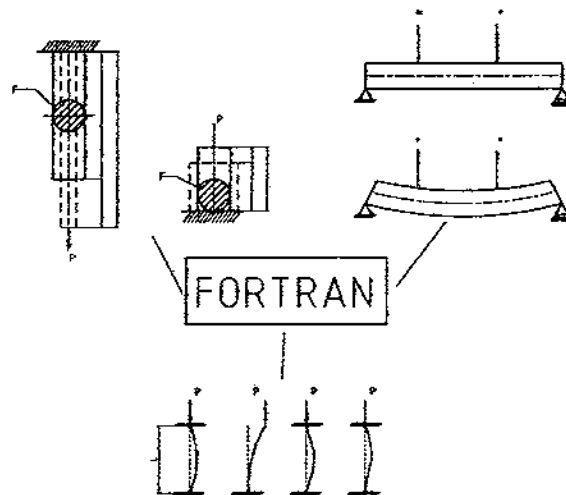




ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΠΑΤΡΑΣ (Α.Τ.Ε.Ι. ΠΑΤΡΑΣ)  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΓΙΑ ΤΗΝ  
ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΑΠΛΩΝ ΕΝΤΑΤΙΚΩΝ  
ΚΑΤΑΣΤΑΣΕΩΝ



ΣΠΟΥΔΑΣΤΗΣ: Αγάπιος Αγαπίου

ΕΙΣΗΓΗΤΗΣ: Δρ. Αναστάσιος Βασιλόπουλος

Πάτρα 2006

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα

ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΓΙΑ ΤΗΝ  
ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΑΠΛΩΝ ΕΝΤΑΤΙΚΩΝ  
ΚΑΤΑΣΤΑΣΕΩΝ

του Αγάπιου Αγαπίου

Πρόεδρος της εποπτεύουσας επιτροπής:

Βασιλόπουλος Αναστάσιος  
Τμήμα μηχανολογίας

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ .....	- 9 -
2. ΑΝΤΟΧΗ ΤΩΝ ΥΛΙΚΩΝ.....	- 10 -
2.1 Μονοαξονικός Εφελκυσμός-Θλίψη.....	- 10 -
2.2 Κάμψη.....	- 18 -
2.3 Λυγισμός.....	- 15 -
3. 3Ο ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ.....	- 18 -
4. 4ΟΔΗΓΙΕΣ ΧΡΗΣΕΩΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	- 47 -

## ΛΙΣΤΑ ΕΙΚΟΝΩΝ

1		<i>Σελίδα2</i>
	a) Εφελκυσμός ράβδου	
	b) Θλίψη ράβδου	
2	Αμφιέριστη δοκός με δύο δυνάμεις	<i>Σελίδα4</i>
3	Αμφιέριστη δοκός παραμορφωμένη από δύο δυνάμεις	<i>Σελίδα5</i>
4	Διαστάσεις αμφιέριστης δοκού	<i>Σελίδα6</i>
5		<i>Σελίδα8</i>
	a) Αμφιαρθρωτή ράβδος	
	b) Μονόπακτη ράβδος, αμφίπακτη ράβδος, πακτωμένη-αρθρωτή ράβδος	

## ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά τον καθηγητή μου Βασιλόπουλο Αναστάσιο για τη σημαντική καθοδήγηση και βοήθεια του στα δύσκολα σημεία της πτυχιακής εργασίας.

## ΓΛΩΣΣΑΡΙ

**Material.** [υλικό]

**tension.** [τάση]

**modulus of elasticity, Young's modulus, Tension modulus, Material stiffness.** [μέτρο ελαστικότητας]

**Proportional limit.** [όριο αναλογίας]

**Elastic limit.** [όριο ελαστικότητας]

**Poisson's ratio.** [λόγος Poisson]

**Ultimate strength.** [όριο αντοχής]

**Yield strength.** [όριο διαρροής]

**density.** [πυκνότητα]

**point.** [σημείο]

**flexion.** [κάμψη]

**reaction.** [αντίδραση]

**radius of curvature.** [ακτίνα καμπυλότητας]

**force.** [δύναμη]

**torque.** [ροπή]

**buckling.** [λυγισμός]

**moment of inertia.** [ροπή αδράνειας]

**critical buckling load.** [κρίσιμο φορτίο λυγισμού]

**critical buckling stress.** [κρίσιμη τάση λυγισμού]

**radius of gyration.** [ελάχιστη ακτίνα λυγισμού]

**tension.** [εφελκυσμός]

**compression.** [θλίψη]

**strain.** [ανοιγμένη παραμόρφωση]

**change in length.** [διαφορά μήκους]

**original length.** [Αρχικό μήκος]

**Cross-sectional area.** [κάθετη διατομή]

## 1 ΕΙΣΑΓΩΓΗ

Στην πτυχιακή αυτή πήραμε απλά προβλήματα αντοχής υλικών, όπως τα βλέπουμε αναλυμένα στο δεύτερο κεφάλαιο και προσπαθήσαμε να αναπτύξουμε απλά προγράμματα, τα οποία να είναι όσο το δυνατόν πιο δυναμικά και όχι παθητικά. Με λίγα λόγια τα προγράμματα, για κάθε τμήμα, έγιναν με δυνατότητα εκτύπωσης σφάλματος και δυνατότητα επανεισαγωγής των δεδομένων ή τη δυνατότητα της σωστής επιλογής από τον χρήστη. Επίσης, έκανε το πρόγραμμα δυναμικό και ο κατάλογος υλικών, έτσι που να μην χρειάζεται ο χρήστης να εισάγει κάθε φορά χειροκίνητα το υλικό που θέλει για να κάνει τους υπολογισμούς, αλλά να το επιλέγει από μια λίστα. Κατά τον προγραμματισμό έγινε μεγάλη προσπάθεια για να χρησιμοποιήσουμε όσο το δυνατόν περισσότερες από τις καινούργιες δυνατότητες της *FORTRAN 90/95*, αποφεύγοντας τη χρήση εντολών της *FORTRAN 77*. Με τη χρήση της *FORTRAN 90/95* το πρόγραμμα ήταν πιο εύκολο να γίνει δυναμικό, αλλά και απλό στη λειτουργία του. Ελπίζω το πρόγραμμα αυτό να αναπτυχθεί σε ένα δυναμικό πολυχρηστικό εργαλείο μάθησης στον τομέα της αντοχής των υλικών αλλά και της *FORTRAN*.

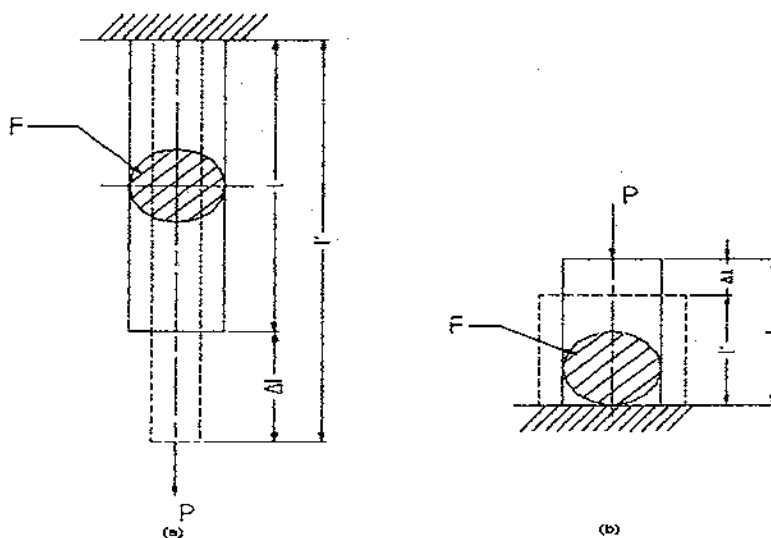


## 2 ΑΝΤΟΧΗ ΤΩΝ ΥΛΙΚΩΝ

Για να μπορέσουμε να αναφερθούμε στην αντοχή των υλικών πρέπει πρώτα να αναφέρουμε κάποια πράγματα που μας ενδιαφέρουν. Όπως τάση  $s$  και ορίζεται ως το πηλίκο της εφαρμοζόμενης δύναμης προς το εμβαδόν της κάθετης διατομής,  $\sigma = \frac{P}{F}$ . Όταν η εφαρμοζόμενη δύναμη είναι κάθετη προς τη διατομή τότε έχουμε  $t = \frac{Q}{F}$  όπου  $Q$  η δύναμη. Η  $s$  και  $t$  έχουν μονάδες μέτρησης  $[Pa]$ ,  $[\frac{N}{m^2}]$ ,  $[\frac{kp}{cm^2}]$ ,  $[\frac{t}{m^2}]$ , αναλογικά βέβαια με τις μονάδες που μετρήσαμε τη δύναμη και την επιφάνεια.

### 2.1 Μονοαξονικός Εφελκυσμός-Θλίψη

Όταν καταπονούμε μια ράβδο υλικού με κάποια δύναμη, έστω  $P$  (σχήμα 1) με χαρακτηριστικά της ράβδου τη διατομή  $F$  και το μήκος  $l$ . Κατά την εφαρμογή της δύναμης αυτής το υλικό θα παραμορφωθεί γραμμικά κατά μήκος του άξονά της κατά  $\Delta l$  με τελικό μήκος  $l'$  ((α) εφελκυσμός, (β) θλίψη). Όταν η παραμορφώσεις



που δημιουργούνται είναι μέσα στην ελαστική περιοχή του υλικού, η παραμόρφωση είναι ανάλογη με τη δύναμη που την προκαλεί. Ο Hooke παρατήρησε πρώτος αυτή

την αναλογία και τη διατύπωσε ως εξής: “η επιμήκυνση  $\Delta l$  είναι ανάλογη τόσο της εφαρμοζόμενης δύναμης αλλά και του αρχικού μήκους  $l$  και αντιστρόφως ανάλογη του εμβαδού της διατομής”, δηλαδή  $\Delta l \propto \frac{P \cdot l}{F}$ . Με μαθηματική ματιά έχουμε

$$\Delta l = \frac{P \cdot l}{E \cdot F} \text{ όπου } \Delta l = l' - l, P \text{ η εφαρμοζόμενη δύναμη, } F \text{ είναι το εμβαδόν της}$$

κάθετης διατομής, και  $E$  είναι ο συντελεστής αναλογίας, όπου εξαρτάται από το είδος του υλικού. Για να ισχύει ο νόμος του Hooke πρέπει να πληρούνται οι παρακάτω παραδοχές:

- 1) Ο άξονας της ράβδου να είναι ευθύγραμμος.
- 2) Η δύναμη να δρα στη διεύθυνση του άξονα της ράβδου και το σημείο εφαρμογής να είναι το κέντρο βάρους.
- 3) Οι τάσεις κατανέμονται ομοιόμορφα στη διατομή.
- 4) Όταν έχουμε θλίψη στο θλιβόμενο μέρος δεν υπάρχει κίνδυνος λυγισμού.
- 5) Όλες οι κατά μήκος ίνες της ράβδου επιμηκύνονται όλες το ίδιο.
- 6) Οι διατομές που είναι αρκετά επίπεδες και κάθετες στον άξονα της ράβδου παραμένουν έτσι και μετά την παραμόρφωση.
- 7) Οι αναπτυσσόμενες τάσεις είναι μικρότερες από την τάση αναλογίας του υλικού.

Οι παραπάνω παραδοχές ισχύουν, όπως μπορεί να καταλάβει κάποιος, για τον εφελκυσμό αλλά και για τη θλίψη.

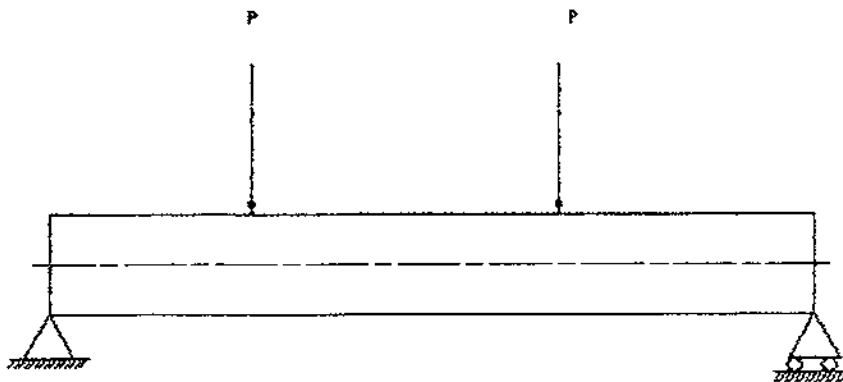
Στο πρόγραμμα μας χρησιμοποιήσαμε τον τύπο  $\Delta l = \frac{P \cdot l}{E \cdot F}$ , γιατί με αυτόν μπορούσε να γραφτεί ένα δυναμικό πρόγραμμα, με δυνατότητα επιλογής από τον χρήστη την παράμετρο που θέλει να υπολογίσει.

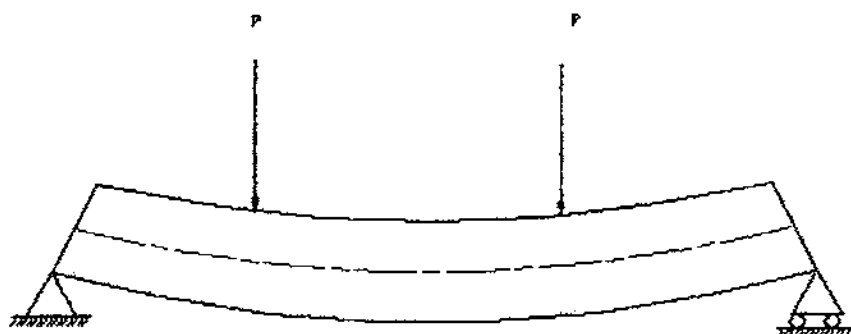
Κατά την διεξαγωγή των πειραμάτων εφελκυσμού και θλίψης, εξάγουμε τα αποτελέσματα της εφαρμοζόμενης δύναμης συναρτήσει της διαφοράς μήκους. Τα αποτελέσματα αυτά για να είναι πιο εύχρηστα αλλά και πιο ευπαρουσίαστα τα μετατρέπουμε σε τάση συναρτήσει της ανοιγμένης επιμήκυνσης. Διαιρώντας τη δύναμη με το εμβαδόν της κάθετης διατομής  $\sigma = \frac{P}{F}$  και για τον υπολογισμό του  $\epsilon$  χρησιμοποιούμε την εξής σχέση:  $\epsilon = \frac{\Delta l}{l} = \frac{l' - l}{l}$ . Με τα δεδομένα αυτά, σχεδιάζουμε γράφημα από το οποίο μπορούμε να διακρίνουμε το όριο αναλογίας, το όριο ελαστικότητας, την περιοχή παραμόρφωσης και το όριο θραύσης του υλικού.

## 2.2 Κάμψη

Κάμψη είναι η εντατική κατάσταση στην οποία βρίσκεται μια δοκός, πάνω στην οποία επιβάλλονται εγκάρσιες δυνάμεις, με αποτέλεσμα η δοκός να παραμορφωθεί όπως βλέπουμε στο σχήμα 2 παρακάτω.

σχήμα 2





Στην περίπτωση της κάμψης μελετήσαμε μόνο καταπόνηση απλής αμφιέριστης δοκού, με δύο φορτία κατανομημένα σε τυχαία θέση. Κατά την εφαρμογή των δυνάμεων σε μια δοκό έχουμε εμφάνιση παραμορφώσεων της δοκού. Δημιουργείται δηλαδή μια καμπύλη, για την οποία έχουμε την ακτίνα καμπυλότητας που τη χαρακτηρίζει. Το αντίστροφό της ονομάζεται καμπυλότητα,

που δίνεται από τη σχέση  $k = \frac{1}{R}$ , όπου  $R$  η ακτίνα καμπυλότητας. Επίσης, ισχύει

$\frac{\Delta l}{\Delta x} = \frac{y}{R}$  όπου  $\frac{\Delta l}{\Delta x}$  είναι η ανοιγμένη παραμόρφωση  $\epsilon$ . Από τα πιο πάνω, προκύπτει

ότι  $\sigma = \frac{E}{R} y$ , που χαρακτηρίζει τη γραμμική μεταβολή των τάσεων  $\sigma$  συναρτήσει

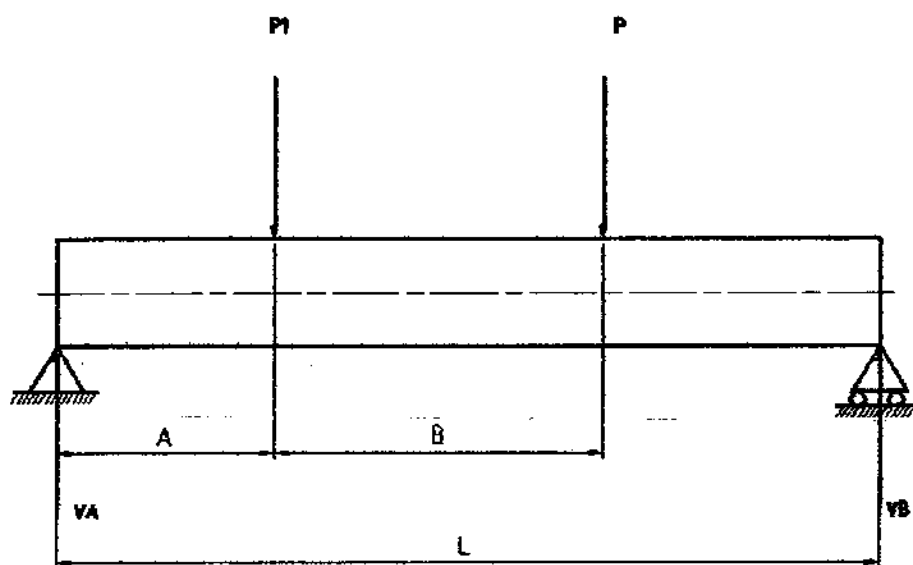
της κάθετης απόστασης  $y$ . Κατά την καταπόνηση μιας ράβδου σε κάμψη δημιουργούνται τάσεις που είναι ανάλογες των καμπτικών ροπών στην κάθε θέση και αντιστρόφως ανάλογες με τη ροπή αδράνειας και ευθέως ανάλογες με την

κάθετη απόσταση από την κεντρική γραμμή, δηλαδή  $\sigma_x = \frac{M_z}{I_z} y$ . Η ακτίνα

καμπυλότητας μπορεί να υπολογιστεί και ως εξής  $\frac{1}{R} = \frac{M_z}{E \cdot I_z}$ . Από την ανάλυση

της προηγούμενης σχέσης έχουμε τη γωνία στρώφης που είναι  $\varphi = \frac{M_z}{EI_z} l$ .

Για τον προσδιορισμό της καμπτικής ροπής σε κάθε θέση χρησιμοποιούμε τις παρακάτω σχέσεις:  $M(x) = V_A \cdot x$  όταν  $0 < X < A$ ,  
 $M(x) = V_A \cdot x - P_1(x - A)$  όταν  $A < x < A + B$  και τέλος  
 $M(x) = V_A \cdot x - P_1(x - A) - P_2(x - (A + B))$  όταν  $A + B < x < L$ . Τα σύμβολα που χρησιμοποιήσαμε στις παραπάνω σχέσεις τα βλέπουμε στο παρακάτω σχήμα διακρίνοντας τι συμβολίζει το κάθε τι. Για την ολοκληρωμένη μελέτη της κάμψης



μιας ράβδου καλό θα ήταν να βρούμε και την ελαστική γραμμή. Η ελαστική γραμμή υπολογίζεται εάν ολοκληρώσουμε τις συναρτήσεις της ροπής δύο φορές και έτσι έχουμε το βέλος κάμψης για κάθε ροπή, αφού η συνάρτηση που διέπει την

κλίση της ελαστικής γραμμής σε κάθε σημείο είναι  $\frac{d^2y}{dx^2} = \pm \frac{M(x)}{EI_z}$ . Επομένως, για

$M(x) = V_A \cdot x$  στο διάστημα  $0 < X < A$  έχουμε

$y = \iint \frac{M(x)}{EI_z} dx = \frac{V_A \cdot x^3}{6EI_z} + C_1x + C_2$ , μελετώντας την εξίσωση στα άκρα

υπολογίζουμε τα  $C_1$  και  $C_2$  όπου καταλήξαμε  $C_1 = -\frac{V_A \cdot l^2}{6EI_z}$  και  $C_2 = 0$ ,

αντικαθιστώντας όπου  $C_1$  και  $C_2$  στην αρχική εξίσωση καταλήγουμε να έχουμε την εξίσωση του βέλους κάμψης συναρτήσει του  $x$  για την περιοχή  $0 < X < A$  και η

σχέση είναι 
$$y = \frac{1}{EI_z} \left( \frac{V_A \cdot x^3}{6} - \frac{V_A \cdot L^2}{6} \right)$$
. Για τις υπόλοιπες περιοχές,

εργαζόμαστε με τον ίδιο τρόπο. Έτσι για την περιοχή  $A < x < A+B$  η σχέση είναι:

$$y = \frac{1}{EI_z} \left( \frac{V_A \cdot x^3}{6} - \frac{P_1 \cdot x^3}{6} + \frac{P_1 \cdot A \cdot x^2}{2} - \frac{V_A \cdot L^2 \cdot x}{6} + \frac{P_1 \cdot L^2 \cdot x}{6} - \frac{P_1 \cdot A \cdot L \cdot x}{2} \right)$$

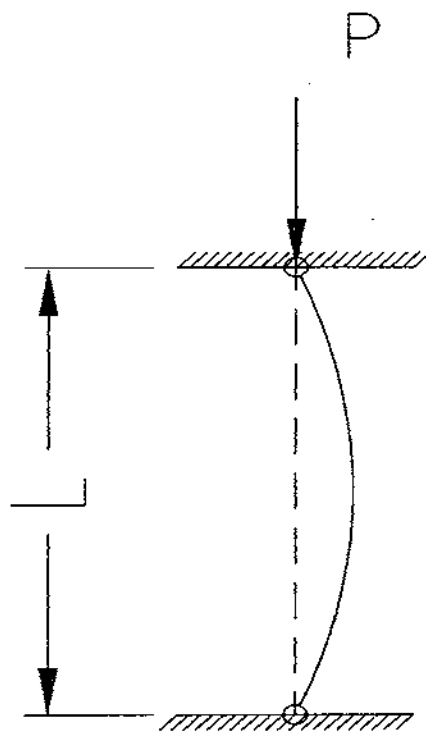
ενώ για την περιοχή  $A+B < x < L$  έχουμε

$$y = \frac{1}{EI_z} \left( \frac{V_A \cdot x^3}{6} - \frac{P_1 \cdot x^3}{6} + \frac{P_1 \cdot A \cdot x^2}{2} - \frac{P_2 \cdot x^3}{6} + \frac{P_2 \cdot A \cdot x^2}{2} + \frac{P_2 \cdot B \cdot x^2}{2} - \frac{V_A \cdot L^2}{6} + \frac{P_1 \cdot L^2}{6} - \frac{P_1 \cdot A \cdot L}{2} + \frac{P_2 \cdot L^2}{6} - \frac{P_2 \cdot A \cdot L}{2} - \frac{P_2 \cdot B \cdot L}{2} \right)$$

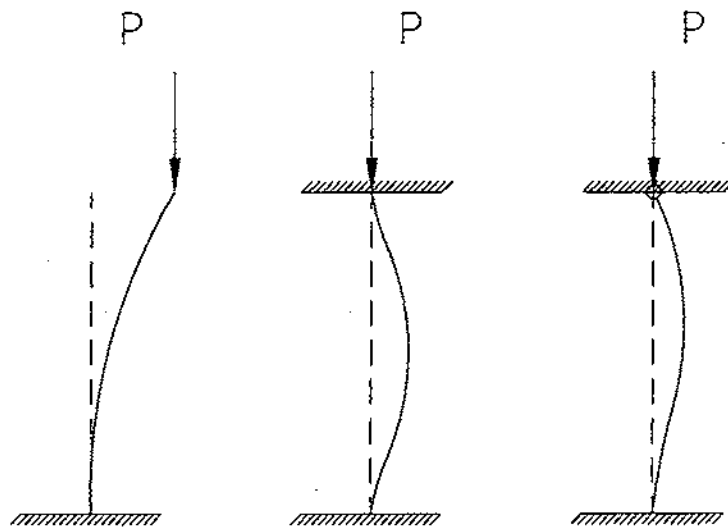
### 2.3 Λυγισμός

Κατά την φόρτιση μίας ράβδου με προοδευτικά αυξανόμενα θλιπτικά φορτία δημιουργούνται τάσεις στο υλικό. Όταν όμως το μήκος της ράβδου είναι κατά πολύ μεγαλύτερο από το πάχος της, τότε κατά τη φόρτιση και από μια τιμή της δύναμης και πέρα η ράβδος συμπεριφέρεται διαφορετικά από ότι στη θλίψη. Συγκεκριμένα, δημιουργούνται κάθετες στη διατομή τάσεις που τείνουν να λυγίσουν τη ράβδο. Η δύναμη που ασκείται στη ράβδο όταν αυτή περνά από ευσταθή κατάσταση θλίψης σε ασταθή κατάσταση λυγισμού, ονομάζεται κρίσιμο φορτίο.

Στον λυγισμό έχουμε τέσσερις διαφορετικές περιπτώσεις, που εξαρτώνται από τον τρόπο στήριξης της κάθε ράβδου. Έχουμε τη θεμελιώδη περίπτωση της αμφιαρθρωτής ράβδου ορθογωνικής διατομής.



Αμφιαρθρωτή ράβδος



Μονόπακτη Ράβδος

Αμφίπακτη Ράβδος

Πακτωμένη-Αρθρωτή Ράβδος.

Η αμφιαρθρωτή ράβδος είναι αυτή που χρησιμοποιήσαμε στο πρόγραμμα. Οι σχέσεις που χρησιμοποιήσαμε για τον προσδιορισμό της κρίσιμης δύναμης είναι  $P_{κρ} = \frac{\pi^2 \cdot E \cdot I_{\min}}{l^2}$ , ενώ για την κρίσιμη τάση έχουμε  $\sigma_{κρ} = \frac{P_{κρ}}{F}$ , για την ελάχιστη ακτίνα έχουμε  $i_{\min} = \sqrt{\frac{I_{\min}}{F}}$ , και για το λόγο λυγρότητας  $\lambda = \frac{l_{κρ}}{i_{\min}}$ . Η κρίσιμη τάση μπορεί επίσης να υπολογιστεί και από τη σχέση  $\sigma_{κρ} = \frac{\pi^2 \cdot E}{\lambda^2}$ .



### 3 Ο ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ

Παρακάτω έχουμε τον πηγαίο κώδικα του προγράμματος οδηγού μέσω του οποίου γίνεται η επιλογή της εφαρμογής που θέλει κάποιος.

```
!KENTRIKO PROGRAMA
PROGRAM SOFM
  CHARACTER(3)      :: SM, LOOP
  INTEGER           :: N, N1, N2, N3
  PRINT*, '=====
  PRINT*, '-----          WELCOME          -----
  PRINT*, '-----          TO          -----
  PRINT*, '-----          STRENGTH OF MATERIALS          -----
  PRINT*, '-----          PROGRAM          -----
  PRINT*, '=====
  PRINT*, ' *
DO
  N1=0
  DO
    PRINT*, '=====
    PRINT*, '-----          FOR TENSION OF MATERIALS          -----
    PRINT*, '-----          PRESS T OR t          -----
    PRINT*, '=====
    PRINT*, '=====
    PRINT*, '-----          FOR FLEXION OF MATERIALS          -----
    PRINT*, '-----          PRESS F OR f          -----
    PRINT*, '=====
    PRINT*, '-----          FOR BUCKLING OF BARS          -----
    PRINT*, '-----          PRESS B OR b          -----
    PRINT*, '=====
    PRINT*, '-----          TO ADD MATERIAL ON DATA LIST          -----
    print*, '-----          PRESS A OR a          -----
    PRINT*, '=====
    PRINT*, '=====
    PRINT*, '-----          TO QUIT          -----
    PRINT*, '-----          PRESS Q OR q          -----
    PRINT*, '=====
  READ*, SM
  N=1
```

```
SELECT CASE (SM)
  CASE ('T','t')
    CALL EFEPL()
  CASE ('F','f')
    CALL FLEXION()
  CASE ('B','b')
    CALL BUCKLING()
  CASE ('A','a')
    CALL AMD()
  CASE ('Q','q')
    N1=1
  CASE DEFAULT
    CALL ERROR()
    N=0
END SELECT
IF (N>0) EXIT
END DO
IF (N1>0) EXIT
END DO
END
```

Παρακάτω έχουμε τον πηγαίο κώδικα της κάθε υπορουτίνας που καλεί το αρχικό πρόγραμμα. Πρώτη έχουμε την υπορουτίνα για τον εφελκυσμό-θλίψη.

```

!      =====TENSION=====
SUBROUTINE TENSION()
  REAL      ::DL, P, L, E, F, TASH
  CHARACTER ::agnostos, LOOP, CH
  CHARACTER(31) ::NAM
  INTEGER    ::M, K, N, N1
  PRINT*, ' '
  PRINT*, ' '
  PRINT*, '=====
  PRINT*, '=====
  PRINT*, '=====
  PRINT*, '          TENSION OF MATERIALS          '
  PRINT*, '=====
  PRINT*, '=====
  PRINT*, '          P          P          '
  PRINT*, '          <----->          '
  PRINT*, '          |<---L--->|          '
  PRINT*, '          |-DL-|          '
  PRINT*, ' '
  PRINT*, '=====
  PRINT*, '=====
  PRINT*, ' '
DO
  M=0
  DO
    N1=0
    PRINT*, '=====
    PRINT*, '====DO YOU WANT TO NAME THE OUTPOUR FILE?===='
    PRINT*, '====FOR YES PRESS (Y)===='
    PRINT*, '====FOR NO PRESS (N)===='
    PRINT*, '====TO RETURN TO MAIN MENU (R)===='
    PRINT*, '=====
    READ*, CH
    SELECT CASE (CH)
      CASE ('Y', 'y')
        PRINT*, ' '
        PRINT*, '=====
        PRINT*, '====GIVE OUTPOURDATA FILE NAME(ONLY 31 CHARACTERS)===='
        PRINT*, '==== EXAMPLE: OUTPOURDATA.TXT, DATA.DAT===='
        PRINT*, '=====
        READ*, NAM
      CASE ('N', 'n')
        PRINT*, ' '
        PRINT*, '=====
        PRINT*, '==== DEFAULT NAME "TENSIONOUTPOUR.TXT"===='
        PRINT*, '=====
        NAM='OUTPOURDATA.TXT'

```

```

CASE ('R','r')
    M=1
CASE DEFAULT

END SELECT
IF (N1==0) EXIT
END DO
IF (M==0) THEN
    OPEN (30,FILE=NAME)
!EKTIPOSI STIN OUGNI
    PRINT*, '=====
    PRINT*, '==== GIVE THE PARAMETER OFF YOU WISH TO CALCULETE ====
    PRINT*, '==== AND PRESS ENTER =====
    PRINT*, '====
    PRINT*, '==== DL P L F E =====
    PRINT*, '-----
    PRINT*, ' '
    READ*,agnostos
    WRITE(30,*) '-----
    WRITE(30,*) ' '
    WRITE(30,*) ' P P
    WRITE(30,*) ' <----->
    WRITE(30,*) ' |<---L--->|
    WRITE(30,*) ' |---DL---|
    WRITE(30,*) ' '
    SELECT CASE (agnostos)
        CASE ('D','d')
            CALL DD()
        CASE ('P','p')
            CALL PP()
        CASE ('L','l')
            CALL LL()
        CASE ('F','f')
            CALL FF()
        CASE ('E','e')
            CALL EE()
        CASE DEFAULT
            CALL ERROR()
    END SELECT
    CLOSE (30)
END IF
IF (M==0) THEN
    DO
        N=0
        PRINT*, '=====
        PRINT*, '===== TO REPEAT PRESS (R) =====
        PRINT*, '===== TO QUIT PRESS (Q) =====
        PRINT*, '-----
        READ*,LOOP
        M=0
    
```

```
SELECT CASE (LOOP)
  CASE ('R','r')
    M=0
  CASE ('Q','q')
    M=1
  CASE DEFAULT
    CALL ERROR();N=1
END SELECT
IF (N .EQ. 0) EXIT
END DO
END IF
IF (M.EQ.1) EXIT
END DO
END
```



```

READ*, CH
SELECT CASE (CH)
  CASE ('M', 'm')
    CALL PORS(N2)
    PRINT*, '-----'
    PRINT*, '---- GIVE INTAKE NUMBER OF DATA ----'
    PRINT*, '-----'
    READ*, J
    ALLOCATE (AFAR(J),DL(J),P(J),S(J),e(J))
    SELECT CASE (N2)
      CASE (1)
        DO I=1,J
          AFAR(I)=I
          PRINT*, '-----'
          PRINT*, '---- (X AXIS) GIVE NUMBER IN POSITION',I,' ----'
          PRINT*, '---- DL IN (m) ----'
          PRINT*, '-----'
          READ*, DL(I)
        END DO
        DO I=1,J
          PRINT*, '-----'
          PRINT*, '---- (Y AXIS) GIVE NUMBER IN POSITION',I,' ----'
          PRINT*, '---- P IN (N) ----'
          PRINT*, '-----'
          READ*, P(I)
        END DO
        OPEN (15,FILE='GRADATA.TXT')
        DO I=1,J
          WRITE(15,*) DL(I),P(I)
        END DO
        CLOSE (15)
      CASE (0)
        DO I=1,J
          AFAR(I)=I
          PRINT*, '-----'
          PRINT*, '---- (X AXIS) GIVE NUMBER IN POSITION',I,' ----'
          PRINT*, '---- e ----'
          PRINT*, '-----'
          READ*, DL(I)
        END DO
        DO I=1,J
          PRINT*, '-----'
          PRINT*, '---- (Y AXIS) GIVE NUMBER IN POSITION',I,' ----'
          PRINT*, '---- S IN (MPa) ----'
          PRINT*, '-----'
          READ*, P(I)
        END DO
        OPEN (15,FILE='GRADATA.TXT')
        DO I=1,J
          WRITE(15,*) e(I),S(I)
        END DO
        CLOSE (15)
    END SELECT
  END CASE

```

```

CASE DEFAULT
END SELECT
CASE ('F','f')
CALL PORS(N2)
PRINT*, '-----'
PRINT*, '==== GIVE INTAKE NUMBER OF DATA ====='
PRINT*, '-----'
READ*, J
PRINT*, '-----'
PRINT*, '==== GIVE FILE NAME (ONLY 31 CHARACTERS) ====='
PRINT*, '==== EXAMPLE: OUTPOURDATA.TXT, DATA.DAT ====='
PRINT*, '-----'
READ*, NAM
ALLOCATE (AFAR(J),DL(J),P(J),e(J),S(J),e1(J),e12(J),e1S(J),S2(J))
IF (N2=-1) THEN
OPEN (15,FILE=NAM)
DO I=1,J
AFAR(I)=I
READ(15,*) DL(I),P(I)
END DO
CLOSE (15)
ELSE
OPEN (15,FILE=NAM)
DO I=1,J
AFAR(I)=I
READ (15,*) e(I),S(I)
END DO
CLOSE (15)
END IF
CASE DEFAULT
CALL ERROR()
N1=1
END SELECT
IF (N1=0) EXIT
END DO
IF (N2=-1) THEN
DO I=1,J
S(I)=P(I)/F ; e(I)=DL(I)/L
END DO
END IF
DO I=1,J
e1(I)=e(I)*MAXVAL(S)/MAXVAL(e)
END DO
ALLOCATE (AA(J-1),AA1(J-1))
e1S=e1*S;e12=e1**2;S2=S**2
DO I=2,J
Se1S=SUM(e1S(1:I))
Se1=SUM(e1(1:I))
SS=SUM(s(1:I))
Se12=SUM(e12(1:I))
SS2=SUM(S2(1:I))
S_ =SS/I

```



```

e1_ =Sel/I
S1e1S=SUM(e1S(I-1:I))
S1e1=SUM(e1(I-1:I))
S1S=SUM(s(I-1:I))
S1e12=SUM(e12(I-1:I))
S1S2=SUM(S2(I-1:I))
S1_ =S1S/2;e11_ =S1e1/2
AA(I-1)=A(SelS, Sel, S_, Sel2, e1_)
AAI(I-1)=A(S1e1S, S1e1, S1_, S1e12, e11_)
END DO
DO
DO I=1,J
IF (AAI(I)<=AA(I)*GC) THEN
PRINT*, ' '
PRINT*, '===== '
PRINT*, '==== THE POINT',I,' IS PROPORTIONAL LIMIT ===== '
PRINT*, '===== '
L=I
END IF
IF (AAI(I)<=AA(I)*GC) EXIT
END DO
PRINT*, '----- '
PRINT*, '==== GRAPH LOOK COEFFICIENT IS ',GC,' ===== '
PRINT*, '----- '
DO
PRINT*, '----- '
PRINT*, '==== IF POINT IS WRONG CHANGE ===== '
PRINT*, '==== GRAPH LOOK COEFFICIENT TO FIXT ===== '
PRINT*, '----- '
PRINT*, '====DO YOU WANT TO CHANGE ===== '
PRINT*, '----- '
PRINT*, '====FOR YES PRESS (Y) ===== '
PRINT*, '====FOR NO PRESS (N) ===== '
PRINT*, '----- '
READ*, YN
N3=1
SELECT CASE (YN)
CASE ('Y','y')
PRINT*, '----- '
PRINT*, '==== GIVE GRAPH LOOK COEFFICIENT ===== '
PRINT*, '----- '
READ*, GC
CASE ('n','N')
N3=1
CASE DEFAULT
CALL ERROR()
N3=0
END SELECT
IF (N3>0) EXIT
END DO
IF (GC<=0.8) EXIT
END DO

```

```

IF (N==0) THEN
  DO
    PRINT*, '===== '
    PRINT*, '==== TO REPEAT PRESS (R)      ===='
    PRINT*, '==== TO QUIT PRESS (Q)      ===='
    PRINT*, '===== '
    READ*, LOOP
    SELECT CASE (LOOP)
      CASE ('R','r')
      CASE ('Q','q')
        N=1
      CASE DEFAULT
        CALL ERROR()
        N4=0
    END SELECT
    IF (N4==1) EXIT
  END DO
  END IF
  IF (N==1) EXIT
  END DO
  CASE ('R','r')
  CASE DEFAULT
    CALL ERROR()
    N5=1
  END SELECT
  IF (N5==0) EXIT
  END DO
END

```

Το επόμενο κομμάτι πηγαίου κώδικα ανήκει στην κάμψη αμφιάξονιας δοκού.

```

!KAMPSI
SUBROUTINE FLEXION()
  REAL      :: L, A, B, P1, P2, VA, VB, E, SMAX, FE1, Y, X, R6
             , I, M, FE(300)=0, MX(300)=0, MMAX(2), XE(300)=0 &
             , STEP1, STEP, STEP3
  INTEGER   :: N, N1, N2, N3
  INTEGER   :: T, N8
  CHARACTER :: LOOP, CH, RC
  CHARACTER(31) :: NAM
  DO
    PRINT*, ' '
    PRINT*, ' '
    PRINT*, '-----'
    PRINT*, '-----'
    PRINT*, '===='
    PRINT*, '----- FLEXION OF MATERIALS -----'
    PRINT*, '===='
    PRINT*, '-----'
    PRINT*, '-----'
    PRINT*, ' '
    PRINT*, '          P1      P2      '
    PRINT*, '          |      |      '
    PRINT*, '-----|-----|-----'
    PRINT*, '          |      |      oo '
    PRINT*, '          V      V      V      V '
    PRINT*, '          -----'
    PRINT*, '          /\                /\ '
    PRINT*, '          |<- A ->|<- B ->|      | '
    PRINT*, '          |<-----L----->| '
    PRINT*, '          |                        | '
    PRINT*, '          VA                VB '
    PRINT*, ' '
    N=0
    PRINT*, '-----'
    PRINT*, '=====>TO RETURN TO MAIN MENU (R) ====='
    PRINT*, '=====>TO CONTINUE (C) ====='
    PRINT*, '-----'
    READ*, RC
    SELECT CASE (RC)
      CASE ('R', 'r')
        N=1
      CASE ('C', 'c')
        PRINT*, '-----'
        PRINT*, '==== GIVE THE PARAMETER ====='
        PRINT*, '==== L(IN m), A(IN m), B(IN m), P1(IN N), P2(IN N), ====='
        PRINT*, '==== IZ(IN m^4) ====='
        PRINT*, '==== AND PRESS ENTER ====='
        PRINT*, '-----'

```

```

PRINT*, ' '
PRINT*, '-----'
PRINT*, '-----          GIVE L (IN m)          -----'
READ*, L
DO
  N1=0
  PRINT*, '-----'
  PRINT*, '-----'
  PRINT*, '-----          GIVE A (IN m)          -----'
  READ*, A
  IF (A>L) THEN
    CALL ERROR()
    PRINT*, '*****          MAST A<L          *****'
  ELSE
    N1=1
  END IF
  IF (N1>0) EXIT
END DO
N1=0
DO
  N2=0
  PRINT*, '-----'
  PRINT*, '-----'
  PRINT*, '-----          GIVE B (IN m)          -----'
  READ*, B
  IF (B.GT.(L-B)) THEN
    CALL ERROR()
    PRINT*, '*****          MAST B<L-A          *****'
  ELSE
    N2=1
  END IF
  IF (N2>0) EXIT
END DO
N2=0
PRINT*, '-----'
PRINT*, '-----'
PRINT*, '-----          GIVE P1 (IN N)          -----'
READ*, P1
PRINT*, '-----'
PRINT*, '-----'
PRINT*, '-----          GIVE P2 (IN N)          -----'
READ*, P2
PRINT*, '-----'
PRINT*, '-----'
PRINT*, '-----          GIVE I (IN m^4)          -----'
READ*, I
PRINT*, '-----'
CALL RETON(A,B,L,P1,P2,VA,VB)
CALL ME_EL(E)
STEP1=A/100;STEP2=B/100;STEP3=(L-(A+B))/100
X=0;N7=1

```



```

PRINT*, '-----'
PRINT*, 'DISTANCE DATA:'
PRINT*, 'L=', L, '[M]'
PRINT*, 'A=', A, '[M]'
PRINT*, 'B=', B, '[M]'
PRINT*, 'FORCES DATA:'
PRINT*, 'P1=', P1, '[N]'
PRINT*, 'P2=', P2, '[N]'
PRINT*, 'REACTIONS DATA:'
PRINT*, 'VA=', VA, '[N]'
PRINT*, 'VB=', VB, '[N]'
PRINT*, 'MAXIMUM TORQUE'
PRINT*, 'MMAX=', M, '[Nm]'
PRINT*, 'RADIUS OF CURRVATURE'
PRINT*, 'R=', R, '[m]'
PRINT*, 'TWIST'
PRINT*, 'f=', FEI, '[*]'
PRINT*, 'MAXIMUM STRES ON MAXIMUM DISTANCE FROM CENTER LINE'
PRINT*, 'SMAX=', SMAX, '[Pa]'
PRINT*, '-----'
PRINT*, ' '
DO
    CALL DATANAME(T, NAM)
    IF (T>0) EXIT
END DO
T=0
OPEN (10, FILE=NAM)
WRITE(10,*) ' '
WRITE(10,*) '-----'
WRITE(10,*) ' '
WRITE(10,*) '          P1      P2      '
WRITE(10,*) '          |      |      '
WRITE(10,*) '          |      |      '
WRITE(10,*) '          |      |      oo '
WRITE(10,*) '          \      \      \      \ '
WRITE(10,*) '          ----- '
WRITE(10,*) '          ^          ^      '
WRITE(10,*) '          |<- A ->|<- B ->|      | '
WRITE(10,*) '          |<-----L----->|      '
WRITE(10,*) '          |          |      '
WRITE(10,*) '          VA          VB      '
WRITE(10,*) ' '
WRITE(10,*) '-----'
WRITE(10,*) 'DISTANCE DATA:'
WRITE(10,*) 'L=', L, '[M]'
WRITE(10,*) 'A=', A, '[M]'
WRITE(10,*) 'B=', B, '[M]'
WRITE(10,*) 'FORCES DATA:'
WRITE(10,*) 'P1=', P1, '[N]'
WRITE(10,*) 'P2=', P2, '[N]'
WRITE(10,*) 'REACTIONS DATA:'
WRITE(10,*) 'VA=', VA, '[N]'

```

```

WRITE(10,*) 'VB=',VB, '[N]'
WRITE(10,*) 'MAXIMUM TORQUE'
WRITE(10,*) 'MMAX=',M, '[Nm]'
WRITE(10,*) 'RADIUS OF CURRVATURE'
WRITE(10,*) 'R=',R, '[M]'
WRITE(10,*) 'TWIST'
WRITE(10,*) 'f=',FE1, '['*]'
WRITE(10,*) 'MAXIMUM STRES ON MAXIMUM DISTANCE FROM CENTER LINE'
WRITE(10,*) 'SMAX=',SMAX, '[Pa]'
WRITE(10,*) '=====
CLOSE (10)
IF (N==0) THEN
DO
N3=1
PRINT*, '=====
PRINT*, '---- TO REPEAT PRESS (R)
PRINT*, '==== TO QUIT PRESS (Q)
PRINT*, '=====
READ*, LOOP
SELECT CASE (LOOP)
CASE ('R','r')
CASE ('Q','q')
N=1
CASE DEFAULT
CALL ERROR()
N3=0
END SELECT
IF (N3==1) EXIT
END DO
END IF
CASE DEFAULT
CALL ERROR()
END SELECT
IF (N==1) EXIT
END DO
END

```





```

PRINT*, '====          GIVE L (IN m)          ====='
READ*, L
PRINT*, '===='
PRINT*, '===='
PRINT*, '====          GIVE F (IN m^2)          ====='
READ*, F
PRINT*, '===='
PRINT*, '===='
PRINT*, '====          GIVE I MIN (IN m^4)          ====='
READ*, I
PRINT*, '=====
CALL ME_EL(E)
P=(PI**2*E*10**9*I)/L**2      !ΥΠΟΛΟΓΙΣΜΟΣ ΚΡΙΣΙΜΟΥ ΦΟΡΤΙΟΥ
TASH=PCR/F                    !ΥΠΟΛΟΓΙΣΜΟΣ ΚΡΙΣΙΜΗΣ ΤΑΣΗΣ
IMIN=SQRT(I/F)                !ΥΠΟΛΟΓΙΣΜΟΣ ΕΛΑΧΙΣΤΗΣ ΑΚΤΙΝΑΣ
LAMDA=L/IMIN                  !ΥΠΟΛΟΓΙΣΜΟΣ ΕΙΝΤΕΛΕΣΤΗ ΑΓΓΕΡΟΠΙΤΑΣ
PRINT*, ' '
PRINT*, '-----'
PRINT*, '          y|          '
PRINT*, '          √          '
PRINT*, '          P  _ _ _ _ _ P          '
PRINT*, '          >O=====O<          '
PRINT*, '          |<-----L----->|          '
PRINT*, '          '
PRINT*, '-----'
PRINT*, 'DISTANCE DATA:'
PRINT*, 'L=',L, ' [M]'
PRINT*, 'CRITICAL BUKLING LOAD'
PRINT*, 'P=',PCR, '[N]'
PRINT*, 'RADIUS OF GYRATION'
PRINT*, 'R=',R, '[m]'
PRINT*, 'CRITICAL BUCKLING STRESS'
PRINT*, 'SCR=',TASH, '[Pa]'
PRINT*, '-----'
DO WHILE (N4>0)
  N4=0
  PRINT*, '-----'
  PRINT*, '===DO YOU WANT TO NAME THE OUTPOUR FILE? ==='
  PRINT*, '===FOR YES PRESS (Y) ==='
  PRINT*, '===FOR NO PRESS (N) ==='
  PRINT*, '-----'
  READ*, CH
  SELECT CASE (CH)
    CASE ('Y','y')
      PRINT*, ' '
      PRINT*, '-----'
      PRINT*, '===GIVE OUTPOURDATA FILE NAME (ONLY 31 CHARACTERS)===>'
      PRINT*, '=== EXAMPLE: OUTPOURDATA.TXT, DATA.DAT =====>'
      PRINT*, '-----'
      READ*, NAM
    CASE ('N','n')

```

```

PRINT*, ' '
PRINT*, '====='
PRINT*, '--- DEFAULT NAME "BUCKLINGOUTPOUR.TXT" ---'
PRINT*, '====='
NAM='BUCKLINGOUTPOUR.TXT'
CASE DEFAULT
CALL ERROR();N4=1
END SELECT
END DO
OPEN (50,FILE=NAM)
WRITE(50,*) ' '
WRITE(50,*) '-----'
WRITE(50,*) '          y|          '
WRITE(50,*) '          \|          '
WRITE(50,*) '          P  _.'''''''''_  P          '
WRITE(50,*) '          <----->O<----->          '
WRITE(50,*) '          |<-----L----->|          '
WRITE(50,*) ' '
WRITE(50,*) ' '
WRITE(50,*) '-----'
WRITE(50,*) 'DISTANCE DATA:'
WRITE(50,*) 'L=',L, '[M]'
WRITE(50,*) 'CRITICAL BUCKLING LOAD'
WRITE(50,*) 'P=',PCR, '[N]'
WRITE(50,*) 'RADIUS OF GYRATION'
WRITE(50,*) 'R=',R, '[m]'
WRITE(50,*) 'CRITICAL BUCKLING STRESS'
WRITE(50,*) 'SCR=',TASH, '[Pa]'
WRITE(50,*) '-----'
CLOSE (50)
DO
PRINT*, '-----'
PRINT*, '===== TO REPEAT PRESS (R) ====='
PRINT*, '----- TO QUIT PRESS (Q) -----'
PRINT*, '-----'
READ*,CH
N1=0;N3=0
SELECT CASE (CH)
CASE ('R','r')
N1=0
CASE ('Q','q')
N1=1
CASE DEFAULT
CALL ERROR()
N3=1
END SELECT
IF (N3 .EQ. 0) EXIT
END DO
IF (N1>0) EXIT
END IF
END

```

Το επόμενο τμήμα που ακολουθεί ανήκει στον πηγαίο κώδικα του προγράμματος για την εισαγωγή υλικών στον κατάλογο υλικών του προγράμματος.

```

!ANAGNOSI KAI APOTHIKESH DEDOMENON APO ARXIO KAI PROS

!*****ADD MATERIAL ON DATA LIST*****
SUBROUTINE AMD()
  INTEGER                ::K,J,I,I1,N1,N2,N3,N4,I2=0
  INTEGER, ALLOCATABLE  ::AFAR(:)
  REAL, ALLOCATABLE     ::YMO(:),GMO(:),N(:),SD(:),
                        ,TD(:),SMEF(:),SMTH(:),TM(:),RO(:)
  CHARACTER(31), ALLOCATABLE ::NAMEOFMA(:)
  CHARACTER(1)          ::LOOP,RC
  REAL                  ::E
  60FORMAT
(1X,I3,2X,' ',A31,' ',1X,F10.4,' ',1X,F10.4,' ',1X,F10.4,' ',1X,F10.4,' ',1X,&
  F10.4,' ',1X,F10.4,' ',1X,F10.4,' ',1X,F10.4,' ',1X,F10.4,' ',1X)
  DO
    N3=1
    PRINT*, '====='
    PRINT*, '====='
    PRINT*, '====='
    PRINT*, '==== ADD MATERIAL ON DATA LIST ====='
    PRINT*, '====='
    PRINT*, '====='
    PRINT*, '====='
    PRINT*, '====TO RETURN TO MAIN MENU (R)====='
    PRINT*, '====TO CONTINUE (C)====='
    PRINT*, '====='
    READ*, RC
    SELECT CASE (RC)
      CASE ('R','r')
      CASE ('C','c')
        N4=1
      CASE DEFAULT
        CALL ERROR()
        N3=0
    END SELECT
    IF (N3==1) EXIT
  END DO
  IF (N4==1) THEN
    DO
      PRINT*, '====='
      PRINT*, '==== GIVE INTAKE NUMBER OF MATERIAL DATA ====='
      PRINT*, '====='
      READ*, J
      if (J==0) call ERROR()
      IF (J>0) EXIT
    DO

```

```

END DO
OPEN (10, FILE='MATERIAL.DAT')
  READ (10, *) K
  REWIND (10)
  ALLOCATE (AFAR(K+J), YMO(K+J), GMO(K+J), N(K+J), SD(K+J), TD(K+J), &
    SMEF(K+J), SMTH(K+J), TM(K+J), RO(K+J), NAMEOFMA(K+J))
  DO I=K, 1, -1
    READ
    AFAR(I), NAMEOFMA(I), YMO(I), GMO(I), N(I), SD(I), TD(I), SMEF(I), SMTH(I), TM(I), RO(I)
  END DO
  REWIND (10)
  I1=K+J
  DO I=K+1, I1, 1
    AFAR(I)=I
    I2=I2+1
    PRINT*, '-----'
    PRINT*, 'GIVE NAME OF', I2, 'INPOUR MATERIAL (ONLY 31 CHARACTERS)'
    PRINT*, '-----'
    READ*, NAMEOFMA(I)
    PRINT*, '-----'
    PRINT*, 'GIVE MODULUS OF ELASTICITY E IN (GPa)'
    PRINT*, '-----'
    READ*, YMO(I)
    PRINT*, '-----'
    PRINT*, 'GIVE SHEAR MODULUS G IN (GPa)'
    PRINT*, '-----'
    READ*, GMO(I)
    PRINT*, '-----'
    PRINT*, 'GIVE POISSONS RATIO  $\mu$ '
    PRINT*, '-----'
    READ*, N(I)
    PRINT*, '-----'
    PRINT*, 'GIVE TENSION YIEND STREGTH IN (GPa)'
    PRINT*, '-----'
    READ*, SD(I)
    PRINT*, '-----'
    PRINT*, 'GIVE SHEAR YIEND STREGTH IN (GPa)'
    PRINT*, '-----'
    READ*, TD(I)
    PRINT*, '-----'
    PRINT*, 'GIVE TENSIONS MAXIMUM STRESS IN (GPa)'
    PRINT*, '-----'
    READ*, SMEF(I)
    PRINT*, '-----'
    PRINT*, 'GIVE COMPRESSIONS MAXIMUM STRESS IN (GPa)'
    PRINT*, '-----'
    READ*, SMTH(I)
    PRINT*, '-----'
    PRINT*, 'GIVE SHEAR MAXIMUM STRESS IN (GPa)'
    PRINT*, '-----'
    READ*, TM(I)
    PRINT*, '-----'

```

```

PRINT*, 'GIVE MATERIALS DENSITY IN (kg/m^3)
PRINT*, '-----'
READ*, RO(I)
END DO
DO I=K+J,1,-1
WRITE
(10, 60)
AFAR(I), NAMEOFMA(I), YMO(I), GMO(I), N(I), SD(I), TD(I), SMEF(I), SMTH(I), TM(I), RO(I)
END DO
WRITE(10,*) 'A/A', NAME, E(GPa), G,
μ, σΔ(MPa), ε
σ τΔ(MPa), σM(TEN) (MPa), σM(comp) (MPa), tM(MPa), ρo(kg/M^3)
CLOSE (10)

DO
N2=0
PRINT*, ' '
PRINT*, '-----'
PRINT*, '==== TO REPEAT PRESS (R) ====='
PRINT*, '==== TO QUIT PRESS (Q) ====='
PRINT*, '-----'
READ*, LOOP
select case (LOOP)
CASE ('R', 'r')
N1=0
CASE ('Q', 'q')
N1=1
CASE DEFAULT
CALL ERROR()
N2=1
END select
IF (N2 .EQ. 0) EXIT
END DO
IF (N1>0) EXIT
END DO
END IF
END

```

Παρακάτω βλέπουμε ένα μικρό αριθμό υλικών που προστέθηκαν με το πρόγραμμα που προηγήθηκε.

7	,copper				119.0000,	42.0000,	0.3500,	266.0000,
161.0000,	385.0000,	0.0000,	0.0000,	8913.0000,				
6	,lead				5.0000,	0.0000,	0.4400,	14.0000,
14.0000,	175.0000,	0.0000,	0.0000,	1400.0000,				
5	,Aluminium(1%Mg)				69.0000,	26.0000,	0.3400,	255.0000,
140.0000,	290.0000,	0.0000,	185.0000,	2710.0000,				
4	,Aluminium(44%Cu)				72.0000,	27.0000,	0.3400,	410.0000,
220.0000,	480.0000,	0.0000,	290.0000,	2800.0000,				
3	,Aluminium(99%)				70.0000,	26.0000,	0.3400,	95.0000,
55.0000,	110.0000,	0.0000,	70.0000,	2710.0000,				
2	,XALIBAS(STIII)				200.0000,	83.0000,	0.0000,	0.3450,
0.2100,	0.4800,	0.0000,	0.0000,	7860.0000,				
1	,XYTOSIDHROS(4.5%C)ASTMA-48				70.0000,	28.0000,	0.2500,	0.2300,
0.0000,	0.3400,	0.6200,	0.3300,	7200.0000,				
A/A	, NAME,				E(GPa),	G,	μ,	σΔ,
τΔ,	σM(TEN),	σM(comp),	tM,	ρo(DENSITY)				

Και τέλος έχουμε όλες τις υπορουτίνες που χρησιμοποιήσαμε συγκεντρωμένες σε ένα αρχείο.

```

!*****SUBROUTINES*****
!METRO ELASTIKOTITAS
SUBROUTINE ME_EL(E)
INTEGER                ::K,J
INTEGER, ALLOCATABLE  ::AFAR(:)
REAL, ALLOCATABLE     ::YMO(:),GMO(:),N(:),SD(:),TD(:)&
                      ,SMEF(:),SMTH(:),TM(:),RO(:)
CHARACTER(30), ALLOCATABLE ::NAMEOFMA(:)
REAL
K=0
11 FORMAT (2X,I4,1X,A30,1X,'E=',F7.2,'GPa')
DO
PRINT*, ' '
PRINT*, '=====!'
PRINT*, '==== GIVE MODULUS OF ELASTICITY (E) IN [GPa] ===='
PRINT*, '==== OR PICK IT FROM LIST ====='
PRINT*, '====TO SEE THE LIST OF MATERIAL PREST 0 AND ENTER ===='
PRINT*, '=====!'
READ*,E
IF ((E.GT.0).AND..NOT.(E.LT.0)) THEN
PRINT*, 'E=',E
ELSE
OPEN (10,FILE='MATERIAL.DAT')
READ (10,*) K
REWIND (10)
ALLOCATE (AFAR(K),YMO(K),GMO(K),N(K),SD(K),TD(K),&
SMEF(K),SMTH(K),TM(K),RO(K),NAMEOFMA(K))
DO I=K,1,-1
READ (10,*) AFAR(I),NAMEOFMA(I),YMO(I),GMO(I)&
,N(I),SD(I),TD(I),SMEF(I),SMTH(I),TM(I),RO(I)
END DO
CLOSE (10)
PRINT*, ' '
PRINT*, '=====!'
PRINT*, 'MATERIAL LIST'
PRINT*, '=====!'
DO I=1,K
WRITE(*,11) AFAR(I),NAMEOFMA(I),YMO(I)
END DO
PRINT*, '=====!'
PRINT*, '==== TO CHOOSE YOUR MATERIAL PRESS THE ORDINAL ===='
PRINT*, '==== NUMBER AND ENTER ====='
PRINT*, '=====!'
READ*, J
PRINT*, ' '
PRINT*, '=====!'
PRINT*, 'CONFIRMATION

```

```

PRINT*, '=====
DO I=1,K
  IF (AFAR(I).EQ.J) THEN
    WRITE(*,11) afar(I),NAMEOFMA(I),YMO(I)
  END IF
E=YMO(I)
END DO
PRINT*, '=====
ENDIF
IF (E.GT.0) EXIT
END DO
END

```

```

!*****
!      YPOLOGISMOS TASEON TENSION
!
!*****

```

```

SUBROUTINE ETASH(P,F,TASH)
  3 FORMAT(1X,' STRESS HAS CALCULETE S=',F11.3,2X,'GPa')
  REAL P,F,TASH
  TASH=P/F
  WRITE(*,3) TASH
  WRITE(30,3) TASH
  WRITE(30,*) '=====
  WRITE(30,*) '=====
  PRINT*, '=====
  PRINT*, '=====
END

```

```

!*****
SUBROUTINE DD()

```

```

  REAL    :: DL,P,L,E,F,TASH
  PRINT*, '=====
  PRINT*, '====          PARAMETERE P,L,F          =====
  PRINT*, '=====
  PRINT*, '====          GIVE P (IN kN)            =====
  PRINT*, '=====
  READ*, P
  PRINT*, '=====
  PRINT*, '====          GIVE L (IN mm)            =====
  PRINT*, '=====
  READ*, L
  PRINT*, '=====
  PRINT*, '====          GIVE F (IN mm^2)          =====
  READ*, F
  PRINT*, '=====
  CALL ME_EL(E)
  DL=(P*L)/((E/10)*F)
  PRINT*, '=====
  PRINT*, 'THE PARAMETERE DL HAS CALCULETE =',DL,'(mm)'
  WRITE(30,*) '=====
  WRITE(30,*) 'E=',E,'[GPa]','P=',P,'[kN]'

```



```

WRITE(30,*) 'L=',L,'[mm]', 'F=',F,'[mm^2]'
WRITE(30,*) '-----'
WRITE(30,*) 'THE PARAMETERE DL HAS CALCULETE =',DL,'(mm)'
CALL ETASH(P,F,TASH)

END

;*****
SUBROUTINE PP()
  REAL    :: DL,P,L,E,F,TASH
  PRINT*, '-----'
  PRINT*, '=====          PARAMETERE DL, L, F          ====='
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE DL (IN mm)          ====='
  PRINT*, '=====          ====='
  READ*, DL
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE L (IN mm)          ====='
  PRINT*, '=====          ====='
  READ*, L
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE F (IN mm^2)          ====='
  READ*, F
  PRINT*, '-----'
  CALL ME_EL(E)
  P=(DL*(E/10)*F)/L
  PRINT*, '-----'
  PRINT*, 'THE PARAMETERE P HAS CALCULETE =',P,'(KN)'
  WRITE(30,*) '-----'
  WRITE(30,*) 'E=',E,'[GPa]', 'DL=',DL,'[mm]'
  WRITE(30,*) 'L=',L,'[mm]', 'F=',F,'[mm^2]'
  WRITE(30,*) '-----'
  WRITE(30,*) 'THE PARAMETERE P HAS CALCULETE =',P,'(KN)'
  CALL ETASH(P,F,TASH)

END

;*****

SUBROUTINE LL()
  REAL    :: DL,P,L,E,F,TASH
  PRINT*, '-----'
  PRINT*, '=====          PARAMETERE DL, P, F          ====='
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE DL (IN mm)          ====='
  PRINT*, '=====          ====='
  READ*, DL
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE P (IN KN)          ====='
  PRINT*, '=====          ====='
  READ*, P
  PRINT*, '=====          ====='
  PRINT*, '=====          GIVE F (IN mm^2)          ====='
  READ*, F

```

```

PRINT*, '=====
CALL ME_EL(E)
L=(DL*(E/10)*F)/P
PRINT*, '=====
PRINT*, 'THE PARAMETERE L HAS CALCULETE =' , L, '(mm)'
WRITE(30,*) '=====
WRITE(30,*) 'E=' , E, '[GPa]', 'DL=' , DL, '[mm]'
WRITE(30,*) 'P=' , P, '[kN]', 'F=' , F, '[mm^2]'
WRITE(30,*) '=====
WRITE(30,*) 'THE PARAMETERE L HAS CALCULETE =' , L, '(mm)'
CALL ETASH(P,F,TASH)
END
|*****

```

```

SUBROUTINE FF()
REAL    :: DL,P,L,E,F,TASH
PRINT*, '=====
PRINT*, '=====          PARAMETERE DL, L, P          =====
PRINT*, '=====
PRINT*, '=====          GIVE DL (IN mm)          =====
PRINT*, '=====
READ*, DL
PRINT*, '=====
PRINT*, '=====          GIVE L (IN mm)          =====
PRINT*, '=====
READ*, L
PRINT*, '=====
PRINT*, '=====          GIVE P (IN KN)          =====
READ*, P
PRINT*, '=====
CALL ME_EL(E)
F=(P*L)/(DL*(E/10))
PRINT*, '=====
PRINT*, 'THE PARAMETERE F HAS CALCULETE =' , F, '(mm^2)'
WRITE(30,*) '=====
WRITE(30,*) 'E=' , E, '[GPa]', 'DL=' , DL, '[mm]'
WRITE(30,*) 'L=' , L, '[mm]', 'P=' , P, '[kN]'
WRITE(30,*) '=====
WRITE(30,*) 'THE PARAMETERE F HAS CALCULETE =' , F, '(mm^2)'
CALL ETASH(P,F,TASH)
END
|*****

```

```

SUBROUTINE EE()
PRINT*, '=====
PRINT*, '=====          PARAMETERE DL, L, F, P          =====
PRINT*, '=====
PRINT*, '=====          GIVE DL (IN mm)          =====
PRINT*, '=====
READ*, DL

```

```

PRINT*, '===='
PRINT*, '====' GIVE L (IN mm) '===='
PRINT*, '===='
READ*, L
PRINT*, '===='
PRINT*, '====' GIVE F (IN mm^2) '===='
PRINT*, '===='
READ*, F
PRINT*, '===='
PRINT*, '====' GIVE P (IN kN) '===='
READ*, P
PRINT*, '-----'
E=(P*L)*10/(F*DL)
PRINT*, '-----'
PRINT*, 'THE PARAMETERE E HAS CALCOLETE =',E,'GPa'
WRITE(30,*) '-----'
WRITE(30,*) 'DL=',DL,' [mm] ', 'L=',L,' [mm] '
WRITE(30,*) 'F=',F,' [mm^2] ', 'P=',P,' [kN] '
WRITE(30,*) '-----'
WRITE(30,*) 'THE PARAMETERE E HAS CALCOLETE =',E,' (kN/mm^2) '
CALL ETASH(P,F,TASH)
END
!*****
!====REACTION====
SUBROUTINE RETON(A,B,L,P1,P2,VA,VB)
REAL ::A,B,L,P1,P2,VA,VB
VB=(P1*A+P2*(B+A))/L
VA=P1+P2-VB
END
!*****
!*****ERROR SUBROUTINE*****
SUBROUTINE ERROR()
PRINT*, '-----'
PRINT*, '-----'
PRINT*, '***** ERROR: WRONG PARAMETERE *****'
PRINT*, '***** TRY AGAIN *****'
PRINT*, '-----'
PRINT*, '-----'
END
!*****
!*****EPILOGH ONOMATOS*****

SUBROUTINE DATANAME(T,NAM)
INTEGER ::T
CHARACTER(31) ::NAM,CH
PRINT*, '-----'
PRINT*, '====DO YOU WANT TO NAME THE OUTPOUR FILE?===='
PRINT*, '====FOR YES PRESS (Y)===='
PRINT*, '====FOR NO PRESS (N)===='
PRINT*, '-----'

```

```

READ*, CH
SELECT CASE (CH)
  CASE ('Y','y')
    PRINT*, ' '
    PRINT*, '====='
    PRINT*, '===GIVE OUTPOURDATA FILE NAME (ONLY 31 CHARACTERS)=== '
    PRINT*, '=== EXAMPLE: OUTPOURDATA.TXT, DATA.DAT === '
    PRINT*, '====='
    READ*, NAM
    T=T+1
  CASE ('N','n')
    PRINT*, ' '
    PRINT*, '====='
    PRINT*, '===          DEFAULT NAME "FLEXIONOUTPOUR.TXT"          === '
    PRINT*, '====='
    NAM='FLEXIONOUTPOUR.TXT'
    T=T+1
  case default
    CALL ERROR()

END SUBROUTINE DATANAME
!*****
!KLISH THS EPTHIAS ELAXISTON TETRAGONON
REAL FUNCTION A(SXY,SX,Y_,SX2,X_)
  REAL  ::SXY,SX,Y_,SX2,X_
  A=(SXY-SX*Y_)/(SX2-SX*X_)
END FUNCTION
!*****
!ROYTINA EPILOGHS TON YSERKOMENON DEDOMENON
SUBROUTINE PORS(N2)
  INTEGER ::N2,N3
  CHARACTER(3) ::PS
  DO
    N3=1
    PRINT*, '====='
    PRINT*, '===          IS YOUR DATA P-DL OR S-E          === '
    PRINT*, '===FOR P-DL PRESS (P)          === '
    PRINT*, '===FOR S-E PRESS (S)          === '
    PRINT*, '====='
    READ*, PS
    SELECT CASE (PS)
      CASE ('P','p')
        N2=1
      CASE ('S','s')
        N2=0
      CASE DEFAULT
        CALL ERROR()
        N3=0
    END SELECT
    IF (N3==1) EXIT
  END DO
END

```

```

!*****
!EPILOGH EFELKISMOY H EVRESH TOY ORIOY ELASTIKOTHTAS
SUBROUTINE EFEPL()
  CHARACTER(3)  ::TP
  INTEGER       ::N
  DO
    N=0
    PRINT*, '=====
    PRINT*, '====          FOR TENSION OF MATERIALS          =====
    PRINT*, '====          PRESS T OR t          =====
    PRINT*, '=====
    PRINT*, '=====
    PRINT*, '====          FOR PROPORTIONAL LIMIT          =====
    PRINT*, '====          PRESS P OR p          =====
    PRINT*, '=====
    PRINT*, '=====
    PRINT*, '====          TO RETURN TO MAIN MENU          =====
    PRINT*, '====          PRESS R OR r          =====
    PRINT*, '=====
    READ*, TP
    SELECT CASE (TP)
      CASE ('T','t')
        CALL TENSION()
      CASE ('p','P')
        CALL PL()
      CASE ('R','r')
        N=1
      CASE DEFAULT
        CALL ERROR()
        N=1
    END SELECT
    IF (N==1) EXIT
  END DO
END

```

Για να κατανοήσετε καλύτερα του κώδικα εγκαταστήστε το πρόγραμμα Scintilla που είναι στο φάκελο EXECUTABLE, τρέξτε τα αρχεία με αυτό. Για καλύτερη παρουσίαση από τη καρτέλα LANGUAGE επιλέξτε *FORTTRAN*. Τα αρχεία είναι στο φάκελο *SOURCE CODE* και είναι τα εξής: *SOFM.F90* (κώδικας οδηγού προγράμματος), *EFE.F90* (κώδικας προγράμματος εφελκυσμού), *PL.F90* (κώδικας προγράμματος για την εύρεση του ορίου αναλογίας), *KAM.F90* (κώδικας προγράμματος για την κάμψη), *LIG.F90* (κώδικας για το πρόγραμμα λογιισμού), *AMD.F90* (κώδικας προγράμματος εισαγωγής υλικών), *SUB.F90* (υποπρογράμματα).

#### 4 ΟΔΗΓΙΕΣ ΧΡΗΣΕΩΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Κάθε κατασκευή, μηχανήμα, εργαλείο και πρόγραμμα πρέπει να έχει τις οδηγίες χρήσεως. Ο λόγος είναι ότι με την απουσία οδηγιών χρήσεως θα είχαμε λανθασμένη χρήση των εργαλείων, με συνέπεια να μην έχουμε τα επιθυμητά αποτελέσματα. Έτσι, με τη λανθασμένη χρήση, υπάρχει μεγάλη πιθανότητα να προκληθούν τραυματισμοί ή ακόμα και θάνατος. Ευτυχώς όμως στην περίπτωση μας με τη χρήση του προγράμματος δεν υπάρχει άμεσος κίνδυνος. Υπάρχει όμως έμμεσος κίνδυνος γιατί, αν χρησιμοποιήσει κάποιος το πρόγραμμα για να κάνει υπολογισμούς για μια κατασκευή που τη χειρίζονται-χρησιμοποιούν άνθρωποι και λόγω λάθους χειρισμού του προγράμματος τα αποτελέσματα που θα προκύψουν δεν ανταποκρίνονται στην πραγματικότητα, τότε σίγουρα θα προκληθούν έστω μικροτραυματισμοί. Επομένως καθίσταται αναγκαίο να γραφτεί ένα εγχειρίδιο χρήσεως, για να μπορέσει να το χρησιμοποιήσει κάποιος, καθώς και για να βγάλει σωστά αποτελέσματα.

Το πρόγραμμα που θα τρέξουμε ονομάζεται SOFM (STRENGTH OF MATERIALS) στα ελληνικά ΑΝΤΟΧΗ ΤΩΝ ΥΛΙΚΩΝ. Κατά το τρέξιμο, βλέπουμε τον τίτλο του προγράμματος και στη συνέχεια τέσσερις επιλογές χρήσης:

- a. Για τον εφελκυσμό-θλίψη (TENSION OF MATERIALS).
  - a. Εφελκυσμός-θλίψη
  - b. Όριο αναλογίας
- b. Για την κάμψη (FLEXION OF MATERIALS).
- c. Για τον λυγισμό (BUCKLING OF BARS).
- d. Και τέλος το πρόγραμμα προσθήκης υλικού στον κατάλογο υλικών του προγράμματος (ADD MATERIAL ON DATA LIST).



Παρακάτω βλέπουμε το μήνυμα λάθους που εμφανίζεται κατά την εισαγωγή λανθασμένου γράμματος. Επίσης, το μήνυμα αυτό εμφανίζεται κάθε φορά που γίνεται κάποια λάθος επιλογή.

```
*****
*****
*****          ERROR: WRONG PARAMETERE          *****
*****          TRY AGAIN                          *****
*****
*****
```

1. Πληκτρολογώντας T ή t και πατώντας ENTER μπαίνουμε στο πρόγραμμα για τον εφελκυσμό-θλίψη, οπότε βλέπουμε μπροστά μας τώρα δύο άλλες επιλογές. Μια για τον εφελκυσμό-θλίψη και μια για την εύρεση του ορίου αναλογίας σε ένα διάγραμμα σ-ε. Παρακάτω βλέπουμε τι ακριβώς θα είναι μπροστά μας.

```
=====
=====
=====          FOR TENSION OF MATERIALS          =====
=====          PRESS T OR t                        =====
=====
=====
=====          FOR PROPORTIONAL LIMIT            =====
=====          PRESS P OR p                        =====
=====
=====
=====          TO RETURN TO MAIN MENU            =====
=====          PRESS R OR r                        =====
=====
```

Εδώ πάλι πληκτρολογούμε το γράμμα, κεφαλαίο ή μικρό, που αντιστοιχεί στην κάθε λειτουργία. Σε περίπτωση λάθους εμφανίζεται μήνυμα λάθους, δίνοντας μας την ευκαιρία να ξαναπροσπαθήσουμε.





```

=====
---      DEFAULT NAME "OUTPOURDATA.TXT"      ---
=====

```

Άσχετα με την επιλογή που κάναμε στο προηγούμενο βήμα, κάθε φορά θα τυπώνεται κάτω από την επιλογή η εξής διεπιφάνεια επιλογών:

```

=====
==== GIVE THE PARAMETER OFF YOU WISH TO CALCULETE ====
====          AND PRESS ENTER          ====
====                                     ====
====          DL P L F E          ====
====          OR Q TO QUIT          ====
=====

```

Η παραπάνω διεπιφάνεια μας δηλώνει ποια από τις παραμέτρους της σχέσης  $\Delta l = \frac{P \cdot l}{E \cdot F}$  θέλουμε να υπολογίσει, δηλαδή ποια μας είναι άγνωστη. Η επιλογή γίνεται πάλι με κεφαλαία ή μικρά, καθώς επίσης μας δίνει τη δυνατότητα επιλογής εξόδου από το πρόγραμμα, με q ή Q. Αν κάνουμε λάθος επιλογή, θα μας τυπώσει μήνυμα λάθους, δίνοντάς μας τη δυνατότητα να ξαναπροσπαθήσουμε. Το μήνυμα λάθους είναι το ίδιο που αναφέραμε και παρουσιάσαμε στην αρχή. Ακολούθως, αφού κάνουμε τη σωστή επιλογή το πρόγραμμα μας ζητά να του δώσουμε τις γνωστές τιμές μία μία. Για παράδειγμα, όταν επιλέξουμε για άγνωστο το DL θα μας τυπώσει το μήνυμα,

```

=====
====          PARAMETERE P, L, F          ====
====                                     ====
====          GIVE P (IN KN)          ====
=====

```

όπου μας ζητάει να του δώσουμε την εφαρμοζόμενη δύναμη σε [kN]. Στη συνέχεια, πληκτρολογούμε την τιμή της δύναμης και πατάμε *ENTER* και ακολούθως θα μας τυπώσει το παρακάτω μήνυμα, με το

οποίο μας ζητά την επόμενη γνωστή παράμετρο, δηλαδή το αρχικό ολικό μήκος της ράβδου σε  $[mm]$ .

```
====
GIVE L (IN mm)
====
```

Το επόμενο μήνυμα μας ζητά να του δώσουμε την επιφάνεια της κάθετης διατομής της ράβδου σε  $[mm^2]$ .

```
====
GIVE F (IN mm^2)
====
```

Το επόμενο μήνυμα που εμφανίζεται, είναι ένα μήνυμα που μας ζητά να δώσουμε το μέτρο ελαστικότητας ή να το επιλέξουμε από πίνακα πατώντας το 0, αφού πρώτα δώσουμε τις τιμές των προηγούμενων. Το μήνυμα είναι το εξής:

```
====
GIVE MODULUS OF ELASTICITY (E) IN [GPa]
OR PICK IT FROM LIST
TO SEE THE LIST OF MATERIAL PRESS 0 AND ENTER
====
```

Εν συνεχεία, τυπώνει τη λίστα με τα υλικά και εμφανίζεται το παρακάτω μήνυμα

```
====
TO CHOOSE YOUR MATERIAL PRESS THE ORDINAL
NUMBER AND ENTER
====
```

όπου μας αναφέρει ότι για την επιλογή του υλικού πατάμε τον αύξοντα αριθμό που είναι μπροστά από το υλικό που θέλουμε και ακολούθως το *ENTER*. Τότε γίνεται μια επιβεβαίωση του υλικού που επιλέξαμε και αμέσως μας τυπώνει την άγνωστη παράμετρο, αλλά και την τάση που εφαρμόζεται.

Η παραπάνω διαδικασία είναι η ίδια για κάθε άγνωστο εκτός της περίπτωσης στην οποία για άγνωστο έχουμε το μέτρο ελαστικότητας  $E$ .

Εκεί η μόνη διαφοροποίηση που έχουμε είναι ότι δεν μας ζητά να του δώσουμε το  $E$  ή να επιλέξουμε υλικό, αφού είναι άγνωστα.

- 1.2. Αν επιλέξαμε να πληκτρολογήσουμε  $P$  ή  $p$ , δηλαδή να εκτελέσουμε το πρόγραμμα για την εύρεση του ορίου αναλογίας, τότε μετά το πάτημα του *ENTER* θα εμφανιστεί το παρακάτω μήνυμα, που αναγγέλλει τη λειτουργία του προγράμματος και μας ζητά να πληκτρολογήσουμε το αρχικό μήκος της ράβδου σε  $[m]$ .

```
=====
=====
====
====          PROPORTIONAL LIMIT          ====
====
=====
=====
====TO RETURN TO MAIN MENU (R)          =====
====TO CONTINUE (C)                    =====
=====
```

Για να συνεχίσουμε πατάμε  $C$  και για να επιστρέψουμε στο κίριος μενού πατάμε  $R$ .

```
=====
====          GIVE ORIGINAL LENGTH L IN (m)          =====
=====
```

Με το πάτημα του *ENTER* και την αποδοχή του αρχικού μήκους, τυπώνεται το ακόλουθο μήνυμα που μας ζητά να δώσουμε το εμβαδόν της κάθετης διατομής της ράβδου.

```
=====
====          GIVE CROSS-SECTIONAL AREA F IN (m^2)          =====
=====
```

Μετά την εισαγωγή του εμβαδού τυπώνεται το παρακάτω μήνυμα που μας ρωτά αν θέλουμε να εισάγουμε δεδομένα από αρχείο ή αν θα του τα δώσουμε χειροκίνητα.

```

=====
---DO YOU WANT TO READ DATA FROM FILE OR   ---
---      TO GIVE IT MANUALLY ?              ---
---FROM FILE PRESS (F)                      ---
---FOR MANUALLY PRESS (M)                   ---
=====

```

Για την επιλογή της κατάλληλης λειτουργίας πληκτρολογούμε  $F$  ή  $f$  για την ανάγνωση από αρχείο και  $M$  ή  $m$  για την χειροκίνητη εισαγωγή δεδομένων. Στην περίπτωση που κάνουμε λάθος τυπώνεται το γνωστό μήνυμα λάθους. Αν κάνουμε την επιλογή για την ανάγνωση από αρχείο θα τυπωθεί ένα μήνυμα που μας κάνει την ερώτηση του τι είδους δεδομένα έχουμε, δηλαδή αν έχουμε δύναμη συναρτήσει της διαφοράς μήκους ή αν έχουμε τάση συναρτήσει της ανοιγμένης παραμόρφωσης. Αυτή η επιλογή εμφανίζεται και όταν επιλέξουμε χειροκίνητη εισαγωγή δεδομένων.

```

=====
---      IS YOUR DATA P-DL OR S-E          ---
---FOR P-DL PRESS (P)                      ---
---FOR S-E PRESS (S)                      ---
=====

```

Μετά από αυτό μας ζητά να του δώσουμε το πλήθος των δεδομένων που θα διαβάσει από το αρχείο. Αυτή η επιλογή εμφανίζεται και κατά την χειροκίνητη εισαγωγή δεδομένων.

```

=====
---      GIVE INTAKE NUMBER OF DATA        ---
=====

```

Ακολούθως μας ζητά να του δώσουμε το όνομα του αρχείου όπου θα εισάγουμε τα δεδομένα.

```

=====
---      GIVE FILE NAME (ONLY 31 CHARACTERS) ---
---      EXAMPLE: OUTPOURDATA.TXT, DATA.DAT ---
=====

```

Τα δεδομένα στο αρχείο πρέπει να είναι σε στήλες με την τελεία για το σημείο των δεκαδικών και κάθε στήλη να απέχει από την άλλη ένα διάκενο. Για την περίπτωση που έχουμε δύναμη συναρτήσει της διαφοράς

μήκους, στην πρώτη στήλη έχουμε τη διαφορά μήκους, ενώ στη δεύτερη έχουμε δύναμη. Στην περίπτωση που έχουμε την τάση συναρτήσει της ανοιγμένης παραμόρφωσης, στην πρώτη στήλη μπαίνει η ανοιγμένη παραμόρφωση και στη δεύτερη η τάση.

Όταν γίνει η χειροκίνητη εισαγωγή δεδομένων και αφού δηλώσουμε το είδος και το πλήθος των δεδομένων, μας ζητά να δίνουμε ένα ένα, με τη σειρά που μας τα ζητά τα δεδομένα. Πρώτα τη διαφορά μήκους και μετά τη δύναμη, αν επιλέξαμε τα δεδομένα να είναι δύναμη συναρτήσει της διαφοράς μήκους και την ανοιγμένη παραμόρφωση πρώτα για την περίπτωση που επιλέξαμε την τάση συναρτήσει της ανοιγμένης παραμόρφωσης μετά την τάση.

Όταν εισάγουμε τα δεδομένα με οποιοδήποτε τρόπο και μετά την επεξεργασία τους, μας τυπώνει ένα μήνυμα με το όριο αναλογίας και τον συντελεστή μορφής, της γραφικής παράστασης που σχηματίζουν τα δεδομένα.

```
=====
==== THE POINT X IS PROPORTIONAL LIMIT      =====
=====
=====
==== GRAPH LOOK COEFFICIENT IS Y            =====
=====
=====
==== IF POINT IS WRONG CHANGE                =====
==== GRAPH LOOK COEFFICIENT TO FIXT         =====
=====
====DO YOU WANT TO CHANGE                    =====
=====
====FOR YES PRESS (Y)                        =====
====FOR NO PRESS (N)                         =====
=====
```

Στην περίπτωση που το σημείο είναι λάθος μπορούμε να αλλάξουμε το συντελεστή μορφής, ώστε να μπορέσουμε να πλησιάσουμε το πραγματικό σημείο.



Ακολούθως, τα μήκη  $A$  και  $B$  σε  $[m]$  όπου οπωσδήποτε το  $A$  πρέπει να είναι μικρότερο του  $L$  και το  $B$  μικρότερο του  $A$  αλλιώς θα τυπωθεί μήνυμα λάθους δίνοντας μας τη δυνατότητα να ξαναπροσπαθήσουμε.

```

=====
=====
=====
=====
GIVE A (IN m)
=====
=====
=====
=====
GIVE B (IN m)
=====
=====

```

Στη συνέχεια μας ζητά τις τιμές των δυνάμεων σε  $[N]$  και τέλος την τιμή της ροπής αδράνειας  $I$  σε  $[m^4]$ .

```

=====
=====
=====
=====
GIVE P1 (IN N)
=====
=====
=====
=====
GIVE P2 (IN N)
=====
=====
=====
=====
GIVE I (IN m^4)
=====
=====

```

Συνεχίζοντας, μας ζητά να δώσουμε μια τιμή του μέτρου ελαστικότητας ή να επιλέξουμε υλικό. Η διαδικασία είναι η ίδια με αυτή που αναφέραμε και στον εφελκυσμό.

Μετά την επιλογή ή δήλωση του μέτρου ελαστικότητας μας ζητά να δώσουμε το μέγιστο μήκος γαλό την κεντρική γραμμή της ράβδου σε  $[m]$ . Ακολούθως, τυπώνονται τα αποτελέσματα των υπολογισμών στην οθόνη, δίνοντας μας και τη δυνατότητα να τα τυπώσουμε σε αρχείο της επιλογής μας. Σε περίπτωση που δεν επιλέξουμε αρχείο, τότε τυπώνει τα αποτελέσματα στο αρχείο OUTPOURDATA.TXT. Στην επόμενη σελίδα βλέπουμε την μορφή των εξαγομένων τιμών.







CRITICAL BUCKLING LOAD

P=' ,PCR, '(N)

RADIUS OF GYRATION

R=' ,R, '(m)

CRITICAL BUCKLING STRESS

SCR=' ,TASH, '(Pa)

-----  
Ακολουθεί μήνυμα με το ερώτημα αν θέλουμε να ονομάσουμε το αρχείο των δεδομένων ή όχι. Στην περίπτωση που θέλουμε να ονομάσουμε το αρχείο, το μέγιστο πλήθος χαρακτήρων είναι 31 μαζί με το διαχωριστικό στίγμα και το ακρώνυμο του είδους των αρχείων. Τέλος, μας ρωτά εάν θέλουμε να επαναλάβουμε την παραπάνω διαδικασία.

4. Η τελευταία επιλογή που απομένει, είναι το πρόγραμμα εισαγωγής υλικών στη λίστα υλικών. Πρώτο μήνυμα που παρουσιάζεται κατά την εκτέλεση του προγράμματος είναι το παρακάτω, που μας ρωτά πόσο είναι το πλήθος των υλικών που θα εισάγουμε.

-----  
---- GIVE INTAKE NUMBER OF MATERIAL DATA ----  
-----

Μετά την εισαγωγή του πλήθους υλικών που θέλουμε να προσθέσουμε στη λίστα των υλικών, μας ζητά το όνομα του πρώτου υλικού και τα δεδομένα που αντιστοιχούν για το κάθε υλικό με τη σειρά που είναι στον πίνακα υλικών (σελίδα 63 στο βιβλίο του κυρίου Βουθούνης). Και τέλος με την εισαγωγή όλων των δεδομένων τυπώνεται μήνυμα εάν θέλουμε να επαναλάβουμε τη διαδικασία εισαγωγής δεδομένων. Τα υλικά που είναι μέσα στον κατάλογο υλικών μπορούμε να τον δούμε στο αρχείο με την ονομασία *MATERIAL.DAT*.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Παναγιώτης Ανδρ. Βουθούνης. Τεχνική Μηχανική: Αντοχή των Υλικών, 6 έκδοση, Αθήνα 2002

Δημήτρης Σ. Ματαράς-Φραγκίσκος Α. Κουτελιέρης, Προγραμματισμός για Επιστήμονες και Μηχανικούς: FORTRAN 90/95, νέα έκδοση, εκδόσεις ΤΖΟΛΑ, Θεσσαλονίκη 200