.D0
ENDDO
ENDDO
DO I=NDI+1,NTENS
DO J=1,NTENS
  CEL(I,J)=CEL(I,J)/2.D0
ENDDO
ENDDO
C
CALL KCOPY(CEL,DDSDDE,NTENS*NTENS)
CALL KCOPY(CEL,CELT,NTENS*NTENS)
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'CEL'
  DO I=1,NTENS
    WRITE(IOUT,1001) (CEL(I,J),J=1,NTENS)
  ENDDO
END IF
C
C*** VALUE OF YIELD FUNCTION AT TIME T
C
POISSON=0.5D0
CALL KESH_Q(Q,POISSON,W1T,W2T,AN1T,AN2T,AN3T,X,W,NGAUS,NTENS,G)
CALL KINV(Q,QINV,NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
ENDDO
ENDDO
DO I=NDI+1,NTENS
DO J=1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
ENDDO
ENDDO
CALL KMTLD(AMTLDT,FT,G,NTENS,AKMX,QINV)
CALL KYFCN(PHIT,STRESST,EBART,FT,AMTLDT,EXPO,E0,SIG0,AUXM1,NTENS,
+ NDI)
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'M-TILDE AT TIME T'
  DO I=1,NTENS
    WRITE(IOUT,1001) (AMTLDT(I,J),J=1,NTENS)
  ENDDO
  WRITE(IOUT,*) 'PHIT'

```

```

        WRITE(IOUT,1001) PHIT
    ENDIF
C
    AUX=0.D0
    DO I=1,3
    DO J=1,3
        AUX=AUX+DABS(DFGRD1(I,J)-DFGRD0(I,J))
    ENDDO
    ENDDO
    IF (IWR.NE.0) THEN
        WRITE(IOUT,*) 'DFGRD1-DFGRD0'
        WRITE(IOUT,1001) AUX
    END IF
    IF (AUX.NE.0.D0) GOTO 29
C
C
C
C
C
C
C*** DE=0 NEEDS DDSDE ONLY
C
    IF (IUNLOAD.NE.0) GOTO 18
    IF (YFLAG.NE.0.D0) GOTO 19
18 CONTINUE
C*** ELASTICITY
    DO I=1,NTENS
    DO J=1,NTENS
        DDSDE(I,J)=DDSDE(I,J)+STRESST(I)*DELTA(J)
    ENDDO
    ENDDO
    IF (IWR.NE.0) THEN
        WRITE(IOUT,*) 'DEMAG=0, ELASTIC DDSDE'
        DO I=1,NTENS
            WRITE(IOUT,1001) (DDSDE(I,J),J=1,NTENS)
        ENDDO
    END IF
    GOTO 9999
C
19 CONTINUE
C*** PLASTICITY
C
    IF (IWR.NE.0) WRITE(IOUT,*) 'DEMAG=0, PLASTIC DDSDE'
C
    CALL KPLJAC(DDSDE,CEL,STRESST,EBART,FT,W1T,W2T,AN1T,AN2T,AN3T,
+ X,W,NGAUS,S,P,Q,QINV,AMTLDT,DQDW1,DQDW2,AIMX,AJMX,AKMX,
+ AMX,CMX,AN,QMX,AUXM1,AUXM2,AUXM3,AUXM4,OMEGA,
+ NTENS,NDI,ANU,G,AK,EXPO,E0,SIG0,FN,EN,SN,IOUT,IWR)
    DO I=1,NTENS
    DO J=1,NTENS
        DDSDE(I,J)=DDSDE(I,J)+STRESST(I)*DELTA(J)
    ENDDO
    ENDDO
    IF (IWR.NE.0) THEN
        WRITE(IOUT,*) 'DEMAG=0, PLASTIC DDSDE'
        DO I=1,NTENS
            WRITE(IOUT,1001) (DDSDE(I,J),J=1,NTENS)
        ENDDO
    END IF
    GOTO 9999
C

```

```

C
C
C
C
C
  29  CONTINUE
C
C*** INTEGRATE ELASTOPLASTIC EQUATIONS
C
  CALL KINV (DFGRD0, AUXM, 3)
  CALL KMULT (DFGRD1, AUXM, DF, 3, 3, 3)
  IF (IWR.NE.0) THEN
    WRITE (IOUT, *) 'DF'
    DO I=1, 3
      WRITE (IOUT, 1001) (DF (I, J), J=1, 3)
    ENDDO
  END IF
C
  CALL KPOLAR (DF, R, U, V)
  IF (IWR.NE.0) THEN
    WRITE (IOUT, *) 'R'
    DO I=1, 3
      WRITE (IOUT, 1001) (R (I, J), J=1, 3)
    ENDDO
    WRITE (IOUT, *) 'U'
    DO I=1, 3
      WRITE (IOUT, 1001) (U (I, J), J=1, 3)
    ENDDO
    WRITE (IOUT, *) 'V'
    DO I=1, 3
      WRITE (IOUT, 1001) (V (I, J), J=1, 3)
    ENDDO
  END IF
C
  CALL KCOPY (U, UMI1, 3*3)
  DO I=1, 3
    UMI1 (I, I)=UMI1 (I, I)-1.DO
  ENDDO
  CALL KMULT (UMI1, UMI1, UMI2, 3, 3, 3)
  CALL KMULT (UMI2, UMI1, UMI3, 3, 3, 3)
  CALL KMULT (UMI3, UMI1, UMI4, 3, 3, 3)
  DO I=1, 3
    DO J=1, 3
      DETENS (I, J)=UMI1 (I, J)-UMI2 (I, J)/2.DO+UMI3 (I, J)/3.DO
+      -UMI4 (I, J)/4.DO
    ENDDO
  ENDDO
  IF (IWR.NE.0) THEN
    WRITE (IOUT, *) 'DETENS'
    DO I=1, 3
      WRITE (IOUT, 1001) (DETENS (I, J), J=1, 3)
    ENDDO
  END IF
C
  DO I=1, NDI
    DE (I)=DETENS (I, I)
  ENDDO
  DO I=1, NSHR
    IF (I.EQ.1) DE (NDI+I)=DETENS (1, 2)
    IF (I.EQ.2) DE (NDI+I)=DETENS (1, 3)
    IF (I.EQ.3) DE (NDI+I)=DETENS (2, 3)
  ENDDO

```

```

ENDDO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'DE VECTOR (WITH TENSOR-SHEAR COMPONENTS) '
  WRITE(IOUT,1001) (DE(I),I=1,NTENS)
END IF
C
CALL KAB(DE,DE,DEMAG,NTENS,NDI)
DEMAG=DSQRT(2.D0*DEMAG/3.D0)
ETOL=DEMAG*1.D-3
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'DEMAG, ETOL'
  WRITE(IOUT,1001) DEMAG,ETOL
END IF
IF (ITESTDE.NE.0) THEN
  EMAX=4.D0*E0
  RATIO=DEMAG/EMAX
  IF (RATIO.GT.1.D0) THEN
    pnewdt=0.5d0
    PNEWDT=0.9D0/RATIO
    WRITE(IOUT,*) 'DEMAG>4*E0. UMAT CUTS BACK. PNEWDT=', pnewdt
    WRITE(IOUT,*) 'NOEL, NPT, KSTEP, KINC'
    WRITE(IOUT,1002) NOEL,NPT,KSTEP,KINC
    RETURN
  END IF
END IF
C
DO I=NDI+1,NTENS
  DE(I)=DE(I)*2.D0
ENDDO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'DE VECTOR (WITH ENNG-SHEAR COMPONENTS) '
  WRITE(IOUT,1001) (DE(I),I=1,NTENS)
END IF
CALL KMULT(CEL,DE,DSEL,NTENS,NTENS,1)
DO I=1,NTENS
  SEL(I)=STRESST(I)+DSEL(I)
ENDDO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'DSEL'
  WRITE(IOUT,1001) (DSEL(I),I=1,NTENS)
  WRITE(IOUT,*) 'SEL'
  WRITE(IOUT,1001) (SEL(I),I=1,NTENS)
END IF
C
C*** CHECK FOR YIELDING
C
YTOL=SIG0*1.D-4
CALL KYFCN(PHIEL,SEL,EBART,FT,AMTLDT,EXPO,E0,SIG0,AUXM1,NTENS,NDI)
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'PHIEL, YTOL'
  WRITE(IOUT,1001) PHIEL,YTOL
END IF
IF (PHIEL.LE.YTOL) GOTO 1000
IF (PHIEL.GT.YTOL) GOTO 2000
C
C*** ELASTICITY
C
1000 CONTINUE
IF (IWR.NE.0) WRITE(IOUT,*) 'ELASTICITY'
CALL KCOPY(SEL,STRESS,NTENS)
CALL KCOPY(AN1T,AN1,3)

```

```

        CALL KCOPY (AN2T, AN2, 3)
        CALL KCOPY (AN3T, AN3, 3)
        IF (IWR.NE.0) WRITE (IOUT, *) 'STRESS HAT'
        IF (IWR.NE.0) WRITE (IOUT, 1001) (STRESS (I), I=1, NTENS)
C
C*** ROTATE STRESS, AN1, AN2 AND AN3
C
        CALL KROTSTRS (STRESS, R, QMX, NTENS)
        CALL KMULT (R, AN1, V1, 3, 3, 1)
        CALL KCOPY (V1, AN1, 3)
        CALL KMULT (R, AN2, V1, 3, 3, 1)
        CALL KCOPY (V1, AN2, 3)
        CALL KMULT (R, AN3, V1, 3, 3, 1)
        CALL KCOPY (V1, AN3, 3)
        THETA=DATAN2 (AN1 (2), AN1 (1)) *180.D0/PI
        STATEV (5)=THETA
        STATEV (6)=0.D0
        IF (IWR.NE.0) THEN
            WRITE (IOUT, *) 'STRESS AFTER ROTATION'
            WRITE (IOUT, 1001) (STRESS (I), I=1, NTENS)
            WRITE (IOUT, *) 'AN1, AN2 AND AN3 AFTER ROTATION'
            WRITE (IOUT, 1001) (AN1 (I), I=1, 3)
            WRITE (IOUT, 1001) (AN2 (I), I=1, 3)
            WRITE (IOUT, 1001) (AN3 (I), I=1, 3)
            WRITE (IOUT, *) 'THETA AFTER ROTATION'
            WRITE (IOUT, 1001) THETA
        END IF
C
C*** ELASTIC JACOBIAN
C
        DO I=1, NTENS
            DO J=1, NTENS
                DDSDE (I, J)=DDSDDE (I, J)+STRESS (I)*DELTA (J)
            ENDDO
        ENDDO
C
        IF (IWR.NE.0) THEN
            WRITE (IOUT, *) 'ELASTIC DDSDE'
            DO I=1, NTENS
                WRITE (IOUT, 1001) (DDSDE (I, J), J=1, NTENS)
            ENDDO
        END IF
        GOTO 9999
C
C*** PLASTICITY
C
        2000 CONTINUE
        IF (IWR.NE.0) WRITE (IOUT, *) 'PLASTICITY'
        FRACT=0.D0
        AUX=0.D0
C
        IF (YFLAG.NE.0) THEN
            CALL KFINDN (ANT, STRESST, FT, AMTLDT, AUXM1, NTENS, NDI)
            ZZ=ANT (1)+ANT (2)+ANT (3)
            CALL KAB (ANT, DSEL, AUX, NTENS, NDI)
            IF (IWR.NE.0) THEN
                WRITE (IOUT, *) 'AN AT T'
                WRITE (IOUT, 1001) (ANT (I), I=1, NTENS)
                WRITE (IOUT, *) 'N_KK AT T'
                WRITE (IOUT, 1001) ZZ
                WRITE (IOUT, *) 'ANT:DSEL'
            END IF
        END IF

```

```

        WRITE(IOUT,1001) AUX
    END IF
END IF
C
IF (YFLAG.EQ.0.DO.OR.AUX.LT.0.DO) THEN
C*** FIND ELASTIC FRACTION
    CALL KELFR(STRESST,DSEL,FT,NTENS,NDI,AMTLDT,AUXM1,PHIT,FRACT)
    DO I=1,NTENS
        STRESST(I)=STRESST(I)+FRACT*DSEL(I)
        DE(I)=(1.DO-FRACT)*DE(I)
    ENDDO
    IF (IWR.NE.0) THEN
        WRITE(7,*) 'ELASTIC FRACTION R'
        WRITE(7,1001) FRACT
        WRITE(IOUT,*) 'STRESST AFTER SCALING'
        WRITE(IOUT,1001) (STRESST(I),I=1,NTENS)
        WRITE(IOUT,*) 'DE AFTER SCALING (ENGENG-SHEAR)'
        WRITE(IOUT,1001) (DE(I),I=1,NTENS)
        CALL KYFCN(PHIT,STRESST,EBART,FT,AMTLDT,EXPO,E0,SIG0,AUXM1,
+           NTENS,NDI)
        WRITE(IOUT,*) 'PHIT AFTER SCALING OF STRESS'
        WRITE(IOUT,1001) PHIT
    END IF
END IF
C
C*** PRELIMINARY CALCULATIONS AT THE START OF THE INCREMENT
C
C FIND N AT T
    CALL KFINDN(ANT,STRESST,FT,AMTLDT,AUXM1,NTENS,NDI)
    CALL KCOPY(ANT,AN,NTENS)
C FIND A, C AT T
    CALL KAC(AMX,CMX,FT,W1T,W2T,AN1T,AN2T,AN3T,S,P,AUXM1,AUXM2,
+ X,W,NGAUS,NTENS,NDI,AIMX)
C
IF (IWR.NE.0) THEN
    WRITE(IOUT,*) 'ESHELBY TENSOR S (ANU=0.5) FROM KAC'
    DO I=1,NTENS
        WRITE(IOUT,1001) (S(I,J),J=1,NTENS)
    ENDDO
    WRITE(IOUT,*) 'A AT T'
    DO I=1,NTENS
        WRITE(IOUT,1001) (AMX(I,J),J=1,NTENS)
    ENDDO
    WRITE(IOUT,*) 'EHSELBY TENSOR PI (ANU=0.5) FROM KAC'
    DO I=1,NTENS
        WRITE(IOUT,1001) (P(I,J),J=1,NTENS)
    ENDDO
    WRITE(IOUT,*) 'C AT T'
    DO I=1,NTENS
        WRITE(IOUT,1001) (CMX(I,J),J=1,NTENS)
    ENDDO
END IF
C
CALL KCOPY(ANT,V1,NTENS)
DO I=NDI+1,NTENS
    V1(I)=V1(I)*2.DO
END DO
CALL KMULT(AMX,V1,V2,NTENS,NTENS,1)
CALL KCOPY(V2,ADOTN,NTENS)
C
C FIND OMEGA AT T

```

```

OM=0.D0
WTOL=1.D-2
IF (DABS(W1T-W2T).GE.WTOL) THEN
  T12=CMX(4,1)*ANT(1)+CMX(4,2)*ANT(2)+CMX(4,3)*ANT(3)+
+ 2.D0*CMX(4,4)*ANT(4)
  B21=(W2T*W2T+W1T*W1T)/(W2T*W2T-W1T*W1T)
  V3(1)=-DSIN(2.D0*THETAT)
  V3(2)=-V3(1)
  V3(3)=0.D0
  V3(4)=DCOS(2.D0*THETAT)
  CALL KAB(V3,ADOTN,AUX,NTENS,NDI)
  OM=-T12-0.5D0*B21*AUX
END IF
C
CALL KINITIA(OMEGAT,3*3)
OMEGAT(1,2)=-OM
OMEGAT(2,1)=OM
CALL KCOPY(OMEGAT,OMEGA,3*3)
C
ZX=W1T-W2T
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'W1T-W2T, WTOL'
  WRITE(IOUT,1001) ZX,WTOL
  WRITE(IOUT,*) 'OMEGA AT T'
  DO I=1,3
    WRITE(IOUT,1001) (OMEGAT(I,J),J=1,3)
  ENDDO
ENDIF
C
R1(1)=2.D0*STRESST(4)*OM
R1(2)=-R1(1)
R1(3)=0.D0
R1(4)=(STRESST(2)-STRESST(1))*OM
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'R1=S.OM-OM.S'
  WRITE(IOUT,1001) (R1(I),I=1,NTENS)
END IF
C
CALL KYCURVE(YIELD,H,EBART,EXPO,E0,SIG0)
AUX=(1.D0-FT)*YIELD
DO I=1,NTENS
  R2(I)=STRESST(I)/AUX
END DO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'R2'
  WRITE(IOUT,1001) (R2(I),I=1,NTENS)
END IF
C
V3(1)=AN3T(1)**2-AN1T(1)**2
V3(2)=AN3T(2)**2-AN1T(2)**2
V3(3)=AN3T(3)**2-AN1T(3)**2
V3(4)=2.D0*(AN3T(1)*AN3T(2)-AN1T(1)*AN1T(2))
CALL KMULT(V3,AMX,R31,1,NTENS,NTENS)
V3(1)=AN3T(1)**2-AN2T(1)**2
V3(2)=AN3T(2)**2-AN2T(2)**2
V3(3)=AN3T(3)**2-AN2T(3)**2
V3(4)=2.D0*(AN3T(1)*AN3T(2)-AN2T(1)*AN2T(2))
CALL KMULT(V3,AMX,R32,1,NTENS,NTENS)
DO I=1,NTENS
  R31(I)=W1T*R31(I)
  R32(I)=W2T*R32(I)

```

```

END DO
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'R31'
  WRITE(IOUT,1001) (R31(I),I=1,NTENS)
  WRITE(IOUT,*) 'R32'
  WRITE(IOUT,1001) (R32(I),I=1,NTENS)
END IF
C
C FIND HH, H1, H2, H3, GG, G1, G2 AT T (needed for DL estimate)
GG=0.D0
DO I=1,NDI
  GG=GG+ANT(I)
END DO
HH=GG*(1.D0-FT)
C nucleation term added below
C
V3(1)=AN3T(1)**2-AN1T(1)**2
V3(2)=AN3T(2)**2-AN1T(2)**2
V3(3)=AN3T(3)**2-AN1T(3)**2
V3(4)=AN3T(1)*AN3T(2)-AN1T(1)*AN1T(2)
CALL KAB(V3,ADOTN,G1,NTENS,NDI)
V3(1)=AN3T(1)**2-AN2T(1)**2
V3(2)=AN3T(2)**2-AN2T(2)**2
V3(3)=AN3T(3)**2-AN2T(3)**2
V3(4)=AN3T(1)*AN3T(2)-AN2T(1)*AN2T(2)
CALL KAB(V3,ADOTN,G2,NTENS,NDI)
H1=G1*W1T
H2=G2*W2T
C
CALL KYCURVE(YIELD,H,EBART,EXPO,E0,SIG0)
CALL KAB(STRESST,ANT,SUM,NTENS,NDI)
H3=SUM/((1.D0-FT)*YIELD)
ANUCL=0.D0
IF (FN.NE.0.D0)
+ ANUCL=(FN/(SN*DSQRT(2.D0*PI)))*DEXP(-0.5D0*((EBART-EN)/SN)**2)
HH=HH+ANUCL*H3
C
HHT=HH
H1T=H1
H2T=H2
H3T=H3
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'HH, H1, H2, H3 AT T'
  WRITE(IOUT,1001) HH,H1,H2,H3
END IF
C
C*** DETERMINE DL
C
C FIRST ESTIMATES FOR DL AND DEP
C
CALL KMTLD(AMTLDT,FT,G,NTENS,AKMX,QINV)
CALL KFINDN(ANT,STRESST,FT,AMTLDT,AUXM1,NTENS,NDI)
CALL KYCURVE(YIELD,H,EBART,EXPO,E0,SIG0)
DPHIDEP=-2.D0*YIELD*H
POISSON=0.5D0
CALL KDQ(DQDW1,DQDW2,POISSON,W1T,W2T,AN1T,AN2T,AN3T,X,W,NGAUS,
+ NTENS,G)
CALL KDFD(DPHIDF,DPHIDW1,DPHIDW2,AKMX,FT,STRESST,DQDW1,DQDW2,
+ AUXM1,AUXM2,AUXM3,AUXM4,NTENS,NDI,G,W1T,W2T,AN1T,AN2T,AN3T,
+ QINV)

```



```

C
CALL KFINDDL(DL, STRESST, EBART, FT, W1T, W2T, AN1T, AN2T, AN3T,
+ ANT, CEL, DE, NTENS, NDI, HH, H1, H2, H3, EXPO, E0, SIG0, G, Q, QINV,
+ DQDW1, DQDW2, AKMX, AUXM1, AUXM2, AUXM3, AUXM4,
+ DPHIDF, DPHIDW1, DPHIDW2, DPHIDEP)
C
DO I=1, NTENS
  DEP(I)=DL*ANT(I)
END DO
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'FIRST ESTIMATE FOR DL'
  WRITE(IOUT,1001) DL
  WRITE(IOUT,*) 'FIRST ESTIMATE FOR DEP'
  WRITE(IOUT,1001) (DEP(I), I=1, NTENS)
END IF
C
C
C
C
C NEWTON LOOP FOR DL AND DEP
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'NEWTON LOOP STARTS'
  WRITE(IOUT,*) '  DL, DEP(I)'
  WRITE(IOUT,*) '-PHI, -GV(I)'
END IF
C
DO ITER=1, 20
C
  CALL KCOPY(DEP, V1, NTENS)
  DO I=NDI+1, NTENS
    V1(I)=V1(I)*2.D0
  END DO
  CALL KMULT(AMX, V1, DEVOID, NTENS, NTENS, 1)
  CALL KUPDATE(DL, STRESS, EBAR, F, W1, W2, AN1, AN2, AN3, STRESST, EBART,
+ FT, W1T, W2T, AN1T, AN2T, AN3T, SEL, CELT, NTENS, NDI, ANT, GG, G1, G2, HH,
+ H1, H2, H3, OMEGAT, V, DEVOID, DEP, R1, R2, R31, R32, ANUCL, IWR, IOUT)
C
  POISSON=0.5D0
  CALL KESH_Q(Q, POISSON, W1, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)
  CALL KINV(Q, QINV, NTENS)
  DO I=1, NTENS
    DO J=NDI+1, NTENS
      QINV(I, J)=QINV(I, J)/2.D0
    ENDDO
  ENDDO
  DO I=NDI+1, NTENS
    DO J=1, NTENS
      QINV(I, J)=QINV(I, J)/2.D0
    ENDDO
  ENDDO
C
  CALL KMTLD(AMTLD, F, G, NTENS, AKMX, QINV)
  CALL KYFCN(PHI, STRESS, EBAR, F, AMTLD, EXPO, E0, SIG0, AUXM1, NTENS,
+ NDI)
  CALL KFINDN(AN, STRESS, F, AMTLD, AUXM1, NTENS, NDI)
  GV(1)=-PHI
  DO I=1, NTENS
    GV(1+I)=- (DEP(I) -DL*AN(I))
  END DO

```

```

C
  IF (IWR.NE.0) THEN
    WRITE (IOUT,2001) DL, (DEP(I), I=1, NTENS)
    WRITE (IOUT,1001) (GV(I), I=1, 1+NTENS)
  END IF

C
  IF (DABS(GV(1)).GT.YTOL) GOTO 100
  DO I=2, 1+NTENS
    IF (DABS(GV(I)).GT.ETOL) GOTO 100
  END DO
  GOTO 2500
100  CONTINUE

C
C*** FIND DPHIDL, DYDEP(I)
  CALL KAB(AN, R1, DPHIDL, NTENS, NDI)
  CALL KYCURVE(YIELD, H, EBAR, EXPO, E0, SIG0)
  DPHIDEP=-2.D0*YIELD*H
  POISSON=0.5D0
  CALL KDQ(DQDW1, DQDW2, POISSON, W1, W2, AN1, AN2, AN3, X, W, NGAUS,
+   NTENS, G)
  CALL KDFF(DPHIDF, DPHIDW1, DPHIDW2, AKMX, F, STRESS, DQDW1, DQDW2,
+   AUXM1, AUXM2, AUXM3, AUXM4, NTENS, NDI, G, W1, W2, AN1, AN2, AN3,
+   QINV)
  CALL KCOPY(AN, V1, NTENS)
  DO I=NDI+1, NTENS
    V1(I)=V1(I)*2.D0
  END DO
  CALL KMULT(CELT, V1, V2, NTENS, NTENS, 1)
  DO I=1, NTENS
    DFDEP(I)=(1.D0-FT)*DELTA(I)+ANUCL*R2(I)
  END DO
  DO I=1, NTENS
    DYDEP(I)=-V2(I)+DPHIDEP*R2(I)+DPHIDF*DFDEP(I)+
+   DPHIDW1*R31(I)+DPHIDW2*R32(I)
  END DO

C
C*** FIND DGDL(I), DGDEP(I, J)
  CALL KCOPY(R1, V1, NTENS)
  DO I=NDI+1, NTENS
    V1(I)=V1(I)*2.D0
  END DO
  CALL KMULT(AMTLD, V1, V2, NTENS, NTENS, 1)
  AUX=2.D0*DL/(1.D0-F)
  DO I=1, NTENS
    DGDL(I)=-AN(I)-AUX*V2(I)
  END DO

C
  CALL KDN(DNDF, DNDW1, DNDW2, F, AKMX, Q, QINV, G, STRESS, NTENS, NDI,
+   AN, W1, W2, AN1, AN2, AN3, X, W, NGAUS, AUXM1, AMTLD)
  CALL KCOPY(CELT, AUXM1, NTENS*NTENS)
  DO I=NDI+1, NTENS
    DO J=1, NTENS
      AUXM1(I, J)=AUXM1(I, J)*2.D0
    END DO
  END DO
  CALL KMULT(AMTLD, AUXM1, AUXM2, NTENS, NTENS, NTENS)
  AUX=2.D0/(1.D0-F)
  DO I=1, NTENS
    DO J=1, NTENS
      DGDEP(I, J)=AIMX(I, J)-DL*(-AUX*AUXM2(I, J)+DNDF(I)*DFDEP(J)+
+   DNDW1(I)*R31(J)+DNDW2(I)*R32(J))
    END DO
  END DO

```

```

        END DO
        END DO
C
        DJAC(1,1)=DPHIDL
        DO J=1,NTENS
            DJAC(1,1+J)=DYDEP(J)
        END DO
        DO I=1,NTENS
            DJAC(1+I,1)=DGDL(I)
        END DO
        DO I=1,NTENS
        DO J=1,NTENS
            DJAC(1+I,1+J)=DGDEP(I,J)
        END DO
        END DO
        DO I=1,1+NTENS
        DO J=1+NDI+1,1+NTENS
            DJAC(I,J)=DJAC(I,J)*2.D0
        END DO
        END DO
        CALL KINV(DJAC,DINV,1+NTENS)
        CALL KMULT(DINV,GV,DUN,1+NTENS,1+NTENS,1)
        DL=DL+DUN(1)
        DO I=1,NTENS
            DEP(I)=DEP(I)+DUN(1+I)
        END DO
    END DO
    WRITE(IOUT,*) 'NEWTON LOOP DOES NOT CONVERGE. PNEWDT SET TO 0.5.'
    WRITE(IOUT,*) 'NOEL, NPT, KSTEP, KINC'
    WRITE(IOUT,1002) NOEL,NPT,KSTEP,KINC
    PNEWDT=0.5D0
    RETURN
2500 CONTINUE
C
    IF (IWR.NE.0) THEN
        WRITE(IOUT,*) 'STRESS'
        WRITE(IOUT,1001) (STRESS(I),I=1,NTENS)
        WRITE(IOUT,*) 'EBAR, F'
        WRITE(IOUT,1001) EBAR, F
        WRITE(IOUT,*) 'W1, W2'
        WRITE(IOUT,1001) W1,W2
        WRITE(IOUT,*) 'AN1(I), AN2(I), AN3(I)'
        WRITE(IOUT,1001) (AN1(I),I=1,3)
        WRITE(IOUT,1001) (AN2(I),I=1,3)
        WRITE(IOUT,1001) (AN3(I),I=1,3)
        WRITE(IOUT,*) 'DEP'
        WRITE(IOUT,1001) (DEP(I),I=1,NTENS)
    END IF
C
C*** ROTATE STRESS, AN1, AN2 AND AN3
C
        CALL KROTSTRS(STRESS,R,QMX,NTENS)
        CALL KMULT(R,AN1,V1,3,3,1)
        CALL KCOPY(V1,AN1,3)
        CALL KMULT(R,AN2,V1,3,3,1)
        CALL KCOPY(V1,AN2,3)
        CALL KMULT(R,AN3,V1,3,3,1)
        CALL KCOPY(V1,AN3,3)
C
C MAKE SURE THAT W1<W2
    IF (W1.GT.W2) THEN

```

```

        AUX=W1
        W1=W2
        W2=AUX
        CALL KCOPY (AN1,V1,3)
        CALL KCOPY (AN2,AN1,3)
        CALL KCOPY (V1,AN2,3)
        AUX=AN1(1)*AN2(2)-AN2(1)*AN1(2)
        IF (AUX.LT.0.D0) THEN
            DO I=1,2
                AN2(I)=-AN2(I)
            END DO
        END IF
    END IF
C
    THETA=DATAN2 (AN1(2),AN1(1))*180.D0/PI
    IF (IWR.NE.0) THEN
        WRITE(IOUT,*) 'ROTATED STRESS VECTOR'
        WRITE(IOUT,1001) (STRESS(I),I=1,NTENS)
        WRITE(IOUT,*) 'ROTATED UNIT VECTORS AN1, AN2, AN3'
        WRITE(IOUT,1001) (AN1(I),I=1,3)
        WRITE(IOUT,1001) (AN2(I),I=1,3)
        WRITE(IOUT,1001) (AN3(I),I=1,3)
        WRITE(IOUT,*) 'THETA IN DEGREES'
        WRITE(IOUT,1001) THETA
    END IF
C
C*** UPDATE STATE VARIABLES
C
C
        STATEV(1)=EBAR
        STATEV(2)=F
        STATEV(3)=W1
        STATEV(4)=W2
        STATEV(5)=THETA
        STATEV(6)=1.D0
C
C*** PLASTIC TANGENT MODULUS
C
        iwrl=iwr
C
        IF (ICONCONSISTENT.EQ.0) THEN
            CALL KPLJAC (ddsdde,CEL,STRESS,EBAR,F,W1,W2,AN1,AN2,AN3,
+ X,W,NGAUS,S,P,Q,QINV,AMTLD,DQDW1,DQDW2,AIMX,AJMX,AKMX,
+ AMX,CMX,AN,QMX,AUXM1,AUXM2,AUXM3,AUXM4,OMEGA,
+ NTENS,NDI,ANU,G,AK,EXPO,E0,SIG0,FN,EN,SN,IOUT,IWR1)
C
C*** PLASTIC JACOBIAN
C
C
        ELSE
            CALL KKPLJAC (DDSDDE,CEL,STRESS,EBAR,F,W1,W2,AN1,AN2,AN3,
+ X,W,NGAUS,S,P,Q,QINV,AMTLD,DQDW1,DQDW2,AIMX,AJMX,AKMX,
+ AMX,CMX,AN,QMX,AUXM1,AUXM2,AUXM3,AUXM4,OMEGA,
+ NTENS,NDI,ANU,G,AK,EXPO,E0,SIG0,IOUT,IWR1,
+ CELT,ANT,HHT,H1T,H2T,H3T,OMEGAT,STRESST)
        END IF
C
        DO I=1,NTENS
            DO J=1,NTENS
                DDSDDDE(I,J)=FRACT*CEL(I,J)+(1.D0-FRACT)*DDSDDE(I,J)
            END DO
        END DO
C

```

```

DO I=1,NTENS
DO J=1,NTENS
  DDSDE (I, J)=DDSDE (I, J)+STRESS (I)*DELTA (J)
ENDDO
ENDDO
C
  IF (IWR.NE.0) THEN
    IF (ICONCONSISTENT.EQ.0) WRITE (IOUT,*)
+   'FINAL TANGENT MODULUS'
    IF (ICONCONSISTENT.NE.0) WRITE (IOUT,*)
+   'FINAL CONSISTENT PLASTIC DDSDE'
    DO I=1,NTENS
      WRITE (IOUT,1001) (DDSDE (I, J), J=1,NTENS)
    END DO
  END IF
C
C
C
9999 CONTINUE
C
  RETURN
1001 FORMAT (1P8E13.5)
2001 FORMAT (2X,1P8E13.5)
1002 FORMAT (10I5)
END
C
C*****
C
  SUBROUTINE KDN (DNDF, DNDW1, DNDW2, F, AKMX, Q, QINV, G, STRESS, NTENS, NDI,
+ AN, W1, W2, AN1, AN2, AN3, X, W, NGAUS, AUXM1, AMTLD)
C
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
  DIMENSION DNDF (NTENS), DNDW1 (NTENS), DNDW2 (NTENS), AKMX (NTENS, NTENS),
+ Q (NTENS, NTENS), QINV (NTENS, NTENS), STRESS (NTENS),
+ AN (NTENS), AN1 (3), AN2 (3), AN3 (3), X (NGAUS), W (NGAUS),
+ AUXM1 (NTENS, NTENS), AMTLD (NTENS, NTENS)
  DIMENSION V1 (6)
C
  AUX=2.D0*(1.D0+F)*G/(1.D0-F)
  DO I=1,NTENS
  DO J=1,NTENS
    AUXM1 (I, J)=AKMX (I, J)+AUX*QINV (I, J)
  END DO
  END DO
  CALL KCOPY (STRESS, V1, NTENS)
  DO I=NDI+1,NTENS
    V1 (I)=V1 (I)*2.D0
  END DO
  CALL KMULT (AUXM1, V1, DNDF, NTENS, NTENS, 1)
  AUX=3.D0/(1.D0-F)**2
  DO I=1,NTENS
    DNDF (I)=AUX*DNDF (I)
  END DO
C
  POISSON=0.5D0
  DW1=0.005D0*W1
  W11=W1-DW1
  CALL KESH_Q (Q, POISSON, W11, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)
  CALL KINV (Q, QINV, NTENS)
  DO I=1,NTENS

```

```

DO J=NDI+1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
END DO
END DO
DO I=NDI+1,NTENS
DO J=1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
ENDDO
ENDDO
CALL KMTLD(AMTLD,F,G,NTENS,AKMX,QINV)
CALL KFINDN(V1,STRESS,F,AMTLD,AUXM1,NTENS,NDI)
DO I=1,NTENS
  DNDW1(I)=(V1(I)-AN(I))/(-DW1)
END DO
C
POISSON=0.5D0
DW2=0.005D0*W2
W21=W2+DW2
CALL KESH_Q(Q,POISSON,W1,W21,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
CALL KINV(Q,QINV,NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
END DO
END DO
DO I=NDI+1,NTENS
DO J=1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
END DO
END DO
CALL KMTLD(AMTLD,F,G,NTENS,AKMX,QINV)
CALL KFINDN(V1,STRESS,F,AMTLD,AUXM1,NTENS,NDI)
DO I=1,NTENS
  DNDW2(I)=(V1(I)-AN(I))/DW2
END DO
C
RETURN
END
C
C*****
C
SUBROUTINE KCHOOSE(W1,W2,NGAUS)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
NGAUS=32
ANMAX=80.D0
C
IF (W1.LE.0.1D0.OR.W2.GE.(10.D0*W1)) THEN
  NGAUS=ANMAX
  RETURN
END IF
C
IF (W1.GT.0.1D0.AND.W1.LE.1.D0.AND.W2.GE.1.D0.AND.
+ W2.LT.(10.D0*W1)) THEN
  AN0=110.167D0-366.855*W1+477.098*W1*W1-210.179*W1*W1*W1
  AUX=(ANMAX-AN0)*(W2-1.D0)/(10.D0*W1-1.D0)
  NGAUS=AN0+AUX
END IF
C
RETURN

```

```

      END
C
C*****
C
      SUBROUTINE KINITIA(A,N)
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
      DIMENSION A(1)
C
      DO I=1,N
         A(I)=0.D0
      ENDDO
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KPOLAR(F,R,U,V)
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
      DIMENSION F(3,3),R(3,3),U(3,3),V(3,3)
      DIMENSION FT(3,3),C(3,3),C2(3,3),AL(3),AI(3,3),UINV(3,3),RT(3,3)
C
      PI=4.D0*DATAN(1.D0)
      CALL KINITIA(AI,3*3)
      DO 10 I=1,3
10      AI(I,I)=1.D0
C
      CALL KTRANSPOSE(F,FT,3,3)
      CALL KMULT(FT,F,C,3,3,3)
C
      CALL KMULT(C,C,C2,3,3,3)
      TR1= C(1,1)+ C(2,2)+ C(3,3)
      TR2=C2(1,1)+C2(2,2)+C2(3,3)
      DET=C(1,1)*(C(2,2)*C(3,3)-C(3,2)*C(2,3))
      *   -C(1,2)*(C(2,1)*C(3,3)-C(3,1)*C(2,3))
      *   +C(1,3)*(C(2,1)*C(3,2)-C(3,1)*C(2,2))
      AI1=TR1
      AI2=0.5D0*(TR1*TR1-TR2)
      AI3=DET
C
      B=AI2-AI1*AI1/3.D0
      D=- (2.D0/27.D0)*AI1**3+AI1*AI2/3.D0-AI3
C
      IF (DABS(B).LT.1.D-6) THEN
         AUX=AI1/3.D0
         AL(1)=AUX
         AL(2)=AUX
         AL(3)=AUX
      ELSE
         AM=2.D0*DSQRT(-B/3.D0)
         AN=3.D0*D/(AM*B)
         AUX=1.D0-AN*AN
         IF (AUX.LT.0.D0) THEN
            IF (AN.GT.1.D0) T=0.D0
            IF (AN.LT.(-1.D0)) T=PI/3.D0
         ELSE
            T=DATAN2(DSQRT(AUX),AN)/3.D0
      ENDIF

```

```

        END IF
        DO I=1,3
            X=AM*DCOS(T+2.D0*(I-1)*PI/3.D0)
            AL(I)=X+AI1/3.D0
        ENDDO
    END IF
C
    DO I=1,3
        AL(I)=DSQRT(AL(I))
    ENDDO
C
    BI1=AL(1)+AL(2)+AL(3)
    BI2=AL(1)*AL(2)+AL(2)*AL(3)+AL(3)*AL(1)
    BI3=AL(1)*AL(2)*AL(3)
    DD=BI1*BI2-BI3
C
    AUX1=BI1*BI1-BI2
    AUX2=BI1*BI3
    DO I=1,3
        DO J=1,3
            U(I,J)=(-C2(I,J)+AUX1*C(I,J)+AUX2*AI(I,J))/DD
            UINV(I,J)=(C(I,J)-BI1*U(I,J)+BI2*AI(I,J))/BI3
        ENDDO
    ENDDO
C
    CALL KMULT(F,UINV,R,3,3,3)
    CALL KTRANPOSE(R,RT,3,3)
    CALL KMULT(F,RT,V,3,3,3)
C
    RETURN
    END
C
C*****
C
    SUBROUTINE KMULT(A,B,C,L,M,N)
C
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
    DIMENSION A(L,M),B(M,N),C(L,N)
C
    DO 10 I=1,L
        DO 10 J=1,N
            AUX=0.D0
            DO 20 K=1,M
                20  AUX=AUX+A(I,K)*B(K,J)
            10  C(I,J)=AUX
C
    RETURN
    END
C
C*****
C
    SUBROUTINE KCOPY(A,B,N)
C
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
    DIMENSION A(1),B(1)
C
    DO 10 I=1,N
        10  B(I)=A(I)
C

```



```

RETURN
END
C
C*****
C
SUBROUTINE KADOTB(A,B,C)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION A(3),B(3)
C
C=A(1)*B(1)+ A(2)*B(2)+ A(3)*B(3)
C
RETURN
END
C
C*****
C
SUBROUTINE KAB(A,B,C,NTENS,NDI)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION A(NTENS),B(NTENS)
C
C=0.D0
DO I=1,NDI
  C=C+A(I)*B(I)
ENDDO
DO I=NDI+1,NTENS
  C=C+A(I)*B(I)*2.D0
ENDDO
C
RETURN
END
C
C*****
C
SUBROUTINE KTRANSPOSE(A,AT,M,N)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION A(M,N),AT(N,M)
C
DO 10 I=1,N
DO 10 J=1,M
10 AT(I,J)=A(J,I)
C
RETURN
END
C
C*****
C
SUBROUTINE KINV(A,AINV,N)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION A(N,N),AINV(N,N)
DIMENSION WORK(100),IPIV(100)
C
IOUT=7
CALL KCOPY(A,AINV,N*N)

```

```

C*** LAPACK library
C*** f77 filename.f -lcomplib.sgimath
M=N
NN=N
LDA=N
LWORK=N
CALL DGETRF (M, NN, AINV, LDA, IPIV, INFO)
IF (INFO.NE.0) THEN
  WRITE (IOUT, *) ' INFO NONZERO FROM DGETRF'
  WRITE (IOUT, *) ' INFO =', INFO
END IF
CALL DGETRI (NN, AINV, LDA, IPIV, WORK, LWORK, INFO)
IF (INFO.NE.0) THEN
  WRITE (IOUT, *) ' INFO NONZERO FROM DGETRI'
  WRITE (IOUT, *) ' INFO =', INFO
END IF

C
  RETURN
END

C
C*****
C
C   SUBROUTINE KYCURVE (YIELD, H, EBAR, EXPO, E0, SIG0)
C
C   IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
C   IF (EXPO.GT.50.D0) THEN
C     YIELD=SIG0
C     H=0.D0
C   ELSE
C     YIELD=SIG0*(EBAR/E0+1.D0)**(1.D0/EXPO)
C     H=(SIG0/(E0*EXPO))*(SIG0/YIELD)**(EXPO-1.D0)
C   END IF
C
C   RETURN
END

C
C*****
C
C   SUBROUTINE KMTLD (AMTLD, F, G, NTENS, AKMX, QINV)
C
C   IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
C   DIMENSION AKMX (NTENS, NTENS), AMTLD (NTENS, NTENS), QINV (NTENS, NTENS)
C
C   AUX=3.D0*F*G/(1.D0-F)
C   DO I=1, NTENS
C     DO J=1, NTENS
C       AMTLD (I, J)=1.5D0*AKMX (I, J)+AUX*QINV (I, J)
C     END DO
C   END DO
C
C   RETURN
END

C
C*****
C
C   SUBROUTINE KYFCN (PHI, STRESS, EBAR, F, AMTLD, EXPO, E0, SIG0, AUXM1, NTENS,
+ NDI)
C

```

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION STRESS (1) , AMTLD (NTENS, NTENS) , AUXM1 (NTENS, NTENS) ,
+ AUX1 (6)
C
      CALL KCOPY (AMTLD, AUXM1, NTENS*NTENS)
      DO I=1, NTENS
      DO J=NDI+1, NTENS
        AUXM1 (I, J) = AUXM1 (I, J) * 2. D0
      ENDDO
      ENDDO
      CALL KMULT (AUXM1, STRESS, AUX1, NTENS, NTENS, 1)
C
      CALL KAB (STRESS, AUX1, SE, NTENS, NDI)
C
      CALL KYCURVE (YIELD, H, EBAR, EXPO, E0, SIG0)
C
      PHI=SE / (1. D0-F) -YIELD*YIELD
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KFINDN (AN, STRESS, F, AMTLD, AUXM1, NTENS, NDI)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION AN (1) , STRESS (1) , AMTLD (NTENS, NTENS) , AUXM1 (NTENS, NTENS)
C
      CALL KCOPY (AMTLD, AUXM1, NTENS*NTENS)
      DO 40 I=1, NTENS
      DO 40 J=NDI+1, NTENS
40      AUXM1 (I, J) = AUXM1 (I, J) * 2. D0
      CALL KMULT (AUXM1, STRESS, AN, NTENS, NTENS, 1)
C
      AUX=2. D0 / (1. D0-F)
      DO 50 I=1, NTENS
50      AN (I) =AN (I) *AUX
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KROTSTRS (STRESS, R, QMX, NTENS)
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      DIMENSION STRESS (NTENS) , R (3, 3) , QMX (NTENS, NTENS) , AUX (6)
C
      CALL KINITIA (QMX, NTENS*NTENS)
      DO I=1, 3
      DO J=1, 3
        QMX (I, J) =R (I, J) **2
      ENDDO
      ENDDO
C
      QMX (1, 4) =2. D0 *R (1, 1) *R (1, 2)
      QMX (2, 4) =2. D0 *R (2, 1) *R (2, 2)
      QMX (3, 4) =2. D0 *R (3, 1) *R (3, 2)

```

```

C
  QMX(4,1)=R(1,1)*R(2,1)
  QMX(4,2)=R(1,2)*R(2,2)
  QMX(4,3)=R(1,3)*R(2,3)
C
  QMX(4,4)=R(1,2)*R(2,1)+R(2,2)*R(1,1)
C
  CALL KMULT(QMX,STRESS,AUX,NTENS,NTENS,1)
  CALL KCOPY(AUX,STRESS,NTENS)
C
  RETURN
  END
C
C*****
C
  SUBROUTINE KELFR(STRESST,DSEL,FT,NTENS,NDI,AMTLDT,AUXM1,PHIT,
+ FRACT)
C
  IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
  DIMENSION STRESST(NTENS),DE(NTENS),AMTLDT(NTENS,NTENS),
+ AUXM1(NTENS,NTENS),DSEL(6),AUX1(6)
C
  CALL KCOPY(AMTLDT,AUXM1,NTENS*NTENS)
  DO I=1,NTENS
  DO J=NDI+1,NTENS
    AUXM1(I,J)=AUXM1(I,J)*2.D0
  ENDDO
  ENDDO
C
  CALL KMULT(AUXM1,DSEL,AUX1,NTENS,NTENS,1)
C***
  AUX1=M_TILDE:DSEL
  CALL KAB(DSEL,AUX1,A,NTENS,NDI)
C
  CALL KAB(STRESST,AUX1,B,NTENS,NDI)
C
  C=(1.D0-FT)*PHIT
C
  D=B*B-A*C
  IF(D.GE.0.D0) THEN
    FRACT=(-B+DSQRT(D))/A
  ELSE
    FRACT=0.D0
  END IF
C
  RETURN
  END
C
C*****
C
  SUBROUTINE KAC(AMX,CMX,F,W1,W2,AN1,AN2,AN3,S,P,AUXM1,AUXM2,
+ X,W,NGAUS,NTENS,NDI,AIMX)
C
  IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
  DIMENSION AMX(NTENS,NTENS),CMX(NTENS,NTENS),S(NTENS,NTENS),
+ P(NTENS,NTENS),AUXM1(NTENS,NTENS),AUXM2(NTENS,NTENS),
+ AN1(1),AN2(1),AN3(1),X(1),W(1),AIMX(NTENS,NTENS)
C
  POISSON=0.5D0
  CALL KESH_S(S,POISSON,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS)

```

```

AUX=1.D0-F
DO I=1,NTENS
DO J=1,NTENS
  AMX(I,J)=AIMX(I,J)-AUX*S(I,J)
ENDDO
ENDDO
CALL KCOPY(AMX,AUXM1,NTENS*NTENS)
CALL KINV(AUXM1,AMX,NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  AMX(I,J)=AMX(I,J)/2.D0
ENDDO
ENDDO
DO I=NDI+1,NTENS
DO J=1,NTENS
  AMX(I,J)=AMX(I,J)/2.D0
ENDDO
ENDDO
C FIND C
CALL KESH_PI(P,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS)
CALL KCOPY(P,AUXM1,NTENS*NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  AUXM1(I,J)=AUXM1(I,J)*2.D0
ENDDO
ENDDO
CALL KMULT(AUXM1,AMX,AUXM2,NTENS,NTENS,NTENS)
AUX=- (1.D0-F)
DO I=1,NTENS
DO J=1,NTENS
  CMX(I,J)=AUX*AUXM2(I,J)
ENDDO
ENDDO
C
RETURN
END
C
C*****
C
SUBROUTINE KFINDDL(DL,STRESS,EBAR,F,W1,W2,AN1,AN2,AN3,
+ AN,CEL,DE,NTENS,NDI,HH,H1,H2,H3,EXPO,E0,SIG0,G,Q,QINV,
+ DQDW1,DQDW2,AKMX,AUXM1,AUXM2,AUXM3,AUXM4,
+ DPHIDF,DPHIDW1,DPHIDW2,DPHIDEP)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION STRESS(1),AN1(1),AN2(1),AN3(1),AN(1),
+ CEL(NTENS,NTENS),DE(1),Q(NTENS,NTENS),
+ QINV(NTENS,NTENS),DQDW1(NTENS,NTENS),DQDW2(NTENS,NTENS),
+ AKMX(NTENS,NTENS),AUXM1(NTENS,NTENS),AUXM2(NTENS,NTENS),
+ AUXM3(NTENS,NTENS),AUXM4(NTENS,NTENS),V1(6)
C
C FIND DL
HC=- (DPHIDF*HH+DPHIDW1*H1+DPHIDW2*H2+DPHIDEP*H3)
CALL KCOPY(CEL,AUXM1,NTENS*NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  AUXM1(I,J)=AUXM1(I,J)*2.D0
ENDDO
ENDDO
CALL KMULT(AUXM1,AN,V1,NTENS,NTENS,1)

```

```

        CALL KAB (AN,V1,AL,NTENS,NDI)
        AL=HC+AL
C*** RECALL THAT 'DE' HAS ENGG SHEAR COMPONENTS
        DL=0.D0
        DO I=1,NTENS
            DL=DL+V1 (I) *DE (I)
        ENDDO
        DL=DL/AL
C
        RETURN
        END
C
C*****
C
        SUBROUTINE KUPDATE (DL,STRESS,EBAR,F,W1,W2,AN1,AN2,AN3,STRESST,
+ EBART,FT,W1T,W2T,AN1T,AN2T,AN3T,SEL,CEL,NTENS,NDI,AN,GG,G1,G2,
+ HH,H1,H2,H3,OMEGA,V,DEVOID,DEP,R1,R2,R31,R32,ANUCL,IWR,IOUT)
C
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
        DIMENSION STRESS (NTENS),AN1 (3),AN2 (3),AN3 (3),STRESST (NTENS),
+ AN1T (3),AN2T (3),AN3T (3),SEL (1),CEL (NTENS,NTENS),AN (1),OMEGA (3,3),
+ V (3,3),DEVOID (NTENS),DEP (NTENS),R1 (NTENS),R2 (NTENS),R31 (NTENS),
+ R32 (NTENS)
        DIMENSION ROT (3,3),V1 (6),V2 (6),V3 (6)
C
        DLTOL=1.D-8
        IF (DABS (DL) .LT. DLTOL) THEN
            CALL KCOPY (SEL,STRESS,NTENS)
            EBAR=EBART
            F=FT
            W1=W1T
            W2=W2T
            CALL KCOPY (AN1T,AN1,3)
            CALL KCOPY (AN2T,AN2,3)
            CALL KCOPY (AN3T,AN3,3)
            RETURN
        END IF
C
C*** UPDATE STRESSES
        CALL KCOPY (DEP,V1,NTENS)
        DO I=NDI+1,NTENS
            V1 (I)=V1 (I) *2.D0
        ENDDO
        CALL KMULT (CEL,V1,V2,NTENS,NTENS,1)
        DO I=1,NTENS
            STRESS (I)=SEL (I) -V2 (I) +DL*R1 (I)
        ENDDO
C
C*** UPDATE EBAR
        CALL KAB (R2,DEP,DEBAR,NTENS,NDI)
        EBAR=EBART+DEBAR
C
C*** UPDATE F
        DEPKK=DEP (1) +DEP (2) +DEP (3)
        DF= (1.D0-FT) *DEPKK+ANUCL*DEBAR
        F=FT+DF
C
C*** UPDATE W1, W2, AN1, AN2, AN3
        WTOL=1.D-2
        IF (DABS (W1T-W2T) .LT. WTOL) GOTO 10

```

```

C
C W1T different from W2T
C
      CALL KAB (R31, DEP, DW1, NTENS, NDI)
      CALL KAB (R32, DEP, DW2, NTENS, NDI)
      W1=W1T+DW1
      W2=W2T+DW2
C
C ROT=EXP(-OMEGA*DL)
      OM=-OMEGA (2, 1) *DL
      ZS=DSIN (OM)
      ZC=DCOS (OM)
      CALL KINITIA (ROT, 3*3)
      ROT (1, 1)=ZC
      ROT (2, 2)=ZC
      ROT (1, 2)=-ZS
      ROT (2, 1)=ZS
      ROT (3, 3)=1.D0
C
      CALL KMULT (ROT, AN1T, AN1, 3, 3, 1)
      CALL KMULT (ROT, AN2T, AN2, 3, 3, 1)
      CALL KMULT (ROT, AN3T, AN3, 3, 3, 1)
      GOTO 999
C
10 CONTINUE
C |W1T-W2T|<1.D-2 needs special treatment
C
      IF (W1T.NE.W2T) THEN
        CALL KAB (R31, DEP, DW1, NTENS, NDI)
        CALL KAB (R32, DEP, DW2, NTENS, NDI)
        W1=W1T+DW1
        W2=W2T+DW2
        CALL KCOPY (AN1T, AN1, 3)
        CALL KCOPY (AN2T, AN2, 3)
        CALL KCOPY (AN3T, AN3, 3)
      END IF
C
      IF (W1T.EQ.W2T) THEN
C
C Update AN1, AN2 and AN3
      CALL KINITIA (AN1, 3)
      CALL KINITIA (AN2, 3)
      CALL KINITIA (AN3, 3)
      AN3 (3)=1.D0
      ZX=1.D-6*DSQRT (DEVOID (1) **2+DEVOID (2) **2+DEVOID (3) **2) /3.D0
      IF (DABS (DEVOID (4)) .LT. ZX) THEN
        DAL1=DEVOID (1)
        DAL2=DEVOID (2)
        AN1 (1)=1.D0
      ELSE
        R=DSQRT ( ( (DEVOID (1) -DEVOID (2)) /2.D0) **2+DEVOID (4) **2)
        AM= (DEVOID (1) +DEVOID (2)) /2.D0
        DAL1=AM+R
        DAL2=AM-R
        AN1 (1)=- (DEVOID (2) -DAL1) /DEVOID (4)
        AN1 (2)=1.D0
        ZZ=DSQRT (AN1 (1) **2+AN1 (2) **2)
        AN1 (1)=AN1 (1) /ZZ
        AN1 (2)=AN1 (2) /ZZ
      END IF
      AN2 (1)=-AN1 (2)

```

```

      AN2 (2) = AN1 (1)
C
C      Update W1 and W2
      DE11 = DEVOID (1) * AN1 (1) ** 2 + DEVOID (2) * AN1 (2) ** 2 +
+      2.0 * DEVOID (4) * AN1 (1) * AN1 (2)
      DE22 = DEVOID (1) * AN2 (1) ** 2 + DEVOID (2) * AN2 (2) ** 2 +
+      2.0 * DEVOID (4) * AN2 (1) * AN2 (2)
      DE33 = DEVOID (3)
      W1 = W1T * DEXP (DE33 - DE11)
      W2 = W2T * DEXP (DE33 - DE22)
C      W1 = W1T + W1T * (DE33 - DE11)
C      W2 = W2T + W2T * (DE33 - DE22)
      END IF
C
C
999  CONTINUE
C
      RETURN
1001 FORMAT (1P8E13.5)
      END
C
C*****
C
      SUBROUTINE KDFF (DPHIDF, DPHIDW1, DPHIDW2, AKMX, F, STRESS, DQDW1, DQDW2,
+      AUXM1, AUXM2, AUXM3, AUXM4, NTENS, NDI, G, W1, W2, AN1, AN2, AN3, QINV)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      DIMENSION AKMX (NTENS, NTENS), STRESS (NTENS), DQDW1 (NTENS, NTENS),
+      DQDW2 (NTENS, NTENS), AUXM1 (NTENS, NTENS), AUXM2 (NTENS, NTENS),
+      AUXM3 (NTENS, NTENS), AUXM4 (NTENS, NTENS), VEC1 (6),
+      AN1 (3), AN2 (3), AN3 (3), QINV (NTENS, NTENS)
C
      CALL KCOPY (QINV, AUXM4, NTENS * NTENS)
C*** AUXM4 = QINV
C
      AUX = 3.0 * G * (1.0 + F) / (1.0 - F)
      DO 10 I = 1, NTENS
      DO 10 J = 1, NTENS
10      AUXM1 (I, J) = 1.5 * D0 * AKMX (I, J) + AUX * AUXM4 (I, J)
      DO 20 I = 1, NTENS
      DO 20 J = NDI + 1, NTENS
20      AUXM1 (I, J) = AUXM1 (I, J) * 2.0
      CALL KMULT (AUXM1, STRESS, VEC1, NTENS, NTENS, 1)
      CALL KAB (STRESS, VEC1, DPHIDF, NTENS, NDI)
      DPHIDF = DPHIDF / (1.0 - F) ** 2
C
      CALL KCOPY (DQDW1, AUXM1, NTENS * NTENS)
      DO 110 I = 1, NTENS
      DO 110 J = NDI + 1, NTENS
110      AUXM1 (I, J) = AUXM1 (I, J) * 2.0
      CALL KMULT (AUXM1, AUXM4, AUXM2, NTENS, NTENS, NTENS)
      CALL KCOPY (AUXM4, AUXM1, NTENS * NTENS)
      DO 120 I = 1, NTENS
      DO 120 J = NDI + 1, NTENS
120      AUXM1 (I, J) = AUXM1 (I, J) * 2.0
      CALL KMULT (AUXM1, AUXM2, AUXM3, NTENS, NTENS, NTENS)
      DO 130 I = 1, NTENS
      DO 130 J = 1, NTENS
130      AUXM3 (I, J) = -AUXM3 (I, J)
      CALL KCOPY (AUXM3, AUXM1, NTENS * NTENS)

```



```

DO 140 I=1,NTENS
DO 140 J=NDI+1,NTENS
140   AUXM1 (I, J)=AUXM1 (I, J)*2.D0
      CALL KMULT (AUXM1, STRESS, VEC1, NTENS, NTENS, 1)
      CALL KAB (STRESS, VEC1, DPHIDW1, NTENS, NDI)
      DPHIDW1=DPHIDW1*3.D0*G*F/(1.D0-F)**2
C
      CALL KCOPY (DQDN2, AUXM1, NTENS*NTENS)
      DO 210 I=1,NTENS
      DO 210 J=NDI+1,NTENS
210   AUXM1 (I, J)=AUXM1 (I, J)*2.D0
      CALL KMULT (AUXM1, AUXM4, AUXM2, NTENS, NTENS, NTENS)
      CALL KCOPY (AUXM4, AUXM1, NTENS*NTENS)
      DO 220 I=1,NTENS
      DO 220 J=NDI+1,NTENS
220   AUXM1 (I, J)=AUXM1 (I, J)*2.D0
      CALL KMULT (AUXM1, AUXM2, AUXM3, NTENS, NTENS, NTENS)
      DO 230 I=1,NTENS
      DO 230 J=1,NTENS
230   AUXM3 (I, J)=-AUXM3 (I, J)
      CALL KCOPY (AUXM3, AUXM1, NTENS*NTENS)
      DO 240 I=1,NTENS
      DO 240 J=NDI+1,NTENS
240   AUXM1 (I, J)=AUXM1 (I, J)*2.D0
      CALL KMULT (AUXM1, STRESS, VEC1, NTENS, NTENS, 1)
      CALL KAB (STRESS, VEC1, DPHIDW2, NTENS, NDI)
      DPHIDW2=DPHIDW2*3.D0*G*F/(1.D0-F)**2
C
      RETURN
      END
C
C*****
C
      SUBROUTINE KDFDN (DPHIDN1, DPHIDN2, DPHIDN3, AKMX, F, STRESS, DQDN1,
+ DQDN2, DQDN3, AUXM1, AUXM2, AUXM3, AUXM4, NTENS, NDI, G, W1, W2, AN1, AN2,
+ AN3, QINV)
C
      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
      DIMENSION DPHIDN1 (3), DPHIDN2 (3), DPHIDN3 (3),
+ AKMX (NTENS, NTENS), STRESS (NTENS), DQDN1 (NTENS, NTENS, 3),
+ DQDN2 (NTENS, NTENS, 3), DQDN3 (NTENS, NTENS, 3),
+ AUXM1 (NTENS, NTENS), AUXM2 (NTENS, NTENS),
+ AUXM3 (NTENS, NTENS), AUXM4 (NTENS, NTENS), VEC1 (6),
+ AN1 (3), AN2 (3), AN3 (3), QINV (NTENS, NTENS)
C
      CALL KCOPY (QINV, AUXM4, NTENS*NTENS)
C*** AUXM4=QINV
C
      AUX=3.D0*G*F/(1.D0-F)**2
      DO 1000 K=1, 3
          CALL KCOPY (DQDN1 (1, 1, K), AUXM1, NTENS*NTENS)
          DO I=1,NTENS
          DO J=NDI+1,NTENS
              AUXM1 (I, J)=AUXM1 (I, J)*2.D0
          END DO
          END DO
          CALL KMULT (AUXM1, AUXM4, AUXM2, NTENS, NTENS, NTENS)
          CALL KCOPY (AUXM4, AUXM1, NTENS*NTENS)
          DO I=1,NTENS
          DO J=NDI+1,NTENS

```

```

      AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , AUXM2 , AUXM3 , NTENS , NTENS , NTENS )
DO I=1 , NTENS
DO J=1 , NTENS
  AUXM3 ( I, J ) =-AUXM3 ( I, J )
END DO
END DO
CALL KCOPY (AUXM3 , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , STRESS , VEC1 , NTENS , NTENS , 1 )
CALL KAB (STRESS , VEC1 , DPHIDN1 ( K ) , NTENS , NDI )
DPHIDN1 ( K ) =DPHIDN1 ( K ) *AUX
C
CALL KCOPY (DQDN2 ( 1, 1, K ) , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , AUXM4 , AUXM2 , NTENS , NTENS , NTENS )
CALL KCOPY (AUXM4 , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , AUXM2 , AUXM3 , NTENS , NTENS , NTENS )
DO I=1 , NTENS
DO J=1 , NTENS
  AUXM3 ( I, J ) =-AUXM3 ( I, J )
END DO
END DO
CALL KCOPY (AUXM3 , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , STRESS , VEC1 , NTENS , NTENS , 1 )
CALL KAB (STRESS , VEC1 , DPHIDN2 ( K ) , NTENS , NDI )
DPHIDN2 ( K ) =DPHIDN2 ( K ) *AUX
C
CALL KCOPY (DQDN3 ( 1, 1, K ) , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO
CALL KMULT (AUXM1 , AUXM4 , AUXM2 , NTENS , NTENS , NTENS )
CALL KCOPY (AUXM4 , AUXM1 , NTENS*NTENS )
DO I=1 , NTENS
DO J=NDI+1 , NTENS
  AUXM1 ( I, J ) =AUXM1 ( I, J ) *2 .DO
END DO
END DO

```

```

CALL KMULT (AUXM1, AUXM2, AUXM3, NTENS, NTENS, NTENS)
DO I=1, NTENS
DO J=1, NTENS
  AUXM3 (I, J) = -AUXM3 (I, J)
END DO
END DO
CALL KCOPY (AUXM3, AUXM1, NTENS*NTENS)
DO I=1, NTENS
DO J=NDI+1, NTENS
  AUXM1 (I, J) = AUXM1 (I, J) * 2. D0
END DO
END DO
CALL KMULT (AUXM1, STRESS, VEC1, NTENS, NTENS, 1)
CALL KAB (STRESS, VEC1, DPHIDN3 (K), NTENS, NDI)
DPHIDN3 (K) = DPHIDN1 (K) * AUX
1000 CONTINUE
C
  RETURN
END
C
C*****
C
  SUBROUTINE KPLJAC (DDSDDE, CEL, STRESS, EBAR, F, W1, W2, AN1, AN2, AN3,
+ X, W, NGAUS, S, P, Q, QINV, AMTLD, DQDW1, DQDW2, AIMX, AJMX, AKMX,
+ AMX, CMX, AN, QMX, AUXM1, AUXM2, AUXM3, AUXM4, OMEGA,
+ NTENS, NDI, ANU, G, AK, EXPO, E0, SIG0, FN, EN, SN, IOUT, IWR)
C
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
  DIMENSION DDSDDE (NTENS, NTENS), STRESS (1), OMEGA (3, 3)
C
  DIMENSION X (1), W (1),
+ S (NTENS, NTENS), P (NTENS, NTENS), Q (NTENS, NTENS), QINV (NTENS, NTENS),
+ AN1 (3), AN2 (3), AN3 (3), CEL (NTENS, NTENS), AMTLD (NTENS, NTENS),
+ DQDW1 (NTENS, NTENS), DQDW2 (NTENS, NTENS),
+ AIMX (NTENS, NTENS), AJMX (NTENS, NTENS), AKMX (NTENS, NTENS),
+ AMX (NTENS, NTENS), CMX (NTENS, NTENS), AN (NTENS), QMX (NTENS, NTENS),
+ AUXM1 (NTENS, NTENS), AUXM2 (NTENS, NTENS), AUXM3 (NTENS, NTENS),
+ AUXM4 (NTENS, NTENS)
C
  DIMENSION V1 (6), V2 (6), V3 (6)
C
  COMMON/KHARD/HC
C
  IF (IWR.NE.0) WRITE (IOUT, *)
+ '*** CALCULATION OF PLASTIC TANGENT MATRIX (zpljac)'
C
  PI=4. D0*DATAN (1. D0)
  CALL KESH_Q (Q, ANU, W1, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)
  CALL KINV (Q, QINV, NTENS)
  DO I=1, NTENS
  DO J=NDI+1, NTENS
    QINV (I, J) = QINV (I, J) / 2. D0
  ENDDO
  ENDDO
  DO I=NDI+1, NTENS
  DO J=1, NTENS
    QINV (I, J) = QINV (I, J) / 2. D0
  ENDDO
  ENDDO
C

```

```

Z1=2.D0*G
Z2=3.D0*AK
Z3=F/(1.D0-F)
DO I=1,NTENS
DO J=1,NTENS
  CEL(I,J)=AKMX(I,J)/Z1+AJMX(I,J)/Z2+QINV(I,J)*Z3
ENDDO
ENDDO
CALL KCOPY(CEL,AUXM1,NTENS*NTENS)
CALL KINV(AUXM1,CEL,NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  CEL(I,J)=CEL(I,J)/2.D0
ENDDO
ENDDO
DO I=NDI+1,NTENS
DO J=1,NTENS
  CEL(I,J)=CEL(I,J)/2.D0
ENDDO
ENDDO
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'CEL'
  DO I=1,NTENS
    WRITE(IOUT,1001) (CEL(I,J),J=1,NTENS)
  ENDDO
END IF
C
C FIND N
CALL KFINDN(AN,STRESS,F,AMTLD,AUXM1,NTENS,NDI)
C
IF (IWR.NE.0) WRITE(IOUT,*) 'AN'
IF (IWR.NE.0) WRITE(IOUT,1001) (AN(I),I=1,NTENS)
C
C FIND DQ/DW1, DQ/DW2
POISSON=0.5D0
CALL KDQ(DQDW1,DQDW2,POISSON,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
C
IF (IWR.NE.0) THEN
  WRITE(IOUT,*) 'DQDW1'
  DO I=1,NTENS
    WRITE(IOUT,1001) (DQDW1(I,J),J=1,NTENS)
  ENDDO
  WRITE(IOUT,*) 'DQDW2'
  DO I=1,NTENS
    WRITE(IOUT,1001) (DQDW2(I,J),J=1,NTENS)
  ENDDO
END IF
C
C FIND DPHI/DF, DPHI/DW1, DPHI/DW2
C
CALL KESH_Q(Q,POISSON,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
CALL KINV(Q,QINV,NTENS)
DO I=1,NTENS
DO J=NDI+1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0
ENDDO
ENDDO
DO I=NDI+1,NTENS
DO J=1,NTENS
  QINV(I,J)=QINV(I,J)/2.D0

```

```

        ENDDO
        ENDDO
        CALL KDFF (DPHIDF,DPHIDW1,DPHIDW2,AKMX,F,STRESS,DQDW1,DQDW2,
+ AUXM1,AUXM2,AUXM3,AUXM4,NTENS,NDI,G,W1,W2,AN1,AN2,AN3,QINV)
C
C FIND DPHI/DEBAR
        CALL KYCURVE (YIELD,H,EBAR,EXPO,E0,SIG0)
        DPHIDEP=-2.D0*YIELD*H
C
        IF (IWR.NE.0) THEN
            WRITE (IOUT,*) 'DPHIDF,DPHIDW1,DPHIDW2,DPHIDEP'
            WRITE (IOUT,1001) DPHIDF,DPHIDW1,DPHIDW2,DPHIDEP
        END IF
C
C FIND A, C
        CALL KAC (AMX,CMX,F,W1,W2,AN1,AN2,AN3,S,P,AUXM1,AUXM2,
+ X,W,NGAUS,NTENS,NDI,AIMX)
C
C FIND HH, H1, H2, H3, GG, G1, G2
        GG=0.D0
        DO I=1,NDI
            GG=GG+AN(I)
        ENDDO
C
        CALL KCOPY (AN,V1,NTENS)
        DO I=NDI+1,NTENS
            V1(I)=V1(I)*2.D0
        END DO
        CALL KMULT (AMX,V1,V2,NTENS,NTENS,1)
C
        V3(1)=AN3(1)**2-AN1(1)**2
        V3(2)=AN3(2)**2-AN1(2)**2
        V3(3)=AN3(3)**2-AN1(3)**2
        V3(4)=AN3(1)*AN3(2)-AN1(1)*AN1(2)
        CALL KAB (V3,V2,G1,NTENS,NDI)
        V3(1)=AN3(1)**2-AN2(1)**2
        V3(2)=AN3(2)**2-AN2(2)**2
        V3(3)=AN3(3)**2-AN2(3)**2
        V3(4)=AN3(1)*AN3(2)-AN2(1)*AN2(2)
        CALL KAB (V3,V2,G2,NTENS,NDI)
C
        HH=GG*(1.D0-F)
        H1=G1*W1
        H2=G2*W2
        CALL KYCURVE (YIELD,H,EBAR,EXPO,E0,SIG0)
        CALL KAB (STRESS,AN,SUM,NTENS,NDI)
        H3=SUM/((1.D0-F)*YIELD)
        ANUCL=0.D0
        IF (FN.NE.0.D0)
+ ANUCL=(FN/(SN*DSQRT(2.D0*PI)))*DEXP(-0.5D0*((EBAR-EN)/SN)**2)
        HH=HH+ANUCL*H3
C
C FIND OMEGA
        OM=0.D0
        WTOL=1.D-2
        THETA=DATAN2 (AN1(2),AN1(1))
        IF (DABS (W1-W2) .GE. WTOL) THEN
            T12=CMX(4,1)*AN(1)+CMX(4,2)*AN(2)+CMX(4,3)*AN(3)+
+ 2.D0*CMX(4,4)*AN(4)
            B21=(W2*W2+W1*W1)/(W2*W2-W1*W1)
            V3(1)=-DSIN(2.D0*THETA)

```

```

V3(2)=-V3(1)
V3(3)=0.D0
V3(4)=DCOS(2.D0*THETA)
CALL KAB(V3,V2,AUX,NTENS,NDI)
OM=-T12-0.5D0*B21*AUX
END IF
CALL KINITIA(OMEGA,3*3)
OMEGA(1,2)=-OM
OMEGA(2,1)=OM
C
IF (IWR.NE.0) THEN
WRITE(IOUT,*) 'A'
DO I=1,NTENS
WRITE(IOUT,1001) (AMX(I,J),J=1,NTENS)
ENDDO
WRITE(IOUT,*) 'C'
DO I=1,NTENS
WRITE(IOUT,1001) (CMX(I,J),J=1,NTENS)
ENDDO
WRITE(IOUT,*) 'OMEGA'
DO I=1,3
WRITE(IOUT,1001) (OMEGA(I,J),J=1,3)
ENDDO
WRITE(IOUT,*) 'HH, H1, H2, H3'
WRITE(IOUT,1001) HH,H1,H2,H3
WRITE(IOUT,*) 'GG, G1, G2'
WRITE(IOUT,1001) GG,G1,G2
END IF
C
C FIND AL
HC=- (DPHIDF*HH+DPHIDW1*H1+DPHIDW2*H2+DPHIDEP*H3)
C
CALL KCOPY(AN,V2,NTENS)
DO I=NDI+1,NTENS
V2(I)=V2(I)*2.D0
ENDDO
CALL KMULT(CEL,V2,V1,NTENS,NTENS,1)
C*** V1=C^(e1):N=N:C^(e1)
CALL KAB(AN,V1,AL,NTENS,NDI)
AL=HC+AL
IF (IWR.NE.0) THEN
WRITE(IOUT,*) 'HC, AL'
WRITE(IOUT,1001) HC,AL
END IF
C
W3=OMEGA(2,1)
V2(1)=2.D0*STRESS(4)*W3
V2(2)=-V2(1)
V2(3)=0.D0
V2(4)=(STRESS(2)-STRESS(1))*W3
IF (IWR.NE.0) THEN
WRITE(IOUT,*) 'S.OM-OM.S'
WRITE(IOUT,1001) (V2(I),I=1,NTENS)
END IF
C
DO I=1,NTENS
V3(I)=(V1(I)-V2(I))/AL
ENDDO
C
DO I=1,NTENS

```

```

DO J=1,NTENS
  DDSDE (I, J)=CEL (I, J)-V3 (I)*V1 (J)
ENDDO
ENDDO
C
IF (IWR.NE.0) THEN
  WRITE (IOUT,*) 'PLASTIC DDSDE'
  DO I=1,NTENS
    WRITE (IOUT,1001) (DDSDE (I, J), J=1,NTENS)
  ENDDO
END IF
C
RETURN
1001 FORMAT (1P8E13.5)
1002 FORMAT (10I5)
END
C
C*****
C
SUBROUTINE KKPLJAC (DDSDE, CEL, STRESS, EBAR, F, W1, W2, AN1, AN2, AN3,
+ X, W, NGAUS, S, P, Q, QINV, AMTLD, DQDW1, DQDW2, AIMX, AJMX, AKMX,
+ AMX, CMX, AN, QMX, AUXM1, AUXM2, AUXM3, AUXM4, OMEGA,
+ NTENS, NDI, ANU, G, AK, EXPO, E0, SIG0, IOUT, IWR,
+ CELT, ANT, HHT, H1T, H2T, H3T, OMEGAT, STRESST)
C
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
DIMENSION DDSDE (NTENS, NTENS), STRESS (1), OMEGA (3, 3), OMEGAT (3, 3)
C
DIMENSION X (1), W (1),
+ S (NTENS, NTENS), P (NTENS, NTENS), Q (NTENS, NTENS), QINV (NTENS, NTENS),
+ AN1 (3), AN2 (3), AN3 (3), CEL (NTENS, NTENS), AMTLD (NTENS, NTENS),
+ DQDW1 (NTENS, NTENS), DQDW2 (NTENS, NTENS),
+ AIMX (NTENS, NTENS), AJMX (NTENS, NTENS), AKMX (NTENS, NTENS),
+ AMX (NTENS, NTENS), CMX (NTENS, NTENS), AN (NTENS), QMX (NTENS, NTENS),
+ AUXM1 (NTENS, NTENS), AUXM2 (NTENS, NTENS), AUXM3 (NTENS, NTENS),
+ AUXM4 (NTENS, NTENS), CELT (NTENS, NTENS), ANT (NTENS), STRESST (NTENS)
C
DIMENSION V1 (6), V2 (6), V3 (6), DETENS (3, 3), DE (6), V1T (6)
C
IF (IWR.NE.0) WRITE (IOUT,*)
+ '*** CALCULATION OF PLASTIC JACOBIAN (kkpljac)'
C
C
C FIND N AT T+DT
CALL KFINDN (AN, STRESS, F, AMTLD, AUXM1, NTENS, NDI)
C
IF (IWR.NE.0) WRITE (IOUT,*) 'AN AT T+DT'
IF (IWR.NE.0) WRITE (IOUT,1001) (AN (I), I=1, NTENS)
C
C FIND DQ/DW1, DQ/DW2 AT T+DT
POISSON=0.5D0
CALL KDQ (DQDW1, DQDW2, POISSON, W1, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)
C
IF (IWR.NE.0) THEN
  WRITE (IOUT,*) 'DQDW1 AT T+DT'
  DO I=1,NTENS
    WRITE (IOUT,1001) (DQDW1 (I, J), J=1,NTENS)
  ENDDO
  WRITE (IOUT,*) 'DQDW2 AT T+DT'
  DO I=1,NTENS

```

```

        WRITE (IOUT,1001) (DQDW2 (I, J), J=1, NTENS)
    ENDDO
END IF
C
C FIND DPHI/DF, DPHI/DW1, DPHI/DW2 AT T+DT
C
    POISSON=0.5D0
    CALL KESH_Q(Q, POISSON, W1, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)
    CALL KINV(Q, QINV, NTENS)
    DO I=1, NTENS
    DO J=NDI+1, NTENS
        QINV(I, J)=QINV(I, J)/2.D0
    ENDDO
    ENDDO
    DO I=NDI+1, NTENS
    DO J=1, NTENS
        QINV(I, J)=QINV(I, J)/2.D0
    ENDDO
    ENDDO
    CALL KDFE(DPHIDF, DPHIDW1, DPHIDW2, AKMX, F, STRESS, DQDW1, DQDW2,
+ AUXM1, AUXM2, AUXM3, AUXM4, NTENS, NDI, G, W1, W2, AN1, AN2, AN3, QINV)
C
C FIND DPHI/DEBAR AT T+DT
    CALL KYCURVE(YIELD, H, EBAR, EXPO, E0, SIG0)
    DPHIDEP=-2.D0*YIELD*H
C
    IF (IWR.NE.0) THEN
        WRITE (IOUT, *) 'DPHIDF, DPHIDW1, DPHIDW2, DPHIDEP AT T+DT'
        WRITE (IOUT, 1001) DPHIDF, DPHIDW1, DPHIDW2, DPHIDEP
    END IF
C
C FIND AL
    HC=- (DPHIDF*HHT+DPHIDW1*H1T+DPHIDW2*H2T+DPHIDEP*H3T)
C
    CALL KCOPY(ANT, V2, NTENS)
    DO I=NDI+1, NTENS
        V2(I)=V2(I)*2.D0
    ENDDO
    CALL KMULT(CELT, V2, V1T, NTENS, NTENS, 1)
C*** V1T=CELT:NT=NT:CT
C
    CALL KCOPY(AN, V2, NTENS)
    DO I=NDI+1, NTENS
        V2(I)=V2(I)*2.D0
    ENDDO
    CALL KMULT(CELT, V2, V1, NTENS, NTENS, 1)
C*** V1=CELT:N=N:CELT
C
    CALL KAB(AN, V1T, AL, NTENS, NDI)
    AL=HC+AL
    IF (IWR.NE.0) THEN
        WRITE (IOUT, *) 'HC, AL'
        WRITE (IOUT, 1001) HC, AL
    END IF
C
    W3=OMEGAT(2, 1)
    V2(1)=2.D0*STRESST(4)*W3
    V2(2)=-V2(1)
    V2(3)=0.D0
    V2(4)=(STRESST(2)-STRESST(1))*W3
    IF (IWR.NE.0) THEN

```



```

        WRITE(IOUT,*) 'S.OM-OM.S'
        WRITE(IOUT,1001) (V2(I),I=1,NTENS)
END IF
C
DO I=1,NTENS
    V3(I)=(V1T(I)-V2(I))/AL
ENDDO
C
DO I=1,NTENS
DO J=1,NTENS
    DDSDE(I,J)=CELT(I,J)-V3(I)*V1(J)
ENDDO
ENDDO
C
IF (IWR.NE.0) THEN
    WRITE(IOUT,*) 'PLASTIC DDSDE FROM KKPLJAC'
    DO I=1,NTENS
        WRITE(IOUT,1001) (DDSDE(I,J),J=1,NTENS)
    ENDDO
END IF
C
RETURN
1001 FORMAT(1P8E13.5)
1002 FORMAT(10I5)
END
C
C*****
C
SUBROUTINE KESH_S(S,ANU,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION X(1),W(1),S(NTENS,NTENS),AN1(3),AN2(3),AN3(3),Q(3,3),
+ SS(6,6)
C
DIMENSION QMX(4,4),AUXM1(4,4),AUXM2(4,4)
C
DIMENSION AJMX(4,4),AKMX(4,4)
COMMON/KNICK/AJMX,AKMX
C
CALL KINITIA(S,NTENS*NTENS)
CALL KINITIA(SS,6*6)
C
C1=1.D0
A1=C1/W1
B1=C1/W2
PI=4.D0*DATAN(1.D0)
C
CALL KORDER(A,B,C,IA,IB,IC,A1,B1,C1)
C
CALL KIAB(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,B,C,
+ IA,IB,IC,X,W,NGAUS)
C
QQ=3.D0/(8.D0*PI*(1.D0-ANU))
R=(1.D0-2.D0*ANU)/(8.D0*PI*(1.D0-ANU))
C
SS(1,1)=QQ*A*A*AI11+R*AI1
SS(2,2)=QQ*B*B*AI22+R*AI2
SS(3,3)=QQ*C*C*AI33+R*AI3
C
SS(1,2)=QQ*B*B*AI12-R*AI1

```

```

SS (1,3)=QQ*C*C*AI13-R*AI1
SS (2,1)=QQ*A*A*AI12-R*AI2
SS (2,3)=QQ*C*C*AI23-R*AI2
SS (3,1)=QQ*A*A*AI13-R*AI3
SS (3,2)=QQ*B*B*AI23-R*AI3
C
SS (4,4)=0.5D0*QQ*(A*A+B*B)*AI12+0.5D0*R*(AI1+AI2)
SS (5,5)=0.5D0*QQ*(A*A+C*C)*AI13+0.5D0*R*(AI1+AI3)
SS (6,6)=0.5D0*QQ*(B*B+C*C)*AI23+0.5D0*R*(AI2+AI3)
C
CALL KARRANGE(S,SS,A1,B1,C1,NTENS)
C
C*** ROTATION
C
DO I=1,3
  Q(I,1)=AN1(I)
  Q(I,2)=AN2(I)
  Q(I,3)=AN3(I)
END DO
C
CALL KINITIA(QMX,NTENS*NTENS)
DO I=1,3
DO J=1,3
  QMX(I,J)=Q(I,J)**2
END DO
END DO
C
QMX(1,4)=Q(1,1)*Q(1,2)
QMX(2,4)=Q(2,1)*Q(2,2)
QMX(3,4)=Q(3,1)*Q(3,2)
C
QMX(4,1)=Q(1,1)*Q(2,1)
QMX(4,2)=Q(1,2)*Q(2,2)
QMX(4,3)=Q(1,3)*Q(2,3)
C
QMX(4,4)=Q(1,1)*Q(2,2)
QMX55=Q(1,2)*Q(2,1)
C
CALL KCOPY(QMX,AUXM1,NTENS*NTENS)
AUXM1(1,4)=2.D0*AUXM1(1,4)
AUXM1(2,4)=2.D0*AUXM1(2,4)
AUXM1(4,4)=AUXM1(4,4)+QMX55
CALL KMULT(AUXM1,S,AUXM2,NTENS,NTENS,NTENS)
CALL KTRANPOSE(AUXM1,S,NTENS,NTENS)
CALL KMULT(AUXM2,S,AUXM1,NTENS,NTENS,NTENS)
CALL KCOPY(AUXM1,S,NTENS*NTENS)
C
RETURN
END
C
C*****
C
SUBROUTINE KESH_PI(P,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION X(1),W(1),P(NTENS,NTENS),AN1(3),AN2(3),AN3(3),Q(3,3)
C
DIMENSION QMX(4,4)
C
DIMENSION AJMX(4,4),AKMX(4,4)

```

```

COMMON/KNICK/AJMX, AKMX
C
CALL KINITIA(P, NTENS*NTENS)
C
C1=1.D0
A1=C1/W1
B1=C1/W2
PI=4.D0*DATAN(1.D0)
C
IF (DABS(A1-B1).LT.0.01D0.AND.DABS(B1-C1).LT.0.01D0) RETURN
C
CALL KORDER(A, B, C, IA, IB, IC, A1, B1, C1)
C
CALL KIAB(AI1, AI2, AI3, AI11, AI22, AI33, AI12, AI13, AI23, A, B, C,
+ IA, IB, IC, X, W, NGAUS)
C
IF (A1.GE.B1.AND.B1.GE.C1) P(4,4)=(AI2-AI1)/(8.D0*PI)
IF (A1.GE.C1.AND.C1.GE.B1) P(4,4)=(AI3-AI1)/(8.D0*PI)
IF (C1.GE.A1.AND.A1.GE.B1) P(4,4)=(AI3-AI2)/(8.D0*PI)
IF (B1.GE.A1.AND.A1.GE.C1) P(4,4)=(AI1-AI2)/(8.D0*PI)
IF (B1.GE.C1.AND.C1.GE.A1) P(4,4)=(AI1-AI3)/(8.D0*PI)
IF (C1.GE.B1.AND.B1.GE.A1) P(4,4)=(AI2-AI3)/(8.D0*PI)
C
C*** ROTATION
C
DO I=1,3
  Q(I,1)=AN1(I)
  Q(I,2)=AN2(I)
  Q(I,3)=AN3(I)
END DO
C
CALL KINITIA(QMX, NTENS*NTENS)
DO I=1,3
DO J=1,3
  QMX(I,J)=Q(I,J)**2
END DO
END DO
C
QMX(1,4)=Q(1,1)*Q(1,2)
QMX(2,4)=Q(2,1)*Q(2,2)
QMX(3,4)=Q(3,1)*Q(3,2)
C
QMX(4,1)=Q(1,1)*Q(2,1)
QMX(4,2)=Q(1,2)*Q(2,2)
QMX(4,3)=Q(1,3)*Q(2,3)
C
QMX(4,4)=Q(1,1)*Q(2,2)
QMX55=Q(1,2)*Q(2,1)
C
A4=P(4,4)
P(4,1)=2.D0*A4*QMX(1,4)*(QMX(4,4)-QMX55)
P(4,2)=2.D0*A4*QMX(2,4)*(QMX(4,4)-QMX55)
P(4,3)=0.D0
P(4,4)=A4*(QMX(4,4)-QMX55)*(QMX(4,4)+QMX55)
C
RETURN
END
C
C*****
C
SUBROUTINE KESH_Q(Q, ANU, W1, W2, AN1, AN2, AN3, X, W, NGAUS, NTENS, G)

```

```

C
  IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
  DIMENSION Q(NTENS,NTENS),AN1(3),AN2(3),AN3(3),X(1),W(1),QQ(3,3),
+ SS(6,6)
C
  DIMENSION QMX(4,4),AUXM1(4,4),AUXM2(4,4)
C
  DIMENSION AJMX(4,4),AKMX(4,4)
  COMMON/KNICK/AJMX,AKMX
C
  CALL KINITIA(Q,NTENS*NTENS)
  CALL KINITIA(SS,6*6)
C
  C1=1.D0
  A1=C1/W1
  B1=C1/W2
  PI=4.D0*DATAN(1.D0)
C
  CALL KORDER(A,B,C,IA,IB,IC,A1,B1,C1)
C
  CALL KIAB(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,B,C,
+ IA,IB,IC,X,W,NGAUS)
C
  SS(1,1)=(G/(2.D0*PI*(1.D0-ANU)))*(4.D0*PI-0.5D0*AI1-
+ 1.5D0*A*A*AI11)
  SS(2,2)=(G/(2.D0*PI*(1.D0-ANU)))*(4.D0*PI-0.5D0*AI2-
+ 1.5D0*B*B*AI22)
  SS(3,3)=(G/(2.D0*PI*(1.D0-ANU)))*(4.D0*PI-0.5D0*AI3-
+ 1.5D0*C*C*AI33)
C
  SS(1,2)=(G/(8.D0*PI*(1.D0-ANU)))*(16.D0*PI*ANU+(1.D0-4.D0*ANU)*
+ (AI1+AI2)-3.D0*(A*A+B*B)*AI12)
  SS(1,3)=(G/(8.D0*PI*(1.D0-ANU)))*(16.D0*PI*ANU+(1.D0-4.D0*ANU)*
+ (AI1+AI3)-3.D0*(A*A+C*C)*AI13)
  SS(2,3)=(G/(8.D0*PI*(1.D0-ANU)))*(16.D0*PI*ANU+(1.D0-4.D0*ANU)*
+ (AI2+AI3)-3.D0*(B*B+C*C)*AI23)
C
  SS(2,1)=SS(1,2)
  SS(3,1)=SS(1,3)
  SS(3,2)=SS(2,3)
C
  SS(4,4)=G-(G/(8.D0*PI*(1.D0-ANU)))*((1.D0-2.D0*ANU)*(AI1+AI2)+
+ 3.D0*(A*A+B*B)*AI12)
  SS(5,5)=G-(G/(8.D0*PI*(1.D0-ANU)))*((1.D0-2.D0*ANU)*(AI1+AI3)+
+ 3.D0*(A*A+C*C)*AI13)
  SS(6,6)=G-(G/(8.D0*PI*(1.D0-ANU)))*((1.D0-2.D0*ANU)*(AI2+AI3)+
+ 3.D0*(B*B+C*C)*AI23)
C
  CALL KARRANGE(Q,SS,A1,B1,C1,NTENS)
C
C*** ROTATION
C
  DO I=1,3
    QQ(I,1)=AN1(I)
    QQ(I,2)=AN2(I)
    QQ(I,3)=AN3(I)
  END DO
C
  CALL KINITIA(QMX,NTENS*NTENS)
  DO I=1,3

```

```

DO J=1,3
  QMX(I,J)=QQ(I,J)**2
END DO
END DO
C
QMX(1,4)=2.D0*QQ(1,1)*QQ(1,2)
QMX(2,4)=2.D0*QQ(2,1)*QQ(2,2)
QMX(3,4)=2.D0*QQ(3,1)*QQ(3,2)
C
QMX(4,1)=QQ(1,1)*QQ(2,1)
QMX(4,2)=QQ(1,2)*QQ(2,2)
QMX(4,3)=QQ(1,3)*QQ(2,3)
C
QMX(4,4)=QQ(1,1)*QQ(2,2)+QQ(1,2)*QQ(2,1)
C
CALL KTRANSPOSE(QMX,AUXM1,NTENS,NTENS)
CALL KMULT(QMX,Q,AUXM2,NTENS,NTENS,NTENS)
CALL KMULT(AUXM2,AUXM1,Q,NTENS,NTENS,NTENS)
C
RETURN
END
C
C*****
C
SUBROUTINE KDQ(DQDW1,DQDW2,ANU,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,
+ G)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION DQDW1(NTENS,NTENS),DQDW2(NTENS,NTENS),
+ AN1(3),AN2(3),AN3(3),X(1),W(1)
C
DIMENSION AUXM1(4,4)
C
CALL KESH_Q(AUXM1,ANU,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
C
DW1=0.005D0
IF (W1.EQ.W2.AND.DABS(W1-1.D0).GT.0.05)
+ DW1=1.1D0*(W1-1.D0/(0.01D0+(1.D0/W1)))
W12=W1-DW1
CALL KESH_Q(DQDW1,ANU,W12,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
DO I=1,NTENS
DO J=1,NTENS
  DQDW1(I,J)=(DQDW1(I,J)-AUXM1(I,J))/(-DW1)
END DO
END DO
C
DW2=0.005
IF (W1.EQ.W2.AND.DABS(W1-1.D0).GT.0.05)
+ DW2=1.1D0*(-W2+1.D0/((1.D0/W2)-0.01D0))
W22=W2+DW2
CALL KESH_Q(DQDW2,ANU,W1,W22,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
DO I=1,NTENS
DO J=1,NTENS
  DQDW2(I,J)=(DQDW2(I,J)-AUXM1(I,J))/DW2
END DO
END DO
C
RETURN
1001 format(1p6e12.4)
1002 format(i6)

```

```

2001 format(2i4)
END
C
C*****
C
SUBROUTINE KDQDN(DQDN1,DQDN2,DQDN3,ANU,W1,W2,AN1,AN2,AN3,X,W,
+ NGAUS,NTENS,G)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
DIMENSION DQDN1(NTENS,NTENS,3),DQDN2(NTENS,NTENS,3),
+ DQDN3(NTENS,NTENS,3),AN1(3),AN2(3),AN3(3),X(1),W(1),
+ BN1(3),BN2(3),BN3(3)
C
DIMENSION AUXM1(4,4),AUXM2(4,4)
C
CALL KESH_Q(AUXM1,ANU,W1,W2,AN1,AN2,AN3,X,W,NGAUS,NTENS,G)
write(7,*)
write(7,*) ' auxm1 in kdqdn1'
do i=1,ntens
write(7,1001) (auxm1(i,j),j=1,ntens)
end do
C
DO I=1,3
CALL KCOPY(AN1,BN1,3)
BN1(I)=1.01D0*AN1(I)
IF (DABS(BN1(I)).LT.1.D-6) BN1(I)=0.005D0
DN=BN1(I)-AN1(I)
CALL KESH_Q(AUXM2,ANU,W1,W2,BN1,AN2,AN3,X,W,NGAUS,NTENS,G)
write(7,*)
write(7,*) ' auxm2 in kdqdn1'
do ii=1,ntens
write(7,1001) (auxm2(ii,j),j=1,ntens)
end do
DO K=1,NTENS
DO L=1,NTENS
DQDN1(K,L,I)=(AUXM2(K,L)-AUXM1(K,L))/DN
END DO
END DO
END DO
C
DO I=1,3
CALL KCOPY(AN2,BN2,3)
BN2(I)=1.01D0*AN2(I)
IF (DABS(BN2(I)).LT.1.D-6) BN2(I)=0.005D0
DN=BN2(I)-AN2(I)
CALL KESH_Q(AUXM2,ANU,W1,W2,AN1,BN2,AN3,X,W,NGAUS,NTENS,G)
DO K=1,NTENS
DO L=1,NTENS
DQDN2(K,L,I)=(AUXM2(K,L)-AUXM1(K,L))/DN
END DO
END DO
END DO
C
DO I=1,3
CALL KCOPY(AN3,BN3,3)
BN3(I)=1.01D0*AN3(I)
DN=BN3(I)-AN3(I)
IF (DABS(BN3(I)).LT.1.D-6) BN3(I)=0.005D0
CALL KESH_Q(AUXM2,ANU,W1,W2,AN1,AN2,BN3,X,W,NGAUS,NTENS,G)
DO K=1,NTENS

```

```

DO L=1,NTENS
  DQDN3 (K, L, I) = (AUXM2 (K, L) -AUXM1 (K, L) ) /DN
END DO
END DO
END DO
C
RETURN
1001 format (1p6e12.4)
1002 format (i6)
2001 format (2i4)
END
C
C*****
C
SUBROUTINE KIAB (AI1, AI2, AI3, AI11, AI22, AI33, AI12, AI13, AI23, A, B, C,
+ IA, IB, IC, X, W, NGAUS)
C
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
DIMENSION X (1), W (1)
C
PI=4.D0*DATAN (1.D0)
C
IF (DABS (A-B) .GE.0.01D0 .AND. DABS (B-C) .GE.0.01D0) THEN
C
CALL KABC (AI1, AI2, AI3, AI11, AI22, AI33, AI12, AI13, AI23, A, B, C,
+ X, W, NGAUS)
RETURN
END IF
C
C
C
C
IF (DABS (A-B) .LT.0.01D0 .AND. DABS (B-C) .LT.0.01D0) THEN
C
IF (IA.EQ.1) THEN
CALL KAAA (AI1, AI2, AI3, AI11, AI22, AI33, AI12, AI13, AI23, 1.D0)
CALL KAAC (AI1Y, AI2Y, AI3Y, AI11Y, AI22Y, AI33Y, AI12Y, AI13Y,
+ AI23Y, 1.D0, 0.99D0)
CALL KACC (AI1Z, AI2Z, AI3Z, AI11Z, AI22Z, AI33Z, AI12Z, AI13Z,
+ AI23Z, 1.D0, 0.99D0)
SH1=(C-0.99D0)/0.01D0
SH2=(B-C)/0.01
SH3=(1.D0-B)/0.01D0
AI1=SH1*AI1+SH2*AI1Y+SH3*AI1Z
AI2=SH1*AI2+SH2*AI2Y+SH3*AI2Z
AI3=SH1*AI3+SH2*AI3Y+SH3*AI3Z
AI11=SH1*AI11+SH2*AI11Y+SH3*AI11Z
AI22=SH1*AI22+SH2*AI22Y+SH3*AI22Z
AI33=SH1*AI33+SH2*AI33Y+SH3*AI33Z
AI12=SH1*AI12+SH2*AI12Y+SH3*AI12Z
AI13=SH1*AI13+SH2*AI13Y+SH3*AI13Z
AI23=SH1*AI23+SH2*AI23Y+SH3*AI23Z
RETURN
END IF
C
IF (IB.EQ.1) THEN
CALL KAAA (AI1, AI2, AI3, AI11, AI22, AI33, AI12, AI13, AI23, 1.D0)
CALL KACC (AI1Y, AI2Y, AI3Y, AI11Y, AI22Y, AI33Y, AI12Y, AI13Y,
+ AI23Y, 1.01D0, 1.D0)
CALL KABC (AI1Z, AI2Z, AI3Z, AI11Z, AI22Z, AI33Z, AI12Z, AI13Z, AI23Z,
+ 1.01D0, 1.D0, 0.99D0, X, W, NGAUS)

```

```

CALL KAAC(AI1X,AI2X,AI3X,AI11X,AI22X,AI33X,AI12X,AI13X,AI23X,
+ 1.D0,0.99D0)
SH1=(C-0.99D0)*(1.01D0-A)/0.0001D0
SH2=(C-0.99D0)*(A-1.D0)/0.0001D0
SH3=(1.D0-C)*(A-1.D0)/0.0001D0
SH4=(1.D0-C)*(1.01D0-A)/0.0001D0
AI1=SH1*AI1+SH2*AI1Y+SH3*AI1Z+SH4*AI1X
AI2=SH1*AI2+SH2*AI2Y+SH3*AI2Z+SH4*AI2X
AI3=SH1*AI3+SH2*AI3Y+SH3*AI3Z+SH4*AI3X
AI11=SH1*AI11+SH2*AI11Y+SH3*AI11Z+SH4*AI11X
AI22=SH1*AI22+SH2*AI22Y+SH3*AI22Z+SH4*AI22X
AI33=SH1*AI33+SH2*AI33Y+SH3*AI33Z+SH4*AI33X
AI12=SH1*AI12+SH2*AI12Y+SH3*AI12Z+SH4*AI12X
AI13=SH1*AI13+SH2*AI13Y+SH3*AI13Z+SH4*AI13X
AI23=SH1*AI23+SH2*AI23Y+SH3*AI23Z+SH4*AI23X
RETURN
END IF
C
IF (IC.EQ.1) THEN
CALL KAAA(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,1.D0)
CALL KACC(AI1Y,AI2Y,AI3Y,AI11Y,AI22Y,AI33Y,AI12Y,AI13Y,
+ AI23Y,1.01D0,1.D0)
CALL KAAC(AI1Z,AI2Z,AI3Z,AI11Z,AI22Z,AI33Z,AI12Z,AI13Z,AI23Z,
+ 1.01D0,1.D0)
SH1=(1.01D0-A)/0.01D0
SH2=(A-B)/0.01D0
SH3=(B-1.D0)/0.01D0
AI1=SH1*AI1+SH2*AI1Y+SH3*AI1Z
AI2=SH1*AI2+SH2*AI2Y+SH3*AI2Z
AI3=SH1*AI3+SH2*AI3Y+SH3*AI3Z
AI11=SH1*AI11+SH2*AI11Y+SH3*AI11Z
AI22=SH1*AI22+SH2*AI22Y+SH3*AI22Z
AI33=SH1*AI33+SH2*AI33Y+SH3*AI33Z
AI12=SH1*AI12+SH2*AI12Y+SH3*AI12Z
AI13=SH1*AI13+SH2*AI13Y+SH3*AI13Z
AI23=SH1*AI23+SH2*AI23Y+SH3*AI23Z
RETURN
END IF
END IF
C
C
C
C
IF (DABS(A-B).LT.0.01D0.AND.DABS(B-C).GE.0.01D0) THEN
CALL KAAC(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,C)
CALL KABC(AI1Z,AI2Z,AI3Z,AI11Z,AI22Z,AI33Z,AI12Z,AI13Z,AI23Z,
+ B+0.01D0,B,C,X,W,NGAUS)
XI=(A-B)/0.01D0
AI1=XI*AI1Z+(1.D0-XI)*AI1
AI2=XI*AI2Z+(1.D0-XI)*AI2
AI3=XI*AI3Z+(1.D0-XI)*AI3
AI11=XI*AI11Z+(1.D0-XI)*AI11
AI22=XI*AI22Z+(1.D0-XI)*AI22
AI33=XI*AI33Z+(1.D0-XI)*AI33
AI12=XI*AI12Z+(1.D0-XI)*AI12
AI13=XI*AI13Z+(1.D0-XI)*AI13
AI23=XI*AI23Z+(1.D0-XI)*AI23
RETURN
END IF
C
C

```



```

C
C   IF (DABS(A-B).GE.0.01D0.AND.DABS(B-C).LT.0.01D0) THEN
C
C       CALL KACC(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,C)
C       CALL KABC(AI1Z,AI2Z,AI3Z,AI11Z,AI22Z,AI33Z,AI12Z,AI13Z,AI23Z,
+   A,C+0.01D0,C,X,W,NGAUS)
C       XI=(B-C)/0.01D0
C       AI1=XI*AI1Z+(1.D0-XI)*AI1
C       AI2=XI*AI2Z+(1.D0-XI)*AI2
C       AI3=XI*AI3Z+(1.D0-XI)*AI3
C       AI11=XI*AI11Z+(1.D0-XI)*AI11
C       AI22=XI*AI22Z+(1.D0-XI)*AI22
C       AI33=XI*AI33Z+(1.D0-XI)*AI33
C       AI12=XI*AI12Z+(1.D0-XI)*AI12
C       AI13=XI*AI13Z+(1.D0-XI)*AI13
C       AI23=XI*AI23Z+(1.D0-XI)*AI23
C       RETURN
C   END IF
C
C   RETURN
C   END
C
C*****
C
C   SUBROUTINE KABC(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,B,C,
+   X,W,NGAUS)
C
C   IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C   DIMENSION X(1),W(1)
C
C   PI=4.D0*DATAN(1.D0)
C   TH=DASIN(DSQRT(1.D0-C*C/(A*A)))
C   AK=DSQRT((A*A-B*B)/(A*A-C*C))
C
C   SUM1=0.D0
C   SUM2=0.D0
C   AA=0.D0
C   BB=TH
C   AUX1=(BB-AA)/2.D0
C   AUX2=(BB+AA)/2.D0
C   DO I=1,NGAUS
C       PHI=AUX1*X(I)+AUX2
C       SUM1=SUM1+W(I)/DSQRT(1.D0-(AK*DSIN(PHI))**2)
C       SUM2=SUM2+W(I)*DSQRT(1.D0-(AK*DSIN(PHI))**2)
C   END DO
C   F=SUM1*AUX1
C   E=SUM2*AUX1
C
C   AI1=4.D0*PI*A*B*C*(F-E)/((A*A-B*B)*DSQRT(A*A-C*C))
C   AI3=4.D0*PI*A*B*C*(B*DSQRT(A*A-C*C)/(A*C)-E)/
+   ((B*B-C*C)*DSQRT(A*A-C*C))
C   AI2=4.D0*PI-AI1-AI3
C   AI12=(AI2-AI1)/(3.D0*(A*A-B*B))
C   AI13=(AI3-AI1)/(3.D0*(A*A-C*C))
C   AI23=(AI3-AI2)/(3.D0*(B*B-C*C))
C   AI11=4.D0*PI/(3.D0*A*A)-AI12-AI13
C   AI22=4.D0*PI/(3.D0*B*B)-AI12-AI23
C   AI33=4.D0*PI/(3.D0*C*C)-AI13-AI23
C
C   RETURN

```

```

END
C
C*****
C
SUBROUTINE KAAC(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,C)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
PI=4.D0*DATAN(1.D0)
AI1=2.D0*PI*A*A*C/( (A*A-C*C)**1.5D0 )*
+ ( DACOS(C/A)-(C/A)*DSQRT(1.D0-C*C/(A*A)) )
AI2=AI1
AI3=4.D0*PI-2.D0*AI1
AI13=(AI3-AI1)/(3.D0*(A*A-C*C))
AI23=AI13
AI12=PI/(3.D0*A*A)-AI13/4
AI11=3.D0*AI12
AI22=AI11
AI33=4.D0*PI/(3.D0*C*C)-AI13-AI23
C
RETURN
END
C
C*****
C
SUBROUTINE KACC(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A,C)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
PI=4.D0*DATAN(1.D0)
Z=A/C
DACOSH=DLOG(Z+DSQRT(Z*Z-1.D0))
AI2=2.D0*PI*A*C*C/( (A*A-C*C)**1.5D0 )*
+ ( (A/C)*DSQRT(A*A/(C*C)-1.D0)-DACOSH )
AI3=AI2
AI1=4.D0*PI-2.D0*AI2
AI13=(AI2-AI1)/(3.D0*(A*A-C*C))
AI12=AI13
AI23=PI/(3.D0*C*C)-AI12/4
AI22=3.D0*AI23
AI33=AI22
AI11=4.D0*PI/(3.D0*A*A)-AI12-AI13
C
RETURN
END
C
C*****
C
SUBROUTINE KAAA(AI1,AI2,AI3,AI11,AI22,AI33,AI12,AI13,AI23,A)
C
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
PI=4.D0*DATAN(1.D0)
AI1=4.D0*PI/3.D0
AI2=AI1
AI3=AI1
AI11=4.D0*PI/(5.D0*A*A)
AI22=AI11
AI33=AI11
AI12=AI11/3.D0
AI13=AI12

```

```

C      AI23=AI12
C
C      RETURN
C      END
C
C*****
C
C      SUBROUTINE KORDER (A, B, C, IA, IB, IC, A1, B1, C1)
C
C      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
C      IA=0
C      IB=0
C      IC=0
C
C      IF (A1.GE.B1.AND.B1.GE.C1) THEN
C          A=A1
C          B=B1
C          C=C1
C          IC=1
C          RETURN
C      END IF
C      IF (A1.GE.C1.AND.C1.GE.B1) THEN
C          A=A1
C          B=C1
C          C=B1
C          IB=1
C          RETURN
C      END IF
C      IF (C1.GE.A1.AND.A1.GE.B1) THEN
C          A=C1
C          B=A1
C          C=B1
C          IA=1
C          RETURN
C      END IF
C      IF (B1.GE.A1.AND.A1.GE.C1) THEN
C          A=B1
C          B=A1
C          C=C1
C          IC=1
C          RETURN
C      END IF
C      IF (B1.GE.C1.AND.C1.GE.A1) THEN
C          A=B1
C          B=C1
C          C=A1
C          IB=1
C          RETURN
C      END IF
C      IF (C1.GE.B1.AND.B1.GE.A1) THEN
C          A=C1
C          B=B1
C          C=A1
C          IA=1
C          RETURN
C      END IF
C
C      RETURN
C      END
C

```

```

C*****
C
C      SUBROUTINE KARRANGE (Q, SS, A1, B1, C1, NTENS)
C
C      IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
C      DIMENSION Q (NTENS, NTENS), SS (6, 6)
C
C      IF (A1.GE.B1.AND.B1.GE.C1) THEN
C          DO I=1, NTENS
C              DO J=1, NTENS
C                  Q (I, J)=SS (I, J)
C              END DO
C          END DO
C      END IF
C
C      IF (A1.GE.C1.AND.C1.GE.B1) THEN
C          Q (1, 1)=SS (1, 1)
C          Q (1, 2)=SS (1, 3)
C          Q (1, 3)=SS (1, 2)
C          Q (2, 1)=SS (3, 1)
C          Q (2, 2)=SS (3, 3)
C          Q (2, 3)=SS (3, 2)
C          Q (3, 1)=SS (2, 1)
C          Q (3, 2)=SS (2, 3)
C          Q (3, 3)=SS (2, 2)
C          Q (4, 4)=SS (5, 5)
C      END IF
C
C      IF (C1.GE.A1.AND.A1.GE.B1) THEN
C          Q (1, 1)=SS (2, 2)
C          Q (1, 2)=SS (2, 3)
C          Q (1, 3)=SS (2, 1)
C          Q (2, 1)=SS (3, 2)
C          Q (2, 2)=SS (3, 3)
C          Q (2, 3)=SS (3, 1)
C          Q (3, 1)=SS (1, 2)
C          Q (3, 2)=SS (1, 3)
C          Q (3, 3)=SS (1, 1)
C          Q (4, 4)=SS (6, 6)
C      END IF
C
C      IF (B1.GE.A1.AND.A1.GE.C1) THEN
C          Q (1, 1)=SS (2, 2)
C          Q (1, 2)=SS (2, 1)
C          Q (1, 3)=SS (2, 3)
C          Q (2, 1)=SS (1, 2)
C          Q (2, 2)=SS (1, 1)
C          Q (2, 3)=SS (1, 3)
C          Q (3, 1)=SS (3, 2)
C          Q (3, 2)=SS (3, 1)
C          Q (3, 3)=SS (3, 3)
C          Q (4, 4)=SS (4, 4)
C      END IF
C
C      IF (B1.GE.C1.AND.C1.GE.A1) THEN
C          Q (1, 1)=SS (3, 3)
C          Q (1, 2)=SS (3, 1)
C          Q (1, 3)=SS (3, 2)
C          Q (2, 1)=SS (1, 3)
C          Q (2, 2)=SS (1, 1)

```

```

        Q(2,3)=SS(1,2)
        Q(3,1)=SS(2,3)
        Q(3,2)=SS(2,1)
        Q(3,3)=SS(2,2)
        Q(4,4)=SS(5,5)
    END IF
C
    IF (C1.GE.B1.AND.B1.GE.A1) THEN
        Q(1,1)=SS(3,3)
        Q(1,2)=SS(3,2)
        Q(1,3)=SS(3,1)
        Q(2,1)=SS(2,3)
        Q(2,2)=SS(2,2)
        Q(2,3)=SS(2,1)
        Q(3,1)=SS(1,3)
        Q(3,2)=SS(1,2)
        Q(3,3)=SS(1,1)
        Q(4,4)=SS(6,6)
    END IF
C
    RETURN
    END
C
C*****
C
    DOUBLE PRECISION FUNCTION FLGAMA(W)
C
C CALCULATES THE LOG(BASE E) GAMMA(W) FOR W REAL AND
C GAMMA(W) POSITIVE. USES STIRLING'S APPROXIMATION.
C ACCURATE TO ABOUT 12 SIGNIFICANT PLACES.
C
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    PI=4.D0*DATAN(1.D0)
    X=W
    M=0
    FK=-1.D0
    IF(X-.5D0) 10,20,20
C
C W LESS THAN .5
C
10  M=1
    XPI=X*PI
    X=1.D0-X
20  FK=FK+1.D0
    IF(X+FK-6.D0) 20,20,30
30  Z=X+FK
    ZZ=Z*Z
C
C LOG GAMMA(Z), Z GREATER 6.
C
    Y=(Z-.5D0)*DLOG(Z)-Z+.9189385332047D0+(((((-4146.D0/ZZ+1820.D0)/ZZ
1-1287.D0)/ZZ+1716.D0)/ZZ-6006.D0)/ZZ+180180.D0)/Z/2162160.D0
    IF(FK) 60,60,40
40  IK=IDINT(FK)
    DO 50 I=1,IK
        FK=FK-1.D0
50  Y=Y-DLOG(X+FK)
60  IF(M) 70,110,70
70  P=PI/DSIN(XPI)
    IF(P) 80,80,100
80  WRITE(6,90)W

```

```

90  FORMAT(2X,'GAMMA(',1PE12.5,') IS NEGATIVE')
    Y=0.D0
    GO TO 110
100  Y=DLOG(P)-Y
110  FLGAMA=Y
    RETURN
    END
C
C*****
C
    SUBROUTINE JACOBI(ALF,BTA,NN,X,A)
C
C    CALCULATES THE ZEROS X(I) OF THE NN-TH ORDER
C    JACOBI POLYNOMIAL PN(ALF,BTA) FOR THE SEGMENT (-1,1).
C    THE LARGEST ZERO WILL BE STORED IN X(1). ALSO
C    CALCULATES THE CORRESPONDING COEFFICIENTS A(I)
C    OF THE NN-TH ORDER GAUSS-JACOBI QUADRATURE FORMULA
C    OF DEGREE 2*NN-1.
C
C      1      ALF      BTA      NN
C    INTEGRAL (1-X)  (1+X)  F(X) DX = SUM A(I)*F(X(I))
C      -1                                I=1
C
C    REFERENCE: "GAUSSIAN QUADRATURE FORMULAS", A. H. STROUD AND
C              DON SECREST, PRENTICE-HALL, INC., 1966
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      DIMENSION X(1),A(1),B(1000),C(1000)
C
C    DEFINE ERROR TOLERANCE EPS ON THE X(I)
C    DEFINE THE SUM OF ALF PLUS BTA AS APB
C
C      EPS=10.D0**(-9)
C      APB=ALF+BTA
C
C    COMPUTE THE COEFFICIENTS B(N) AND C(N) FOR THE
C    RECURSION RELATION OF THE P(N)'S
C
C      P(N) = (X - B(N))*P(N-1) - C(N)*P(N-2)
C
C      DO 5 I=2,NN
C        XI=FLOAT(I)
C        ABI=APB+2.D0*XI
C        B(I)=(BTA*BTA-ALF*ALF)/((ABI)*(ABI-2.D0))
C        IF(I.EQ.2) THEN
C          CNUM=4.D0*(ALF+1.D0)*(BTA+1.D0)
C          CDEN=(APB+3.D0)*(APB+2.D0)*(APB+2.D0)
C        ELSE
C          CNUM=4.D0*(XI-1.D0)*(ALF+XI-1.D0)*(BTA+XI-1.D0)*(APB+XI-1.D0)
C          CDEN=(ABI-1.D0)*(ABI-2.D0)*(ABI-2.D0)*(ABI-3.D0)
C        ENDIF
C        C(I)=CNUM/CDEN
C      5  CONTINUE
C
C    BEGIN COMPUTATION OF ZEROS OF JACOBI POLYNOMIAL
C    THE SECTIONS BELOW LABELED LARGEST ZERO THROUGH
C    LAST ZERO COMPUTE THE INITIAL APPROXIMATIONS TO
C    THE ROOTS. THE CALL TO ROOT IMPROVES UPON THIS
C    INITIAL APPROXIMATION.
C
C      FN=FLOAT(NN)

```

```

      BETA=DEXP (FLGAMA (ALF+1.D0) +FLGAMA (BTA+1.D0) -FLGAMA (APB+2.D0) )
      CC=2.D0** (APB+1.D0) *BETA
      DO 10  J=2,NN
      CC=CC*C (J)
10    CONTINUE
      DO 120 I=1,NN
      IF (I-1) 120, 20, 30
C
C  LARGEST ZERO
C
20    AN=ALF/FN
      BN=BTA/FN
      R1=(1.D0+ALF) * (2.78D0/ (4.D0+FN*FN) +.768D0*AN/FN)
      R2=1.D0+1.48D0*AN+.96D0*BN+.452D0*AN*AN+.83D0*AN*BN
      XT=1.D0-R1/R2
      GO TO 110
30    IF (I-2) 120, 40, 50
C
C  SECOND ZERO
C
40    R1=(4.1D0+ALF) / ( (1.D0+ALF) * (1.D0+.156D0*ALF) )
      R2=1.D0+.06D0* (FN-8.D0) * (1.D0+.12D0*ALF) /FN
      R3=1.D0+.012*BTA* (1.D0+.25D0*DABS (ALF) ) /FN
      RATIO=R1*R2*R3
      XT=XT-RATIO* (1.D0-XT)
      GO TO 110
50    IF (I-3) 120, 60, 70
C
C  THIRD ZERO
C
60    R1=(1.67D0+.28D0*ALF) / (1.D0+.37D0*ALF)
      R2=1.D0+.22D0* (FN-8.D0) /FN
      R3=1.D0+8.D0*BTA/ ( (6.28D0+BTA) *FN*FN)
      RATIO=R1*R2*R3
      XT=XT-RATIO* (X (1) -XT)
      GO TO 110
70    IF (NN-I-1) 100, 90, 80
C
C  MIDDLE ZEROS
C
80    XT=3.D0*X (I-1) -3.D0*X (I-2) +X (I-3)
      GO TO 110
C
C  SECOND TO LAST ZERO
C
90    R1=(1.D0+.235D0*BTA) / (.766D0+.119D0*BTA)
      R2=1.D0/ (1.D0+.639D0* (FN-4.D0) / (1.D0+.71D0* (FN-4.D0) ) )
      R3=1.D0/ (1.D0+20.D0*ALF/ ( (7.5D0+ALF) *FN*FN) )
      RATIO=R1*R2*R3
      XT=XT+RATIO* (XT-X (I-2) )
      GO TO 110
C
C  LAST ZERO
C
100   R1=(1.D0+.37D0*BTA) / (1.67D0+.28D0*BTA)
      R2=1.D0/ (1.D0+.22D0* (FN-8.D0) /FN)
      R3=1.D0/ (1.D0+8.D0*ALF/ ( (6.28D0+ALF) *FN*FN) )
      RATIO=R1*R2*R3
      XT=XT+RATIO* (XT-X (I-2) )
C
110   CALL ROOT (XT, NN, ALF, BTA, DPN, PN1, B, C, EPS)

```

```

        X(I)=XT
        A(I)=CC/(DPN*PN1)
120    CONTINUE
        RETURN
        END
C
C*****
C
        SUBROUTINE RECUR(PN, DPN, PN1, X, NN, ALF, BTA, B, C)
C
C CALCULATES P(N) BASED UPON RECURSION RELATION
C
C   P(N) = (X - B(N)) * P(N-1) - C(N) * P(N-2)
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        DIMENSION B(1),C(1)
        P1=1.D0
        P=X+(ALF-BTA)/(ALF+BTA+2.D0)
        DP1=0.D0
        DP=1.D0
        DO 10 J=2,NN
        Q=(X-B(J))*P-C(J)*P1
        DQ=(X-B(J))*DP+P-C(J)*DP1
        P1=P
        P=Q
        DP1=DP
        DP=DQ
10    CONTINUE
        PN=P
        DPN=DP
        PN1=P1
        RETURN
        END
C
C*****
C
        SUBROUTINE ROOT(X, NN, ALF, BTA, DPN, PN1, B, C, EPS)
C
C IMPROVES UPON THE APPROXIMATE ROOT X
C IN ADDITION WE ALSO OBTAIN
C DPN = DERIVATIVE OF P(N) AT X
C PN1 = VALUE OF P(N-1) AT X
C
C NEWTON-RAPHSON METHOD IS USED
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        DIMENSION B(1),C(1)
        ITER=0
10    ITER=ITER+1
        CALL RECUR(P, DP, PN1, X, NN, ALF, BTA, B, C)
        D=P/DP
        X=X-D
        IF(DABS(D)-EPS)30,30,20
20    IF(ITER-10)10,30,30
30    DPN=DP
        RETURN
        END
C
C*****
C
        SUBROUTINE MPC(UE, A, JDOF, MDOF, N, JTYPE, X, U, UUNIT, MAXDOF,

```



```

+ LMPC, KSTEP, KINC, TIME, NT, NF, TEMP, FIELD, LTRAN, TRAN)
C
  INCLUDE 'ABA_PARAM.INC'
c
  IMPLICIT DOUBLE PRECISION (A-H, O-Z)
C
  DIMENSION A(N), JDOF(N), X(6,N), U(MAXDOF,N), UINIT(MAXDOF,N),
+ TIME(2), TEMP(NT,N), FIELD(NF,NT,N), LTRAN(N), TRAN(3,3,N)
C
  RR=1214.2D0
  JDOF(1)=JTYPE
  JDOF(2)=1
  A(1)=1.D0
C
  E=300.D0
  ANU=0.3D0
  G=E/(2.D0*(1.D0+ANU))
  AK=3.D0-4.D0*ANU
  PI=4.D0*DATAN(1.D0)
  SIF=TIME(2)
C
  X1=X(1,1)
  X2=X(2,1)
  R=DSQRT(X1*X1+X2*X2)
  TH=DATAN2(X2,X1)
  IF (TH.LT.0.D0) TH=TH+PI
  THG=TH*180.D0/PI
C
  IF (JTYPE.EQ.1) THEN
    AUX=(1.D0/(2.D0*G))*DSQRT(0.5D0*R/PI)*DCOS(TH/2.D0)*
+ (AK-1.D0+2.D0*(DSIN(TH/2.D0)**2))
    UE=SIF*AUX-SIF*(1.D0-2.D0*ANU)/G*DSQRT(0.5D0*RR/PI)
    A(2)=-AUX
  END IF
C
  IF (JTYPE.EQ.2) THEN
    AUX=(1.D0/(2.D0*G))*DSQRT(0.5D0*R/PI)*DSIN(TH/2.D0)*
+ (AK+1.D0-2.D0*(DCOS(TH/2.D0)**2))
    UE=SIF*AUX
    A(2)=-AUX
  END IF
C
  WRITE(7,*) 'R, TH, SIF, UE, A(1), A(2) '
C
  WRITE(7,1001) R, TH, SIF, UE, A(1), A(2)
C
  WRITE(7,1001) R, TH, SIF, UE, A(1), A(2)
C
  RETURN
C
1001 FORMAT(1P6E12.4)
1002 FORMAT(6I5)
END

```