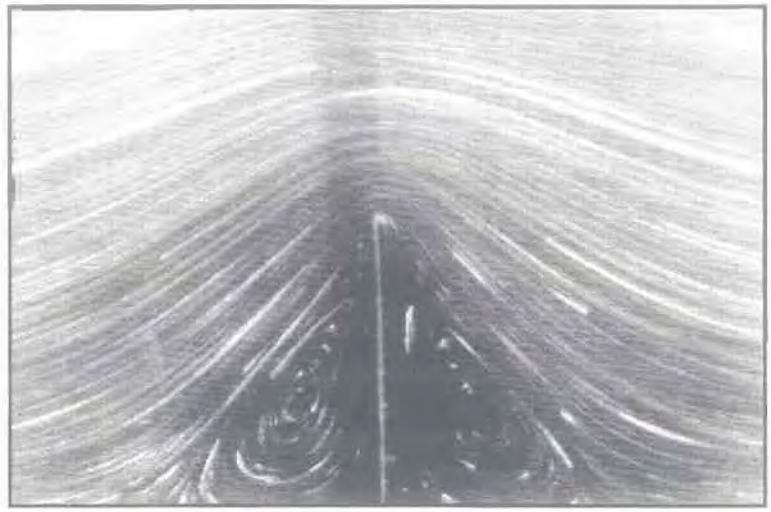
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ



ΑΡΙΘΜΗΤΙΚΑ ΠΕΙΡΑΜΑΤΑ ΕΡΠΟΥΣΑΣ ΡΟΗΣ ΓΥΡΩ ΑΠΟ ΕΔΡΑΣΜΕΝΟ ΕΜΠΟΔΙΟ ΜΕ ΤΗΝ ΜΕΘΟΔΟ ΤΩΝ ΠΕΠΕΡΑΣΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ.

ΕΠΙΜΕΛΕΙΑ:
ΠΑΖΗΣ ΔΗΜΗΤΡΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΜΑΑΑΜΑΤΑΡΗΣ ΝΙΚΟΑΑΟΣ

ΒΟΑΟΣ, ΙΟΥΛΙΟΣ 2000

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΠΕΡΙΛΗΨΗ

Στην εργασία αυτή μελετάται η έρπουσα ροή γύρω από κεκλιμένο φράκτη, μεταβαλλόμενης κλίσεως, με την μέθοδο των πεπερασμένων στοιχείων. Έχει γραφεί πρόγραμμα στη FORTRAN που προσομοιάζει αρχικά την ροή γύρω από κεκλιμένο φράκτη γωνίας 75° και συγκρίνεται με το εργαστηριακό πείραμα που πραγματοποίησε ο Taneda το 1979, προβλέποντας τις ίδιες τάσεις. Στην συνέχεια γίνεται πρόβλεψη της ροής για μεταβαλλόμενη γωνία κλίσεως του φράκτη. Από τον υπολογισμό των ροϊκών γραμμών προκύπτουν χρήσιμα συμπεράσματα για την έρπουσα ροή, όσον αφορά το μέγεθος των στροβίλων που δημιουργούνται και την αλληλεπίδρασή τους. Έγιναν περαιτέρω υπολογισμοί των διατμητικών τάσεων στην βάση του υπολογιστικού πεδίου καθώς και των ισοβαρών και ισοστροβιλωδών καμπυλών γύρω από τον φράκτη, επιτρέποντας την πληρέστερη δυνατή μελέτη των ροϊκών φαινομένων. Επίσης παρουσιάζονται εν συντομία αποτελέσματα για έρπουσα ροή γύρω από εδρασμένο εμπόδιο μεταβαλλόμενου πλάτους, όπου παρατηρούνται συνθήκες ροής ***backward facing step***, όταν το πλάτος του εμποδίου είναι μεγαλύτερο του 4L (L:ύψος του εμποδίου). Το μήκος ανακυκλοφορίας μεταβάλλεται εκθετικά όταν το πάχος του εμποδίου από πολύ λεπτό (0,01L) καταλήγει στα 4L.

2.ΕΙΣΑΓΩΓΗ

2.1) ΓΕΝΙΚΑ ΠΕΡΙ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΡΕΥΣΤΟΜΗΧΑΝΙΚΗΣ

Η υπολογιστική ρευστομηχανική είναι ένας σχετικά νέος κλάδος στην επιστήμη των Ρευστών. Μέχρι το τέλος της δεκαετίας του 1960 οι λύσεις των προβλημάτων της ρευστομηχανικής προέρχονται είτε από πειραματική προσομοίωση των προβλημάτων, είτε από αναλυτικές λύσεις απλοποιημένων εξισώσεων με παραδοχές των οποίων η αξιοπιστία είναι αμφισβητήσιμη(10). Με το πέρασμα των χρόνων όμως, οι υπολογιστές αναδείχθηκαν σε ένα ισχυρό εργαλείο σε όλες τις επιστήμες που χρόνο με τον χρόνο η χρησιμότητα και η ισχύ τους αυξάνεται με γεωμετρική πρόοδο. Έτσι και στην επιστήμη των ρευστών, η εισαγωγή των ηλεκτρονικών υπολογιστών έγινε ακαριαία, επιλύνοντας σημαντικά προβλήματα με την αριθμητική προσέγγιση των διαφορικών εξισώσεων που εκφράζουν μαθηματικά τα πεδία ροής.

Η εφαρμογή αυτή, η οποία συνεχώς εξελίσσεται παράλληλα με την αύξηση της ισχύος των ηλεκτρονικών υπολογιστών, έχει σαν αποτέλεσμα τον σχεδιασμό και τη βελτίωση των μηχανολογικών ή αεροπορικών κατασκευών καθώς επίσης και την εκτίμηση της αποδοτικότητας των θερμικών διεργασιών, κ.ο.κ. Η επίλυση των διαφορικών εξισώσεων μας επιτρέπει πέρα από την επαλήθευση των πειραματικών αποτελεσμάτων, να μελετήσουμε σε μεγαλύτερο βάθος τα ροϊκά φαινόμενα και φυσικά με μικρότερο κόστος, λόγω του γεγονότος ότι το κόστος των εργαστηριακών πειραμάτων είναι αρκετά υψηλό σε σύγκριση με το αριθμητικό.

2.2) ΚΙΝΗΤΡΟ ΤΗΣ ΕΡΓΑΣΙΑΣ

Από την στιγμή που είναι εφικτή η μελέτη των ροϊκών φαινομένων σε βάθος, είναι απαραίτητο να δημιουργηθεί μία τράπεζα δεδομένων στην οποία να έχουν πρόσβαση τα διεθνή ερευνητικά κέντρα, όπου να είναι καταχωρημένα όλα τα δεδομένα τα οποία απορρέουν από την ενδελεχέστερη μελέτη των φαινομένων της ροής. Έχοντας σαν κίνητρο την συνεισφορά σ' αυτήν την τράπεζα δεδομένων, πραγματοποιήθηκε αυτή η εργασία μελετώντας την ροή γύρω από κεκλιμένο φράκτη σε πολύ μικρό αριθμό Reynolds.

2.3) ΕΡΠΟΥΣΑ ΡΟΗ

Οι ροές αυτές που χαρακτηρίζονται από μικρούς αριθμούς Re ονομάζονται **έρπουσες ροές (creeping flows).** Το χαρακτηριστικό της ροής αυτής είναι ότι λόγω του μικρού αριθμού Re, ο όρος της αδράνειας, $\rho * Du/Dt$, δεν παίζει σημαντικό ρόλο στην εξίσωση Navier-Stokes. Η κατανομή της στροβιλότητας σ' αυτές τις ροές εξαρτάται από τη διάχυση, ενώ ο ρόλος της συναγωγής γίνεται λιγότερο σημαντικός και συχνά απαλείφεται (γίνεται μηδενικός).

Το ενδιαφέρον των μελετητών επικεντρώνεται συνήθως σε ροές που έχουν υψηλό αριθμό Re: για παράδειγμα σε ροές γύρω από αεροπλάνα, πλοία, ποτάμια ή μηχανοκίνητες ρουκέτες. Παρόλα αυτά, πολλές πρακτικές εφαρμογές, όπως η ρευστομηχανική συμπεριφορά των λιπαντικών, ως επίσης καινούργιες βιομηχανικές διαδικασίες, όπως η κατασκευή του έγχρωμου φιλμ και των μαγνητικών κασετών ή διεργασίες μικροηλεκτρονικής λαμβάνουν χώρα σε συνθήκες έρπουσας ροής. Αν το αντικείμενο το επεκτείνουμε και σε άλλες μορφές όπως η ροή της βροχής με τη μορφή σταγόνας, η ροή του γυαλιού, οι διεργασίες πολυμερών είναι αντιληπτό ότι η μελέτη ερπουσών ροών έχει ευρύ αντικείμενο και έχει γίνει απαραίτητη στις μέρες μας λόγω των ποικίλων εφαρμογών στην βιομηχανία.

2.4) ΠΡΟΗΓΟΥΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Παρ' όλο που υπάρχουν εκτενείς εργαστηριακές πειραματικές εργασίες για τις έρπουσες ροές (πχ S. Taneda 1979), η μελέτη αυτών με μεθόδους της Υπολογιστικής Ρευστομηχανικής δεν έχουν τύχει ανάλογης προσοχής. Συνήθως η δραστηριότητα περιορίζεται στην δοκιμή κατά πόσο μία υπολογιστική μέθοδος αναπαράγει τα εργαστηριακά πειράματα στα ποιοτικά του χαρακτηριστικά. Το 1981 οι Gresho & Leone προσπάθησαν να προσεγγίσουν αριθμητικά ένα από τα πειράματα του Taneda, για ροή γύρω από οριζόντιο εδρασμένο εμπόδιο, με την μέθοδο των πεπερασμένων στοιχείων και στην συνέχεια ο Higdon το 1985, την ίδια ροή, με την Boundary Integral μέθοδο. Τα αποτελέσματα των δύο αυτών εργασιών προσέγγιζαν μόνο ποιοτικά το εργαστηριακό πείραμα, κάτι που δικαιολογείται αν αναλογιστούμε την ελλιπή υπολογιστική ισχύ της εποχής. Ακόμη και σήμερα, σε εργασίες έγκριτων επιστημόνων (11) αναφέρονται υπολογισμοί για έρπουσες ροές χωρίς την ανάλογη σύγκριση με διαθέσιμα εργαστηριακά πειράματα. Στην συνέχεια το 1997 ο Φράγκος και οι συνεργάτες του, με την μέθοδο των πεπερασμένων στοιχείων και εκμεταλλευόμενοι την αλματώδη ανάπτυξη των υπολογιστών και φυσικά της υπολογιστικής ισχύος, είχαν ικανοποιητική συμφωνία με το εργαστηριακό πείραμα του Taneda. Αυτή η εργασία έγινε στην κατεύθυνση να μελετήσουμε σε βάθος τα εργαστηριακά αυτά πειράματα, που έγιναν πριν από 20 χρόνια και που μας δίνουν πληροφορίες μόνο για τις ροϊκές γραμμές.

2.5) ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ

Έμφαση στην συγγραφή αυτής της εργασίας δόθηκε στην παρουσίαση των αποτελεσμάτων και στην στρατηγική του προγραμματισμού της μεθόδου των πεπερασμένων στοιχείων, μιας και στην ελληνική βιβλιογραφία είναι πολύ σπάνιες τέτοιες αναφορές κυρίως ως προς τον προγραμματισμό.

Στο κεφάλαιο 3 παρουσιάζονται οι διέπουσες εξισώσεις του προβλήματος καθώς και οι οριακές συνθήκες. Εξηγείται ο σχεδιασμός του υπολογιστικού πεδίου και ο τρόπος αδιαστατοποιήσεως των εξισώσεων. Στο κεφάλαιο 4 γίνεται η διατύπωση των διεπουσών εξισώσεων με την μέθοδο των πεπερασμένων στοιχείων και στο επόμενο κεφάλαιο εξηγείται η αριθμητική επίλυση του προβλήματος με την μέθοδο Newton –Raphson. Στα κεφάλαια 4 και 5 γίνεται εκτενής αναφορά και στην στρατηγική του προγραμματισμού μιας και η εφαρμογή της μεθόδου των πεπερασμένων στοιχείων είναι αλληλένδετη με το πρόγραμμα. Στο κεφάλαιο 6 παρουσιάζονται τα αποτελέσματα της εργασίας υπό μορφή γραμμών ροής, κατανομής διατμητικών τάσεων, ισοβαρών και ισοστροβιλωδών καμπυλών και συγκρίνονται με δεδομένα από διαθέσιμα εργαστηριακά πειράματα και υπολογιστικά αποτελέσματα άλλων ερευνητών. Στο κεφάλαιο 7 παρουσιάζεται το πρόγραμμα αυτής της εργασίας με κάθε δυνατή λεπτομέρεια. Έγινε προσπάθεια να εξηγηθεί το πρόγραμμα με συνεχείς αναφορές στα κεφάλαια 4 και 5, ώστε να μπορεί ο αναγνώστης της εργασίας να αντιληφθεί τους γενικούς κανόνες προγραμματισμού και να τους χρησιμοποιήσει σε άλλες εφαρμογές. Στο κεφάλαιο 8 διατυπώνονται τα συμπεράσματα και γίνεται αναφορά σε μελλοντική εργασία. Η βιβλιογραφία ακολουθεί στο κεφάλαιο 9 και στο παράρτημα δίδεται η λίστα του προγράμματος, το αρχείο δεδομένων, το υπολογιστικό πλέγμα και υπάρχει **δισκέτα** με διαθέσιμο το πρόγραμμα για χρήση κάθε ενδιαφερομένου.

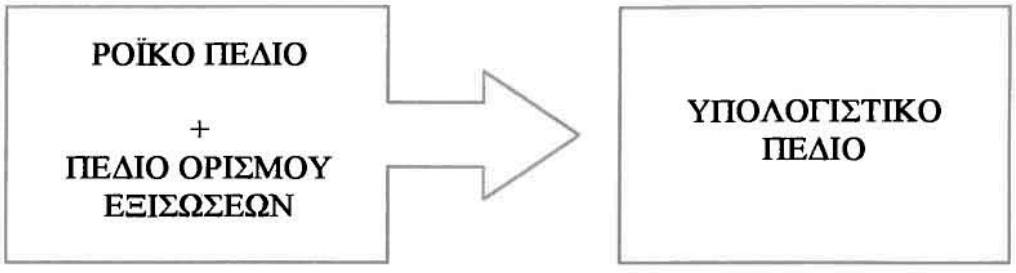
Για τυχόν περαιτέρω ερωτήσεις όσον αφορά τον προγραμματισμό και το «τρέξιμο» του προγράμματος, ο ενδιαφερόμενος αναγνώστης μπορεί να απευθυνθεί στις παρακάτω διευθύνσεις.

Δ. ΠΑΖΗΣ
ΙΠΠΟΚΡΑΤΟΥΣ 7
62122, ΣΕΡΡΕΣ
ΤΗΛ.0321-27307
e-mail: di_pa@hotmail.com

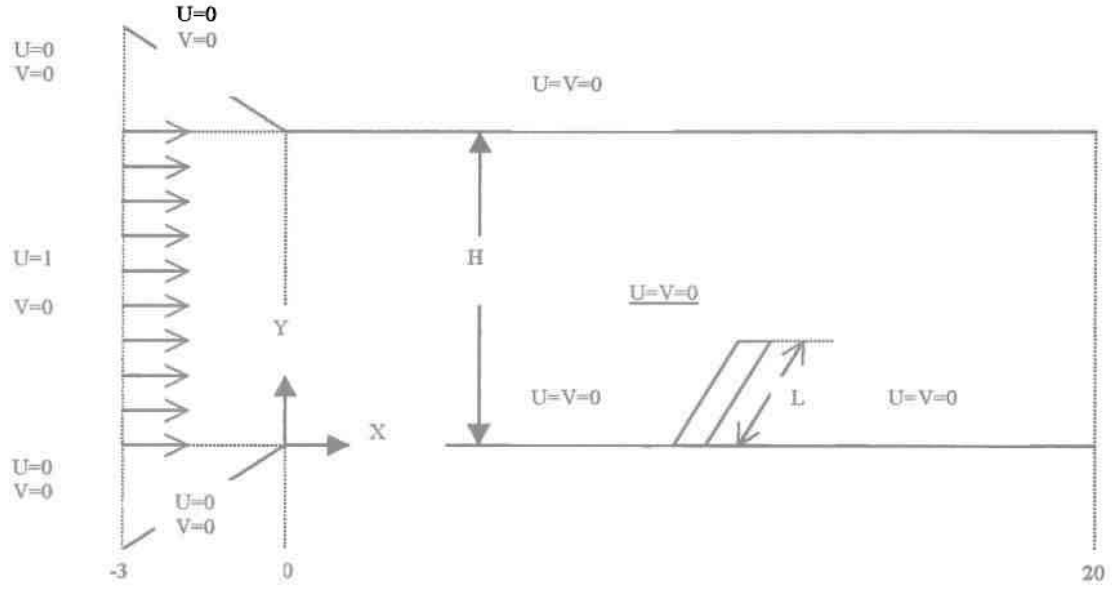
Ν. ΜΑΛΑΜΑΤΑΡΗΣ
ΒΕΡΟΙΑΣ 96
57008 ΙΩΝΙΑ, ΘΕΣ/ΝΙΚΗ
ΤΗΛ.031-781463
Fax:031-782670
e-mail: nikolaos@eng.auth.gr

3. ΔΙΕΠΟΥΣΕΣ ΕΞΙΣΩΣΕΙΣ & ΟΡΙΑΚΕΣ ΣΥΝΘΗΚΕΣ

Προτού προχωρήσουμε στην περαιτέρω παρουσίαση της εργασίας, θα πρέπει να απαντήσουμε στο ερώτημα για το πώς σχεδιάζεται ένα αριθμητικό πείραμα. Αρχικά θα πρέπει να ορίσουμε το υπολογιστικό πεδίο. ***Υπολογιστικό πεδίο*** είναι ο γεωμετρικός χώρος που λαμβάνει χώρα η ροή. Για να γίνει κατανοητός ο ορισμός ας δούμε το παρακάτω σχήμα:



Η έννοια του υπολογιστικού πεδίου έχει σχέση με τον σχεδιασμό του προβλήματος, ο οποίος θα καθορίσει και τις οριακές συνθήκες που διέπουν το πρόβλημα. Ο σχεδιασμός του υπολογιστικού πεδίου έγινε με σκοπό να προσομοιωθούν διαθέσιμα εργαστηριακά πειράματα που έλαβαν χώρα σε αεροσήραγγα. Γι' αυτόν τον λόγο επελέγη το υπολογιστικό πεδίο του σχήματος 3.1.



Σχήμα 3.1 Ροϊκό πεδίο και οριακές συνθήκες

Το νευτωνικό ρευστό ρέει αδιατάρακτα προς την είσοδο της αεροσήραγγας. Η είσοδος είναι κατάλληλα διαμορφωμένη (συγκλίνουσα), ώστε να μην διαταράσσεται η ροή κατά την πρόσκρουση στα τοιχώματα της αεροσήραγγας. Το δε ύψος αυτής, Η, είναι δέκα φορές μεγαλύτερο από το ύψος του εμποδίου, L, ώστε η ροή γύρω από τον κεκλιμένο φράκτη να είναι ανεπηρέαστη από την οροφή.

Οι εξισώσεις Navier-Stokes σε μόνιμη κατάσταση σ' αυτό το πρόβλημα είναι:

$$\rho(\mathbf{u}\cdot\nabla\mathbf{u}) = \nabla\{-\mathbf{pI} + \mu[\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}]\} \quad (3\text{-}1)$$

Για να μελετηθεί το πρόβλημα στη γενικότητα του επιλέγουμε την αδιαστατοποίηση των εξισώσεων αυτών η οποία γίνεται ως εξής:

- Εισάγουμε μια χαρακτηριστική ταχύτητα $u = Uu^*$, μια χαρακτηριστική πίεση $p = p_{ch}p^*$ και μια χαρακτηριστική μεταβλητή για το μήκος $\nabla = \frac{1}{L}\nabla^*$ δηλαδή $x = Lx^*, y = Ly^*, z = Lz^*$. Με την αντικατάσταση των αδιάστατων αυτών μεγεθών στην εξίσωση Navier-Stokes έχουμε:

$$\rho(\frac{U^2}{L}u^* \cdot \nabla^* u^*) = \frac{1}{L}\nabla^*\{-p_{ch}p^* I + \frac{\mu U}{L}[\nabla^* u^* + (\nabla^* u^*)^T]\} \quad (3\text{-}2)$$

κάνοντας απαλοιφή του L και του αστερίσκου (*) και με διαίρεση με p_{ch} προκύπτει:

$$\frac{\rho U^2}{p_{ch}}(u \cdot \nabla u) = \nabla\{-pI + \frac{\mu U}{L p_{ch}}[\nabla u + (\nabla u)^T]\}$$

- Θέτοντας $\mathbf{p_{ch} = \rho U^2 \rightarrow \frac{\rho U^2}{\frac{\mu U}{L}} = \frac{\rho UL}{\mu} = Re}$ και με αντικατάσταση στην εξίσωση (3-2) έχουμε την τελική αδιάστατη εξίσωση Navier- Stokes για έρπουσα ροή:

$$\mathbf{(u \cdot \nabla u) = \nabla\{-pI + \frac{1}{Re}[\nabla u + (\nabla u)^T]\}} \quad (3\text{-}3)$$

Ανακεφαλαιώνοντας οι αδιάστατες εξισώσεις του προβλήματός μας είναι οι ακόλουθες.

$$\mathbf{\nabla \bullet u = 0} \qquad \textbf{εξίσωση της συνέχειας} \qquad \mathbf{(3\text{-}4)}$$

$$\mathbf{u \bullet \nabla u = -\nabla p + \frac{1}{Re}\nabla^2 u} \qquad \textbf{εξίσωση της ορμής} \qquad \mathbf{(3\text{-}5)}$$

όπου: $\mathbf{Re = \frac{\rho u L}{\mu}}$ με L το μήκος του εμποδίου, μ ιξώδες του ρευστού, ρ πυκνότητα του ρευστού και u μέση ταχύτητα ($u = \frac{Q}{L}, Q : παροχη$).

Οι οριακές συνθήκες του προβλήματος, οι οποίες παρουσιάζονται και στο σχήμα 3.1:

στην είσοδο: x=-3,	0<y≤10	u=1, v=0	(3-6)
	10<y≤13	u=v=0	(3-7)
	-3<y<0	u=v=0	(3-8)
στην οροφή: -3≤x≤20,	y=10	u=v=0	(3-9)
στο δάπεδο: -3<x<20,	y=0	u=v=0	(3-10)
γύρω από το εμπόδιο:		u=v=0	(3-11)

στην έξοδο: ελεύθερη οριακή συνθήκη.

Σύμφωνα με τις εξισώσεις (3-7)-(3-11) ισχύει η οριακή συνθήκη μη ολισθήσεως στα τοιχώματα του υπολογιστικού πεδίου. Στην είσοδο, με την εξίσωση (3-6), επιβάλλεται η παροχή της μάζας που εισέρχεται στο υπολογιστικό πεδίο και στην έξοδο εφαρμόζεται η

ελεύθερη οριακή συνθήκη, η οποία επιτρέπει την εξέλιξη των φαινομένων στην έξοδο χωρίς διαταραχές (4).

Η παραδοχή της μονοδιάστατης αδιατάρακτης ροής στην είσοδο δεν έχει συνέπειες στην εξέλιξη του φαινομένου στο υπολογιστικό πεδίο, σε αντίθεση με τις οριακές συνθήκες εξόδου η επιλογή των οποίων έχει θεμελιώδη σημασία και καθοριστική επιρροή στην εξέλιξη των ροϊκών φαινομένων. Επελέγη αυτή η κατανομή της ταχύτητας στην είσοδο, ώστε να υπολογισθεί επακριβώς η μάζα που εισέρχεται στο υπολογιστικό πεδίο.

Μετά τον ορισμό του υπολογιστικού πεδίου και την καταγραφή των εξισώσεων που διέπουν το πρόβλημα, καθώς επίσης και τον καθορισμό των οριακών συνθηκών, προβαίνουμε στην παρουσίαση της μεθόδου των Πεπερασμένων Στοιχείων η οποία καθιστά εφικτή την επίλυση του μη-γραμμικού συστήματος των διεπουσών εξισώσεων.

4. ΔΙΑΤΥΠΩΣΗ ΤΩΝ ΕΞΙΣΩΣΕΩΝ ΜΕ ΤΗΝ ΜΕΘΟΔΟ ΤΩΝ ΠΕΠΕΡΑΣΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ

Για να διατυπώσουμε τις διέπουσες εξισώσεις με την μέθοδο των πεπερασμένων στοιχείων εργαζόμαστε ως εξής:

Οι άγνωστοι u, v, p προσεγγίζονται από τους κομβικούς αγνώστους u^k, v^k, p^k και στα ενδιάμεσα σημεία του υπολογιστικού πεδίου υπολογίζονται από την γραμμική παρεμβολή των κομβικών αγνώστων με τις συναρτήσεις βάσεως:

$$\mathbf{u} = \sum \mathbf{u^k \varphi^k}\ ,\quad \mathbf{v} = \sum \mathbf{v^k \varphi^k}\ ,\ \mathbf{p} = \sum \mathbf{p^k \psi^k} \qquad \textbf{(4-1)}$$

Για τις ***ταχύτητες u, v*** επιλέγονται ***τετραγωνικές*** συναρτήσεις βάσεως $\boldsymbol{\varphi^k}$, διότι η ταχύτητα παραγωγίζεται στις εξισώσεις Navier-Stokes δύο φορές, ενώ για την ***πίεση, γραμμικές*** συναρτήσεις βάσεως $\boldsymbol{\psi^k}$ διότι η πίεση παραγωγίζεται μία φορά. Αυτές οι συναρτήσεις βάσεως έχουν την ακόλουθη μορφή:

$$\phi^1 = \frac{1}{4}\xi(\xi-1)\eta(\eta-1) \qquad \textbf{(4-2)}$$

$$\phi^2 = \frac{1}{2}\xi(\xi-1)(1-\eta*\eta) \qquad \textbf{(4-3)}$$

$$\phi^3 = \frac{1}{4}\xi(\xi-1)\eta(\eta+1) \qquad \textbf{(4-4)}$$

$$\phi^4 = \frac{1}{2}(1-\xi*\xi)\eta(\eta-1) \qquad \textbf{(4-5)}$$

$$\phi^5 = (1-\xi*\xi)(1-\eta*\eta) \qquad \textbf{(4-6)}$$

$$\phi^6 = \frac{1}{2}(1-\xi*\xi)\eta(\eta+1) \qquad \textbf{(4-7)}$$

$$\phi^7 = \frac{1}{4}\xi(\xi+1)\eta(\eta-1) \qquad \textbf{(4-8)}$$

$$\phi^8 = \frac{1}{2}\xi(\xi+1)(1-\eta*\eta) \qquad \textbf{(4-9)}$$

$$\phi^9 = \frac{1}{4}\xi(\xi+1)\eta(\eta+1) \qquad \textbf{(4-10)}$$

και

$$\psi^1 = \frac{1}{4}(1-\xi)(1-\eta) \qquad \textbf{(4-11)}$$

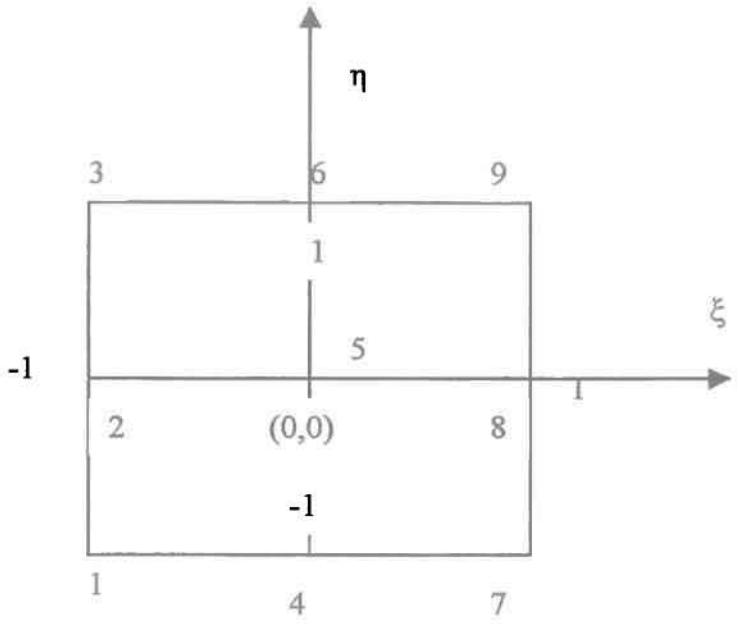
$$\psi^2 = \frac{1}{4}(1-\xi)(1+\eta) \qquad \textbf{(4-12)}$$

$$\psi^3 = \frac{1}{4}(1+\xi)(1-\eta) \qquad \textbf{(4-13)}$$

$$\psi^4 = \frac{1}{4}(1+\xi)(1+\eta) \qquad \textbf{(4-14)}$$

Στο πρόγραμμα που βρίσκεται στο παράρτημα οι συναρτήσεις βάσεως $\boldsymbol{\varphi^i}$ ορίζονται στην SUBROUTINE TFUNC(C,E) και συμβολίζονται με **PHI(1)**, …., **PHI(9)**, οι δε μεταβλητές $\boldsymbol{\xi}$ και $\boldsymbol{\eta}$ συμβολίζονται με ***C, E***. Αντίστοιχα οι γραμμικές συναρτήσεις βάσεως $\boldsymbol{\psi^i}$ ορίζονται στην SUBROUTINE TPRES(C,E) και συμβολίζονται με **PHI(1)**, …., **PHI(4).** Οι TFUNC(C,E) και TPRES(C,E) καλούνται στην SUBROUTINE ABFIND (σελ.55) για τον υπολογισμό των αθροισμάτων (4-1).

Τα αθροίσματα (4-1) υπολογίζονται για κάθε στοιχείο του υπολογιστικού πεδίου ξεχωριστά διότι οι συναρτήσεις βάσεως ορίζονται μόνο σε όμορα πεπερασμένα στοιχεία ενώ στο υπόλοιπο πεδίο είναι μηδέν. Συνεπώς οι κόμβοι **k** αριθμούνται ολικά στο πεδίο και μάλιστα κάθε κόμβος έχει έναν και μόνο έναν αριθμό, η δε αρίθμηση αυτή επιτυγχάνεται στην FUNCTION NODNOR (σελ.37). Όμως οι συναρτήσεις βάσεως μέσα στο πρόγραμμα έχουν τοπική αρίθμηση 1,..., 9 ή 1,..., 4. Αυτό γίνεται διότι η πράξη μας οδήγησε στην ισοπαραμετρική απεικόνιση και –όπως θα δούμε παρακάτω- ορίζεται ένα πρότυπο στοιχείο πάνω στο οποίο απεικονίζονται τα τυχαία στοιχεία του υπολογιστικού πεδίου. Για τις τετραγωνικές συναρτήσεις βάσεως το πρότυπο στοιχείο έχει ως εξής:



Σχήμα: 4-1.Τετραγωνικές συναρτήσεις βάσεως στο πρότυπο στοιχείο.

Στο σχήμα 4–1, είναι εμφανής η ανάγκη εισαγωγής των τοπικών συντεταγμένων ξ και η. Η μεταβολή από τις συντεταγμένες X, Y του υπολογιστικού πεδίου στις ξ, η του πρότυπου στοιχείου γίνεται με τον ίδιο τρόπο που προσεγγίζονται οι μεταβλητές u, v, ρ με τους κομβικούς αγνώστους. Δηλαδή:

$$\mathbf{x} = \sum \mathbf{x^k} \varphi^{\mathbf{k}} \quad , \mathbf{y} = \sum \mathbf{y^k} \varphi^{\mathbf{k}} \qquad \textbf{(4-15)}$$

όπου x^k και y^k οι συντεταγμένες των κόμβων του υπολογιστικού πεδίου. Οι φ^i παίρνουν τον αριθμό 1 ως 9 ανάλογα σε ποιον κόμβο έχουν την τιμή 1. Για παράδειγμα η φ^5ισούται με την μονάδα για ξ=η=0. Σε όλους τους άλλους κόμβους η φ^5 παίρνει την τιμή μηδέν. Το ίδιο ισχύει για όλες τις τετραγωνικές συναρτήσεις βάσεως. Ανάλογα, για τις γραμμικές συναρτήσεις βάσεως, η αρίθμηση 1, 2, 3, 4 αντιστοιχεί στους κόμβους 1, 3, 7, 9 του προτύπου στοιχείου. Γι' αυτό στην SUBROUTINE ABFIND ορίζεται ο πίνακας NNPRES(I) που επιτυγχάνει την παραπάνω αντιστοιχία. Η μετατροπή της ολικής αριθμήσεως των κόμβων k στους τοπικούς κόμβους του προτύπου στοιχείου επιτυγχάνεται με τον πίνακα NOP(NK,KK) όπου NK είναι ο αριθμός του στοιχείου που σαρώνεται, KK είναι ο τοπικός αριθμός του κόμβου του πρότυπου στοιχείου, και η τιμή του πίνακα NOP είναι ο ολικός αριθμός του κόμβου k. Ο πίνακας NOP(NK,KK) υπολογίζεται στην SUBROUTINE CALNOP (σελ.41).

Οι προσεγγίσεις (4-1) εισάγονται στις διέπουσες εξισώσεις (3-4 και 3-5) και ζυγίζονται ολοκληρωτικά με τις συναρτήσεις βάσεως και έτσι προκύπτουν τα ζυγισμένα υπόλοιπα:

$$\mathbf{R}_{\mathbf{c}}^{\mathbf{j}} = \int_{\mathbf{V}} \nabla \cdot \mathbf{u}\psi^{\mathbf{j}} d\mathbf{V} \qquad \textbf{(4-16)}$$

$$\mathbf{R}_{\mathbf{M}}^{\mathbf{i}} = \int_{\mathbf{V}} [\mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \{-\mathbf{pI} + \frac{1}{\mathbf{Re}}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}]\}]\varphi^{\mathbf{i}} d\mathbf{V} \qquad \textbf{(4-17)}$$

όπου $\mathbf{R}_{\mathbf{c}}^{\mathbf{j}}$ και $\mathbf{R}_{\mathbf{M}}^{\mathbf{i}}$ τα ζυγισμένα υπόλοιπα της εξισώσεως της συνέχειας και της ορμής αντιστοίχως. Ο δείκτης ***j*** αντιστοιχεί στους ακραίους κόμβους κάθε πεπερασμένου στοιχείου του υπολογιστικού πεδίου (κόμβοι 1, 3, 7, 9 στο πρότυπο στοιχείο), ενώ ο δείκτης ***i*** αντιστοιχεί σε όλους τους κόμβους του υπολογιστικού πεδίου. Στην εξίσωση (4-17) εφαρμόζεται το θεώρημα της αποκλίσεως (θεώρημα Gauss-Ostragardski) για να ελαττωθεί η τάξη της παραγώγου. Σύμφωνα με τον κανόνα παραγωγίσεως του γινομένου ισχύει:

$$\nabla \cdot \{(-\mathbf{pI} + \frac{1}{\mathbf{Re}}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}])\varphi^{\mathbf{i}}\} = \nabla \cdot \{-\mathbf{pI} + \frac{1}{\mathbf{Re}}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}]\}\varphi^{\mathbf{i}} +$$
$$+ \{-\mathbf{pI} + \frac{1}{\mathbf{Re}}[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}]\} \cdot \nabla \varphi^{\mathbf{i}}\} \qquad \textbf{(4-18)}$$

κατά συνέπεια η εξίσωση της ορμής παίρνει τη μορφή:

$$\mathbf{R}_{\mathbf{M}}^{\mathbf{i}} = \int_{\mathbf{V}} \left[\mathbf{u} \cdot \nabla \mathbf{u} \varphi^{\mathbf{i}} + \left\{ -\mathbf{pI} + \frac{1}{\mathbf{Re}} \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}} \right] \right\} \cdot \nabla \varphi^{\mathbf{i}} \right] d\mathbf{V} -$$
$$- \int_{\mathbf{S}} \mathbf{n} \left\{ -\mathbf{pI} + \frac{1}{\mathbf{Re}} \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}} \right] \varphi^{\mathbf{i}} \right\} d\mathbf{S} \qquad \textbf{(4-19)}$$

Έχοντας υπ' όψιν ότι:

$$\nabla = \frac{\partial}{\partial \mathbf{x}}\mathbf{i} + \frac{\partial}{\partial \mathbf{y}}\mathbf{j} \qquad \textbf{(4-20)}$$

$$\mathbf{u} = \mathbf{ui} + \mathbf{vj} \qquad \textbf{(4-21)}$$

όπου ***i*** και ***j*** τα μοναδιαία διανύσματα στους άξονες X, Y του υπολογιστικού πεδίου. Προκύπτουν οι εξής όροι:

$$\nabla \cdot \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{y}} \qquad \textbf{(4-22)}$$

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} & \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \\ \frac{\partial \mathbf{u}}{\partial \mathbf{y}} & \frac{\partial \mathbf{v}}{\partial \mathbf{y}} \end{bmatrix} \textbf{(4-23)}, \quad (\nabla \mathbf{u})^{\mathbf{T}} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} & \frac{\partial \mathbf{u}}{\partial \mathbf{y}} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{x}} & \frac{\partial \mathbf{u}}{\partial \mathbf{y}} \end{bmatrix} \textbf{(4-24)}, \quad \mathbf{u} \cdot \nabla \mathbf{u} = \begin{bmatrix} \mathbf{u}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{v}\frac{\partial \mathbf{u}}{\partial \mathbf{y}} \\ \mathbf{u}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \end{bmatrix} \textbf{(4-25)}$$

$$\nabla \varphi^{\mathbf{i}} = \frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{x}}\mathbf{i} + \frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{y}}\mathbf{j} \textbf{(4-26)}, \quad \mathbf{I} \cdot \nabla \varphi^{\mathbf{i}} = \begin{bmatrix} \frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{x}} \\ \frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{y}} \end{bmatrix} \textbf{(4-27)},$$

$$[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathbf{T}}] \cdot \nabla \varphi^{\mathbf{i}} = \begin{bmatrix} 2\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{x}} + (\frac{\partial \mathbf{u}}{\partial \mathbf{y}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}})\frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{y}} \\ (\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}}{\partial \mathbf{y}})\frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{x}} + 2\frac{\partial \mathbf{v}}{\partial \mathbf{y}}\frac{\partial \varphi^{\mathbf{i}}}{\partial \mathbf{y}} \end{bmatrix} \textbf{(4-28).}$$

Παρατηρούμε ότι το ζυγισμένο υπόλοιπο της συνέχειας $\mathbf{R}_{\mathbf{c}}^{\mathbf{j}}$ είναι βαθμωτό και έχει την τελική μορφή:

$$R_c^j = \int_V (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})\psi^j dV \qquad \textbf{(4-29)}$$

$$\text{όπου } \frac{\partial u}{\partial x} = \sum u^k \frac{\partial \varphi^k}{\partial x}, \qquad \frac{\partial v}{\partial y} = \sum u^k \frac{\partial \varphi^k}{\partial y} \qquad \textbf{(4-30)}$$

ενώ το ζυγισμένο υπόλοιπο της ορμής $\mathbf{R'_M}$ είναι διανυσματικό. Βαθμωτές εξισώσεις για το $\mathbf{R'_M}$ προκύπτουν από το εσωτερικό γινόμενο του $\mathbf{R'_M}$ με τα διανύσματα ***i, j***. Έτσι προκύπτουν:

$$\mathbf{R'_{M,u} = i \cdot R'_M} = \int_V \{(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y})\varphi^i + [-p\frac{\partial \varphi^i}{\partial x} + \frac{1}{Re}(2\frac{\partial u}{\partial x}\frac{\partial \varphi^i}{\partial x} + (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\frac{\partial \varphi^i}{\partial y})]\}dV - \qquad \textbf{(4-31)}$$
$$- \int_S \{-p + \frac{1}{Re}(2\frac{\partial u}{\partial x})\}\varphi^i dS$$

$$R^i_{M,v} = j \cdot R^i_M = \int_V \{(u\frac{\partial v}{\partial x} + v\frac{\partial u}{\partial y})\varphi^i + [-p\frac{\partial \varphi^i}{\partial y} + \frac{1}{Re}(2\frac{\partial v}{\partial y}\frac{\partial \varphi^i}{\partial y} + (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\frac{\partial \varphi^i}{\partial y})]\}dV - \qquad \textbf{4-32)}$$
$$- \int_S \{-p + \frac{1}{Re}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})\}\varphi^i dS$$

Έχοντας υπ' όψιν τις ιδιαιτερότητες των κόμβων, δηλαδή ότι στους ακραίους κόμβους του κάθε στοιχείου, οι άγνωστοι είναι μόνον οι u^k, v^k, p^k, ενώ στους υπολοίπους οι άγνωστοι είναι μόνον οι u^k, v^k, η κατασκευή του ολικού πίνακα R' με τα ζυγισμένα υπόλοιπα έχει ως εξής:

$$\mathbf{R^i} = \begin{bmatrix} \mathbf{R^1_{M,u}} \\ \mathbf{R^1_{M,v}} \\ \mathbf{R^1_{M,p}} \\ \mathbf{R^2_{M,u}} \\ \mathbf{R^2_{M,v}} \\ \mathbf{R^3_{M,u}} \\ \mathbf{R^3_{M,v}} \\ \mathbf{R^3_{M,p}} \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \mathbf{R^{NH}_{M,u}} \\ \mathbf{R^{NH}_{M,v}} \\ \mathbf{R^{NH}_{M,p}} \end{bmatrix}$$

όπου $\mathbf{R^1_{M.u}}$ είναι η εξίσωση που υπολογίζει την ταχύτητα u στον πρώτο κόμβο του υπολογιστικού πεδίου κ.ο.κ. Με NH ορίζεται ο τελευταίος κόμβος του υπολογιστικού πεδίου. Αυτή η κατασκευή του πίνακα με τα ζυγισμένα υπόλοιπα δημιουργεί την ανάγκη ορισμού των πινάκων NOPP(I) και MDF(I) στην SUBROUTINE MSHSTR (σελ.38). Και οι δύο πίνακες έχουν διαστάσεις I=1,…,NH. Ο πίνακας NOPP(I) υπολογίζει την θέση της πρώτης εξισώσεως σε κάθε κόμβο του υπολογιστικού πεδίου. Δηλαδή: NOPP(1)=1, NOPP(2)=4,

NOPP(3)=6, ..., NOPP(NH)= NP-2 όπου NP ο συνολικός αριθμός των εξισώσεων. Ο πίνακας MDF(I) υπολογίζει τον αριθμό των εξισώσεων ανά κόμβο. Δηλαδή: MDF(1)=3,MDF(2)=2, MDF(3)=3 ,..., MDF(NH)=3. Ο πίνακας των ζυγισμένων υπολοίπων συμβολίζεται στην SUBROUTINE ABFIND με τον R1(I) με διαστάσεις NP.

Εν συνεχεία παρουσιάζεται η αριθμητική επίλυση των εξισώσεων του προβλήματος, περιλαμβάνοντας τις εξισώσεις της Ιακωβιανής και εν μέρει την δομή της και φυσικά τον τρόπο με τον οποίο συγκλίνει ο αλγόριθμος μας με την μέθοδο Newton – Raphson.

5.ΑΡΙΘΜΗΤΙΚΗ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Για να επιλυθεί το σύστημα εξισώσεων που προκύπτουν από τα ζυγισμένα υπόλοιπα $\mathbf{R}_c^J$, $R^i_{M,u}, \bar{\kappa}^i_{M,v}$ του προηγουμένου κεφαλαίου, πρέπει να επιλεγεί μία κατάλληλη επαναληπτική μέθοδος, διότι οι εξισώσεις (4-29 ως 4-32) είναι μη- γραμμικές λόγω της εμφανίσεως όρων όπως $\mathbf{u}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ κ.λ.π. Επιλέγουμε την μέθοδο Newton – Raphson λόγω της τετραγωνικής συγκλίσεως του αλγορίθμου και των πληροφοριών περί ευστάθειας της λύσεως από τις πληροφορίες που παρέχει η Ιακωβιανή. Το αλγεβρικό πρόβλημα που προκύπτει από την ακολουθεί μορφή:

$$\Delta \mathbf{u}^i = -(\mathbf{J}^{ik})^{-1}\mathbf{R}^i \qquad \textbf{(5-1)}$$

όπου Δu^I ο συνολικός πίνακας της τιμής της διαφοράς μεταξύ της τρέχουσας (ι+1) και της προηγούμενης (ι) επαναλήψεως για κάθε κομβικό άγνωστο. Δηλαδή:

$$\Delta \mathbf{u} = \begin{bmatrix} \mathbf{u}^1_{i+1} - \mathbf{u}^1_i \\ \mathbf{v}^1_{i+1} - \mathbf{v}^1_i \\ \mathbf{p}^1_{i+1} - \mathbf{p}^1_i \\ \mathbf{u}^2_{i+1} - \mathbf{u}^2_i \\ \mathbf{v}^2_{i+1} - \mathbf{v}^2_i \\ \mathbf{u}^3_{i+1} - \mathbf{u}^3_i \\ \mathbf{v}^3_{i+1} - \mathbf{v}^3_i \\ \mathbf{p}^3_{i+1} - \mathbf{p}^3_i \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \\ \cdots\cdots\cdots \end{bmatrix}$$

Οι δείκτες 1, 2, 3, …., NH δηλώνουν τον ολικό αριθμό του κόμβου του υπολογιστικού πεδίου Ο πίνακας Δu συμβολίζεται με SK(I) στην SUBROUTINE UPDATE όπου λαμβάνει χώρα ο υπολογισμός της τρέχουσας τιμής για κάθε κομβικό άγνωστο σύμφωνα με την σχέση (5-1). Η Ιακωβιανή J^{ik} από την ίδια σχέση θα πρέπει να έχει διαστάσεις NP*NP. Κάτι τέτοιο όμως θα δημιουργούσε τεράστιο πρόβλημα μνήμης για την επίλυση του προβλήματος, έχοντας υπ' όψιν ότι ο αριθμός NP σ' αυτήν την εργασία είναι της τάξεως του 10^5 όπως θα δούμε στο επόμενο κεφάλαιο. Κατά συνέπεια η μνήμη που χρειάζεται ο υπολογιστής θα ήταν: $8*10^5*10^5$ RAM=800Gbyte RAM!!!

Αυτά τα προβλήματα εντοπίσθηκαν εγκαίρως και στα μέσα της δεκαετίας του 1970 επινοήθηκαν τεχνάσματα, ώστε να αποφεύγεται ο ορισμός μιας τέτοιας μεγάλης Ιακωβιανής. Επειδή η αντιστροφή της Ιακωβιανής- που είναι και η πιο χρονοβόρα διαδικασία στην εκτέλεση του προγράμματος – γίνεται με την μέθοδο απαλοιφής Gauss και επειδή κατά την σάρωση των εξισώσεων των ζυγισμένων υπολοίπων προκύπτον μη-μηδενικές ζώνες στην Ιακωβιανή (bounded matrix), αποφασίστηκε οι διαστάσεις της να είναι NCN(I)*NCN(I) (βλέπε SUBROUTINE MSHSTR), όπου NCN(I) ο αριθμός των αγνώστων σε κάθε στοιχείο I του υπολογιστικού πεδίου. Στο πρόβλημα που μελετούμε σ' αυτήν την εργασία ισχύει NCN(I)=22 και έτσι αντιλαμβανόμαστε την τεράστια εξοικονόμηση μνήμης. Αυτό το τέχνασμα καθώς και ο τρόπος που γίνεται η αντιστροφή της Ιακωβιανής (SUBROUTINE FRONT) έχει συζητηθεί εκτενώς στην εργασία του Hood (12) χρησιμοποιείται επιτυχώς την

τελευταία 25ετία απ' όλους τους ερευνητές μηχανικούς που επιλύουν προβλήματα πεπερασμένων στοιχείων με την μέθοδο Newton- Raphson και δεν πρόκειται να επεκταθούμε περαιτέρω. Απλά συμφωνούμε σ' αυτό το σημείο, ότι η SUBROUTINE FRONT είναι η μόνη υπορουτίνα του προγράμματος που δεν γράφτηκε από τον συγγραφέα της παρούσης εργασίας. Οι επιμέρους όροι της Ιακωβιανής J^{ik} έχουν ως εξής:

$$\frac{\partial \mathbf{R}_c^j}{\partial \mathbf{u}^k} = \int_V \frac{\partial \varphi^k}{\partial \mathbf{x}} \cdot \psi^j \mathbf{dV} \qquad \textbf{(5-2)}$$

$$\frac{\partial \mathbf{R}_c^j}{\partial \mathbf{v}^k} = \int_V \frac{\partial \varphi^k}{\partial \mathbf{y}} \cdot \psi^j \mathbf{dV} \qquad \textbf{(5-3)}$$

$$\frac{\partial \mathbf{R}_{M\,u}^i}{\partial \mathbf{u}^k} = \int_V \{(\varphi^k \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{v}\frac{\partial \varphi^k}{\partial \mathbf{y}} + \mathbf{u}\frac{\partial \varphi^k}{\partial \mathbf{x}}) \cdot \varphi^i + \frac{1}{\mathrm{Re}}[2\frac{\partial \varphi^k}{\partial \mathbf{x}} \cdot \frac{\partial \varphi^i}{\partial \mathbf{x}} + \frac{\partial \varphi^k}{\partial \mathbf{y}} \cdot \frac{\partial \varphi^i}{\partial \mathbf{y}}]\}\mathbf{dV} -$$

$$- \int_S \frac{1}{\mathrm{Re}} 2\frac{\partial \varphi^k}{\partial \mathbf{x}} \varphi^i \mathbf{dS} \qquad \textbf{(5-4)}$$

$$\frac{\partial \mathbf{R}_{M,u}^i}{\partial \mathbf{u}^k} = \int_V \{\varphi^k \frac{\partial \mathbf{u}}{\partial \mathbf{y}} \varphi^i + \frac{1}{\mathrm{Re}} \frac{\partial \varphi^k}{\partial \mathbf{x}} \cdot \frac{\partial \varphi^i}{\partial \mathbf{y}}\}\mathbf{dV} \qquad \textbf{(5-5)}$$

$$\frac{\partial \mathbf{R}_{M,u}^i}{\partial \mathbf{p}^k} = \int_V (-\psi^k \frac{\partial \varphi^i}{\partial \mathbf{x}})\mathbf{dV} - \int_S (-\psi^k \varphi^i)\mathbf{dS} \qquad \textbf{(5-6)}$$

$$\frac{\partial \mathbf{R}_{M,v}^i}{\partial \mathbf{u}^k} = \int_V \{\varphi^k \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \varphi^i + \frac{1}{\mathrm{Re}} \frac{\partial \varphi^k}{\partial \mathbf{y}} \cdot \frac{\partial \varphi^i}{\partial \mathbf{x}}\}\mathbf{dV} - \int_S \frac{1}{\mathrm{Re}} \frac{\partial \varphi^k}{\partial \mathbf{y}} \varphi^i \mathbf{dS} \qquad \textbf{(5-7)}$$

$$\frac{\partial \mathbf{R}_{M\,v}^i}{\partial \mathbf{v}^k} - \int_V \{(\mathbf{v}\frac{\partial \varphi^k}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial \varphi^k}{\partial \mathbf{x}} + \varphi^k \frac{\partial \mathbf{v}}{\partial \mathbf{x}}) \cdot \varphi^i + \frac{1}{\mathrm{Re}} \frac{\partial \varphi^k}{\partial \mathbf{x}} \cdot \frac{\partial \varphi^i}{\partial \mathbf{x}}\}\mathbf{dV} - \int_S \frac{1}{\mathrm{Re}} \frac{\partial \varphi^k}{\partial \mathbf{x}} \varphi^i \mathbf{dS} \qquad \textbf{(5-8)}$$

$$\frac{\partial \mathbf{R}_{M,v}^i}{\partial \mathbf{p}^k} = \int_V (-\psi^k \frac{\partial \varphi^i}{\partial \mathbf{y}})\mathbf{dV} - \int_S (-\psi^k \varphi^i)\mathbf{dS} \qquad \textbf{(5-9)}$$

Για τον υπολογισμό των επιμέρους όρων της Ιακωβιανής (εξισώσεις 5-2→5-9) τέθηκαν υπ' όψιν οι εξής σχέσεις:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{u}^k} = \frac{\partial}{\partial \mathbf{u}^k} \sum \mathbf{u}^i \varphi^i = \varphi^k \qquad \textbf{(5-10)}$$

$$\frac{\partial}{\partial \mathbf{u}^k}(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}) = \frac{\partial}{\partial \mathbf{u}^k}(\sum \mathbf{u}^i \frac{\partial \varphi^i}{\partial \mathbf{x}}) = \frac{\partial \varphi^k}{\partial \mathbf{x}} \qquad \textbf{(5-11)}$$

$$\frac{\partial}{\partial \mathbf{u}^k}(\frac{\partial \mathbf{u}}{\partial \mathbf{y}}) = \frac{\partial}{\partial \mathbf{u}^k}(\sum \mathbf{u}^i \frac{\partial \varphi^i}{\partial \mathbf{y}}) = \frac{\partial \varphi^k}{\partial \mathbf{y}} \qquad \textbf{(5-12)}$$

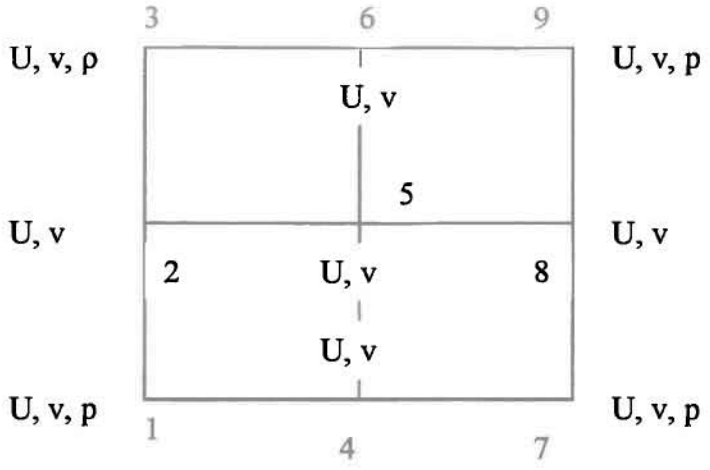
$$\frac{\partial \mathbf{v}}{\partial \mathbf{v}^k} = \frac{\partial}{\partial \mathbf{v}^k}(\sum \mathbf{v}^i \varphi^i) = \varphi^k \qquad \textbf{(5-13)}$$

$$\frac{\partial}{\partial \mathbf{v}^k}(\frac{\partial \mathbf{v}}{\partial \mathbf{x}}) = \frac{\partial}{\partial \mathbf{v}^k}(\sum \mathbf{v}^i \frac{\partial \varphi^i}{\partial \mathbf{x}}) = \frac{\partial \varphi^k}{\partial \mathbf{x}} \qquad \textbf{(5-14)}$$

$$\frac{\partial}{\partial \mathbf{v}^k}(\frac{\partial \mathbf{u}}{\partial \mathbf{y}}) = \frac{\partial}{\partial \mathbf{v}^k}(\sum \mathbf{v}^i \frac{\partial \varphi^i}{\partial \mathbf{y}}) = \frac{\partial \varphi^k}{\partial \mathbf{y}} \qquad \textbf{(5-15)}$$

$$\frac{\partial \mathbf{p}}{\partial \mathbf{p}^k} = \frac{\partial}{\partial \mathbf{p}^k}(\sum \mathbf{p}^j \psi^j) = \psi^k \qquad \textbf{(5-16)}$$

Για την σωστή καταχώριση των όρων (5-2) ως (5-16) στον πίνακα της Ιακωβιανής, με τοπική αρίθμηση και διαστάσεις NCN(I)*NCN(I), λαμβάνεται υπ' όψιν το ακόλουθο σχήμα:



Σχήμα 5.1 Το πρότυπο Στοιχείο.

Γι' αυτό ορίζονται οι πίνακες IU(I), IV(I) για I=1,...., 9 και IP(I) για I=1,...., 4 στην SUBROUTINE ABFIND που αντιστοιχούν στις εξισώσεις $\mathbf{R}^{i}_{M,u}$, $\mathbf{R}^{i}_{M,v}$, $\mathbf{R}^{i}_{c}$. Σε κάθε κόμβο I δίδεται στους πίνακες IU(I), IV(I) και IP(I) η σειρά της αντίστοιχης εξίσωσης στο πρότυπο στοιχείο του σχήματος 5.1. Δηλαδή: IU(1)=1, IV(1)=2, IP(1)=3 κ.ο.κ. Αφού ορίστηκαν αυτοί οι βοηθητικοί πίνακες, τότε η καταχώρηση των όρων 5-2 ως 5-16 απλουστεύεται ως εξής: Ο αριθμητής κάθε όρου λαμβάνει τον αριθμό της γραμμής και ο παρονομαστής της στήλης της Ιακωβιανής. Δηλαδή για τον όρο: $\frac{\partial \mathbf{R}^{j}_{c}}{\partial \mathbf{u}^{k}}$ υπάρχουν 4*9 καταχωρήσεις στην Ιακωβιανή, διότι 4 είναι οι εξισώσεις της συνέχειας σε κάθε στοιχείο του υπολογιστικού πεδίου ενώ 9 είναι οι κομβικοί άγνωστοι για την ταχύτητα u. Σύμφωνα δε με το σχήμα 5-1 οι τέσσερις γραμμές στις οποίες θα καταχωρηθούν οι όροι $\frac{\partial \mathbf{R}^{j}_{c}}{\partial \mathbf{u}^{k}}$ είναι οι: 3, 8, 17, 22 και σε κάθε γραμμή οι αντίστοιχες στήλες θα είναι οι:1, 4, 6, 9, 11, 13, 15, 18, 20. Ακολουθώντας την ίδια λογική είναι φανερό ότι οι όροι $\frac{\partial \mathbf{R}^{j}_{c}}{\partial \mathbf{v}^{k}}$ θα καταχωρηθούν τις ίδιες γραμμές αλλά στις στήλες 2, 5, 7, 10, 12, 14, 16, 19, 21, οι όροι $\frac{\partial \mathbf{R}^{i}_{M,u}}{\partial \mathbf{u}^{k}}$ στις γραμμές και στις στήλες 1, 4, 6, 9, 11, 13, 15, 18, 20, οι όροι $\frac{\partial \mathbf{R}^{i}_{M,u}}{\partial \mathbf{p}^{k}}$ στις ίδιες γραμμές αλλά στις στήλες 3, 8, 17, 22 κ.ο.κ.

Οι εξισώσεις οι οποίες ορίζουν τα ζυγισμένα υπόλοιπα $\mathbf{R}^{i}$ (στο 4° κεφάλαιο) και την Ιακωβιανή $\mathbf{J}^{ik}$ (εξισώσεις 5-2 ως 5-16) είναι ολοκληρώματα τα οποία ορίζονται σε όλο το υπολογιστικό πεδίο. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, οι υπολογισμοί γίνονται σε πρότυπο στοιχείο ώστε να αυτοματοποιηθεί ο αλγόριθμος και να εκμεταλλευτούμε τα πλεονεκτήματα του προγραμματισμού. Έτσι οδηγούμαστε στην σάρωση του υπολογιστικού πεδίου στοιχείο προς στοιχείο με ταυτόχρονη καταγραφή των ζυγισμένων υπολοίπων σε κάθε κόμβο. Λεπτομερέστερα, εκμεταλλευόμενοι τον κανόνα αθροίσεως της ολοκληρώσεως,

$\int_V (..)\mathbf{dV} = \sum_{i=1}^{NE} (..)\mathbf{dV_i}$ όπου $\mathbf{V_i}$ είναι ο όγκος κάθε πεπερασμένου στοιχείου και NE το σύνολο των πεπερασμένων στοιχείων του υπολογιστικού πεδίου (SUBROUTINE MSHSTR), υπολογίζουμε τα ολοκληρώματα των ζυγισμένων υπολοίπων της Ιακωβιανής σε κάθε στοιχείο. Η αυτοματοποίηση των υπολογισμών επιτυγχάνεται περαιτέρω με την ισοπαραμετρική απεικόνιση.

Πιο συγκεκριμένα, σε τυχαίο στοιχείο NELL για την εξίσωση της συνέχειας ο αλγόριθμος αυτοματοποιήσεως της προαναφερθείσας στρατηγικής έχει ως εξής:

$$\int_{V_{NELL}} (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})\psi^j \mathbf{dV}_{NELL} = \int_{V_{NELL}} (\sum \mathbf{u}^k \frac{\partial \varphi^k}{\partial \mathbf{x}} + \sum \mathbf{v}^k \frac{\partial \varphi^k}{\partial \mathbf{y}})\psi^j \mathbf{dV}_{NELL} = \int_{-1}^{1}\int_{-1}^{1} \{(\sum \mathbf{u}^k \frac{\partial \varphi^{kk}}{\partial x}) + (\sum \mathbf{v}^k \frac{\partial \varphi^{kk}}{\partial \mathbf{y}})\}\psi^{jj}|\mathbf{J}|d\xi d\xi \quad \beta\text{-}17$$

Στην σχέση 5-17 ο όρος $|\mathbf{J}|$ είναι η ορίζουσα της ισοπαραμετρικής απεικονίσεως που μετατρέπει το τυχαίο στοιχείο σε πρότυπο με αλλαγή των συντεταγμένων (x, y) σε (ξ, η). Οι σχέσεις για την ορίζουσα $|\mathbf{J}|$ καθώς και για τον τρόπο υπολογισμού των παραγώγων των συναρτήσεων βάσεως ως προς τις συντεταγμένες x και y δίδονται στις σημειώσεις του καθ.Α. Μπουντουβή (13). Στο πρόγραμμα αυτής της εργασίας η ισοπαραμετρική απεικόνιση λαμβάνει χώρα στην SUBROUTINE DER(C,E). Οι δείκτες κκ και jj είναι τοπικοί δείκτες και ο μεν κκ αλλάζει από 1 ως 9, ο δε jj από 1 ως 4. Ο δείκτης κ είναι ολικός και αναφέρεται στον κομβικό άγνωστο του στοιχείου NELL που σαρώνεται. Σ' αυτό το σημείο γίνεται η αλλαγή από ολική σε τοπική αρίθμηση των συναρτήσεων βάσεως (k→kk, j→jj), ενώ σε όλες τις προαναφερθείσες σχέσεις οι δείκτες ***i***, ***k*** και ***j*** των βάσεων αναφέρονταν σε ολική αρίθμηση. Αφού τα ζυγισμένα υπόλοιπα μετατραπούν μέσω της ισοπαραμετρικής απεικονίσεως, υπολογίζονται αριθμητικά με την μέθοδο ολοκληρώσεως Gauss (quadrature formula)(6). Έτσι ολοκληρώνεται ο αλγόριθμος υπολογισμού των ζυγισμένων υπολοίπων που ενδεικτικά για την σχέση 5-17 δίδεται από τις ακόλουθες εντολές FORTRAN:

```
      DO 102      J=1, 9
      XE(J)        =X(NOP(NELL,J))
      YE(J)        =Y(NOP(NELL,J))
102   CONTINUE
      DO 103     K=1, 9
      C            =GPX(K)
      E            =GPY(K)
      CALL TFUNC(C,E)
      CALL TPRES(C,E)
      CALL DER(C,E)
      WE           =GWEI(K)*Y1
      UEVX         =FEVX(U,NELL)
      VEVY         =FEVY(V,NELL)
      DO 107  KI =1, 4
      IC           =NOPP(NOP(NELL,NNPRES(KI)))+2
      R1(IC)       =R1(IC)-WE*PHIP(KI)*(UEVX+VEVY)
107   CONTINUE
103   CONTINUE
```

Στο DO 102 δίδονται οι συντεταγμένες των κόμβων X, Y (XE(J), YE(J)) του στοιχείου που σαρώνεται (NELL). Στο DO 103 αρχίζει ο αριθμητικός υπολογισμός του ολοκληρώματος 5-

17. Πρώτα δίδονται τα σημεία ξ (C) και η(E) για την αριθμητική ολοκλήρωση. Μετά καλούνται οι υπορουτίνες όπου υπολογίζονται οι συναρτήσεις βάσεως (TFUNC για τις τετραγωνικές και TPRES για τις γραμμικές) στα σημεία ξ και η, και υπολογίζεται η ισοπαραμετρική απεικόνιση στην DER(C,E). Y1 είναι η ορίζουσα $|\mathbf{J}|$ και GWEI(K) το βάρος της μεθόδου ολοκληρώσεως Gauss. UEVX και VEVX είναι οι όροι $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ και $\frac{\partial \mathbf{v}}{\partial \mathbf{y}}$. Ενώ οι συναρτήσεις βάσεως έχουν τοπική αρίθμηση, οι κομβικοί άγνωστοι U(I), V(I) έχουν πάντα ολική αρίθμηση. Στο DO 107 υπολογίζονται τα ζυγισμένα υπόλοιπα της εξίσωσης της συνέχειας. Ο δείκτης IC εντοπίζει την ολική θέση του υπολοίπου στον πίνακα R1 και ο δείκτης NNPRES(KI) αναφέρεται στους τέσσερις κόμβους του προτύπου στοιχείου όπου αντιστοιχεί η εξίσωση R1(IC). Τέλος δίδεται ο ορισμός της σχέσης 5-17, όπου το αρνητικό πρόσημο αναφέρεται στο αντίστοιχο πρόσημο της εξίσωσης 5-1.

Για την επιτυχή επίλυση του αλγεβρικού προβλήματος (εξίσωση 5-1), πρέπει να ληφθούν οι αναγκαίες και φυσικές οριακές συνθήκες (κεφάλαιο 4°). Οι αναγκαίες οριακές συνθήκες επιβάλλονται με απόλυτη ακρίβεια. Δηλαδή, το $\Delta \mathbf{u}^1$, που αντιστοιχεί στην εξίσωση όπου υπολογίζεται η τιμή της ταχύτητας u ή v (σ' αυτό το πρόβλημα οι αναγκαίες οριακές συνθήκες αναφέρονται στις ταχύτητες), πρέπει σε κάθε επανάληψη να ισούται με το μηδέν, ικανοποιώντας έτσι τις αναγκαίες οριακές συνθήκες που επιβάλλονται στην SUBROUTINE INIGES (αρχική πρόβλεψη). Γι' αυτόν τον λόγο μηδενίζονται οι αντίστοιχες γραμμές των ζυγισμένων υπολοίπων και της Ιακωβιανής, εκτός από την στήλη της Ιακωβιανής που αντιστοιχεί στην γραμμή του ζυγισμένου υπολοίπου, όπως βλέπουμε στο σχήμα 5.2.

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \\ 0 \\ \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{xxxxxxx} \\ \mathbf{xxxxxxx} \\ 0001000 \\ \mathbf{xxxxxxx} \\ \mathbf{xxxxxxx} \end{bmatrix} * \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \\ 0 \\ \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \end{bmatrix}$$

Σχήμα 5.2: Επιβολή αναγκαίων οριακών συνθηκών.

Μέσα στο πρόγραμμα αυτές οι ενέργειες γίνονται στην SUBROUTINE FRONT. Παρ' όλα αυτά όμως, πρέπει να δοθεί προηγουμένως η εντολή από τον προγραμματιστή για το ποιες γραμμές των ζυγισμένων υπολοίπων $\mathbf{R}^1$ πρέπει να μηδενιστούν. Αυτό επιτυγχάνεται στην SUBROUTINE BNDRYC, όπου εισάγεται ο μονοδιάστατος πίνακας NCOD(I) με διάσταση NP, ίση με αυτή των ζυγισμένων υπολοίπων. Σ' αυτήν την υπορουτίνα εντοπίζονται πρώτα οι κόμβοι, όπου υπάρχουν αναγκαίες οριακές συνθήκες, μετά υπολογίζεται η σειρά των αντίστοιχων εξισώσεων στα ζυγισμένα υπόλοιπα μέσω του πίνακα NOPP(I) και μετά η τιμή αυτή δίδεται στον πίνακα NCOD(I).

Για την επιβολή των φυσικών οριακών συνθηκών στο συγκεκριμένο πρόβλημα, προκύπτει ανάγκη μόνο στην έξοδο του υπολογιστικού πεδίου, αφού στα υπόλοιπα όρια οι οριακές συνθήκες είναι αναγκαίες. Δηλαδή το επικαμπύλιο ολοκλήρωμα, όπως προκύπτει από την εφαρμογή του θεωρήματος της αποκλίσεως (εξίσωση 4-19), θα συνυπολογισθεί μαζί με αυτό του όγκου μόνο στην έξοδο του υπολογιστικού πεδίου, ενώ σε όλους τους άλλους κόμβους αγνοείται γιατί είναι εσωτερικοί. Περισσότερες λεπτομέρειες όσον αφορά την επιβολή του

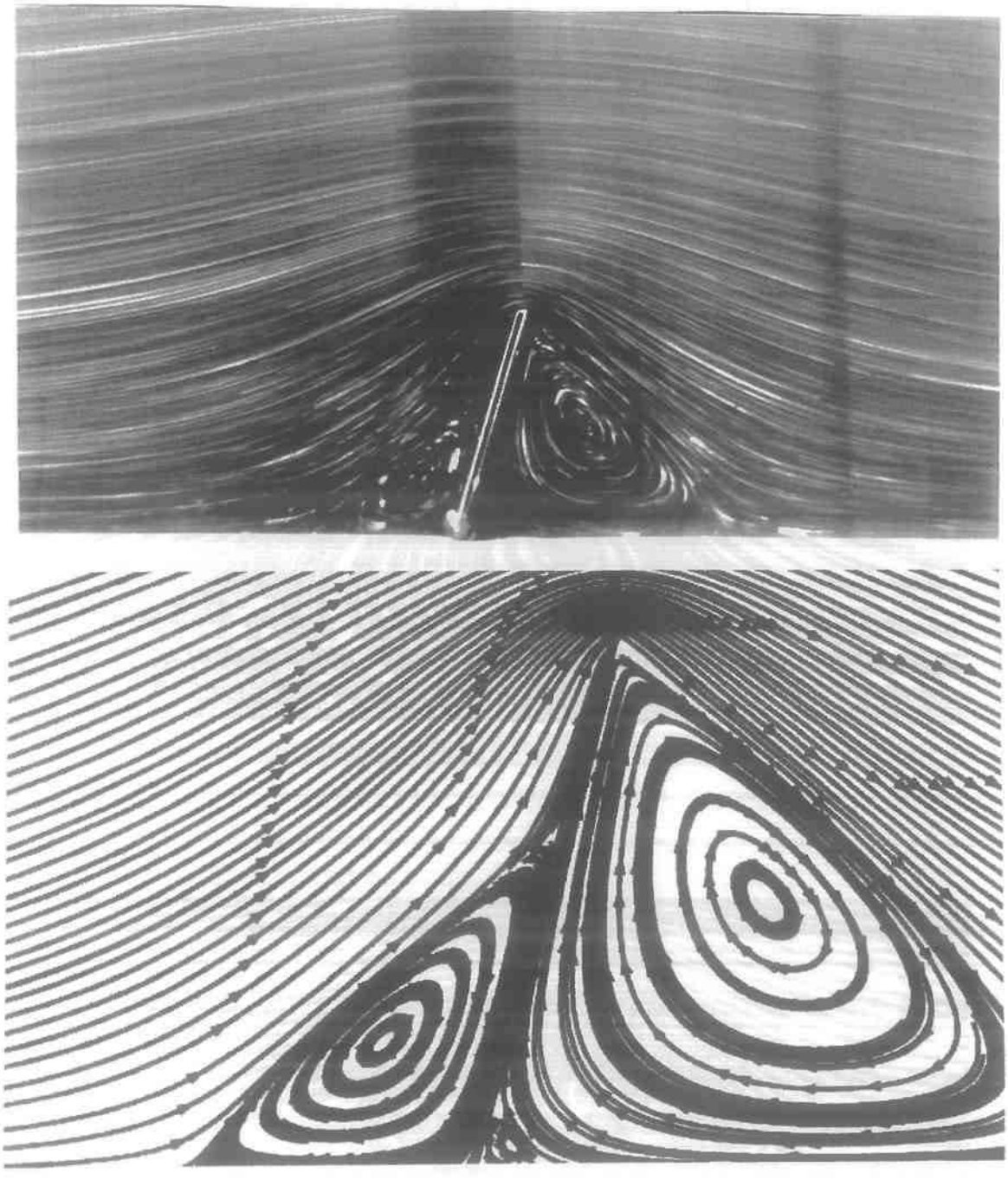
επικαμπύλιου ολοκληρώματος στην έξοδο, υπάρχουν στις εργασίες των Renardy(4) και Μαλαματάρη(10) και δεν πρόκειται να αναλυθεί περαιτέρω σ' αυτήν την εργασία.

Στο επόμενο κεφάλαιο παρουσιάζονται τα αποτελέσματα της εργασίας όπως προέκυψαν από την αριθμητική επίλυση των διεπουσών εξισώσεων με την μέθοδο των πεπερασμένων στοιχείων και την εφαρμογή της επαναληπτικής μεθόδου Newton – Raphson όπως περιγράφηκε παραπάνω.

6.1) ΡΟΗ ΓΥΡΩ ΑΠΟ ΚΕΚΛΙΜΕΝΟ ΦΡΑΚΤΗ.

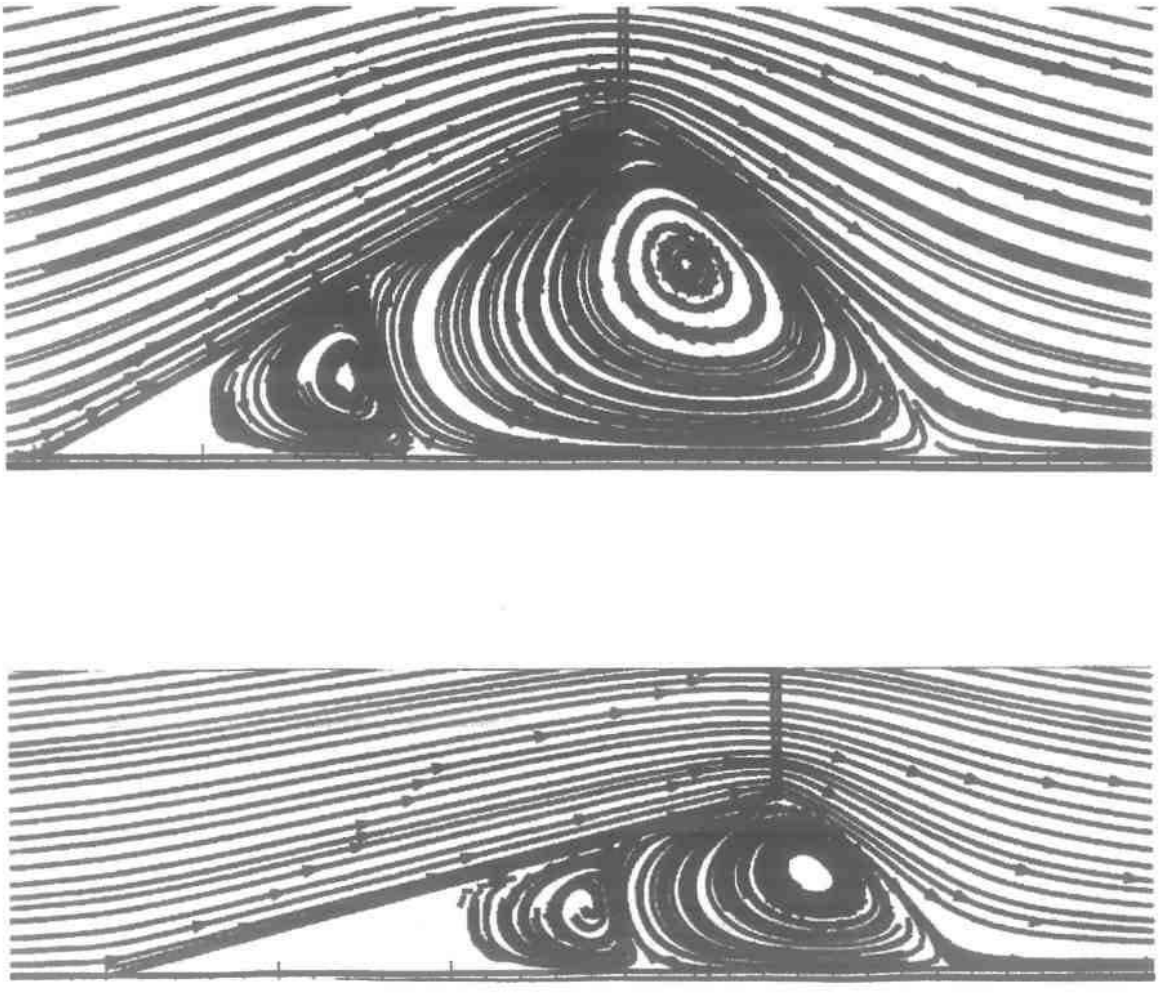
6.1.1) Ροϊκές γραμμές.

Μετά την αριθμητική επίλυση των εξισώσεων και αφού εκτελέσουμε τον κώδικα, τα αποτελέσματα που προκύπτουν για τις γραμμές ροής είναι αυτά που φαίνονται στο σχήμα 6.1.1 σε σύγκριση με το εργαστηριακό πείραμα του Taneda για γωνία κλίσης ίση με 75°.

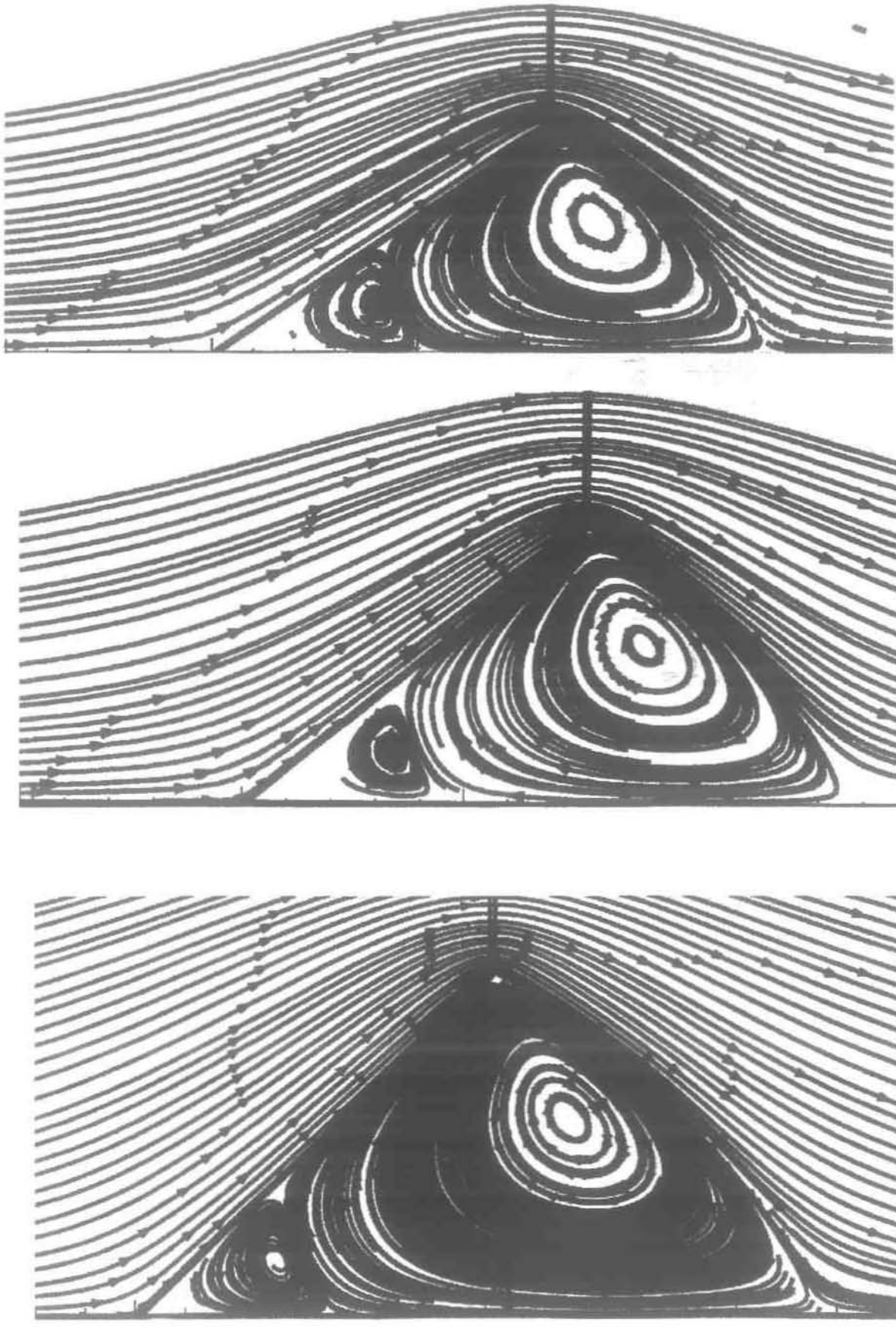


Σχήμα 6.1.1 Σύγκριση ροϊκών γραμμών του εργαστηριακού έναντι του αριθμητικού πειράματος για τη ροή γύρω από φράκτη κλίσεως 75°.

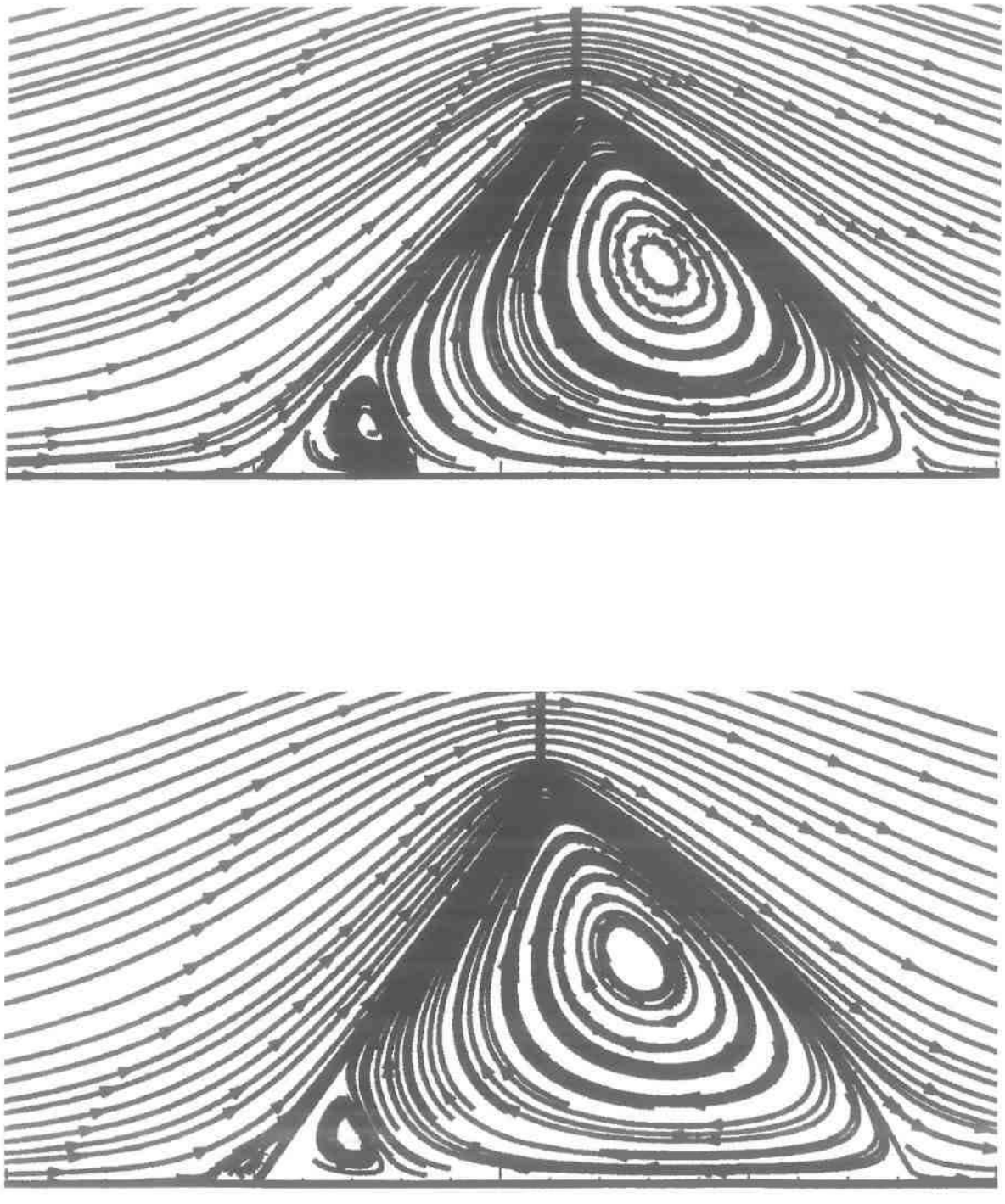
Η ομοιότητα των δύο πειραμάτων είναι φανερή. Παρατηρούμε ότι ο στρόβιλος ανάντη του εμποδίου είναι μικρότερος απ' ότι κατάντη, λόγω της ελλείψεως συμμετρίας που προκύπτει από την κλίση του φράκτη. Όπως θα δούμε στην περίπτωση της έρπουσας ροής γύρω από κατακόρυφο φράκτη η συμμετρία της ροής διατηρείται. Χαρακτηριστική διαφορά μεταξύ της φωτογραφίας του εργαστηριακού πειράματος και του αριθμητικού είναι η ύπαρξη του δεύτερου μικρού στροβίλου κατάντη του κεκλιμένου φράκτη και στη βάση αυτού, κάτι το οποίο δεν φαίνεται στην φωτογραφία του εργαστηριακού πειράματος. Επίσης θα πρέπει να παρατηρήσουμε τα βέλη των ροϊκών γραμμών απ' όπου φαίνεται ότι οι δύο στρόβιλοι μετά το εμπόδιο έχουν αντίθετη φορά. Ο μεγάλος είναι δεξιόστροφος, ενώ ο μικρός αριστερόστροφος. Κάτι ανάλογο έχει παρατηρήσει και ο Taneda στο πείραμα με γωνιακή κοιλότητα. Με το αριθμητικό πείραμα μπορούμε να δούμε με κάθε λεπτομέρεια καθετί που εξελίσσεται ανάντη και κατάντη του εμποδίου. Παρακάτω παρουσιάζεται μια πρόβλεψη για την συμπεριφορά των ροϊκών φαινομένων για χαρακτηριστικές κλίσεις 15°, 30°, 35°, 40° , 45°, 50° , 55°, 60° , 70°, 80°, 85° και 89° στα σχήματα 6.1.2-6.1.3-6.1.4-6.1.5-6.1.6-6.1.7.



Σχήμα 6.1.2 Προβλέψεις για κλίσεις 30° και 15°.

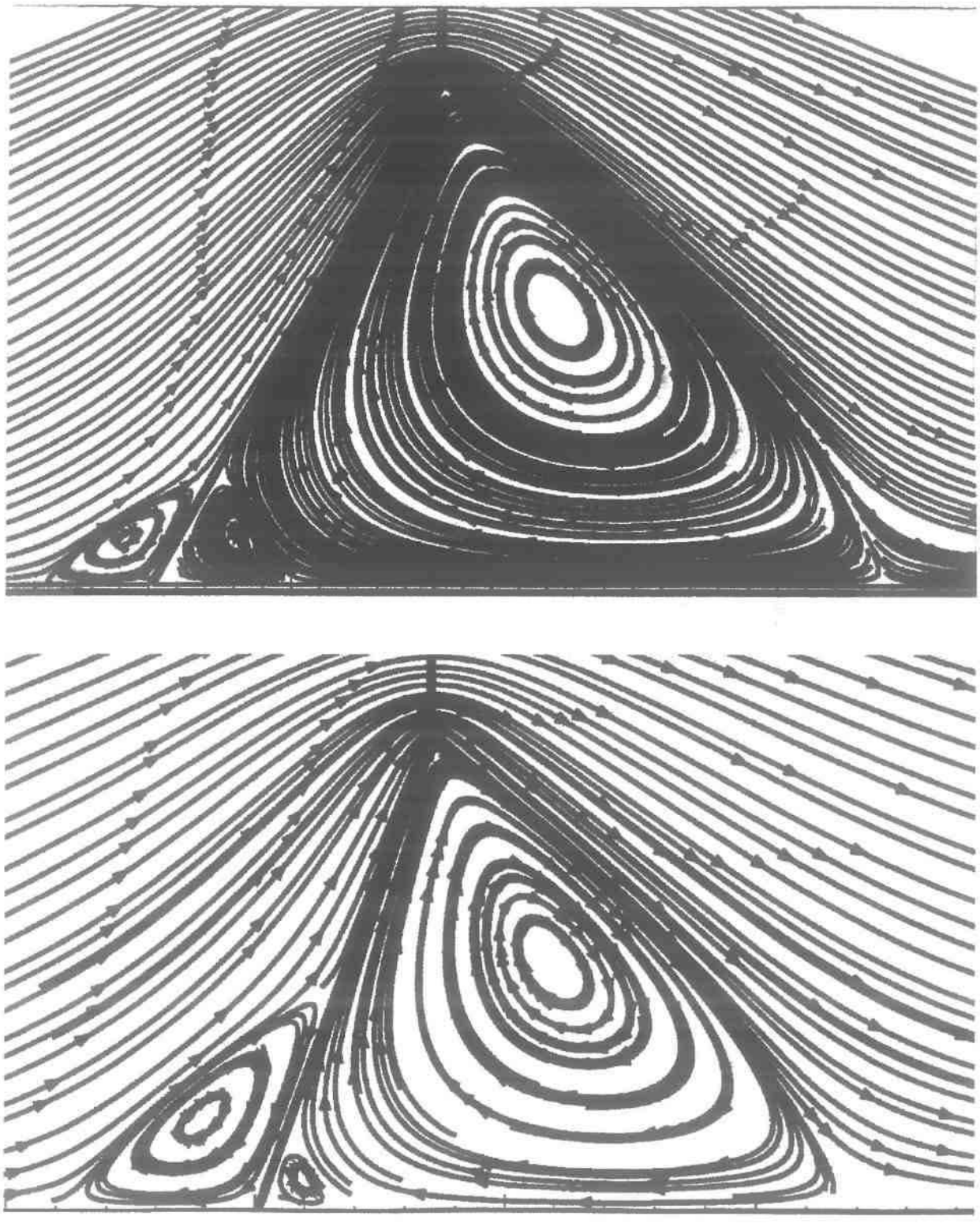


Σχήμα 6.1.3 Προβλέψεις για γωνίες κλίσεως 35°, 40° και 45°.

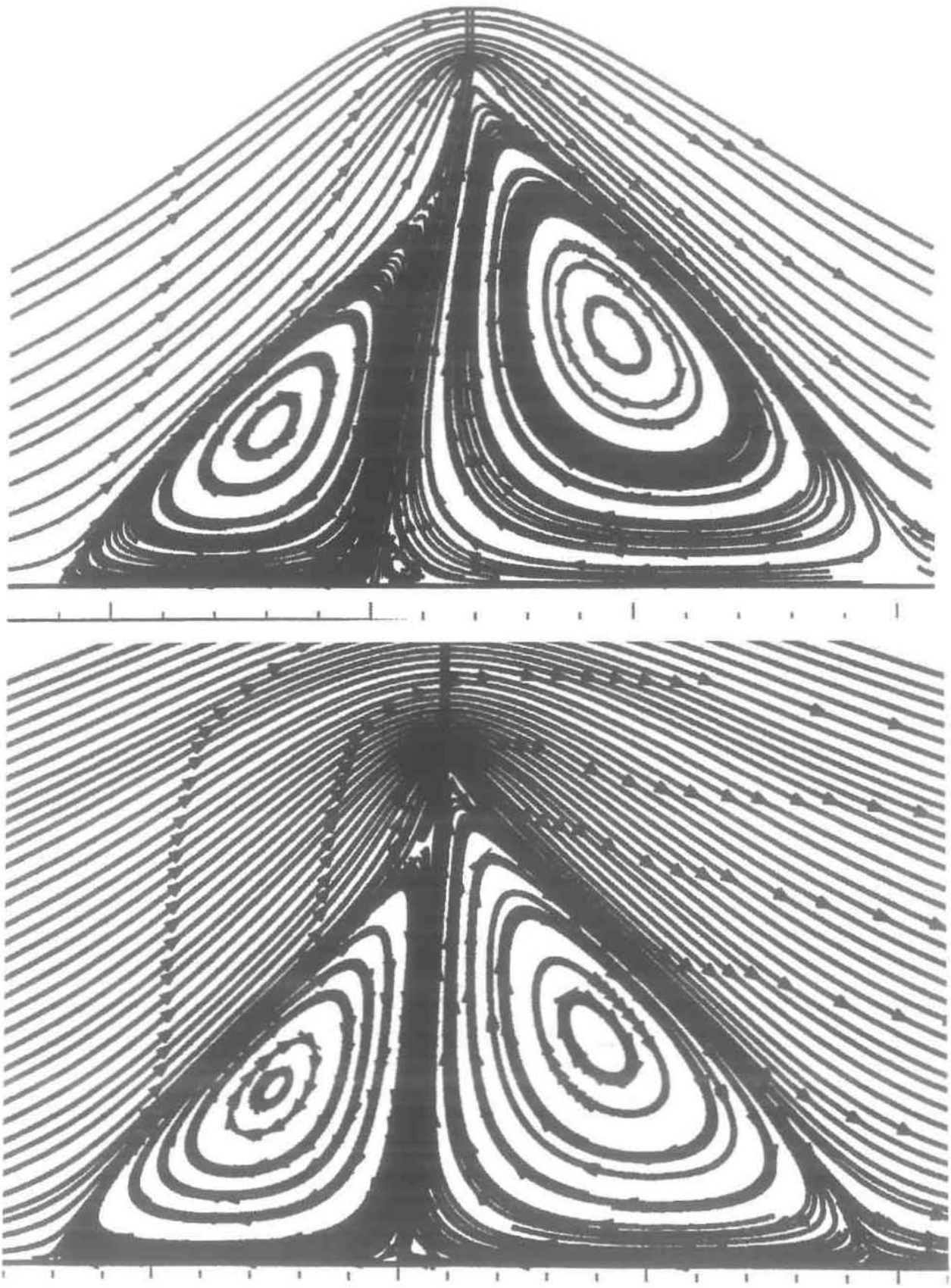


Σχήμα 6.1.4 Προβλέψεις για γωνίες κλίσεως 50^{o} και 55^{o} .

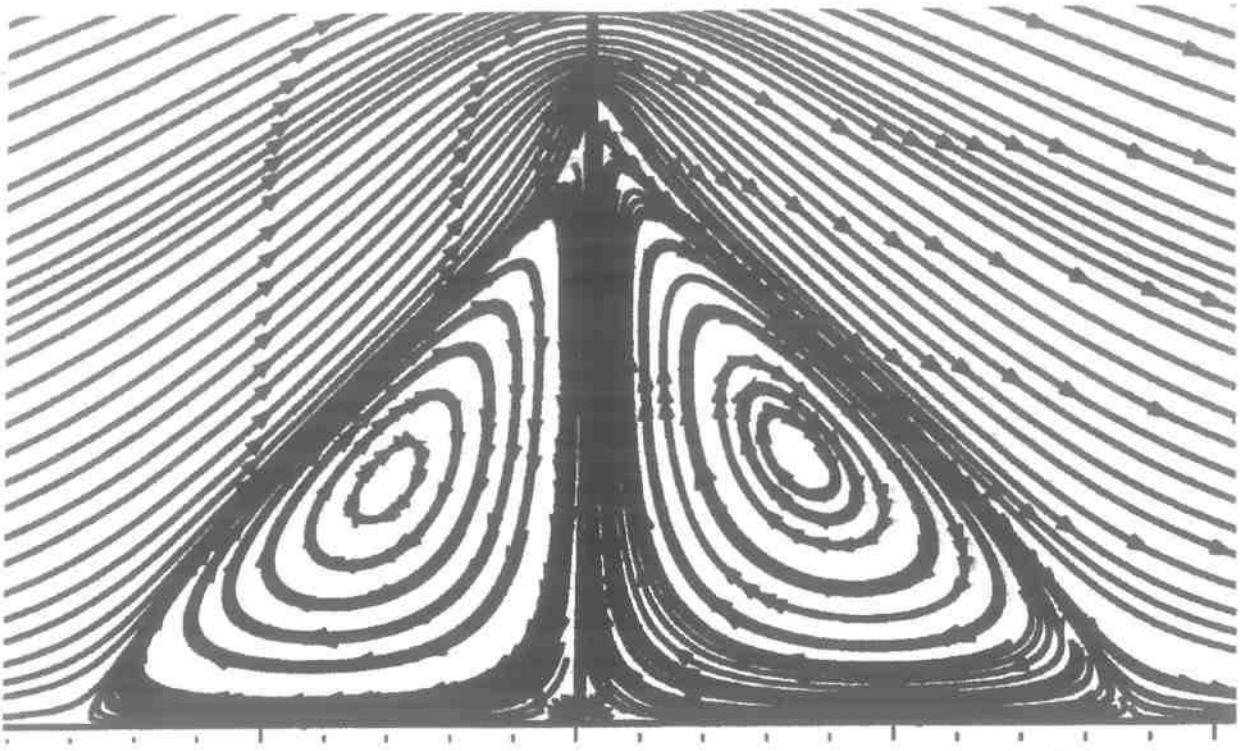
Από τις ροϊκές γραμμές παρατηρήσαμε ότι η δημιουργία του στροβίλου ανάντη του κεκλιμένου φράκτη σχηματίζεται περί των 50°, και όσο η γωνία αυξάνει τόσο ο στρόβιλος αυτός μεγαλώνει όπως θα δούμε παρακάτω.



Σχήμα 6.1.5 Προβλέψεις για γωνίες κλίσεως 60° και 70°.



Σχήμα 6.1.6 Προβλέψεις για γωνίες κλίσεως 80° και 85°.



Σχήμα 6.1.7 Πρόβλεψη για γωνία κλίσεως 89°.

Από τις ροϊκές γραμμές συμπεραίνουμε ότι ο στρόβιλος ανάντη του κεκλιμένου φράκτη εξαφανίζεται για γωνίες κλίσεως μικρότερες των 50° διότι το ρευστό δεν συναντά δυσκολία να υπερπηδήσει τον φράκτη. Όταν όμως η γωνία είναι μεγαλύτερη των 50°, τότε το ρευστό αδυνατεί να υπερπηδήσει το εμπόδιο σχηματίζοντας ανακυκλοφορία.

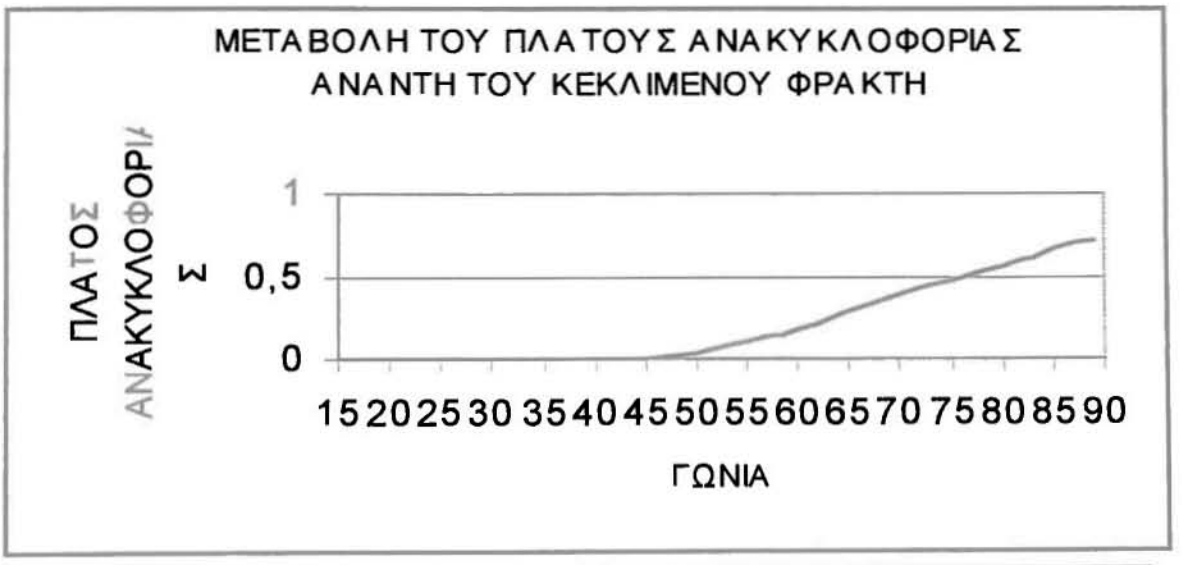
Θα πρέπει επίσης να μελετήσουμε τον μικρό στρόβιλο κατάντη του φράκτη. Όσο η γωνία μικραίνει τόσο αυτός μεγαλώνει διότι μικραίνει ο μεγάλος στρόβιλος. Οι δύο στρόβιλοι έχουν αντίθετη φορά περιστροφής και αυτό είναι που δικαιολογεί την ύπαρξή τους. Ο χώρος στην γωνία κάτω από τον κεκλιμένο φράκτη είναι λιμνάζοντα ρευστό με μηδενική ταχύτητα. Απ' όλα τα παραπάνω γίνεται αντιληπτός ο μηχανισμός διαχωρισμού των δύο στροβίλων. Δηλαδή, επειδή ο μεγάλος στρόβιλος βρίσκεται σε συνεχή επαφή με το λιμνάζοντα ρευστό σε μεγάλες γωνίες, καθώς μικραίνει η γωνία και κατά συνέπεια ο χώρος κάτω από το εμπόδιο, αναπτύσσονται δυνάμεις που προκαλούν τον διαχωρισμό του μεγάλου σε δύο στροβίλους με αντίθετη φορά περιστροφής.

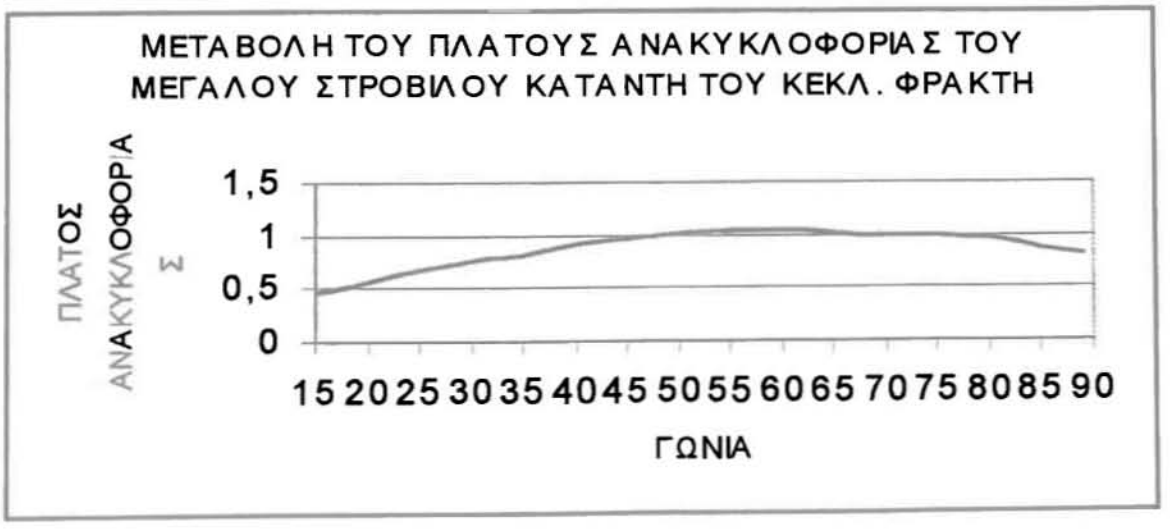
6.1.2 Διατμητικές τάσεις.

Με τα αριθμητικά πειράματα έχουμε την δυνατότητα πέρα από τον υπολογισμό των ροϊκών γραμμών, να υπολογίσουμε την διατμητική τάση στην βάση του υπολογιστικού πεδίου αλλά και κατά μήκος του εμποδίου και εν συνεχεία να προσδιορίσουμε τα σημεία αρχής και τέλους της ανακυκλοφορίας.

Η διατμητική τάση στην βάση του υπολογιστικού πεδίου είναι ίση με: $\frac{\partial \mathbf{u}}{\partial \mathbf{y}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$ αλλά επειδή στην βάση του υπολογιστικού πεδίου $\mathbf{u} = \mathbf{v} = 0 \Rightarrow \left.\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right|_{\mathbf{y}=\mathbf{o}} = 0$, έτσι η υπολογιζόμενη διατμητική τάση είναι: $\left.\frac{\partial \mathbf{u}}{\partial \mathbf{y}}\right|_{\mathbf{y}=0}$.

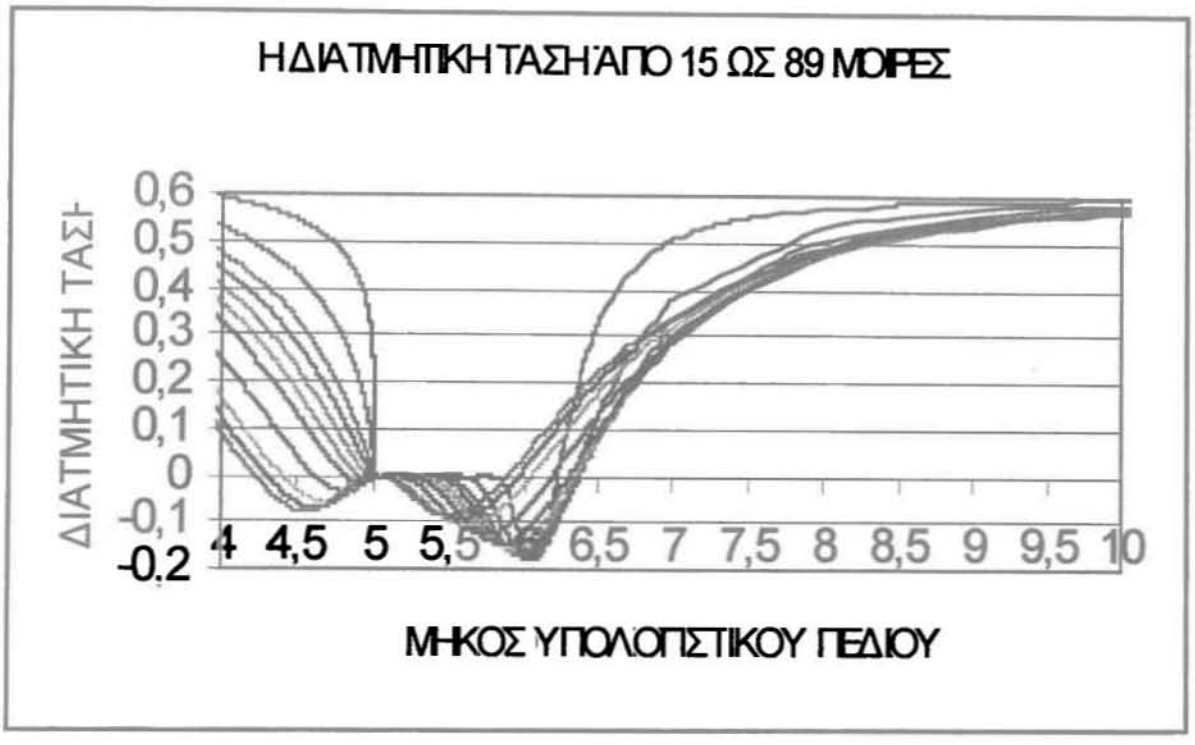
Από την επεξεργασία των αποτελεσμάτων υπολογίζονται τα ακόλουθα διαγράμματα για την μεταβολή του πλάτους ανακυκλοφορίας ανάντη και κατάντη του κεκλιμένου φράκτη και το πώς αυτό μεταβάλλεται συναρτήσει της γωνίας κλίσεως.





Όπως παρατηρούμε από τα διαγράμματα ο στρόβιλος ανάντη της ροής μικραίνει έως ότου να εξαλειφθεί περί των 45 με 50 μοιρών κλίσεως. Ταυτόχρονα ο μεγάλος στρόβιλος κατάντη της ροής παρουσιάζει μέγιστο μήκος ανακυκλοφορίας περί τις 60 μοίρες και από εκείνο το σημείο και έπειτα φθίνει όσο η γωνία κλίσεως τείνει στο μηδέν.

Παρακάτω ακολουθεί πίνακας όπου φαίνεται η κατανομή της διατμητικής τάσης για γωνίες από 15° ως 89°.



Από το παραπάνω διάγραμμα διαπιστώνουμε την μεταβολή της διατμητικής τάσης ανάντη του φράκτη και πώς αυτή μεταβάλλεται με την γωνία κλίσης. Παρατηρούμε, δηλαδή, ότι η διατμητική τάση ανάντη για γωνία κλίσεως 15° μηδενίζεται ακριβώς πάνω στην έναρξη του εμποδίου, γιατί δεν υπάρχει ανακυκλοφορία, λόγω της μικρής γωνίας, και επειδή το μήκος όπου η ταχύτητα είναι πολύ μικρή είναι σχεδόν μηδενικό. Όσο όμως η γωνία αυξάνεται τόσο η δυσκολία του ρευστού να υπερπηδήσει το εμπόδιο μεγαλώνει και κατά συνέπεια το μήκος όπου η διατμητική τάση γίνεται μηδέν είναι μεγαλύτερο. Για γωνίες κλίσεως μεγαλύτερες των 45°, που έχουμε εμφάνιση στροβίλου ανάντη, η διατμητική τάση παίρνει αρνητική τιμή, παρουσιάζει ελάχιστο στο μέσο του μήκους της ανακυκλοφορίας, για να μηδενιστεί πάλι κοντά στην αρχή του εμποδίου.

Από το διάγραμμα συμπεραίνουμε επίσης ότι όταν η γωνία κλίσεως είναι μικρή, η κατανομή της ταχύτητας ανάντη, παραμένει αναλλοίωτη από την ύπαρξη του εμποδίου. Όσο όμως η γωνία αυξάνεται, τόσο η μεταβολή της κατανομής της ταχύτητας αλλοιώνεται και μάλιστα σε απόσταση από το εμπόδιο, διότι σχηματίζεται ο στρόβιλος. Παρόμοια εξελίσσεται η κατανομή της ταχύτητας κατάντη. Σε μικρές γωνίες κλίσεως η ταχύτητα αποκτά το παραβολικό προφίλ της αμέσως μετά το τέλος του εμποδίου. Με την αύξηση της γωνίας η κατανομή της ταχύτητας αποκαθίσταται σε μεγαλύτερη απόσταση από το τέλος του εμποδίου.

Στη συνέχεια παρατηρούμε ότι κατάντη του φράκτη τον κυρίαρχο ρόλο στον υπολογισμό της διατμητικής τάσης παίζει ο μεγάλος στρόβιλος διότι ο μικρός είναι είτε πολύ μικρός σε σχέση με τον μεγάλο, είτε επειδή είναι δίπλα σε χώρο με λιμνάζον ρευστό η ταχύτητά του είναι πολύ μικρή, άρα και η διατμητική τάση αμελητέα. Το ελάχιστο της διατμητικής τάσης,

όπως φαίνεται στο διάγραμμα, δηλώνει την μέγιστη μεταβολή της ταχύτητας, η οποία παρατηρείται στο μέσο του μήκους της ανακυκλοφορίας του στροβίλου, όπου λαμβάνουν χώρα οι εντονότερες αλλαγές. Όταν η γωνία αυξάνεται ο στρόβιλος πλησιάζει τον κεκλιμένο φράκτη όπως φαίνεται και από τις ροϊκές γραμμές, γι' αυτό και μετατοπίζεται ανάλογα και το ελάχιστο της διατμητικής τάσης.

Η διατμητική τάση που υπολογίζεται ανάντη και κατάντη του κεκλιμένου φράκτη υπολογίζεται με τον ίδιο τρόπο, λαμβάνοντας όμως την στροφή των αξόνων. Η διαδικασία είναι η ακόλουθη: ο τανυστής για το νέο σύστημα αξόνων θα είναι $[\sigma'] = [Q][\sigma][Q]^T$ όπου $[\sigma] = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ ο τανυστής για ορθοκανονικό σύστημα αξόνων. Ο πίνακας Q είναι αναγκαίος για την στροφή των αξόνων και ισούται με $[Q] = \begin{bmatrix} \cos a & \sin a \\ -\sin a & \cos a \end{bmatrix}$ όπου α η γωνία κλίσεως του φράκτη. Έπετα από μια σειρά υπολογισμών η διατμητική τάση ισούται με $(D - A)\sin a \cos a + B(\cos^2 a - \sin^2 a)$ όπου $A = 2\frac{\partial u}{\partial x}, D = 2\frac{\partial v}{\partial y}, B = C = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$.

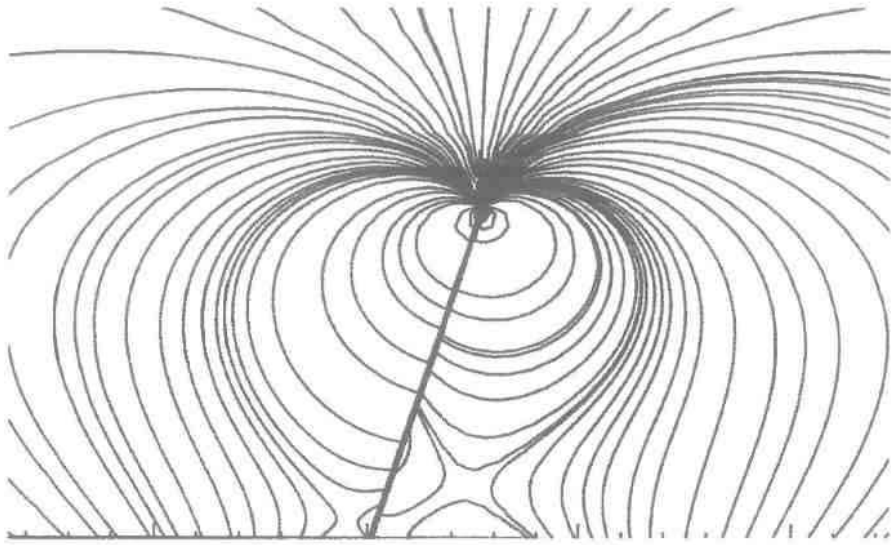
Τα αποτελέσματα που προκύπτουν από τους υπολογισμούς μας για γωνία κλίσεως από 15° μέχρι και 80° είναι τα ακόλουθα.

Από το διάγραμμα αυτό συμπεραίνουμε ότι ανάντη του εμποδίου η διατμητική τάση είναι αρνητική μόνο σε 80° γεγονός που σημαίνει ότι ο στρόβιλος είναι αρκετά μεγάλος μια και είναι και η γωνία. Οι μικροδιακυμάνσεις που δημιουργούνται στο τέλος του μήκους οφείλονται στην αλλαγή των ροϊκών γραμμών προκειμένου να υπερπηδήσει το ρευστό το εμπόδιο. Στη συνέχεια υπολογίζεται η διατμητική τάση κατάντη του κεκλιμένου φράκτη.

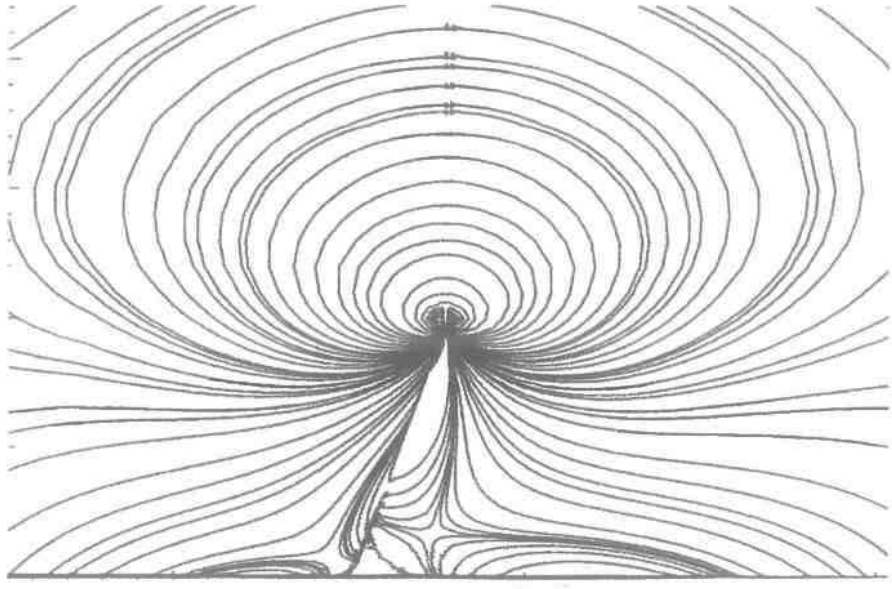
Η διατμητική τάση παρουσιάζει μεγάλες διακυμάνσεις στο τέλος του μήκους του εμποδίου, για τις διάφορες γωνίες. Όταν ο στρόβιλος είναι μεγαλύτερος από το εμπόδιο κυρίως για μικρές γωνίες κλίσεως η διατμητική τάση είναι αρνητική λόγο της ανακυκλοφορίας που δημιουργείται. Η θετική τιμή της διατμητικής τάσης εκφράζει ότι το μήκος του στροβίλου είναι μικρότερο από το μήκος του φράκτη και συναντάται σε μεγάλες γωνίες κλίσεως.

6.1.3 ΙΣΟΒΑΡΕΙΣ ΚΑΙ ΙΣΟΣΤΡΟΒΙΛΩΔΕΙΣ ΚΑΜΠΥΛΕΣ.

Μια ακόμη δυνατότητα που μας παρέχει η υπολογιστική ρευστομηχανική είναι να υπολογίζουμε τις ισοβαρείς και τις ισοστροβιλώδεις καμπύλες. Οι ισοβαρείς καμπύλες δείχνουν σε ποια σημεία του υπολογιστικού πεδίου η πίεση είναι ίδια. Αξίζει να παρουσιάσουμε στο σημείο αυτό τις ισοβαρείς και τις ισοστροβιλώδεις καμπύλες για το αρχικό πείραμα του Taneda και για γωνία κλίσεως 75°.



Παρατηρούμε τον τρόπο με τον οποίο απλώνονται οι καμπύλες γύρω από τον κεκλιμένο φράκτη. Στη συνέχεια πριν από την παρουσίαση των ισοστροβιλωδών καμπυλών θα πρέπει να δώσουμε τον ορισμό της στροβιλότητας: $\omega = -\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$.

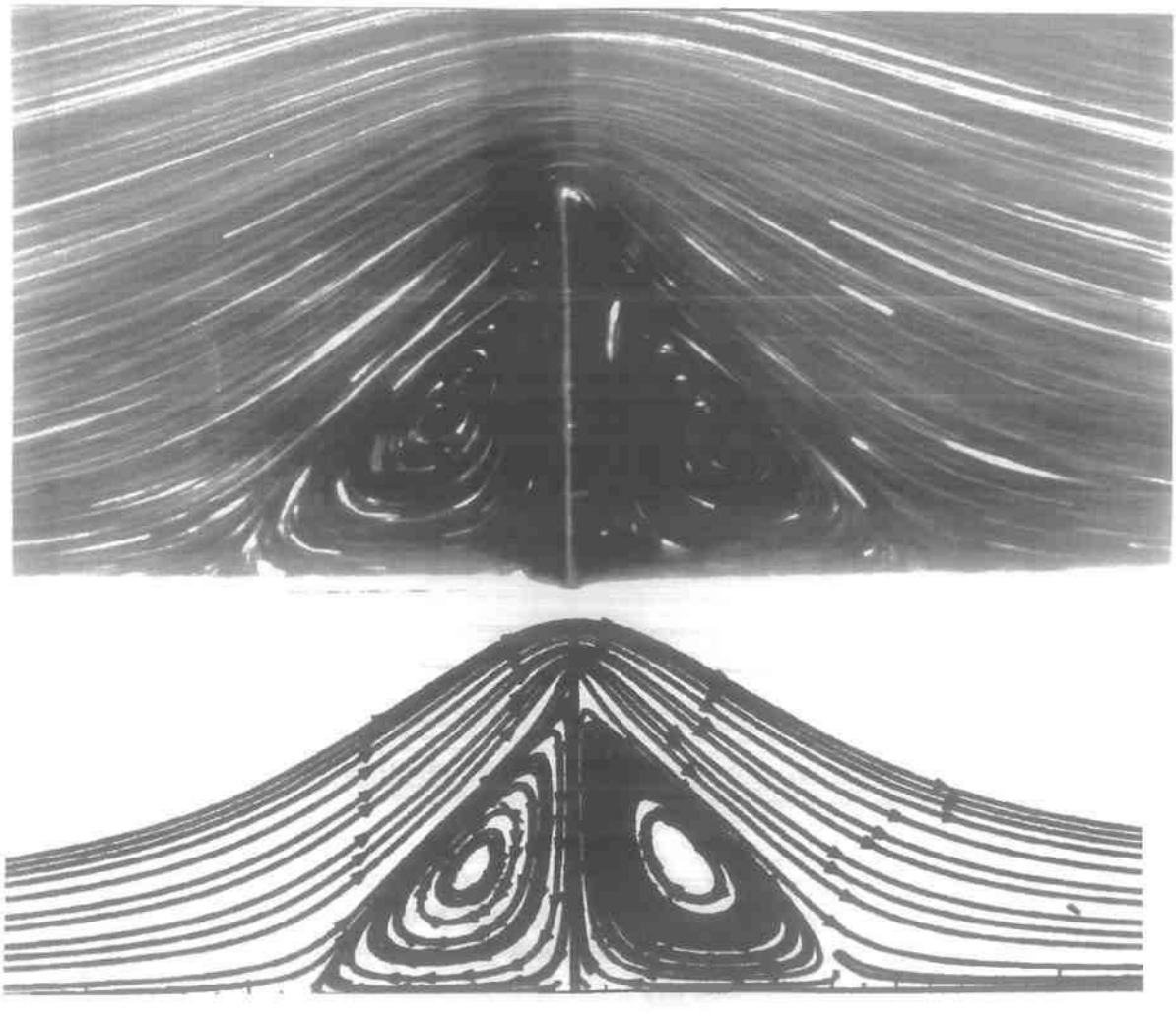


Από τις παραπάνω καμπύλες παρατηρούμε την ομοιομορφία και την συμμετρία που επικρατεί μακριά από το εμπόδιο και την αναταραχή που επικρατεί κοντά και πάνω σ' αυτό.

6.2) ΡΟΗ ΓΥΡΩ ΑΠΟ ΕΔΡΑΣΜΕΝΟ ΕΜΠΟΔΙΟ ΜΕΤΑΒΑΛΛΟΜΕΝΟΥ ΠΑΑΤΟΥΣ

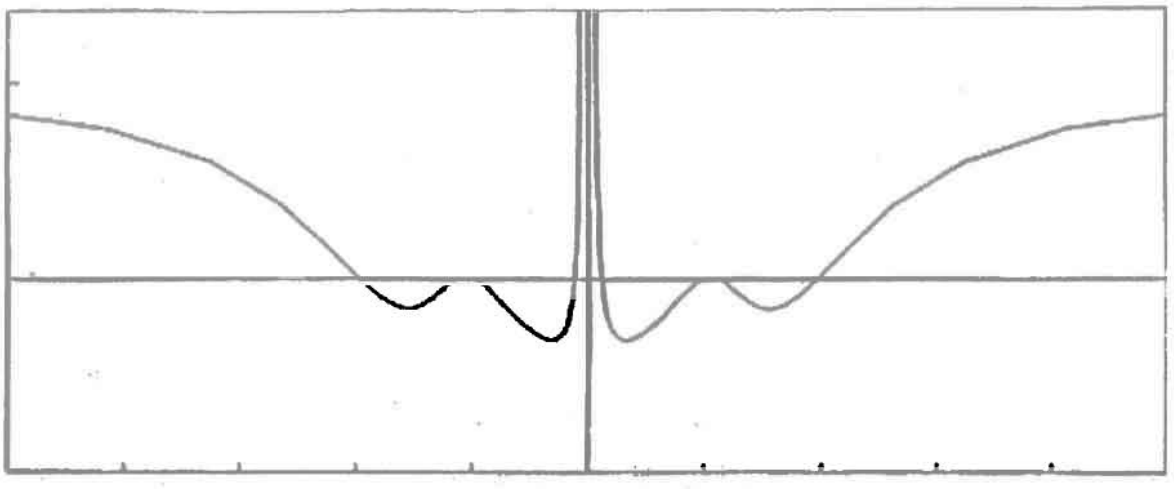
Πριν από το σχεδιασμό του προγράμματος για ροή γύρω από κεκλιμένο φράκτη, χρησιμοποιώντας έτοιμο κώδικα (8), ο οποίος υπολόγιζε ροή γύρω από εδρασμένο εμπόδιο, μελετήσαμε την ροή μεταβάλλοντας το πλάτος του εμποδίου. Ο υπολογισμός αυτός θεωρήθηκε αναγκαίος γιατί ο Taneda στο εργαστηριακό πείραμα του σε κεκλιμένο φράκτη δεν αναφέρει το πλάτος του εμποδίου. Κάνοντας μια σειρά από υπολογισμούς, μεταβάλλοντας το πλάτος του εμποδίου προκειμένου να προσεγγίσουμε αυτό που χρησιμοποιήθηκε και από τον Taneda, καταλήξαμε σε αποτελέσματα ιδιαίτερα ενδιαφέροντα, γι' αυτό και αποφασίσαμε να τα παρουσιάσουμε. Υπολογίσαμε το μήκος της ανακυκλοφορίας πριν και μετά το εμπόδιο και το πώς αυτό μεταβάλλεται με την διακύμανση του πλάτους του εμποδίου. Επίσης έγινε υπολογισμός της διατμητικής τάσης στην βάση σε όλο το μήκος της αεροσήραγγας και σύγκριση με τα αποτελέσματα του Higdon. Τέλος υπολογίσαμε τις ισοβαρείς και ισοστροβιλώδεις καμπύλες γύρω από το εδρασμένο εμπόδιο.

Αναλύοντας ξεχωριστά καθεμία από της παραπάνω εργασίες έχουμε πρώτα την σύγκριση του εργαστηριακού πειράματος με το αριθμητικό καταλήγοντας στο πλάτος του εμποδίου που θα χρησιμοποιήσουμε στον κεκλιμένο φράκτη. Η προσέγγιση είναι πολύ καλή σε σημείο που το αριθμητικό πείραμα μας δίνει με μεγαλύτερη σαφήνεια τις γραμμές ροής, όπως φαίνεται στο σχήμα 6.2.1.

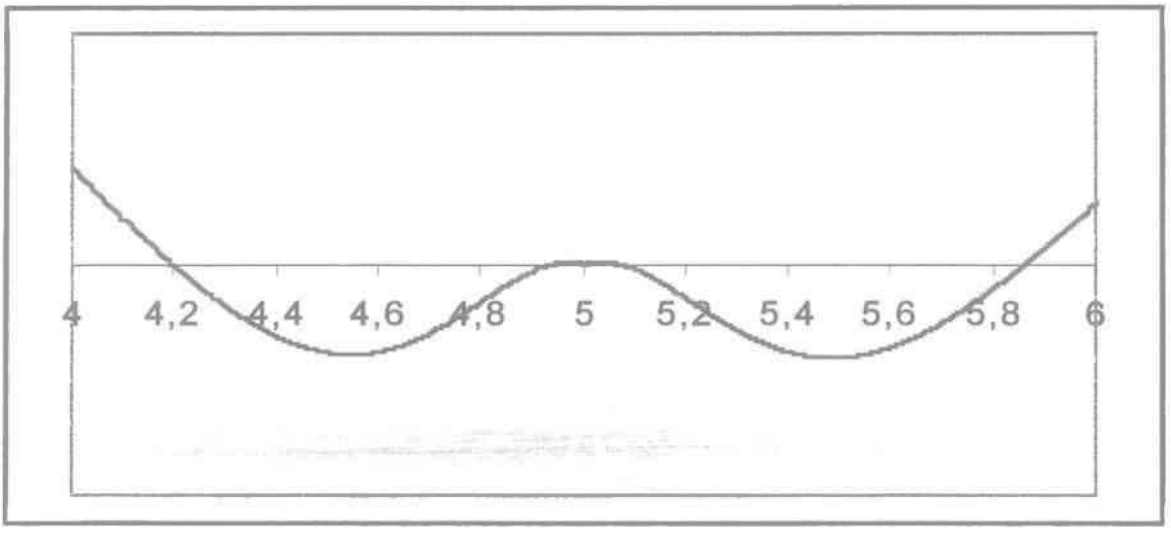


Σχήμα 6.2.1 Σύγκριση των ροϊκών γραμμών μεταξύ του εργαστηριακού πειράματος και του αριθμητικού.

Αξίζει να παρουσιάσουμε στο σημείο αυτό το διάγραμμα της διατμητικής τάσης στη βάση της αεροσήραγγας επικεντρωμένο στην περιοχή γύρω από το κατακόρυφο εμπόδιο όπως προέκυψε από τους υπολογισμούς μας σε σύγκριση με τους υπολογισμούς του Higdon, σχήμα 6.6.2.



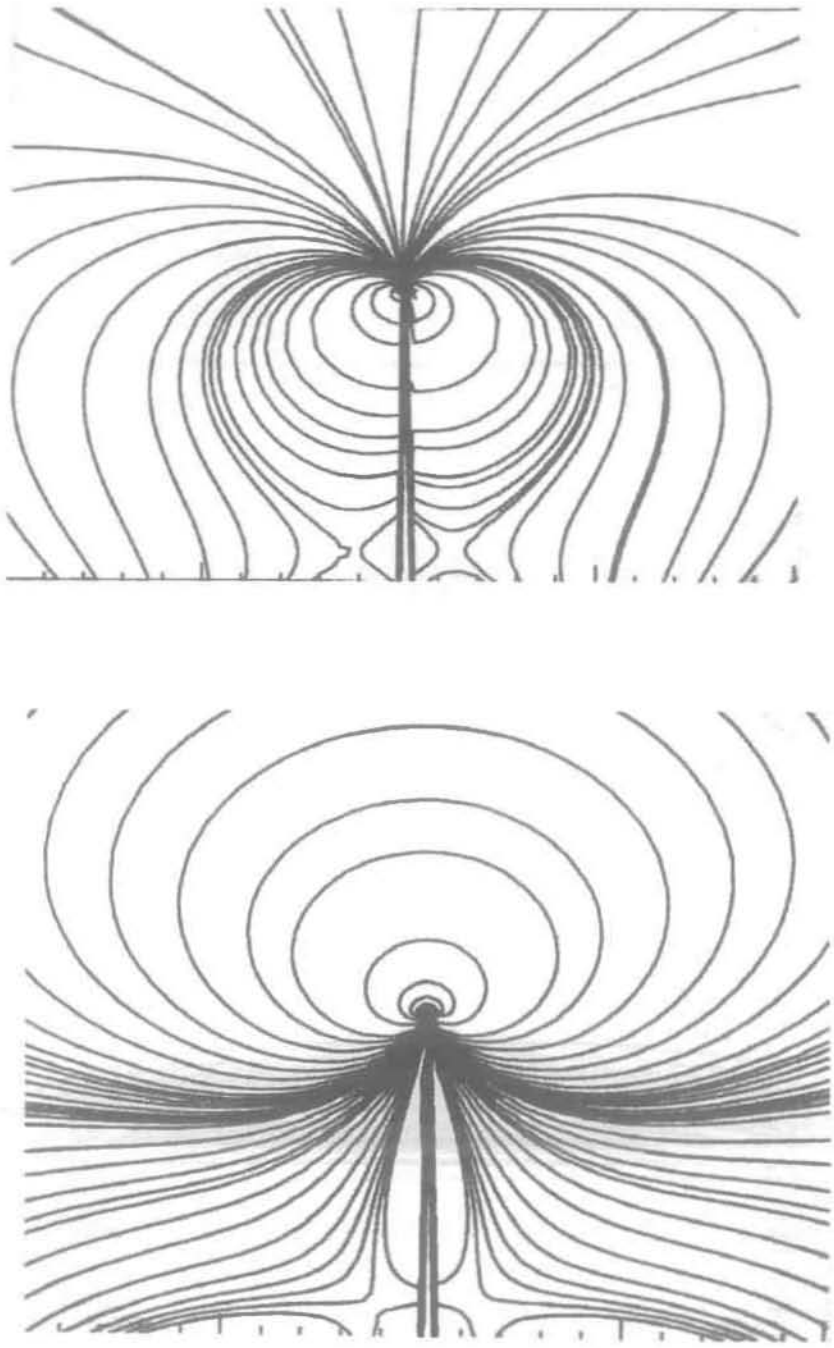
J. J. L. Higdon



Σχήμα 6.2.2 Διατμητική τάση στην βάση της αεροσήραγγας..

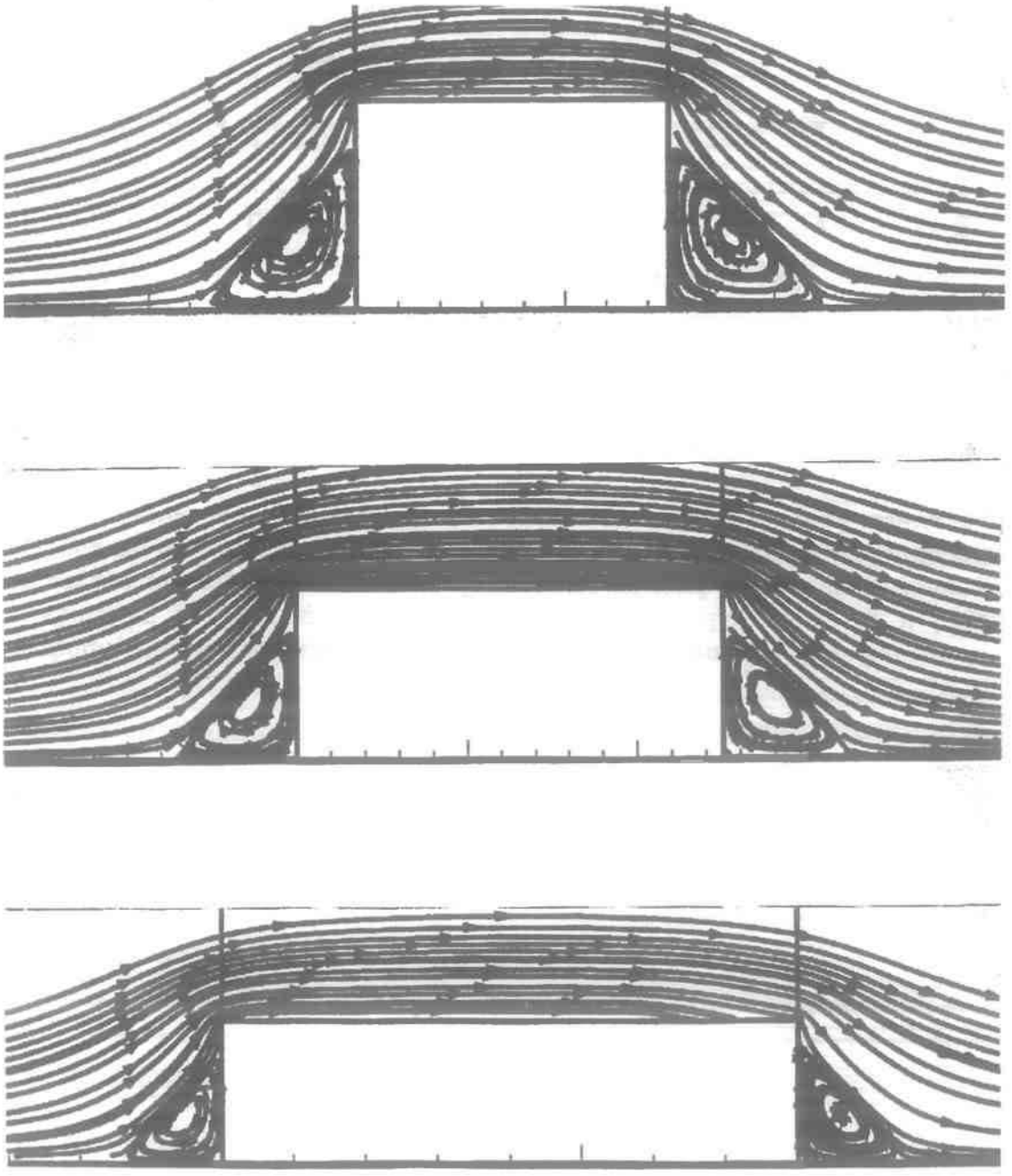
Παρατηρώντας τους υπολογισμούς του Higdon διαπιστώνουμε ότι υπάρχουν δύο περιοχές εκατέρωθεν του εμποδίου όπου η διατμητική τάση είναι αρνητική. Αυτό όμως δεν ισχύει γιατί τότε θα έπρεπε να υπάρχουν και δύο στρόβιλοι ανάντη και κατάντη του εμποδίου ενώ στην πραγματικότητα υπάρχει μόνο ένας ανάντη και ένας κατάντη. Συμπεραίνεται λοιπόν ότι η μέθοδος των πεπερασμένων στοιχείων υπολογίζει με μεγαλύτερη ακρίβεια τις διατμητικές τάσεις από άλλες μεθόδους.

Μετά τους υπολογισμούς αυτούς θα ήταν χρήσιμο να παρουσιάσουμε τις ισοβαρείς και ισοστροβιλώδεις καμπύλες για να έχουμε μια πλήρη άποψη για την κατανομή της πίεσης (ισοβαρείς) και της ταχύτητας (ισοστροβιλώδεις) γύρω από το εμπόδιο. Όπως φαίνεται στο σχήμα 6.2.3 υπάρχει συμμετρία ανάντη και κατάντη του εμποδίου, σε αντιθεση με ότι συμβαίνει όταν η κλίση μεταβληθεί.



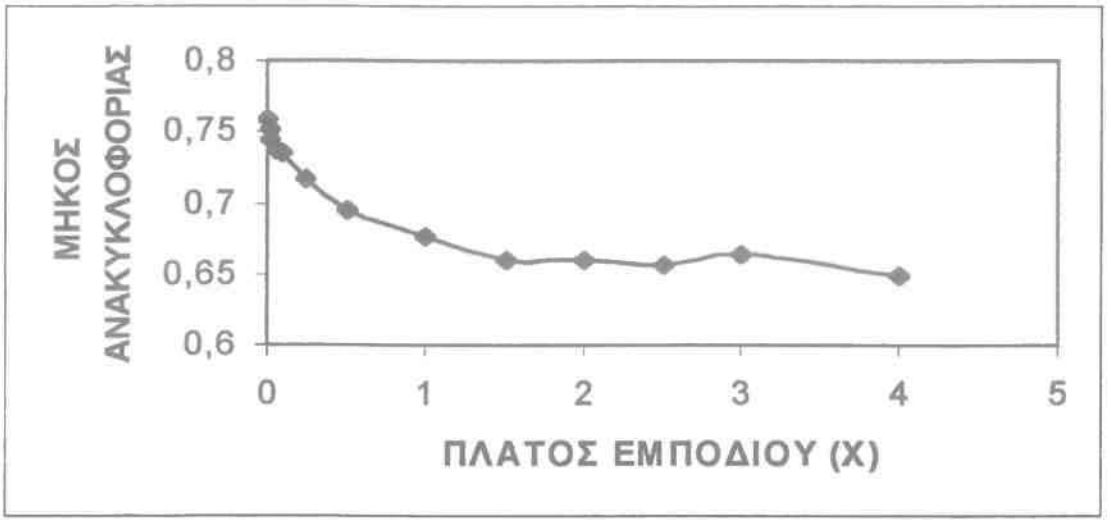
Σχήμα 6.2.3 Ισοβαρείς και ισοστροβιλώδεις καμπύλες.

Στη συνέχεια παρουσιάζονται οι γραμμές ροής γύρω από εδρασμένο εμπόδιο μεταβαλλόμενου πλάτους. Όσο αυξάνει το πλάτος του εμποδίου, τόσο το μήκος της ανακυκλοφορίας της ροής ελαττώνεται εκθετικά και για πλάτος 4L παρατηρείται η δημιουργία ροϊκών συνθηκών του ***backward facing step***. Σε όλες τις περιπτώσεις η ροή ανάντη και κατάντη του εμποδίου είναι συμμετρική, όπως φαίνεται στην εικόνα 6.2.4.



Εικόνα 6.2.4 Ροή γύρω από εδρασμένο εμπόδιο μεταβαλλόμενου πλάτους – Μεταβολή του μήκους ανακυκλοφορίας

Παρακάτω απεικονίζεται αυτή η μεταβολή του μήκους ανακυκλοφορίας συναρτήσει του πλάτους του εμποδίου στο σχήμα. 6.2.5.



Σχήμα 6.2.5 Μεταβολή του μήκους ανακυκλοφορίας ως προς το πλάτος οριζοντίως εδρασμένου εμποδίου.

Όπως παρατηρούμε για πλάτος εμποδίου μεγαλύτερο από 1,5 το μήκος ανακυκλοφορίας θα μπορούσαμε να ισχυρισθούμε ότι παραμένει σταθερό, ενώ για το σύνολο έχουμε μεταβολή της τάξεως του 15%!!! Ποιοτικά ο Higdon έκανε τις ίδιες παρατηρήσεις, χωρίς όμως την σύγκριση με πειραματικά αποτελέσματα, ενισχύοντας την άποψη ότι η μέθοδος των πεπερασμένων στοιχείων έχει την μεγαλύτερη δυνατή αξιοπιστία.

7.ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το κυρίως πρόγραμμα συντονίζει τις επιμέρους ενέργειες. Κάθε βήμα είναι καταχωρημένο σε μία ξεχωριστή υπορουτίνα. Η πλήρης λίστα του προγράμματος βρίσκεται στο παράρτημα. Στη συνέχεια παρουσιάζεται το κύριο πρόγραμμα (PROGRAM ABC) καθώς και κάθε κύρια υπορουτίνα με σχόλια.

```
**********************************************************************
      PROGRAM ABC
C
      PARAMETER (NCRIT1=939,NG=501,NXNP=173000,NL=17000,ND=65000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/  SK(NXNP),AA(25,25)
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
      COMMON /NODCOO/ XE(9),YE(9)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /IITT/   ICOUNT
C
C NCR reads the input data in INPAT which are declared in COMMON PAR and INOT
C NCO saves the solution of the steady state problem
C NPP stores the streamlines of the solution
C NLP saves the isovorticity and isobar lines of the solution
C NLI saves the results of the shear rate along the wall and the fence
C
      NCR = 34
      NCO = 36
      NPP = 37
      NLP = 39
      NLI = 38
C
      OPEN(UNIT=NCR,FILE='pa5.dat',STATUS- OLD')
      OPEN(UNIT=NCO,FILE='pa5.con',STATUS='OLD')
      OPEN(UNIT=NPP,FILE='pa5.pst',STATUS- OLD')
      OPEN(UNIT=NLI,FILE='pa5.out',STATUS='OLD')
      OPEN(UNIT=NLP,FILE='pa5.ite',STATUS- OLD')
C
      CALL INPAT
      CALL MSHSTR
      CALL MSHCOO
      CALL BNDRYC
      CALL INIGES
C
      CALL ITRATE
      CALL SVSOL
      CALL POSTPR
C
      END
```

****

Για την καλύτερη αναγνωσιμότητα του προγράμματος επελέγησαν κοινές μεταβλητές σε κάθε υπορουτίνα. Γι' αυτό στην αρχή κάθε υπορουτίνας και του κυρίου προγράμματος υπάρχουν εντολές COMMON.

Στην αρχή (μετά τις εντολές COMMON) δίδονται οι τιμές των μονάδων των αρχείων που χρειάζονται για τη λειτουργία του προγράμματος. Μετά ανοίγονται τα αρχεία η χρησιμότητα των οποίων παρουσιάζεται στις υπορουτίνες INPUT(αρχείο pa5.dat), SVSOL(αρχείο pa5.con) και POSTPR(αρχεία pa5.out, pa5.pst).

Στη συνέχεια καλείται η υπορουτίνα INPUT όπου δίδονται τα δεδομένα του προβλήματος. Στην υπορουτίνα MSHSTR υπολογίζονται τα χαρακτηριστικά του πλέγματος (Mesh-strategy), όπως οι διάφορες τιμές των παραμέτρων που είναι απαραίτητες για την αντιστροφή του πίνακα, δηλ. αριθμός στοιχείων, αριθμός αγνώστων κ.λ.π. Στην υπορουτίνα MSHCOO υπολογίζονται οι συντεταγμένες των κόμβων του υπολογιστικού πλέγματος (mesh-coordinates). Στην υπορουτίνα BNDRYC ορίζονται οι αναγκαίες οριακές συνθήκες του προβλήματος (boundary conditions) όπως αναφέρθηκε εκτεταμένα στο κεφάλαιο 3. Μετά τίθεται η αρχική πρόβλεψη στην υπορουτίνα INIGES (initial guess). Στην υπορουτίνα ITRATE ξεκινά η επίλύση του προβλήματος με την επαναληπτική μέθοδο Newton – Raphson. Στην υπορουτίνα POSTPR(post processing) επεξεργάζονται τα αποτελέσματα της λύσεως (π.χ. γραμμές ροής, διατμητικές τάσεις, κ.α).

Λεπτομερέστερη περιγραφή των επιμέρους υπορουτινών έχει ως εξής:

Ξεκινώντας πρώτα καλούμε την υπορουτίνα INPUT την οποία βλέπουμε παρακάτω:

****

```
SUBROUTINE INPUT
C
   PARAMETER (NG=501)
C
   IMPLICIT REAL*8 (A-H,O-Z)
C
   COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
   COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
  *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
   COMMON /XYWGT/ XWGHT(NG),YWGHT(NG),XAGHT(NG)
   COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
   COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
   COMMON /PAR/   RE,YY1,PE1,FR1,UO,YD,PL
   COMMON /ESSE/  IESSL
   COMMON /PARI/  ALPHA,DX,XL
C
   DATA PI /3.141592654/
C
   READ(NCR,*)
   READ(NCR,*)   MAXITE,ERRMAX,,RE,YY1
   WRITE(*,*)     MAXITE,ERRMAX, RE, YY1
C
   READ(NCR,*)
   READ(NCR,*)   NEX1A,NEX1B,NEX2,NEX3,NEX4,NEY1,NEY2
   WRITE(*,*)    NEX1A,NEX1B,NEX2,NEX3,NEX4,NEY1,NEY2
C
   READ(NCR,*)
   READ(NCR,*)   ALPHA,DX,XL
   WRITE(*,*)    ALPHA,DX,XL
C
   ALPHA=ALPHA*PI/180.
C
   NEX1 =      NEX1A + NEX1B
   NEY  =      NEY1 + NEY2
```

```
      NY1  =      1 + 2*NEY1
      NY2  =      2*NEY2
      NY   =      NY1 + NY2
      NX1  =      1+2*(NEX1A+NEX1B)
      NX2  =      1+2*(NEX1A+NEX1B+NEX2)
      NX3  =      NX2+1+2*NEX3+3
      NX   =      1+2*(NEX1A+NEX1B+NEX2+NEX3+NEX4)+4
C
      READ(NCR,*)
      READ(NCR,*)   (XWGHT(I), I=1,NX,2)
      NN1 =       1+2*(NEX1A+NEX1B+NEX2)+1+1+2*NEX3
      NN2 =       NN1+2
      XWGHT(NN2)    = XWGHT(NN1)+XL*COS(ALPHA)
      WRITE(*,*)    (XWGHT(I), I=1,NX,2)
C
      DO 11    I =    2, NX-1, 2
      XWGHT(I) =    (XWGHT(I-1)+XWGHT(I+1))/2.
   11 CONTINUE
C
      READ(NCR,*)
      READ(NCR,*)   (YWGHT(J),J=1,NY,2)
      YWGHT(NY1)    = XL*SIN(ALPHA)
      WRITE(*,*)    (YWGHT(I),I=1,NY,2)
C
      DO 18    I =    2, NY-1, 2
      YWGHT(I) =    0.5*(YWGHT(I+1)+YWGHT(I-1))
   18 CONTINUE
C
      RETURN
      END
**********************************************************************
```

Όλα τα δεδομένα που διαβάζονται είναι καταχωρημένα στο αρχείο pa5.dat όπως ορίσθηκε στο PROGRAM ABC. Στην πρώτη εντολή READ διαβάζουμε τον αριθμό Reynolds [Re] , τον μέγιστο αριθμό επαναλήψεων για τη μέθοδο Newton-Raphson (MAXITE), το μέγιστο λάθος που επιτρέπουμε για να έχουμε σύγκλιση (ERRMAX) και το ύψος εισόδου YY1.

Στην δεύτερη εντολή READ γίνεται η διακριτοποίηση του υπολογιστικού πεδίου σε πεπερασμένα στοιχεία στην X και στην Y διεύθυνση. Στο σχήμα που παρατίθεται στο παράρτημα, απεικονίζεται η σημασία των μεταβλητών NEX1A,…., NEY2, η οποία επελέγη καταλλήλως για να διευκολυνθεί ο προγραμματισμός (Sook-keeping) όπως θα δούμε παρακάτω.

Και στην τρίτη εντολή READ διαβάζουμε την γωνία του κεκλιμένου εμποδίου (ALPHA), το πλάτος του εμποδίου (DX) και το μήκος αυτού (XL). (βλέπε σχήμα στο παράρτημα).
Στη συνέχεια η γωνία ALPHA μετατρέπεται σε ακτίνια (rad) για τον ορθό υπολογισμό των τριγωνομετρικών μεγεθών.

Στις γραμμές οπού αναφέρονται οι όροι NX και NY είναι οι συνολικοί κόμβοι στον άξονα X και Y αντίστοιχα.

Στη συνέχεια το πρόγραμμα διαβάζει τις συντεταγμένες των τετμημένων και τεταγμένων του υπολογιστικού πεδίου οι οποίες πρέπει να γράφονται σε αύξουσα σειρά στο αρχείο pa5.dat. Οι τετμημένες και τεταγμένες αναφέρονται στην αρχή και το τέλος του κάθε στοιχείου, δηλαδή στους ακριανούς κόμβους. Στο τέλος υπολογίζονται οι συντεταγμένες των ενδιαμέσων κόμβων των τετμημένων και τεταγμένων (DO 11 & DO 18).

Πριν καλέσουμε την επόμενη υπορουτίνα πρέπει να υπολογίσουμε την FUNCTION NODNOR η οποία δίδει την ολική αρίθμηση των κόμβων του υπολογιστικού πλέγματος. Η ολική αρίθμηση γίνεται από «κάτω προς τα πάνω» και από αριστερά προς τα δεξιά, δίδοντας στον πρώτο κάτω αριστερά κόμβο, μετά την είσοδο, την τιμή 1. Η ιδέα του σχεδιασμού της

NODNOR σ' αυτήν την περίπτωση είναι όπως και στον υπολογισμό του NE και NX, δηλαδή χωρίζουμε το υπολογιστικό πεδίο σε απλά ορθογώνια παραλληλόγραμμα και ξεκινάμε την αρίθμηση.

```
**********************************************************************
FUNCTION NODNOR (I,J)
C
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
   *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
C
C Node numbering before the step, at the entrance.
C
    IF (I .LE. 2*NEX1A+1)
   *NODNOR = J+(I-1)*(NY+2*2*NEX1A)-2*(I-1)*(I-2)/2
C
C Node numbering before the step in the rectangular part of the discretized
C domain.
C
    IF (I .GT. 2*NEX1A+1  .AND.  I .LE. NX1)
   *NODNOR = J+(I-1)*NY+2*NEX1A*(1+2*NEX1A)
C
C Node numbering before the step in the part of the discretized domain parallel
C to the inclined obstacle.
C
    IF (I .GE. NX1+1  .AND.  I .LE. NX2)
   *NODNOR = (I-(NX1+1))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1
C
C Node numbering on the step.
C
    IF (I .EQ. NX2+1  .AND.  J .GE. NY1)
   *NODNOR = J-(NY1-1) + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
   *              2*NEX2*NY
C
C Node numbering after the step in the part of the discretized domain parallel
C to the inclined obstacle.
C
    IF (I .GE. NX2+2  .AND.  I .LE. NX2+2*NEX3+2)
   *NODNOR = (I-(NX2+2))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
   *                  2*NEX2*NY+NY2+1
C
C Node numbering after the step in the rectangular part of the discretized
C domain.
C
    IF (I .GT. NX3)
   *NODNOR = (I-(NX3+1))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
   *                  2*NEX2*NY+NY2+1+(1+2*NEX3)*NY+NY1-2+
   *                  NY1-1
C
    RETURN
    END
**********************************************************************
```

Μετά την ολική αρίθμηση του πλέγματος καλούμε την υπορουτίνα MSHSTR.

```
**********************************************************************
    SUBROUTINE  MSHSTR
C
    PARAMETER  (NXNP=173000,ND=65000,NL=17000)
C
    IMPLICIT REAL*8 (A-H, O-Z)
```

```
C
      COMMON /MESH/  NEX1A, NEX1B, NEX1,NEX2, NEX3, NEX4, NEX, NEY1, NEY2,
     *           NEY, NX1A, NX1B, NX1, NX2, NX3, NXX1, NX,NY1,NY2,NY
      COMMON /FRON1/ NP, NH, NHADD, NE, NBN(NL), NCN(NL), ND1, NWGA, NELL, NTRA
      COMMON /FRON2/ R1(NXNP), NOP(NL,9), NOPP(ND), MDF(ND), NCOD(NXNP)
      COMMON /BDYEL/ NET(NL), NEO(NL), IQ, LB
C
C     NE : #-of elements
C     NH : #-of nodes
C
      NE     = NEX1A*(NEX1A+1) + (NEX1A+NEX1B+NEX2+NEX3+NEX4+2)*NEY
      NH     = 2*NEX1A*(2*NEX1A+1) + NX1*NY + NY-2 + NY-1 + NY2+1 +
     *       2*NEX2*NY + (1+2*NEX3)*NY + NY1-2 + NY1-1 + 2*NEX4*NY
      NHADD  = NH
C
C Specify outflow elements
C
      DO 13 I = NE-NEY+1, NE
      NEO(I)  = 1
 13   CONTINUE
C
C Triangular elements at the entrance.
C
      DO 11 I = 1, NEX1A
      LB      = 1+(I-1)*(NEY+2*NEX1A)-2*(I-1)*(I-2)/2
      LT      = NEY+2*(NEX1A-(I-1))+(I-1)*(NEY+2*NEX1A)-2*(I-1)*(I-2)/2
      NET(LB) = 1
      NET(LT) = 1
 11   CONTINUE
C
C Triangular elements in the interior of the domain
C
      LIB     = NEX1A*(1+NEX1A)+(NEX1A+NEX1B)*NEY+1
      NET(LT) = 1
      LIA     = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+NEX2+NEX3+2)*NEY
      NET(LT) = 1
C
C Specify the #-of nodes NBN(I) and the #-of unknowns NCN(I) in the element
C
      DO 12 IEL= 1, NE
      IF (NET(IEL).EQ.1) THEN
      NBN(IEL) = 6
      NCN(IEL) = 15
      ELSE
      NBN(IEL) = 9
      NCN(IEL) = 22
      END IF
 12   CONTINUE
C
C Specify the degree of freedom in the element
C Remember: MDF(I)=3 at corner nodes
C           MDF(I)=2 at midnodes
C
      DO 5 I  = 1, NH
      MDF(I)  = 2
 5    CONTINUE
C
C Fix the edge nodes of quadrilateral elements
C At the entrance
C
```

```
      DO 61 I = 1, 2*NEX1A+1, 2
      DO 61 J = 1, NY-2*(I-1)+2*2*NEX1A, 2
L       = NODNOR(I, J)
      MDF(L)  = 3
 61   CONTINUE
C
C In the rectangular part of the domain upstream the fence
C
      DO 67 I = 1+2*NEX1A+2, NX1, 2
      DO 67 J = 1, NY, 2
L       = NODNOR(I, J)
      MDF(L)  = 3
 67   CONTINUE
C
C In the part of the domain upstream the fence with parallel inclination
C First the correction line
C
      DO 62 J = 2, NY-1, 2
      L       = NODNOR(NX1,NY)+NY-2 + J
      MDF(L)  = 3
 62   CONTINUE
C
C Then the main part of the domain with the parallel inclination
C
      DO 63 I = NX1+2, NX2+2+2*NEX3, 2
      DO 63 J = 1, NY, 2
L       = NODNOR(I, J)
      MDF(L)  = 3
 63   CONTINUE
C
      DO 64 J = 1, NY1-2, 2
      L       = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      MDF(L)  = 3
 64   CONTINUE
C
      DO 65 I = NX3+2, NX, 2
      DO 65 J = 1, NY, 2
      L       = NODNOR(I,J)
      MDF(L)  = 3
 65   CONTINUE
C
      NOPP(1) = 1
      DO 71 I = 2, NH
      NOPP(I) = NOPP(I-1) + MDF(I-1)
 71   CONTINUE
C
      NP      = 0
      DO 8  I = 1, NH
      NP      = NP + MDF(I)
 8    CONTINUE
C
      WRITE(*.101) NE, NH, NP
C
      IF (NOPP(NH)+MDF(NH)-1 .EQ. NP) THEN
      PRINT*, 'NP CORRECT!'
      ELSE
      PRINT*, 'CHECK NP!'
      STOP
      END IF
C
```

```
      CALL CALNOP
C
 101  FORMAT(' NE, NH, NP : ', 3I7)
      RETURN
      END
*****************************************************************
```

Αρχικά γίνεται ο υπολογισμός των συνολικών στοιχείων NE και κόμβων NH. Η ιδέα για τον υπολογισμό αυτόν είναι να τμήσουμε το υπολογιστικό πεδίο σε απλά γεωμετρικά σχήματα και μετά αθροιζόμενα να δώσουν τον συνολικό αριθμό NE, NH. Στο DO 13 δηλώνονται ποια είναι τα τετραγωνικά στοιχεία της εξόδου εισάγοντας το διάνυσμα NEO(I) και δίδοντας σ' αυτό την τιμή 1 στα κατάλληλα στοιχεία.

Στο DO 11 δηλώνουμε τα πάνω (LT) και κάτω (LB) τριγωνικά στοιχεία στην είσοδο αξιοποιώντας την ίδια ιδέα όπως και για τον χαρακτηρισμό των στοιχείων της εξόδου. Η σημασία της εισαγωγής των πινάκων NEO(I) και NET(I) θα φανεί στην παρουσίαση της υπορουτίνας ABFIND.

Στη συνέχεια ορίζονται τα δύο τριγωνικά στοιχεία στο εσωτερικό του υπολογιστικού πεδίου (σχήμα στο παράρτημα). Ο όρος (NEX1A+NEX1B)*NEY είναι τα στοιχεία στο ορθογώνιο τμήμα της εισόδου και ανάντη του εμποδίου. Οι όροι NEX2*NEY και NEX3*NEY είναι ο αριθμός των στοιχείων ανάντη και κατάντη του εμποδίου στο τμήμα του υπολογιστικού πεδίου που είναι παράλληλα με την κλίση του εμποδίου. Ήδη, από την ανάγκη για τον εντοπισμό των τριγωνικών στοιχείων LIA, LIB είναι προφανής η χρησιμότητα του ορισμού των μεταβλητών NEX1A, ... , NEY2 που εισήχθη στην INPUT.

Στο DO 12 ορίζεται ο αριθμός των κόμβων και των αγνώστων ανά κόμβο για κάθε είδος στοιχείου.Οι πίνακες NBN(I) και NCN(I) χρησιμοποιούνται μόνο στην FRONT, δηλαδή στην υπορουτίνα που αντιστρέφει την Ιακωβιανή (12), και δεν πρόκειται να εξηγηθούν στην παρούσα εργασία εκτενέστερα. Έπειτα δηλώνεται ο αριθμός των αγνώστων ανά κόμβο στο DO 5, ενώ από στα DO 61-67-62-63-64-65 δηλώνουμε ποιοι από τους κόμβους έχουν τρεις αγνώστους. Όπως οι πίνακες NBN(I) και NCN(I), έτσι και ο πίνακας MDF(I) χρησιμοποιείται μόνο στην FRONT. Στο DO 71 ορίζεται η θέση της εξισώσεως R'_u (κεφάλαιο 4) στον πίνακα των υπολοίπων. Ο πίνακας NOPP(I) εκτός από την FRONT χρησιμοποιείται και στις υπορουτίνες BNDRYC και ABFIND διευκολύνοντας τον προγραμματισμό. Και τέλος στο DO 8 υπολογίζεται ο συνολικός αριθμός αγνώστων NP προσθέτοντας συνεχώς τους αγνώστους σε κάθε κόμβο (MDF(I)).

Το IF-block ελέγχει κατά πόσον οι υπολογισμοί που έγιναν μέχρι τώρα σ' αυτήν την υπορουτίνα καθώς και η εισαγωγή των δεδομένων στην INPUT είναι ορθοί. Έτσι προλαμβάνονται τυχόν λάθη στο πρόγραμμα πριν προβούμε στην καθ' αυτό επίλυση του προβλήματος.

Στο τέλος της MSHSTR καλείται η CALNOP για τον υπολογισμό του πίνακα NOP(I,J). Λόγω του εκτενέστερου μεγέθους των υπολογισμών του NOP(I,J) κρίθηκε απαραίτητη η δημιουργία ξεχωριστής υπορουτίνας. Ο πίνακας NOP(I,J) χρησιμοποιείται στην FRONT, αλλά έχει κεντρική παρουσία και στην ABFIND για την διευκόλυνση του προγραμματισμού.

Η CALNOP έχει ως εξής:

```
*****************************************************************
SUBROUTINE CALNOP
C
      PARAMETER (NXNP=173000,NL=17000,ND=65000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
```

```
      COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /INOUT/ NCR,NLP,NCO,NSA,NPP,NLI
C
C K locates the global node of the bottom left corner of the NK element
C JJ-1 takes care of the nodes above K
C NY*(II-1) takes care of the adjacent nodes to K
C KK is the local node
C
C Triangular elements in the entrance
C
      DO 1 I   = 1, NEX1A
      NEI      = 2*((I-1)*(I-2)/2)
      LB       = 1+(I-1)*(NEY+2*NEX1A)-NEI
      NOP(LB,1) = NODNOR(2*(I-1)+1, 1)
      NOP(LB,2) = NODNOR(2*(I-1)+3, 1)
      NOP(LB,3) = NOP(LB,1)+2
      NOP(LB,4) = NODNOR(2*(I-1)+2, 1)
      NOP(LB,5) = NOP(LB,4)+1
      NOP(LB,6) = NOP(LB,1)+1
      LT       = NEY+2*(NEX1A-(I-1))+(I-1)*(NEY+2*NEX1A)-NEI
      NOP(LT,1) = NODNOR(2*(I-1)+1,NY+2*2*(NEX1A-I)+4)
      NOP(LT,2) = NOP(LT,1)-2
      NOP(LT,3) = NODNOR(2*(I-1)+3,NY+2*2*(NEX1A-I))
      NOP(LT,4) = NOP(LT,1)-1
      NOP(LT,6) = NODNOR(2*(I-1)+2,NY+2*2*(NEX1A-I)+2)
      NOP(LT,5) = NOP(LT,6)-1
    1 CONTINUE
C
C Quadrilateral elements in the entrance
C
      DO 51 I   = 1, NEX1A
      DO 51 J   = 1, NEY+2*(NEX1A-I)
      NK        = 1+(I-1)*NEY+J+2*(NEX1A*(NEX1A+1)/2)-
     *            2*((NEX1A-(I-1))*(NEX1A-(I-1)+1)/2)
      DO 51 II  = 1, 3
      K         = NODNOR(2*(I-1)+II,3-(II-1)+2*(J-1))
      DO 51 JJ  = 1, 3
      KK        = (II-1)*3+JJ
      NOP(NK,KK) = K+JJ-1
   51 CONTINUE
C
C Quadrilateral elements upstream the obstacle
C
      DO 52 I   = 1, NEX1B
      DO 52 J   = 1, NEY
      K         = 1+2*(I-1)*NY+2*(J-1)+2*(2*NEX1A*(2*NEX1A+1))/2+
     *            2*NEX1A*NY
      NK        = (I-1)*NEY+J+2*(NEX1A*(NEX1A+1)/2)+NEX1A*NEY
      DO 52 II  = 1, 3
      DO 52 JJ  = 1, 3
      KK        = (II-1)*3+JJ
      NOP(NK,KK) = K+(II-1)*NY+JJ-1
   52 CONTINUE
C
C Triangular elements in the interior of the domain
C
      LIB       = NEX1A*(1+NEX1A) + (NEX1A+NEX1B)*NEY + 1
      NOP(NK,1) = NODNOR(NX1,1)
      NOP(NK,2) = NODNOR(NX1,NY)+NY-2 + 2
      NOP(NK,3) = NOP(NK,1) + 2
```

```
      NOP(NK,4) = NOP(NK,2) - 1
      NOP(NK,5) = NODNOR(NX1,NY) + 1
      NOP(NK,6) = NOP(NK,1) + 1
      LIA       = NEX1A*(1+NEX1A) + (NEX1A+NEX1B+NEX2+NEX3+2)*NEY
      NOP(NK,1) = NODNOR(NX2+2+2*NEX3,NY1)
      NOP(NK,2) = NOP(NK,1) - 2
      NOP(NK,3) = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + NY1-2
      NOP(NK,4) = NOP(NK,1) - 1
      NOP(NK,5) = NODNOR(NX2+2+2*NEX3,NY) + NY1-2
      NOP(NK,6) = NOP(NK,3) + 1
C
C Quadrilateral elements above the first triangle in the domain
C
      DO 41  J = 2, NEY
      K         = NODNOR(NX1,2*(J-1)+1)
      NK        = NEX1A*(1+NEX1A)+(NEX1A+NEX1B)*NEY + J
      DO 41 II  = 1, 3
      DO 41 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      IF (II .EQ. 1)  THEN
      NOP(NK,KK) = K + JJ-1
      ELSE IF (II .EQ. 2)  THEN
      NOP(NK,KK) = K + JJ-1 + NY-2
      ELSE
      NOP(NK,KK) = K + JJ-1 + NY-2 + NY-1
      END IF
   41 CONTINUE
C
C The first quadrilateral element to right of the trianlge
C
      NK        = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + 1
      NOP(NK,1) = NODNOR(NX1,1)
      NOP(NK,2) = NODNOR(NX1,NY)+NY-2 + 1
      NOP(NK,3) = NOP(NK,2) + 1
      DO 42 II  = 2, 3
      DO 42 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      NOP(NK,KK) = NODNOR(NX1,NY)+NY-2+NY-1 + 1 + (II-2)*NY + JJ-1
   42 CONTINUE
C
C The rest of the quadrilateral elements above
C
      DO 43  J = 2, NEY
      NK        = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + J
      K         = NODNOR(NX1,NY)+NY-2 + 2*(J-1)
      DO 43 II  = 1, 3
      DO 43 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   43 CONTINUE
C
C Quadrilateral elements upstream the fence with parallel orientation
C
      DO 44  1 = 2, NEX2
      DO 44  J = 1, NEY
      NK        = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + (I-1)*NEY+J
      K         = NODNOR(NX1+2*(I-1),1+2*(J-1))
      DO 44 II  = 1, 3
      DO 44 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
```

```
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   44 CONTINUE
C
C Quadrilateral elements above the fence
C
      DO 45  J = 1, NEY2
      NK       = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1+NEX2)*NEY + J
      K        = NODNOR(NX2,2*(J-1)+NY1)
      DO 45 II  = 1, 3
      DO 45 JJ  = 1, 3
      KK       = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*(NY2+1)
      IF (II .EQ. 3)
     *NOP(NK,KK) = NOP(NK,KK) + NY1-1
   45 CONTINUE
C
C Quadrilateral elements downstream the fence with parallel irientation
C
      NEPR     = NEX1A*(1+NEX1A) + (NEX1A+NEX1B+1+NEX2)*NEY + NEY2
      DO 46  I = 1, NEX3
      DO 46  J = 1, NEY
      NK       = NEPR + (I-1)*NEY+J
      K        = NODNOR(NX2+2*(I-1)+2,2*(J-1)+1)
      DO 46 II  = 1, 3
      DO 46 JJ  = 1, 3
      KK       = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   46 CONTINUE
C
C Correction-quadrilateral elements below the triangle
C
      NEPR     = NEPR + NEX3*NEY
      DO 47  J = 1, NEY1-1
      NK       = NEPR + J
      K        = NODNOR(NX2+2+2*NEX3,2*(J-1)+1)
      DO 47 II  = 1, 3
      DO 47 JJ  = 1, 3
      KK       = 3*(II-1)+JJ
      IF (II .LE. 2)
     *NOP(NK,KK) = K + JJ-1 + (II-1)*NY
      IF (II .EQ. 3)
     *NOP(NK,KK) = K + JJ-1 + NY + NY1-2
   47 CONTINUE
C
C Quadrilateral elements next to the correction-quadrilateral elements
C
      NEPR     = NEPR + NEY1
      DO 48  J = 1, NEY1-1
      NK       = NEPR + J
      K        = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + 2*(J-1)+1
      DO 48 II  = 1, 3
      DO 48 JJ  = 1, 3
      KK       = 3*(II-1)+JJ
      IF (II .LE. 2)
     *NOP(NK,KK) = K + JJ-1 + (II-1)*(NY1-1)
      IF (II .EQ. 3)
     *NOP(NK,KK) = K + JJ-1 + NY1-1 + NY
   48 CONTINUE
C
      NK       = NEPR+NEY1
```

```
      NOP(NK,1) = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + NY1-2
      NOP(NK,2) = NOP(NK,1)+1
      NOP(NK,3) = NODNOR(NX2+2+2*NEX3,NY1)
      DO 57 II  = 2, 3
      DO 57 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      IF (II .EQ. 2)
     *NOP(NK,KK) = NOP(NK,1) + NY1-1 + JJ-1
      IF (II .EQ. 3)
     *NOP(NK,KK) = NOP(NK,1) + NY1-1 + NY + JJ-1
   57 CONTINUE
C
      NEPR      = NK
      DO 49  J  = 1, NEY2
      NK        = NEPR + J
      K         = NODNOR(NX2+2+2*NEX3,2*(J-1)+NY1)
      DO 49 II  = 1, 3
      DO 49 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      IF (II .EQ. 1)
     *NOP(NK,KK) = K + JJ-1
      IF (II .GE. 2)
     *NOP(NK,KK) = K + JJ-1 + NY1-2+NY1-1 + (II-1)*NY
   49 CONTINUE
C
C Rectangular elements up to the outflow
C
      NEPR      = NEPR + NEY2
      DO 50  I  = 2, NEX4
      DO 50  J  = 1, NEY
      NK        = NEPR + (I-2)*NEY+J
      K         = NODNOR(NX3+2*(I-1),2*(J-1)+1)
      DO 50 II  = 1, 3
      DO 50 JJ  = 1, 3
      KK        = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   50 CONTINUE
C
      RETURN
      END
*****************************************************************
```

Στην υπορουτίνα αυτή αντιστοιχούμε κάθε κόμβο του υπολογιστικού πεδίου (ολική αρίθμηση), με τον κόμβο κάθε στοιχείου στο οποίο ανήκει. Δηλαδή υπάρχουν δύο τρόποι ταυτοποίησης του υπολογιστικού πεδίου. Ο ένας είναι αριθμώντας τους κόμβους ολικά ενώ ο άλλος είναι αριθμώντας τα στοιχεία ολικά και δίδοντας σε κάθε κόμβο του στοιχείου τον αριθμό του, που του αντιστοιχεί στην ολική αρίθμηση. Δηλαδή, θέλουμε για το κάθε στοιχείο του υπολογιστικού πεδίου να αντιστοιχίσουμε τους εννέα κόμβους του, εάν πρόκειται για τετραγωνικό και έξι αν πρόκειται για τριγωνικό στοιχείο, με τον αριθμό που έχει δοθεί σ' αυτούς τους κόμβους στην ολική αρίθμηση [που έγινε στην FUNCTION NODNOR(I,J)] για να διευκολύνουμε τους υπολογισμούς των εξισώσεων εκμεταλλευόμενοι την ισοπαραμετρική απεικόνιση και το πρότυπο στοιχείο για τα οποία μιλήσαμε στα κεφάλαια 4 και 5.

Εξάλλου η χρησιμότητα του πίνακα NOP(NK,KK) θα φανεί στην υπορουτίνα ABFIND όπου μαζεύονται οι εξισώσεις στοιχείο προς στοιχείο και γίνεται φανερή η ανάγκη της ισοπαραμετρικής απεικόνισης καθώς και η αποτελεσματικότητα της εισαγωγής του NOP πίνακα.

Το NOP(NK,KK) μας δίνει τον ολικό αριθμό του κόμβου του στοιχείου NK, όπου το NK είναι ο αριθμός του στοιχείου και K ο τοπικός αριθμός του κόμβου του στοιχείου NK (που

είναι από 1-9 για τετραγωνικά και 1-6 για τριγωνικά). Για τον αποτελεσματικότερο προγραμματισμό, τα τετραπλευρικά στοιχεία τα δηλώνουμε με τρόπο αυτοματοποιημένο (όπου αυτό είναι δυνατό) σε αντίθεση με τα τριγωνικά στοιχεία, τα οποία τα βάζουμε με «το χέρι».

Στο DO 1 δίδονται τα τριγωνικά στοιχεία στην είσοδο (δείκτης I από 1 ως NEX1A). LB είναι τα στοιχεία στο κάτω μέρος και LT τα στοιχεία στο άνω μέρος της εισόδου (σχήμα στο παράρτημα). Το K είναι ο κάτω αριστερά κόμβος του τετράπλευρου στοιχείου (NOP(NK,1)=K) και η αυτοματοποίηση έγκειται στον εντοπισμό του K για κάθε NK στοιχείο. Οι υπόλοιποι κόμβοι KK του στοιχείου NK είναι είτε επάνω από τον K (JJ=1, 3) είτε δίπλα στον K (II=1, 3). Στο DO 51 δίδονται τα τετράπλευρα στοιχεία στην είσοδο (δείκτης I από 1 ως NEX1A). Στο DO 52 η αυτοματοποίηση είναι η ίδια με το DO 51 και μάλιστα ακόμη πιο απλά λόγω της μη υπάρξεως των τριγωνικών στοιχείων. Μετά ορίζονται τα δύο τριγωνικά στοιχεία στο εσωτερικό του υπολογιστικού πεδίου (σχήμα στο παράρτημα). Στο DO 41 ορίζονται τα τετράπλευρα στοιχεία πάνω από το τριγωνικό στοιχείο LIB. Μετά ορίζεται το πρώτο τετράπλευρο στοιχείο στα δεξιά του LIB και στη συνέχεια στο DO 43 τα υπόλοιπα στοιχεία πάνω από αυτό. Στο DO 44 υπολογίζονται τα τετράπλευρα στοιχεία ανάντη του φράκτη που έχουν παράλληλη κατεύθυνση με το εμπόδιο. Στο DO 45 υπολογίζονται τα τετράπλευρα στοιχεία πάνω από τον φράκτη. Στο DO 46 ορίζονται τα τετράπλευρα στοιχεία κατάντη ου φράκτη με παράλληλη κατεύθυνση προς αυτό. Στο DO 47 ορίζονται τα τετράπλευρα στοιχεία κάτω από το τρίγωνο LIA. Στο DO 48 ορίζονται τα τετράπλευρα στοιχεία δίπλα και δεξιά από αυτά του DO 47 και στο DO 57 ορίζεται το τετράπλευρο στοιχείο δίπλα και δεξιά από το LIA. Στο DO 49 ορίζονται τα υπόλοιπα στοιχεία άνωθεν αυτού και στο DO 50 τα υπόλοιπα τετράπλευρα στοιχεία μέχρι την έξοδο.
Η επόμενη υπορουτίνα είναι η BNDRYC (boundary conditions).

```
**********************************************************************
SUBROUTINE BNDRYC
C
   PARAMETER (NXNP=173000,NL=17000,ND=65000)
C
   IMPLICIT REAL*8 (A-H,O-Z)
C
   COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
  *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
   COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
   COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
   COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
   COMMON /ESSE/  IESSL
C
   DO 1 I = 1, NP
   NCOD(I) = 0
 1  CONTINUE
C
C Essential b.c's for velocity at the entrance
C
   DO 2 J       = 1, NY+2*2*NEX1A
   L            = NODNOR(1,J)
   NCOD(NOPP(L))   = 1
   NCOD(NOPP(L)+1) = 1
 2  CONTINUE
C
C Essential b.c's for velocity along the top
C
   DO 3 I       = 1, 1+2*NEX1A
   LT           = NODNOR(I,NY+2*(2*NEX1A-(I-1)))
   NCOD(NOPP(LT)) = 1
```

```
      NCOD(NOPP(LT)+1) = 1
 3    CONTINUE
C
      DO 31 I       = 1+2*NEX1A+1, NX1
      LT            = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 31   CONTINUE
C
      LT            = NODNOR(NX1,NY) + NY-2
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
      LT            = NODNOR(NX1,NY)+NY-2 + NY-1
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
C
      DO 32 I       = NX1+1, NX2+2+2*NEX3
      LT            = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 32   CONTINUE
C
      DO 33 I       = NX3+1, NX
      LT            = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 33   CONTINUE
C
C Essential b.c's for velocity along the bottom
C
      DO 4 I        = 1, NX1
      LB            = NODNOR(I,1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
 4    CONTINUE
C
      DO 5 I        = NX1+1, NX2
      LB            = NODNOR(I,1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
 5    CONTINUE
C
      DO 6 J        = 1, NY1
      LH            = NODNOR(NX2,J)
      NCOD(NOPP(LH))   = 1
      NCOD(NOPP(LH)+1) = 1
 6    CONTINUE
C
      LB            = NODNOR(NX2+1,NY1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
C
      DO 7 J        = 1, NY1
      LH            = NODNOR(NX2+2,J)
      NCOD(NOPP(LH))   = 1
      NCOD(NOPP(LH)+1) = 1
 7    CONTINUE
C
      DO 9 I        = NX2+2, NX2+2+2*NEX3
      LB            = NODNOR(I,1)
```

```
      NCOD(NOPP(LB))  = 1
      NCOD(NOPP(LB)+1) = 1
    9 CONTINUE
C
      LB          = NODNOR(NX2+2+2*NEX3,NY) + 1
      NCOD(NOPP(LB))  = 1
      NCOD(NOPP(LB)+1) = 1
      LB          = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + 1
      NCOD(NOPP(LB))  = 1
      NCOD(NOPP(LB)+1) = 1
C
      DO 8 I      = NX3+1, NX
      LB          = NODNOR(I,1)
      NCOD(NOPP(LB))  = 1
      NCOD(NOPP(LB)+1) = 1
    8 CONTINUE
C
      RETURN
      END
**********************************************************************
```

Εκ των προτέρων γνωρίζουμε την τιμή ορισμένων μεταβλητών (ταχυτήτων, πιέσεων, θερμοκρασιών, συγκεντρώσεων) σε συγκεκριμένα σημεία του υπολογιστικού πεδίου.
Αυτές είναι οι οριακές συνθήκες μη-ολισθήσεως, όπως φαίνεται στο σχήμα 3.1. Πρέπει λοιπόν αυτές τις τιμές να τις επιβάλλουμε στη λύση του προγράμματος. Γι' αυτό ορισμένα Δu^k μηδενίζονται (στη λογική της μεθόδου Newton-Raphson) και αυτό γίνεται εφικτό με τις εντολές στην υπορουτίνα BNDRYC. Για να μηδενιστούν τα κατάλληλα Δu^k πρέπει αντίστοιχα να μηδενιστούν τα R^k και η γραμμή k της Ιακωβιανής J^{ik} να μηδενιστεί επίσης πλην του σημείου kk το οποίο πρέπει να είναι μη-μηδενικό, γιατί αλλιώς δεν αντιστρέφεται ο πίνακας J (βλέπε κεφάλαιο 5). Τα κατάλληλα σημεία k εντοπίζονται ως εξής:

- Πρώτα εντοπίζεται ο κατάλληλος κόμβος του υπολογιστικού πεδίου στον οποίο γνωρίζω την τιμή της μεταβλητής από την οριακή συνθήκη (Dirichlet).
- Αφού γίνει αυτό, πρέπει να εντοπισθεί το σημείο k στην στήλη Δu στον οποίο αντιστοιχεί ο κόμβος με την αναγκαία οριακή συνθήκη.

Επειδή σε κάθε κόμβο αντιστοιχούν περισσότερες από μία εξισώσεις συνήθως έχουμε:
Το MDF(L) έχει τον αριθμό των εξισώσεων σε κάθε κόμβο.
Το NOPP(L) έχει τον αριθμό της γραμμής στον πίνακα R^k που αντιστοιχεί στην πρώτη εξίσωση του κόμβου L.
Οπότε ορίζουμε ένα αρχείο NCOD(NP) το οποίο όταν παίρνει την τιμή 1 στην γραμμή k (που αντιστοιχεί στο $\Delta u^k=0$, $R^k=0$) δίνει εντολή στην SUBROUTINE FRONT να επιβάλλει $\Delta u^k=0$, $R^k=0$, $J^{ik}=0$ εκτός του $J^{kk}\neq 0$.
Οι συντεταγμένες του εδρασμένου εμποδίου, που υπάρχει μέσα στο υπολογιστικό πεδίο, καθώς επίσης και οι συντεταγμένες των υπολοίπων κόμβων, καθορίζεται στην υπορουτίνα MSHCOO.

```
**********************************************************************
SUBROUTINE MSHCOO
C
      PARAMETER (NXNP=173000,ND=65000,NG=501,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
```

```
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /PAR1/   ALPHA,DX,XL
C
      DIMENSION XHELP(NG)
C
      YT      =  YY1+YWGHT(NY)
      YB      = -YY1+YWGHT(1)
      XI      =  XWGHT(1)
      X0      =  XWGHT(1+2*NEX1A)
      NX1A    =  1+2*NEX1A
C
      DO 11 I = 1, 1+2*NEX1A
C
      DO 18 J = 1, NY
      L       = NODNOR(I,J+2*NEX1A-(I-1))
      X(L)    = XWGHT(I)
      Y(L)    = YWGHT(J)
   18 CONTINUE
C
      IF (I .LT. 1+2*NEX1A) THEN
      DO 17 J1 = 1,2*NEX1A-(I-1)
      LB      = NODNOR(I,J1)
      X(LB)   = XWGHT(I)
      Y(LB)   = (YB-YWGHT(1))*(XWGHT((I-1)+J1)-X0)/(X1-X0)+YWGHT(1)
      LT      = NODNOR(I,NY+2*NEX1A-(I-1)+J1)
      X(LT)   = XWGHT(I)
      Y(LT)   = (YT-YWGHT(NY))*(XWGHT(NX1A-J1)-X0)/(X1-X0)+YWGHT(NY)
   17 CONTINUE
      END IF
C
   11 CONTINUE
C
      DO 19 I = 1+2*NEX1A+1, NX1
      DO 19 J = 1, NY
      L       = NODNOR(I,J)
      X(L)    = XWGHT(I)
      Y(L)    = YWGHT(J)
   19 CONTINUE
C
      DO 21 J = 1, NY-1
      L       = NODNOR(NX1,NY)+NY-2 + J
      Y(L)    = YWGHT(J+1)
      IF (J .LE. NY1-1)  THEN
      X(L)    = XWGHT(NX1) + YWGHT(J+1)/TAN(ALPHA)
      ELSE
      X(L)    = XWGHT(NX1) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   21 CONTINUE
C
      DO 22 J = 3, NY
      L1      = NODNOR(NX1,J)
      L2      = NODNOR(NX1,NY)+NY-2 + J-1
      L       = NODNOR(NX1,NY) + J-2
      X(L)    = (X(L1)+X(L2))/2.
      Y(L)    = YWGHT(J)
   22 CONTINUE
C
      DO 23 I = NX1+1, NX2
```

```
      DO 23 J = 1, NY
      L     = NODNOR(I,J)
      Y(L)  = YWGHT(J)
      IF (J .LE. NY1) THEN
      X(L)  = XWGHT(I) + YWGHT(J)/TAN(ALPHA)
      ELSE
      X(L)  = XWGHT(I) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   23 CONTINUE
C
      DO 24 J = NY1, NY
      L     = NODNOR(NX2+1,J)
      Y(L)  = YWGHT(J)
      X(L)  = XWGHT(NX2+1) + YWGHT(NY1)/TAN(ALPHA)
   24 CONTINUE
C
      DO 25 I = NX2+2, NX2+2+2*NEX3
      DO 25 J = 1, NY
      L     = NODNOR(I,J)
      Y(L)  = YWGHT(J)
      IF (J .LE. NY1) THEN
      X(L)  = XWGHT(I) + YWGHT(J)/TAN(ALPHA)
      ELSE
      X(L)  = XWGHT(I) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   25 CONTINUE
C
      DO 26 J = 1, NY1-1
      L     = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      Y(L)  = YWGHT(J)
      X(L)  = XWGHT(NX2+2*NEX3+4)
   26 CONTINUE
C
      DO 27 J = 1, NY1-2
      L     = NODNOR(NX2+2+2*NEX3,NY) + J
      L1    = NODNOR(NX2+2+2*NEX3,J)
      L2    = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      Y(L)  = YWGHT(J)
      X(L)  = (X(L1)+X(L2))/2.
   27 CONTINUE
C
      DO 13 I = NX3+1, NX
      DO 13 J = 1, NY
      L     = NODNOR(I,J)
      X(L)  = XWGHT(I)
      Y(L)  = YWGHT(J)
   13 CONTINUE
C
      RETURN
      END
*****************************************************************
```

Στο DO 11 ορίζονται οι συντεταγμένες των κόμβων της εισόδου, στο δε DO 17 οι συντεταγμένες στο συγκλίνον μέρος της εισόδου. Στο DO 19 δίδονται οι συντεταγμένες ανάντη του φράκτη στο ορθογώνιο τμήμα του υπολογιστικού πεδίου. Στο DO 21 δίδονται οι συντεταγμένες της πρώτης γραμμής των κόμβων που είναι παράλληλες με το εμπόδιο. Στο DO 22 υπολογίζονται οι ενδιάμεσοι κόμβοι μεταξύ της τελευταίας γραμμής του ορθογώνιου τμήματος (L1) και της πρώτης γραμμής του παράλληλου τμήματος (L2). Στο DO 23 υπολογίζονται οι συντεταγμένες των κόμβων στο παράλληλο τμήμα ανάντη του φράκτη, στο DO 24 πάνω από τον φράκτη και στο DO 25 οι συντεταγμένες στο παράλληλο τμήμα

κατάντη του εμποδίου. Οι συντεταγμένες της πρώτης κάθετης γραμμής στο ορθογώνιο τμήμα κατάντη του φράκτη υπολογίζονται στο DO 26, ενώ στο DO 27 υπολογίζονται οι ενδιάμεσοι κόμβοι κατάντη. Τέλος στο DO 13 δίδονται οι συντεταγμένες στο ορθογώνιο τμήμα της εξόδου.

Στη συνέχεια ακολουθεί η υπορουτίνα INIGES (initial guess), όπου γίνεται η αρχική πρόβλεψη για την επαναληπτική μέθοδο Newton-Raphson.

```
**********************************************************************
SUBROUTINE INIGES
C
    PARAMETER (ND=65000,NG=501,NL=17000)
C
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
    COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
   *            NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
    COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
    COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
    COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
    COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
    COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
    COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
    COMMON /MESOLD/ NX1OLD, NXXOLD, NXOLD, NY1OLD, NYOLD, NHOLD
C
DO 7 I = 1, NH
    V(I)   = 0.
    U(I)   = 0.
    P(I)   = 0.
  7 CONTINUE
C
    DO 8 I = 1, 2*NEX1A
    DO 8 J = 1, NY+2*2*NEX1A-2*(I-1)
    L      = NODNOR(I,J)
    IF (J .GT. 2*NEX1A-(I-1) .AND. J .LE. NY+2*NEX1A-(I-1)) THEN
    U(L)   = 1.
    ELSE
    U(L)   = 0.
    END IF
  8 CONTINUE
C
    XL     = YWGHT(NY)
C
C Parabolic velocity profile upstream the obstacle
C
    DO 9 I = 1+2*NEX1A, NX1
    DO 9 J = 1, NY
    L      = NODNOR(I,J)
    A      = -6./(XL**2.)
    B      = 6./XL
    U(L)   = A*(Y(L)**2.) + B*Y(L)
    IF (J .EQ. 1  .OR. J .EQ. NY)
   *U(L)   = 0.
  9 CONTINUE
C
C Parabolic velocity profile above the obstacle
C
    DO 21 I = NX1+1, NX2+2+2*NEX3
    DO 21 J = NY1, NY
    A1     = Y(NODNOR(I,NY1))
```

```
      L     = NODNOR(I,J)
      A     = -6.*XL/((XL-A1)**3)
      B     = 6.*XL*(XL+A1)/((XL-A1)**3)
      C     = -6.*A1*XL*XL/((XL-A1)**3)
      U(L)  = A*Y(L)**2 + B*Y(L) + C
      IF (J .EQ. NY1  .OR.  J .EQ. NY)
     *U(L)  = 0.
   21 CONTINUE
C
C Parabolic velocity profile downstream the obstacle
C
      DO 11 I = NX3+1, NX
      DO 11 J = 1, NY
      L     = NODNOR(I,J)
      A     = -6./(XL**2)
      B     = 6./XL
      U(L)  = A*Y(L)**2 + B*Y(L)
      IF (J .EQ. 1  .OR.  J .EQ. NY)
     *U(L)  = 0.
   11 CONTINUE
C
      RETURN
      END
*****************************************************************
```

Πριν την ανάλυση της υπορουτίνας θα πρέπει να αναφέρουμε τον προβληματισμό βάση του οποίου έγινε αυτή.

- Αρχικά πρέπει το πρόγραμμα να «καταλάβει» ότι εισέρχεται μάζα στο πεδίο και αυτό επιτυγχάνεται ορίζοντας την ταχύτητα και κατά συνέπεια την παροχή (μάζα) στην είσοδο του υπολογιστικού πεδίου.
- Και τέλος ότι η αρχική πρόβλεψη πρέπει να είναι «κοντά» στη λύση.

Δεν υπάρχει κανένας κανόνας που να μας λέει ποια είναι η πιο ενδεδειγμένη αρχική πρόβλεψη για μια ροή πέραν του ότι αυτή πρέπει να ικανοποιεί την αρχή διατήρησης της μάζας. Υπάρχουν όμως τόσο σύνθετα προβλήματα ροής, όπως αυτά με ελεύθερες επιφάνειες (κυματοειδής ροές), όπου το πρόβλημα της αρχικής προβλέψεως είναι θεμελιώδες, διότι ακόμη και για μικρούς αριθμούς Re η ροή είναι άκρως μη-γραμμική (7). Ζήτημα αρχικής πρόβλεψης τίθεται σε μη-γραμμικά προβλήματα διότι λύνονται σε περισσότερες από μία επαναλήψεις. Στα γραμμικά προβλήματα σύγκλιση έχουμε στην πρώτη επανάληψη, συνεπώς δεν τίθεται τέτοιο πρόβλημα σε μη γραμμικά προβλήματα. Άρα λοιπόν, η πιο πετυχημένη «συνταγή» (κανόνας) για αρχική πρόβλεψη είναι η λύση της γραμμικής, ή οιονεί γραμμικής, περίπτωσης του προβλήματος που μελετούμε δηλ της έρπουσας ροής (ροή Stokes).

Στη συνέχεια θα εξηγήσουμε τον τρόπο με τον οποίο επιλέγουμε τις κατανομές των ταχυτήτων στην είσοδο. Στην είσοδο θέλουμε η ταχύτητα να προσομοιάζει τις συνθήκες του εργαστηριακού πειράματος, δηλαδή τις συνθήκες της αδιατάραχτης ροής. Η κατανομή της ταχύτητας στην είσοδο έχει για μοναδικό κριτήριο (από την μέχρι τώρα εμπειρία) το ποσό της μάζας που εισέρχεται και προκαθορίζεται από τον μηχανικό. Για λόγους λοιπόν ακριβείας θα πρέπει να επιλέγουμε έτσι την κατανομή ώστε να είμαστε σίγουροι για το ποσό της μάζας που εισάγουμε στο υπολογιστικό πεδίο. Γι' αυτό οι κατανομές είναι ή παραβολές ή ομοιόμορφες, διότι γνωρίζουμε ακριβώς, όταν ολοκληρώνουμε, το ποσό της μάζας. Γι αυτό στο DO 8 επιβάλλουμε στην είσοδο ομοιόμορφη κατανομή της ταχύτητας. Στο DO 9 επιβάλλουμε παραβολική ταχύτητα (ροή Poisseuille) ανάντη του εμποδίου, ενώ στο DO 21 επιβάλουμε το προφίλ της ταχύτητας πάνω από το εμπόδιο. Και τέλος στο DO 11 έχουμε το παραβολικό προφίλ της ταχύτητας κατάντη του εμποδίου.

Συνοψίζοντας λέμε: Σε ροές μέσα σε αγωγούς, η κατανομή της ταχύτητας στην είσοδο, δεν επηρεάζει τα φαινόμενα της ροής κατάντη, απ' όσο γνωρίζουμε. Στο κύριο τμήμα της

αεροσήραγγας επιλέγουμε παραβολική κατανομή της ταχύτητας. Προσοχή: Επειδή πρέπει να ικανοποιείται η *Αρχή Διατήρησης της Μάζας*, η παροχή που υπολογίζεται στις περιοχές με παραβολική κατανομή της ταχύτητας είναι ίδια μ' αυτήν που επιβάλλεται στην είσοδο με την ομοιόμορφη κατανομή της ταχύτητας.

Αυτό που μας ενδιαφέρει στη συνέχεια είναι η σύγκλιση των αποτελεσμάτων, με την επαναληπτική μέθοδο Newton- Raphson, που λαμβάνει χώρα στην υπορουτίνα ITRATE.

```
**********************************************************************
SUBROUTINE ITRATE
C
    PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
    COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
    COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
    COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
    COMMON /ER/     ERR
    COMMON /IITT/   ICOUNT
    COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
    COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
    COMMON /ITIT/   ITER
C
    NTRA  = 1
    ITER  = 0
C
    IF (ICOUNT .EQ. 1)  NTRA=0
C

    CALL FRONT
C
    NTRA  = 0
    ITER  = ITER+1
C
    CALL UPDATE
    WRITE(*,'(A13,I5,1PE11.4)')  'ITER, ERR : ', ITER, ERR
C
    IF (ERR .GT. ERRMAX) THEN
    GO TO 50
    ELSE IF (ERR .LE. ERRMAX) THEN
    WRITE(*,*)  'CONVERGENCE'
    GO TO 200
    END IF
C
  50 IF(ITER .LT. MAXITE) THEN
    GO TO 2
    ELSE
    WRITE(*,*) 'MAXIMUM ITERATION NUMBER'
    STOP
    END IF
C
  200 CONTINUE
C
    RETURN
    END
**********************************************************************
```

Μετά καλείται η υπορουτίνα UPDATE όπου ελέγχεται το μέγιστο λάθος Δu^k σε κάθε κόμβο του υπολογιστικού πεδίου και για κάθε άγνωστο ξεχωριστά. Αν αυτό το λάθος ERR

είναι μικρότερο από ένα ελάχιστο επιτρεπτό λάθος ERRMAX, επιτυγχάνεται σύγκλιση του αλγορίθμου. Αλλιώς οι επαναλήψεις συνεχίζονται (ITER=ITER+1) μέχρις ότου επιτευχθεί σύγκλιση ή να υπερβεί ένας μέγιστος αριθμός επαναλήψεων (MAXITE). Πρέπει μετά από κάθε επανάληψη να παρακολουθείται το λάθος (WRITE(*,'(A13,I5,1PE11.4)') 'ITER, ERR : ', ITER, ERR) ώστε να ελέγχεται η τετραγωνική σύγκλιση της επαναληπτικής μεθόδου Newton – Raphson. Έτσι υπάρχει ένας επιπλέον έλεγχος για την ορθότητα του προγράμματος.

```
**********************************************************************
SUBROUTINE UPDATE
C
   PARAMETER (NXNP=173000,ND=65000,NL=17000,NCRIT1=939)
C
   IMPLICIT REAL*8 (A-H,O-Z)
C
   COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
   COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
   COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
   COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
  *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
   COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
   COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
   COMMON /FRON3/  SK(NXNP),AA(25,25)
   COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
   COMMON /ER/    ERR
   COMMON /PAR/   RE,YY1,PE1,FR1,UO,YD,PL
   COMMON /ITIT/  ITER
C
   ERR    = 0.
   DO 50 I = 1, NH
   CAN    = SK(NOPP(I))
   CAN2   = CAN*CAN
   U(I)   = U(I)+CAN
   IF (CAN2 .GT. ERR) ERR = CAN2
   CAN    = SK(NOPP(I)+1)
   CAN2   = CAN*CAN
   V(I)   = V(I)+CAN
   IF (CAN2 .GT. ERR) ERR = CAN2
   IF (MDF(I) .EQ. 3) THEN
   CAN    = SK(NOPP(I)+2)
   CAN2   = CAN*CAN
   IF (CAN2 .GT. ERR) ERR = CAN2
   P(I)   = P(I)+CAN
   END IF
  50 CONTINUE
   ERR    = SQRT(ERR)
C
   RETURN
   END
**********************************************************************
```

Ο πίνακας SK(NOPP(I)) έχει τη λύση $-\mathbf{J}^{ik}\mathbf{R}^{k}$ όπως υπολογίζεται από την FRONT μετά από κάθε επανάληψη. Η νέα τιμή των κομβικών αγνώστων U(I), V(I) και P(I) υπολογίζεται σύμφωνα με την σχέση $\Delta\mathbf{u} - \mathbf{J}^{ik}\mathbf{R}^{k}$ (κεφάλαιο 5) που στην γλώσσα προγραμματισμού FORTRAN μεταφράζεται ως U(I)=U(I)+CAN (CAN=SK(NOPP(I)) κ.τ.λ. Προσοχή στον συνδυασμό του δείκτη I στους πίνακες U(I), V(I) και P(I) με τον δείκτη (I) στον πίνακα NOPP(I). Ο πίνακας SK έχει διάσταση NP και NP=NOPP(NH)+2, όπου NH ο αριθμός των κόμβων. Δηλαδή οι πίνακες NOPP(I), MDF(I), NOPP(NK,KK) παρέχουν αποτελεσματική βοήθεια στην εκτέλεση του προγράμματος.

Στη συνέχεια θα αναφέρουμε την υπορουτίνα ABFIND που είναι η «καρδιά» του προγράμματος.

```
**********************************************************************
SUBROUTINE ABFIND
C
    PARAMETER (NXNP=173000,ND=65000,NL=17000,NCRIT1=939)
C
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /JACOB/  XC,XEDA,YC,YEDA,Y1,AY1,CX,CY,EX,EY
    COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
    COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
   *            NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
    COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
    COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
    COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
    COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
    COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
    COMMON /FRON3/  SK(NXNP),AA(25,25)
    COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
    COMMON /OLDVAR/ UOLD(ND),VOLD(ND),POLD(ND)
    COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
    COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
    COMMON /NODCOO/ XE(9),YE(9)
    COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD.PL
    COMMON /PRES/   NNPRES(4),IL
C
    DIMENSION R(22,22)
    DIMENSION IU(9),IV(9),IP(4),IUQ(9),IVQ(9),IPQ(4),IUT(6),IVT(6),
   *        IPT(3),NNPRQ(4),NNPRT(3)
C
    DATA IUT   /1,4,7,10,12,14/
    DATA IVT   /2,5,8,11,13,15/
    DATA IPT   /3,6,9/
    DATA IUQ   /1,4,6,9,11,13,15,18,20/
    DATA IVQ   /2,5,7,10,12,14,16,19,21/
    DATA IPQ   /3,8,17,22/
    DATA NNPRT /1,2,3/
    DATA NNPRQ /1,3,7,9/
C
C Distinguish between triangular and quadrilateral elements
C
    IF (NET(NELL) .EQ. 1) THEN
    NGP      = 7
    IQ       = 6
    IL       = 3
    DO 391 I  = 1, IQ
    IU(I)     = IUT(I)
    IV(I)     = IVT(I)
 391  CONTINUE
    DO 393 I  = 1, IL
    IP(I)     = IPT(I)
 393  CONTINUE
    DO 395 I  = 1, IL
    NNPRES(I) = NNPRT(I)
 395  CONTINUE
    ELSE
    NGP      - 9
    IQ       = 9
    IL       = 4
```

```
      DO 392 I  = 1, IQ
      IU(I)     = IUQ(I)
      IV(I)     = IVQ(I)
  392 CONTINUE
      DO 394 I  = 1, IL
      IP(I)     = IPQ(I)
  394 CONTINUE
      DO 396 I  = 1, IL
      NNPRES(I) = NNPRQ(I)
  396 CONTINUE
      END IF
C
C Initialize Jacobian ,R, and remember residual ,R1, is initialized in FRONT
C
      DO 101 I = 1, 22
      DO 101 J = 1, 22
      R(I,J)  = 0.
  101 CONTINUE
C
C Give global coordinates to the master element for isoparametric mapping
C
      DO 102 J = 1, IQ
      LN      = IABS(NOP(NELL,J))
      XE(J)   = X(LN)
      YE(J)   = Y(LN)
  102 CONTINUE
C
C Assembly of R and R1
C
      DO 103 K  = 1, NGP
      CALL GAUSCT
      C       = GPX(K)
      E       = GPY(K)
      CALL TFUNC(C,E)
      CALL TPRES(C,E)
      CALL DER(C,E)
      UEVX      = FEVX(U,NELL)
      UEVY      = FEVY(U,NELL)
      VEVX      = FEVX(V,NELL)
      VEVY      = FEVY(V,NELL)
      UEV       = FEV(U,NELL)
      VEV       = FEV(V,NELL)
      PEV       = FEVP(P,NELL)
      WE        = GWEI(K)*Y1
C
C Compute the equations of momentum conversation
C
      DO 104 KI  = 1, IQ
      IX        = NOPP(IABS(NOP(NELL,KI)))
      IY        = IX+1
      R1(IX)    = R1(IX) - WE*((UEV*UEVX+VEV*UEVY)*PHI(KI)        +
     *                 (-PEV+(2./RE)*UEVX)*PHIX(KI)       +
     *                 (1./RE)*(UEVY+VEVX)*PHIY(KI))
C
      R1(IY)    = R1(IY)  - WE*((UEV*VEVX+VEV*VEVY)*PHI(KI)       +
     *                  (-PEV+(2./RE)*VEVY)*PHIY(KI)      +
     *                  (1./RE)*(UEVY+VEVX)*PHIX(KI))
C
      DO 105 KJ  = 1, IQ
      KKI       = IV(KI)
```

```
      KKJ      = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*(PHI(KJ)*VEVX*PHI(KI)          +
     *                    (1./RE)*PHIY(KJ)*PHIX(KI))
      KKJ      = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*((UEV*PHIX(KJ)+VEV*PHIY(KJ)     +
     *                     VEVY*PHI(KJ))*PHI(KI)         +
     *                     (2./RE)*PHIY(KJ)*PHIY(KI)     +
     *                     (1./RE)*PHIX(KJ)*PHIX(KI))
C
      KKI      = IU(KI)
      KKJ      = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*((VEV*PHIY(KJ)+UEV*PHIX(KJ)     +
     *                     UEVX*PHI(KJ))*PHI(KI)         +
     *                     (2./RE)*PHIX(KJ)*PHIX(KI)     +
     *                     (1./RE)*PHIY(KJ)*PHIY(KI))
       KKJ      = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*(PHI(KJ)*UEVY*PHI(KI)          +
     *                    (1./RE)*PHIX(KJ)*PHIY(KI))
 105  CONTINUE
C                                                  C
      DO 106 KJ  = 1, IL
      KKI      = IU(KI)
      KKJ      = IP(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WE*PHIP(KJ)*PHIX(KI)
C
      KKI      = IV(KI)
      R(KKI,KKJ) = R(KKI,KKJ) - WE*PHIY(KI)*PHIP(KJ)
 106  CONTINUE
C
 104  CONTINUE
C
C Compute the equation of mass conversation
C
      DO 107 KI  = 1, IL
      IC       = NOPP(IABS(NOP(NELL,NNPRES(KI)))) + 2
      R1(IC)   = R1(IC) - WE*PHIP(KI)*(UEVX+VEVY)
      DO 107 KJ  = 1, IQ
      KKI      = IP(KI)
      KKJ      = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*PHIP(KI)*PHIX(KJ)
      KKJ      = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*PHIP(KI)*PHIY(KJ)
  107 CONTINUE
C
  103 CONTINUE
C
C Free boundary condition at the outflow
C
      IF (NEO(NELL) .EQ. 1) THEN
C
      DO 108 K   = 1, 3
      CALL GAUSCT
      C        = 1.
      E        = GPT(K)
      CALL TFUNC(C,E)
      CALL TPRES(C,E)
      CALL DER(C,E)
      UEVX     = FEVX(U,NELL)
      UEVY     = FEVY(U,NELL)
      VEVX     = FEVX(V,NELL)
```

```
      PEV     = FEVP(P,NELL)
      WT      = GWE(K)*YEDA
C                                                C
      DO 109 KI = 7, 9
      IX      = NOPP(IABS(NOP(NELL,KI)))
      IY      = IX+1
      R1(IX)   = R1(IX) + WT*(-PEV+(2./RE)*UEVX)*PHI(KI)
      R1(IY)   = R1(IY) + WT*(1./RE)*(UEVY+VEVX)*PHI(KI)
C
      DO 110 KJ = 1, IQ
      KKI     = IU(KI)
      KKJ     = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(2./RE)*PHIX(KJ)*PHI(KI)
C
      KKI     = IV(KI)
      KKJ     = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(1./RE)*PHIX(KJ)*PHI(KI)
      KKJ     = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(1./RE)*PHIY(KJ)*PHI(KI)
C
 110  CONTINUE
C
      DO 111 KJ = 1, IL
      KKI     = IU(KI)
      KKJ     = IP(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WT*PHIP(KJ)*PHI(KI)
 111  CONTINUE
C
 109  CONTINUE
C
 108  CONTINUE
C
      END IF
C
C Transfer the Jacobian to the Frontal solver
C
      DO 112 I  = 1, 22
      DO 112 J  = 1, 22
      AA(J,I)   = R(J,I)
 112  CONTINUE
C
      RETURN
      END
**********************************************************************
```

Η υπορουτίνα αυτή είναι η «καρδιά» του προγράμματος, διότι εδώ ορίζονται οι διέπουσες εξισώσεις του προβλήματος, οι φυσικές οριακές συνθήκες, η κατασκευή της Ιακωβιανής και των υπολοίπων. Δηλαδή, όλες οι άλλες υπορουτίνες του προγράμματος που παρουσιάστηκαν μέχρι τώρα, έχουν δορυφορικό χαρακτήρα σε σχέση με την ABFIND, και αποσκοπούν αποκλειστικώς στην απλούστευση του προγραμματισμού. Θα περίμενε κανείς η ABFIND να «καλείται» σε κεντρικό σημείο του προγράμματος (π.χ. PROGRAM ABC). Επειδή όμως οι εξισώσεις στην μέθοδο των Πεπερασμένων Στοιχείων μαζεύονται στοιχείο προς στοιχείο, θεωρήθηκε σκόπιμο από την εμπειρία η ABFIND να καλείται από την FRONT, όπου αμέσως μετά το μάζεμα των εξισώσεων κάθε στοιχείου να προβαίνει η τμηματική αντιστροφή της Ιακωβιανής και η επιτάχυνση της λύσεως του αλγεβρικού συστήματος $-\mathbf{J}^{ih}\mathbf{R}^{h}$.

Αναλυτικότερα η δομή και οι επιμέρους εντολές της ABFIND έχουν ως εξής:

- Στο πρώτο IF-block δίδεται ο αριθμός των σημείων **Gauss** (NGP) για την αριθμητική ολοκλήρωση των υπολοίπων καθώς και ο αριθμός των κόμβων του στοιχείου (IQ) και ο

αριθμός των κόμβων του στοιχείου όπου ορίζονται οι γραμμικές βάσεις (IL) για την πίεση. Επειδή το πρόγραμμα έχει και τριγωνικά και τετραπλευρικά στοιχεία πρέπει να γίνει διάκριση για το κάθε στοιχείο ξεχωριστά (NET(NELL)) όπου NELL είναι το τρέχον στοιχείο του υπολογιστικού πεδίου του οποίου μαζεύονται οι εξισώσεις.

- Στο DO 102 δίδονται οι συντεταγμένες XE(J) και YE(J) των κόμβων του κάθε στοιχείου. Αυτές οι εντολές μας χρησιμεύουν στην ισοπαραμετρική απεικόνιση οποιουδήποτε στοιχείου στο πρότυπο στοιχείο όπως αναφέρθηκε στο κεφάλαιο (5). Ο πίνακας NOP(NELL,J) είναι σε απόλυτη τιμή, διότι στην FRONT αλλάζει πρόσημο.
- Στο DO 103 μαζεύονται οι διέπουσες εξισώσεις για την αρχή διατήρησης της ορμής. Πρώτα καλείται η υπορουτίνα GAUSCT όπου είναι αποθηκευμένα τα σημεία GauB για την αριθμητική ολοκλήρωση καθώς και τα αντίστοιχα βάρη. C και E είναι τα σημεία ξ και η του πρότυπου στοιχείου (κεφάλαιο 5). Μετά καλούνται οι υπορουτίνες TFUNC, TPRES και DER όπου υπολογίζονται οι τετραγωνικές (TFUNC) και γραμμικές (TPRES) συναρτήσεις βάσεως καθώς και η ορίζουσα (DER) της ισοπαραμετρικής απεικονίσεως για κάθε σημείο ξ,η Gauss. Μετά υπολογίζονται οι όροι $\frac{\partial u}{\partial x}$**(UEVX)**, $\frac{\partial u}{\partial y}$**(UEVY)**, $\frac{\partial v}{\partial x}$**(VEVX)**, $\frac{\partial v}{\partial y}$**(VEVY)**, **u(UEV)**, **v(VEV)**, και **u(PEV)** όπως ορίζονται στο κεφάλαιο (5) ως ανάπτυγμα των κομβικών αγνώστων σε κάθε στοιχείο ζυγισμένο με την κατάλληλη συνάρτηση βάσεως. WE είναι το γινόμενο βάρους GWEI(K) και της Ιακωβιανής Y1 της ισοπαραμετρικής απεικονίσεως και έτσι ολοκληρώνεται ο ορισμός όλων των μεταβλητών και παραμέτρων και είμαστε έτοιμοι για τον υπολογισμό των υπολοίπων που γίνεται στο DO 104. IX είναι το σημείο του πίνακα που ορίζει την πρώτη εξίσωση σε κάθε κόμβο και αντιστοιχεί στην επίλυση του κομβικού αγνώστου u^k όπου k=NOP(NELL,KI). IY=IX+1 αντιστοιχεί στον κομβικό άγνωστο v^k. Το αρνητικό πρόσημο είναι για να είμαστε συμβατοί με τον ορισμό $\Delta \mathbf{u}^k = -\mathbf{J}^{\mathbf{th}}\mathbf{R}'$. Για την σημασία του κάθε όρου στα υπόλοιπα R1(IX), R1(IY) βλέπε κεφάλαιο (5).
- Στο DO 105 δίδεται η Ιακωβιανή R(KKI,KKJ). Οι δείκτες KKI και KKJ ορίζονται για την διάκριση των μερικών παραγώγων. Για την μερική παράγωγο $\frac{\partial \mathbf{Ru}^i}{\partial \mathbf{u}^k}$ ο δείκτης KKI διακρίνει τον όρο $\mathbf{Ru}^i(\mathbf{KKI} = \mathbf{IU(KI)})$ και ο δείκτης KKJ διακρίνει τον όρο $\mathbf{u}^k(\mathbf{KKJ} = \mathbf{IU(KJ)})$. Δηλαδή για την μερική παράγωγο $\frac{\partial \mathbf{Rv}^i}{\partial \mathbf{u}^k}$ οι δείκτες θα είναι: $\mathbf{KKI} = \mathbf{IV(KI)}$ και $\mathbf{KKJ} = \mathbf{IU(KJ)}$.
- Στο DO 107 υπολογίζεται η εξίσωση της *αρχής διατήρησης της μάζας*. Εδώ φαίνεται η αποτελεσματικότητα του ορισμού του πίνακα NNPRES(KI). Ενώ ο δείκτης KI αλλάζει από 1 έως 4 ο NNPRES(KI) γίνεται 1, 3, 7 και 9 διότι σ' αυτούς τους κόμβους ορίζεται η πίεση ως κομβικός άγνωστος. Ο δείκτης IC ορίζει την θέση της πιέσεως στον ολικό πίνακα των υπολοίπων προσθέτοντας τον αριθμό 2 σε κάθε πρώτη εξίσωση του αντίστοιχου κόμβου όπου ορίζεται η πίεση. Οι μερικές παράγωγοι $\frac{\partial \mathbf{Rc}^i}{\partial \mathbf{u}^k}$ και $\frac{\partial \mathbf{Rc}^i}{\partial \mathbf{v}^k}$ δίδονται με τους δείκτες (KKI=IP(KI), KKJ=IU(KJ)) και (KKI=IP(KI), KKJ=IV(KJ)) αντιστοίχως.
- Μετά ακολουθεί η επιβολή της ελεύθερης οριακής συνθήκης της οποίας ο προγραμματισμός έχει την ίδια στρατηγική με προηγουμένως και δεν πρόκειται να εξηγηθεί λεπτομερέστερα.

➢ Και τέλος ακολουθούν δύο υπορουτίνες οι: SVSOL και POSTPR. Στην μεν πρώτη υπορουτίνα λέμε στο πρόγραμμα να σώσει την λύση, ενώ στην δεύτερη έχουμε την διαδικασία αποθήκευσης των γραμμών ροής και της διατμητικής τάσης.

```
**********************************************************************
SUBROUTINE SVSOL
C
   PARAMETER (ND=65000,NL=17000)
C
   IMPLICIT REAL*8 (A-H,O-Z)
C
   COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
   COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
   COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
   COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
   COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
  *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
   COMMON /PAR/   RE,YY1,PE1,FR1,UO,YD,PL
C
   REWIND NCO
C
   IF (ICONT .LE. 1)  WRITE(NCO,101) RE

C
   DO 103 I = 1, NH
   WRITE(NCO,104) I, X(I), Y(I), U(I), V(I), P(I)
 103  CONTINUE
C
101  FORMAT ('RE : ', F12.5)
 104  FORMAT (I5,4F12.6,1PE19.10)
C
 999  RETURN
   END
**********************************************************************
```

Και

```
SUBROUTINE POSTPR
C
    IMPLICIT REAL*8 (A-H,O-Z)
C
    PARAMETER (NXNP=173000,NG=501,ND=65000,NL=17000)
C
    COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
    COMMON /JACOB/  XC,XEDA,YC,YEDA,Y1,AY1,CX,CY,EX,EY
    COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
    *          NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
    COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
    COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
    COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
    COMMON                                                    /FRON1/
NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
    COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
    COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
    COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
    COMMON /NODCOO/ XE(9),YE(9)
    COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
    COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
```

```
C
C Compute streamlines for tecplot!
C
C At the entrance!
C
      IF (ICONT .GE. 2)
     *WRITE(NPP,393) TIME, NEY+1, NEX1A+1
      IF (ICONT .LE. 1)
     *WRITE(NPP,391) RE,   NEY+1, NEX1A+1
      DO 46 I = 1, 2*NEX1A+1, 2
      DO 46 J = 1+2*NEX1A-(I-1), NY+2*NEX1A-(I-1), 2
      LL     = NODNOR(I,J)
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
 46   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEX1A+1, NEX1A+1
      DO 47 I = 1, 1+2*NEX1A, 2
      DO 47 J = 1, 2*NEX1A+1, 2
      JJ     = 1+2*NEX1A-(I-1)-(J-1)
      IF (JJ .LE. 1  .AND.  I .GT. 1)  THEN
      L1     = NODNOR(I,1)
      WRITE(NPP,'(4F12.6)')  X(L1), Y(L1), U(L1), V(L1)
      ELSE
      LL     = NODNOR(I,JJ)
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
      END IF
 47   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEX1A+1, NEX1A+1
      DO 48 I = 1, 2*NEX1A+1, 2
      DO 48 J = 1, 2*NEX1A+1, 2
      IF (J .GT. 2*NEX1A-(I-1))  THEN
      L1     = NODNOR(I,NY+2*(2*NEX1A-(I-1)))
      WRITE(NPP,'(4F12.6)')  X(L1), Y(L1), U(L1), V(L1)
      ELSE
      LL     = NODNOR(I,J-1+NY+2*NEX1A-(I-1))
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
      END IF
 48   CONTINUE
C
C Upstream the obstacle!
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX1B+1
      DO 49 I = 1+2*NEX1A, NX1, 2
      DO 49 J = 1, NY, 2
      LL     = NODNOR(I,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
```

```
 49   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX2+2
      DO 59 J = 1, NY, 2
      LL     = NODNOR(NX1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 59   CONTINUE
      DO 61 I = 1, 1+2*NEX2, 2
      IF (I .EQ. 1)  THEN
      DO 62 J = I, NY, 2
      IF (J .EQ. 1)  THEN
      LL     = NODNOR(NX1,1)
      ELSE
      LL     = NODNOR(NX1,NY)+NY-2 + J-1
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 62   CONTINUE
      ELSE
      DO 63 J = 1, NY, 2
      LL     = NODNOR(NX1+I-1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 63   CONTINUE
      END IF
 61   CONTINUE
C
C Over the obstacle!
C
      NNN    = 2
      WRITE(NPP,*)
      WRITE(NPP,392) NEY2+1, NNN
      DO 50 I = 1, NNN
      IF (I .EQ. 1)  THEN
      DO 51 J = NY1, NY, 2
      LL     = NODNOR(NX2,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 51   CONTINUE
      ELSE
      DO 52 J = NY1, NY, 2
      LL     = NODNOR(NX2+2,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 52   CONTINUE
      END IF
 50   CONTINUE
C
C Downstream the obstacle!
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX3+2
      DO 53 I = 1, 1+2*NEX3, 2
```

```
      DO 53 J = 1, NY, 2
      LL     = NODNOR(NX2+1+I,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 53   CONTINUE
      DO 57 J = 1, NY, 2
      IF (J .LE. NY1-2)  THEN
      LL     = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      ELSE
      LL     = NODNOR(NX2+2+2*NEX3,J)
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 57   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX4+1
      DO 7I I = 1, I+2*NEX4, 2
      IF (I .EQ. 1)  THEN
      DO 72 J = 1, NY, 2
      IF (J .LE. NY1-2)  THEN
      LL     = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      ELSE
      LL     = NODNOR(NX2+2+2*NEX3,J)
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 72   CONTINUE
      ELSE
      DO 73 J = 1, NY, 2
      LL     = NODNOR(NX3+I-1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 73   CONTINUE
      END IF
 71   CONTINUE
C
      CLOSE(UNIT=NPP,STATUS='KEEP')
C
 392  FORMAT('ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
 391  FORMAT('TITLE = "Re=', F9.5, '", VARIABLES="XE","YE","UEV","VEV",'
     * / 'ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
 393  FORMAT('TITLE = "t =', F9.2, '", VARIABLES="XE","YE","UEV","VEV",'
     * / 'ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
C
      IQ       = 9
      EEE      = 0.
C
      WRITE(NLP,999)  RE, NEY1+NEY2+1, NEX2+NEX3+2
      DO 102 NELL = 1, NEY
      NEL1     = NELL + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 103 KI  = 1, 9
      LN       = IABS(NOP(NEL1,KI))
      XE(KI)   = X(LN)
```

```
      YE(KI)    = Y(LN)
  103 CONTINUE
      PSIE      = 0.
      IF (NELL .EQ. 1) THEN
      C         = -1.
      E         = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NEL1,EEE)
      UEVY      = FEVY(U,NEL1,EEE)
      ZETA1     = VEVX-UEVY
      LN1       = IABS(NOP(NEL1,1))
      END IF
      C         = -1.
      E         = 1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NEL1,EEE)
      UEVY      = FEVY(U,NEL1,EEE)
      ZETA      = VEVX-UEVY
      LN        = IABS(NOP(NEL1,3))
      IF (NELL .EQ. 1)
     *WRITE(NLP,901) XE(1),YWGHT(1),ZETA1,P(LN1)
      WRITE(NLP,901) XE(3),YE(3),ZETA,P(LN)
  102 CONTINUE
C
      DO 101 I    = 1, NEX2
      DO 101 J    = 1, NEY
      NELL      = (I-1)*NEY+J + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 106 KI   = 1, 9
      LN        = IABS(NOP(NELL,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  106 CONTINUE
      IF (J .EQ. 1) THEN
      C         = 1.
      E         = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NELL,EEE)
      UEVY      = FEVY(U,NELL,EEE)
      ZETA1     = VEVX-UEVY
      END IF
      C         = 1.
      E         = 1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NELL,EEE)
      UEVY      = FEVY(U,NELL,EEE)
      ZETA      = VEVX-UEVY
```

```
      L1        = IABS(NOP(NELL,7))
      L         = IABS(NOP(NELL,9))
      IF (J .EQ. 1)
     *WRITE(NLP,901) XE(7),YWGHT(1),ZETA1,P(L1)
      WRITE(NLP,901) XE(9),YE(9),ZETA,P(L)
  101 CONTINUE
C
      DO 302 NELL = 1, NEY
      NEL1      = NELL + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *            + NEY2+NEX2*NEY
      DO 303 KI   = 1, 9
      LN        = IABS(NOP(NEL1,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  303 CONTINUE
      PSIE      = 0.
      IF (NELL .EQ. 1) THEN
      C         = -1.
      E         = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NEL1,EEE)
      UEVY      = FEVY(U,NEL1,EEE)
      ZETA1     = VEVX-UEVY
      LN1       = IABS(NOP(NEL1,1))
      END IF
      C         = -1.
      E         =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX      = FEVX(V,NEL1,EEE)
      UEVY      = FEVY(U,NEL1,EEE)
      ZETA      = VEVX-UEVY
      LN        = IABS(NOP(NEL1,3))
      IF (NELL .EQ. 1)
     *WRITE(NLP,901) XE(1),YWGHT(1),ZETA1,P(LN1)
      WRITE(NLP,901) XE(3),YE(3),ZETA,P(LN)
  302 CONTINUE
C
      DO 301 I    = 1, NEX3
      DO 301 J    = 1, NEY
      NELL      = (I-1)*NEY+J + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                + NEY2+NEX2*NEY
      DO 306 KI   = 1, 9
      LN        = IABS(NOP(NELL,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  306 CONTINUE
      IF (J .EQ. 1) THEN
      C         =  1.
```

```
      E       = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX    = FEVX(V,NELL,EEE)
      UEVY    = FEVY(U,NELL,EEE)
      ZETA1   = VEVX-UEVY
      END IF
      C       = 1.
      E       = 1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX    = FEVX(V,NELL,EEE)
      UEVY    = FEVY(U,NELL,EEE)
      ZETA    = VEVX-UEVY
      L1      = IABS(NOP(NELL,7))
      L       = IABS(NOP(NELL,9))
      IF (J .EQ. 1)
     *WRITE(NLP,901) XE(7),YWGHT(1),ZETA1,P(L1)
      WRITE(NLP,901) XE(9),YE(9),ZETA,P(L)
  301 CONTINUE
C
      DO 701 I  = 1, NEX1B
      NELL    = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+NEX1A*NEY
      DO 706 KI = 1, 9
      LN      = IABS(NOP(NELL,KI))
      XE(KI)  = X(LN)
      YE(KI)  = Y(LN)
  706 CONTINUE
      DO 716 J  = 1, 11
      C       = -1. + (J-1)*0.2
      E       = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY    = FEVY(U,NELL,EEE)
      XX      = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  716 CONTINUE
  701 CONTINUE
C
      NEL1    = 1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 503 KI = 1, 9
      LN      = IABS(NOP(NEL1,KI))
      XE(KI)  = X(LN)
      YE(KI)  = Y(LN)
  503 CONTINUE
      DO 513 J  = 1, 11
      C       = -1. + (J-1)*0.2
      E       = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
```

```
      UEVY      = FEVY(U,NEL1,EEE)
      XX        = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  513 CONTINUE
C
      DO 501 I  = 1, NEX2
      NELL      = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 506 KI = 1, 9
      LN        = IABS(NOP(NELL,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  506 CONTINUE
      DO 516 J  = 1, 11
      C         = -1. + (J-1)*0.2
      E         = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY      = FEVY(U,NELL,EEE)
      XX        = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  516 CONTINUE
  501 CONTINUE
C
      NEL1      = 1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *          + NEY2+NEX2*NEY
      DO 603 KI = 1, 9
      LN        = IABS(NOP(NEL1,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  603 CONTINUE
      DO 613 J  = 1, 11
      C         = -1. + (J-1)*0.2
      E         = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY      = FEVY(U,NEL1,EEE)
      XX        = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  613 CONTINUE
C
      DO 601 I  = 1, NEX3
      NELL      = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                + NEY2+NEX2*NEY
      DO 606 KI = 1, 9
      LN        = IABS(NOP(NELL,KI))
      XE(KI)    = X(LN)
      YE(KI)    = Y(LN)
  606 CONTINUE
      DO 616 J  = 1, 11
      C         = -1. + (J-1)*0.2
```

```
      E       = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY    = FEVY(U,NELL,EEE)
      XX      = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  616 CONTINUE
  601 CONTINUE
C
      DO 801 I  = 1, NEX4
      NELL    = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *              + (NEX2+NEX3+1)*NEY
      DO 806 KI  = 1, 9
      LN      = IABS(NOP(NELL,KI))
      XE(KI)  = X(LN)
      YE(KI)  = Y(LN)
  806 CONTINUE
      C       = -1.
      E       = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY    = FEVY(U,NELL,EEE)
      WRITE(NLI,901) XE(1), UEVY
  801 CONTINUE
C
      WRITE(NLI,'(A50)')  'Shear stress along upstream wall'
      DO 671 K  = 1, NEY1
      NELL    = K + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY+(NEX2-1)*NEY
      DO 676 KI  = 1, 9
      LN      = IABS(NOP(NELL,KI))
      XE(KI)  = X(LN)
      YE(KI)  = Y(LN)
  676 CONTINUE
      DO 686 J  = 1, 11
      E       = -1. + (J-1)*0.2
      C       =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY    = FEVY(U,NELL,EEE)
      UEVX    = FEVX(U,NELL,EEE)
      VEVX    = FEVX(V,NELL,EEE)
      VEVY    = FEVY(V,NELL,EEE)
      A       = 2.*UEVX
      B       = UEVY+VEVX
      D       = 2.*VEVY
      TAU     = B*(COS(ALPHA)**2.-SIN(ALPHA)**2.)+
     *          (D-A)*SIN(ALPHA)*COS(ALPHA)
      YY      = YE(7) + (E+1.)*(YE(9)-YE(7))/2.
      WRITE(NLI,901) YY/SIN(ALPHA), TAU
  686 CONTINUE
```

```
 671  CONTINUE
C
      WRITE(NLI,'(A50)') 'Shear stress along downstream wall'
      DO 771 K   = 1, NEY1
      NELL       = K + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+NEX2+1)*NEY+NEY2
      DO 776 KI  = 1, 9
      LN         = IABS(NOP(NELL,KI))
      XE(KI)     = X(LN)
      YE(KI)     = Y(LN)
 776  CONTINUE
      DO 786 J   = 1, 11
      E          = -1. + (J-1)*0.2
      C          = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY       = FEVY(U,NELL,EEE)
      UEVX       = FEVX(U,NELL,EEE)
      VEVX       = FEVX(V,NELL,EEE)
      VEVY       = FEVY(V,NELL,EEE)
      A          = 2.*UEVX
      B          = UEVY+VEVX
      D          = 2.*VEVY
      TAU        = B*(COS(ALPHA)**2.-SIN(ALPHA)**2.)+
     *             (D-A)*SIN(ALPHA)*COS(ALPHA)
      YY         = YE(1) + (E+1.)*(YE(3)-YE(1))/2.
      WRITE(NLI,901) YY/SIN(ALPHA), TAU
 786  CONTINUE
 771  CONTINUE
C
 901  FORMAT(4F12.6)
 999  FORMAT('TITLE = "Re=', F9.5, '", VARIABLES="XE","YE","UEV","VEV",'
     * / 'ZONE T="ZONE 1", Z=0, I=', 15, ', J=', 15, ', F=POINT')
C
      RETURN
      END
*****************************************************************
```

Τα αποτελέσματα που παίρνουμε από τον υπορουτίνα POSTPR τα επεξεργαζόμαστε στο πρόγραμμα TECPLOT, όπου και βλέπουμε τις γραμμές ροής μέσα στην αεροσήραγγα. Αυτό το κομμάτι του προγράμματος είναι πάγιο και αποτελείται από τα DO-loop 46-47-48-49-59-61-50-53-57 και 71. Από τα αποτελέσματα που αφορούν την διατμητική τάση, βγάζουμε συμπεράσματα για το μέγεθος της ανακυκλοφορίας που λαμβάνει χώρα πριν και μετά το εμπόδιο. Επίσης στην υπορουτίνα αυτή υπολογίζουμε τις ισοβαρείς και τις ισοστροβιλώδεις καμπύλες γύρω από το εμπόδιο και τυπώνονται στο αρχείο pa5.ite

8.ΣΥΜΠΕΡΑΣΜΑΤΑ-ΕΠΙΑΟΓΟΣ

Ανακεφαλαιώνοντας, μελετήσαμε την έρπουσα ροή γύρω από έναν κεκλιμένο φράκτη με γωνία κλίσεως αρχικά 75° και μεταβαλλόμενη στη συνέχεια. Συγκρίναμε τα αριθμητικά αποτελέσματα των ροϊκών γραμμών με αυτές του εργαστηριακού πειράματος και στη συνέχεια προχωρήσαμε στην περαιτέρω μελέτη και εμβάνθυνση του ροϊκού φαινομένου. Γνωρίσαμε την ισχύ των αριθμητικών πειραμάτων που από την μια δεν μπορούμε να ισχυρισθούμε ότι υποκαθιστούν πλήρως τα εργαστηριακά πειράματα, αλλά επιτρέπουν την ενδελεχέστερη μελέτη της ροής υπολογίζοντας σε κάθε σημείο του υπολογιστικού πεδίου την πίεση και την ταχύτητα, σχηματίζοντας τις ισοβαρείς και τις ισοστροβιλώδεις καμπύλες, καθώς επίσης και την μεταβολή της διατμητικής τάσης κατά μήκος των τοιχωμάτων της αεροσήραγγας.

Παρατηρήθηκε ότι στην ροή γύρω από κεκλιμένο φράκτη δημιουργούνται για γωνίες κλίσεως μικρότερες των 90° δύο στρόβιλοι, οι οποίοι αλλάζουν μέγεθος ανάλογα με την κλίση του φράκτη. Για γωνία κλίσεως μικρότερης των 50° ο στρόβιλος ανάντη του κεκλιμένου φράκτη εξαλείφεται. Επίσης παρατηρήθηκε από την επεξεργασία των αποτελεσμάτων ότι το πλάτος του στροβίλου κατάντη παρουσιάζει μέγιστο πλάτος γύρω από τις 60° και φθίνει καθώς η γωνία κλίσεως μειώνεται μέχρις ότου γίνει ίση με μηδέν, οπότε και ομαλοποιείται η ροή.

Στη ροή γύρω από οριζόντιο φράκτη παρατηρήθηκε ότι το μήκος ανακυκλοφορίας δεν αλλάζει όταν το πλάτος του εμποδίου είναι πολύ μικρό (μικρότερο από 1% του ύψους του εμποδίου). Επίσης για λόγο πλάτος ως προς ύψος του εμποδίου μεγαλύτερο από 2 η ροή προσομοιάζει τις συνθήκες backward facing step και ενδιαμέσως η μεταβολή του μήκους ανακυκλοφορίας είναι εκθετική.

Μέσω της εργασίας αυτής αναδεικνύεται για μια ακόμη φορά η αξιοπιστία της μεθόδου των πεπερασμένων στοιχείων, μιας και τα αποτελέσματα είναι σε πολύ καλή συμφωνία με τις εργαστηριακές μετρήσεις. Η επαλήθευση των εργαστηριακών πειραμάτων αριθμητικά, δίνει την δυνατότητα να μελετηθουν ροϊκά φαινόμενα μεταβάλλοντας και τον αριθμό Re χωρίς να είναι απαραίτητο να γίνουν εργαστηριακά πειράματα, που το κόστος τους είναι αρκετά υψηλό. Επίσης η ταχύτητα των εργαστηριακών πειραμάτων είναι μικρή, αν αναλογιστούμε ότι καθώς η υπολογιστική ισχύ ολοένα αυξάνει, τα αριθμητικά αποτελέσματα θα εξάγονται σε χρονικό διάστημα μόλις λίγων ωρών. Έτσι λοιπόν σήμερα που είναι και εποχή της ταχύτητας αλλά και της τεχνολογίας, ο κλάδος αυτός της επιστήμης των ρευστών θα γνωρίσει τεράστια ανάπτυξη τα επόμενα χρόνια.

Ευχή, λοιπόν, όλων είναι η δημιουργία μιας τράπεζας δεδομένων όπου όλα τα ερευνητικά κέντρα του κόσμου να έχουν πρόσβαση. Το ζήτημα όμως δεν είναι η έρευνα να σταματήσει σε μια τράπεζα δεδομένων αλλά θα πρέπει η υπολογιστική ρευστομηχανική να επεκταθεί στη βιομηχανία τόσο των αναπτυγμένων χωρών όσο και σ' αυτή της Ελλάδος ώστε άπαντες να γνωρίζουν και να εφαρμόζουν, όπου είναι εφικτό, τις τελευταίες ερευνητικές δραστηριότητες.

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. J. M. Leone and P. M. Gresho, 'Finite element simulation of steady, two-dimensional, viscous incompressible flow over a step', *J. Comput. Phys.*, **41**, 167-191 (1981).
2. S. Taneda, 'Visualization of separating Stokes flows', *J. Phys. Soc. Jpn.*, **46**, 1935-1942 (1979)
3. V. P. Fragos, 'Computer-aided analysis of flow over a surface mounted obstacle', *M. Sc. Thesis.*, Aristotle University of Thessaloniki, 1995
4. M. Renardy, 'Imposing "No" Boundary Condition at Outflow: Why Does it Work?', *Int. J. numer. Methods fluids*, **24**, 413-417 (1977)
5. Jonathan J. L. Higdon, 'Stokes flow in arbitrary two-dimensional domains: shear flow over ridges and cavities', *J. Fluid Mech. (1985), vol.159, pp.195-226*
6. C.Taylor, T.G.Hughes, 'Finite Element Programming of The Navier-Stokes Equation', *Pineridge Press*, page 44.
7. N.A.Malamataris and V.Bontozoglou, 'Computer Aided Analysis of Viscous Film Flow along an Inclined Wavy Wall', *Journal of Computational Physics, **154**, 372-392(1999)*
8. V.P. Fragos, S. P. Psychoudaki and N.A.Malamataris, 'Computer-Aided Analysis of Flow Past a Surface- Mounted Obstacle', *International Journal For Numerical Methods In Fluids, Vol.25, 495-512(1997)*
9. N. Malamataris, *Computer-Aided Analysis of Flows on Moving and Unbounded Domains: Phase-Change Fronts and Liquid Leveling*, Ph. D.thesis, University of Michigan, Ann Arbor,MI, 1991.
10. Γ. Μπεργελες, Καθ. ΕΜΠ 'Υπολογιστική Ρευστομηχανική', Εκδόσεις Συμεών.
11. C. Pozrikidis, 'Little Book of Streamlines', 1999, Acad Pr, San Diego, CA-USA.
12. P. Hood, 'Frontal solution program for unsymmetric matrices', *Int. j. numer. methods eng.*,**10**, 379 (1974).
13. Α. Μπουντουβής, 'Μέθοδος Πεπερασμένων Στοιχείων', Πανεπιστημιακές σημειώσεις, Αθήνα 1995, ΕΜΠ.

10. ΠΑΡΑΡΤΗΜΑ

ΛΙΣΤΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΧΕΔΙΟ ΠΛΕΓΜΑΤΟΣ

ΑΡΧΕΙΟ 'pa5.dat'

```
      PROGRAM ABC
C
      PARAMETER (NCRIT1=939,NG=501,NXNP=173000,NL=17000,ND=65000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/  SK(NXNP),AA(25,25)
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
      COMMON /NODCOO/ XE(9),YE(9)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /IITT/   ICOUNT
C
C NCR reads the input data in INPAT which are declared in COMMON PAR and INOT
C NCO saves the solution of the steady state problem
C NPP stores the streamlines of the solution
C NLP saves the isovorticity and isobar lines of the solution
C NLI saves the results of the shear rate along the wall and the fence
C
      NCR = 34
      NCO = 36
      NPP = 37
      NLP = 39
      NLI = 38
C
      OPEN(UNIT=NCR,FILE='pa5.dat',STATUS='OLD')
      OPEN(UNIT=NCO,FILE='pa5.con',STATUS='OLD')
      OPEN(UNIT=NPP,FILE='pa5.pst',STATUS='OLD')
      OPEN(UNIT=NLI,FILE='pa5.out',STATUS='OLD')
      OPEN(UNIT=NLP,FILE='pa5.ite',STATUS='OLD')
C
      CALL INPAT
      CALL MSHSTR
      CALL MSHCOO
      CALL BNDRYC
      CALL INIGES
C
      CALL ITRATE
      CALL SVSOL
      CALL POSTPR
C
      END
C
C*********************************************************************
C
      SUBROUTINE INPAT
C
      PARAMETER (NG=501)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /PAR1/   ALPHA,DX,XL
C
      DATA PI /3.141592654/
C
      READ(NCR,*)
      READ(NCR,*)     MAXITE,ERRMAX,IFPRNT,RE,YY1
      WRITE(*,*)      MAXITE,ERRMAX,IFPRNT,RE,YY1
C
      READ(NCR,*)
      READ(NCR,*)     NEX1A,NEX1B,NEX2,NEX3,NEX4,NEY1,NEY2
```

```
      WRITE(*,*)       NEX1A,NEX1B,NEX2,NEX3,NEX4,NEY1,NEY2
C
      READ(NCR,*)
      READ(NCR,*)      ALPHA,DX,XL
      WRITE(*,*)       ALPHA,DX,XL
C
      ALPHA=ALPHA*PI/180.
C
      NEX1 =           NEX1A + NEX1B
      NEY  =           NEY1 + NEY2
      NY1  =           1 + 2*NEY1
      NY2  =           2*NEY2
      NY   =           NY1 + NY2
      NX1  =           1+2*(NEX1A+NEX1B)
      NX2  =           1+2*(NEX1A+NEX1B+NEX2)
      NX3  =           NX2+1+2*NEX3+3
      NX   =           1+2*(NEX1A+NEX1B+NEX2+NEX3+NEX4)+4
C
      READ(NCR,*)
      READ(NCR,*)      (XWGHT(I), I=1,NX,2)
      NN1  =           1+2*(NEX1A+NEX1B+NEX2)+1+1+2*NEX3
      NN2  =           NN1+2
      XWGHT(NN2)       = XWGHT(NN1)+XL*COS(ALPHA)
      WRITE(*,*)       (XWGHT(I), I=1,NX,2)
C
      DO 11  I =       2, NX-1, 2
      XWGHT(I) =       (XWGHT(I-1)+XWGHT(I+1))/2.
   11 CONTINUE
C
      READ(NCR,*)
      READ(NCR,*)      (YWGHT(J),J=1,NY,2)
      YWGHT(NY1)       = XL*SIN(ALPHA)
      WRITE(*,*)       (YWGHT(I),I=1,NY,2)
C
      DO 18 I  =       2, NY-1, 2
      YWGHT(I) =       0.5*(YWGHT(I+1)+YWGHT(I-1))
   18 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      FUNCTION NODNOR (I,J)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *               NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
C
C Node numbering before the step, at the entrance.
C
      IF (I .LE. 2*NEX1A+1)
     *NODNOR = J+(I-1)*(NY+2*2*NEX1A)-2*(I-1)*(I-2)/2
C
C Node numbering before the step in the rectangular part of the discretized
C domain.
C
      IF (I .GT. 2*NEX1A+1  .AND.  I .LE. NX1)
     *NODNOR = J+(I-1)*NY+2*NEX1A*(1+2*NEX1A)
C
C Node numbering before the step in the part of the discretized domain parallel
C to the inclined obstacle.
C
      IF (I .GE. NX1+1  .AND.  I .LE. NX2)
     *NODNOR = (I-(NX1+1))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1
C
C Node numbering on the step.
C
      IF (I .EQ. NX2+1  .AND.  J .GE. NY1)
     *NODNOR = J-(NY1-1) + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
     *                     2*NEX2*NY
C
C Node numbering after the step in the part of the discretized domain parallel
C to the inclined obstacle.
C
      IF (I .GE. NX2+2  .AND.  I .LE. NX2+2*NEX3+2)
     *NODNOR = (I-(NX2+2))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
```

```
     *                              2*NEX2*NY+NY2+1
C
C Node numbering after the step in the rectangular part of the discretized
C domain.
C
      IF (I .GT. NX3)
     *NODNOR = (I-(NX3+1))*NY+J + NX1*NY+2*NEX1A*(1+2*NEX1A)+NY-2+NY-1+
     *                            2*NEX2*NY+NY2+1+(1+2*NEX3)*NY+NY1-2+
     *                            NY1-1
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE MSHSTR
C
      PARAMETER  (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
C NE  : #-of elements
C NH  : #-of nodes
C
      NE       = NEX1A*(NEX1A+1) + (NEX1A+NEX1B+NEX2+NEX3+NEX4+2)*NEY
      NH       = 2*NEX1A*(2*NEX1A+1) + NX1*NY + NY-2 + NY-1 + NY2+1 +
     *           2*NEX2*NY + (1+2*NEX3)*NY + NY1-2 + NY1-1 + 2*NEX4*NY
      NHADD    = NH
C
C Specify outflow elements
C
      DO 13 I  = NE-NEY+1, NE
      NEO(I)   = 1
 13   CONTINUE
C
C Triangular elements at the entrance.
C
      DO 11 I  = 1, NEX1A
      LB       = 1+(I-1)*(NEY+2*NEX1A)-2*(I-1)*(I-2)/2
      LT       = NEY+2*(NEX1A-(I-1))+(I-1)*(NEY+2*NEX1A)-2*(I-1)*(I-2)/2
      NET(LB)  = 1
      NET(LT)  = 1
 11   CONTINUE
C
C Triangular elements in the interior of the domain
C
      LT       = NEX1A*(1+NEX1A)+(NEX1A+NEX1B)*NEY+1
      NET(LT)  = 1
      LT       = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+NEX2+NEX3+2)*NEY
      NET(LT)  = 1
C
C Specify the #-of nodes NBN(I) and the #-of unknowns NCN(I) in the element
C
      DO 12 IEL= 1, NE
      IF (NET(IEL).EQ.1) THEN
      NBN(IEL) = 6
      NCN(IEL) = 15
      ELSE
      NBN(IEL) = 9
      NCN(IEL) = 22
      END IF
 12   CONTINUE
C
C Specify the degree of freedom in the element
C Remember: MDF(I)=3 at corner nodes
C           MDF(I)=2 at midnodes
C
      DO 5 I   = 1, NH
      MDF(I)   = 2
 5    CONTINUE
C
C Fix the edge nodes of quadrilateral elements
```

```
C At the entrance
C
      DO 61 I   = 1, 2*NEX1A+1, 2
      DO 61 J   = 1, NY-2*(I-1)+2*2*NEX1A, 2
      L         = NODNOR(I,J)
      MDF(L)    = 3
 61   CONTINUE
C
C In the rectangular part of the domain upstream the fence
C
      DO 67 I   = 1+2*NEX1A+2, NX1, 2
      DO 67 J   = 1, NY, 2
      L         = NODNOR(I,J)
      MDF(L)    = 3
 67   CONTINUE
C
C In the part of the domain upstream the fence with parallel inclination
C First the correction line
C
      DO 62 J   = 2, NY-1, 2
      L         = NODNOR(NX1,NY)+NY-2 + J
      MDF(L)    = 3
 62   CONTINUE
C
C Then the main part of the domain with the parallel inclination
C
      DO 63 I   = NX1+2, NX2+2+2*NEX3, 2
      DO 63 J   = 1, NY, 2
      L         = NODNOR(I,J)
      MDF(L)    = 3
 63   CONTINUE
C
      DO 64 J   = 1, NY1-2, 2
      L         = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      MDF(L)    = 3
 64   CONTINUE
C
      DO 65 I   = NX3+2, NX, 2
      DO 65 J   = 1, NY, 2
      L         = NODNOR(I,J)
      MDF(L)    = 3
 65   CONTINUE
C
      NOPP(1)   = 1
      DO 71 I   = 2, NH
      NOPP(I)   = NOPP(I-1) + MDF(I-1)
 71   CONTINUE
C
      NP        = 0
      DO 8  I   = 1, NH
      NP        = NP + MDF(I)
  8   CONTINUE
C
      WRITE(*,101) NE, NH, NP
C
      IF (NOPP(NH)+MDF(NH)-1 .EQ. NP) THEN
      PRINT*, 'NP CORRECT!'
      ELSE
      PRINT*, 'CHECK NP!'
      STOP
      END IF
C
      CALL CALNOP
C
 101  FORMAT(' NE, NH, NP : ', 3I7)
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE CALNOP
C
      PARAMETER (NXNP=173000,NL=17000,ND=65000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
```

```
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /INOUT/ NCR,NLP,NCO,NSA,NPP,NLI
C
C K locates the global node of the bottom left corner of the NK element
C JJ-1 takes care of the nodes above K
C NY*(II-1) takes care of the adjacent nodes to K
C KK is the local node
C
C Triangular elements in the entrance
C
      DO 1  I     = 1, NEX1A
      NEI         = 2*((I-1)*(I-2)/2)
      LB          = 1+(I-1)*(NEY+2*NEX1A)-NEI
      NOP(LB,1)   = NODNOR(2*(I-1)+1, 1)
      NOP(LB,2)   = NODNOR(2*(I-1)+3, 1)
      NOP(LB,3)   = NOP(LB,1)+2
      NOP(LB,4)   = NODNOR(2*(I-1)+2, 1)
      NOP(LB,5)   = NOP(LB,4)+1
      NOP(LB,6)   = NOP(LB,1)+1
      LT          = NEY+2*(NEX1A-(I-1))+(I-1)*(NEY+2*NEX1A)-NEI
      NOP(LT,1)   = NODNOR(2*(I-1)+1,NY+2*2*(NEX1A-I)+4)
      NOP(LT,2)   = NOP(LT,1)-2
      NOP(LT,3)   = NODNOR(2*(I-1)+3,NY+2*2*(NEX1A-I))
      NOP(LT,4)   = NOP(LT,1)-1
      NOP(LT,6)   = NODNOR(2*(I-1)+2,NY+2*2*(NEX1A-I)+2)
      NOP(LT,5)   = NOP(LT,6)-1
  1   CONTINUE
C
C Quadrilateral elements in the entrance
C
      DO 51 I     = 1, NEX1A
      DO 51 J     = 1, NEY+2*(NEX1A-I)
      NK          = 1+(I-1)*NEY+J+2*(NEX1A*(NEX1A+1)/2)-
     *              2*((NEX1A-(I-1))*(NEX1A-(I-1)+1)/2)
      DO 51 II    = 1, 3
      K           = NODNOR(2*(I-1)+II,3-(II-1)+2*(J-1))
      DO 51 JJ    = 1, 3
      KK          = (II-1)*3+JJ
      NOP(NK,KK)  = K+JJ-1
  51  CONTINUE
C
C Quadrilateral elements upstream the obstacle
C
      DO 52 I     = 1, NEX1B
      DO 52 J     = 1, NEY
      K           = 1+2*(I-1)*NY+2*(J-1)+2*(2*NEX1A*(2*NEX1A+1))/2+
     *              2*NEX1A*NY
      NK          = (I-1)*NEY+J+2*(NEX1A*(NEX1A+1)/2)+NEX1A*NEY
      DO 52 II    = 1, 3
      DO 52 JJ    = 1, 3
      KK          = (II-1)*3+JJ
      NOP(NK,KK)  = K+(II-1)*NY+JJ-1
  52  CONTINUE
C
C Triangular elements in the interior of the domain
C
      NK          = NEX1A*(1+NEX1A) + (NEX1A+NEX1B)*NEY + 1
      NOP(NK,1)   = NODNOR(NX1,1)
      NOP(NK,2)   = NODNOR(NX1,NY)+NY-2 + 2
      NOP(NK,3)   = NOP(NK,1) + 2
      NOP(NK,4)   = NOP(NK,2) - 1
      NOP(NK,5)   = NODNOR(NX1,NY) + 1
      NOP(NK,6)   = NOP(NK,1) + 1
      NK          = NEX1A*(1+NEX1A) + (NEX1A+NEX1B+NEX2+NEX3+2)*NEY
      NOP(NK,1)   = NODNOR(NX2+2+2*NEX3,NY1)
      NOP(NK,2)   = NOP(NK,1) - 2
      NOP(NK,3)   = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + NY1-2
      NOP(NK,4)   = NOP(NK,1) - 1
      NOP(NK,5)   = NODNOR(NX2+2+2*NEX3,NY) + NY1-2
      NOP(NK,6)   = NOP(NK,3) + 1
C
C Quadrilateral elements above the first triangle in the domain
C
      DO 41   J   = 2, NEY
      K           = NODNOR(NX1,2*(J-1)+1)
      NK          = NEX1A*(1+NEX1A)+(NEX1A+NEX1B)*NEY + J
```

```
      DO 41 II    = 1, 3
      DO 41 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      IF (II .EQ. 1)  THEN
      NOP(NK,KK) = K + JJ-1
      ELSE IF (II .EQ. 2)  THEN
      NOP(NK,KK) = K + JJ-1 + NY-2
      ELSE
      NOP(NK,KK) = K + JJ-1 + NY-2 + NY-1
      END IF
   41 CONTINUE
C
C The first quadrilateral element to right of the trianlge
C
      NK          = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + 1
      NOP(NK,1)   = NODNOR(NX1,1)
      NOP(NK,2)   = NODNOR(NX1,NY)+NY-2 + 1
      NOP(NK,3)   = NOP(NK,2) + 1
      DO 42 II    = 2, 3
      DO 42 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = NODNOR(NX1,NY)+NY-2+NY-1 + 1 + (II-2)*NY + JJ-1
   42 CONTINUE
C
C The rest of the quadrilateral elements above
C
      DO 43   J   = 2, NEY
      NK          = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + J
      K           = NODNOR(NX1,NY)+NY-2 + 2*(J-1)
      DO 43 II    = 1, 3
      DO 43 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   43 CONTINUE
C
C Quadrilateral elements upstream the fence with parallel orientation
C
      DO 44   I   = 2, NEX2
      DO 44   J   = 1, NEY
      NK          = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1)*NEY + (I-1)*NEY+J
      K           = NODNOR(NX1+2*(I-1),1+2*(J-1))
      DO 44 II    = 1, 3
      DO 44 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   44 CONTINUE
C
C Quadrilateral elements above the fence
C
      DO 45   J   = 1, NEY2
      NK          = NEX1A*(1+NEX1A)+(NEX1A+NEX1B+1+NEX2)*NEY + J
      K           = NODNOR(NX2,2*(J-1)+NY1)
      DO 45 II    = 1, 3
      DO 45 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*(NY2+1)
      IF (II .EQ. 3)
     *NOP(NK,KK) = NOP(NK,KK) + NY1-1
   45 CONTINUE
C
C Quadrilateral elements downstream the fence with parallel irientation
C
      NEPR        = NEX1A*(1+NEX1A) + (NEX1A+NEX1B+1+NEX2)*NEY + NEY2
      DO 46   I   = 1, NEX3
      DO 46   J   = 1, NEY
      NK          = NEPR + (I-1)*NEY+J
      K           = NODNOR(NX2+2*(I-1)+2,2*(J-1)+1)
      DO 46 II    = 1, 3
      DO 46 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   46 CONTINUE
C
C Correction-quadrilateral elements below the triangle
C
      NEPR        = NEPR + NEX3*NEY
      DO 47   J   = 1, NEY1-1
      NK          = NEPR + J
```

```
      K           = NODNOR(NX2+2+2*NEX3,2*(J-1)+1)
      DO 47 II    = 1, 3
      DO 47 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      IF (II .LE. 2)
     *NOP(NK,KK) = K + JJ-1 + (II-1)*NY
      IF (II .EQ. 3)
     *NOP(NK,KK) = K + JJ-1 + NY + NY1-2
   47 CONTINUE
C
C Quadrilateral elements next to the correction-quadrilateral elements
C
      NEPR        = NEPR + NEY1
      DO 48   J   = 1, NEY1-1
      NK          = NEPR + J
      K           = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + 2*(J-1)+1
      DO 48 II    = 1, 3
      DO 48 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      IF (II .LE. 2)
     *NOP(NK,KK) = K + JJ-1 + (II-1)*(NY1-1)
      IF (II .EQ. 3)
     *NOP(NK,KK) = K + JJ-1 + NY1-1 + NY
   48 CONTINUE
C
      NK          = NEPR+NEY1
      NOP(NK,1)   = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + NY1-2
      NOP(NK,2)   = NOP(NK,1)+1
      NOP(NK,3)   = NODNOR(NX2+2+2*NEX3,NY1)
      DO 57 II    = 2, 3
      DO 57 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      IF (II .EQ. 2)
     *NOP(NK,KK) = NOP(NK,1) + NY1-1 + JJ-1
      IF (II .EQ. 3)
     *NOP(NK,KK) = NOP(NK,1) + NY1-1 + NY + JJ-1
   57 CONTINUE
C
      NEPR        = NK
      DO 49   J   = 1, NEY2
      NK          = NEPR + J
      K           = NODNOR(NX2+2+2*NEX3,2*(J-1)+NY1)
      DO 49 II    = 1, 3
      DO 49 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      IF (II .EQ. 1)
     *NOP(NK,KK) = K + JJ-1
      IF (II .GE. 2)
     *NOP(NK,KK) = K + JJ-1 + NY1-2+NY1-1 + (II-1)*NY
   49 CONTINUE
C
C Rectangular elements up to the outflow
C
      NEPR        = NEPR + NEY2
      DO 50   I   = 2, NEX4
      DO 50   J   = 1, NEY
      NK          = NEPR + (I-2)*NEY+J
      K           = NODNOR(NX3+2*(I-1),2*(J-1)+1)
      DO 50 II    = 1, 3
      DO 50 JJ    = 1, 3
      KK          = 3*(II-1)+JJ
      NOP(NK,KK) = K + JJ-1 + (II-1)*NY
   50 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE BNDRYC
C
      PARAMETER (NXNP=173000,NL=17000,ND=65000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *               NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
```

```
      COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /INOT/  TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
C
      DO 1 I  = 1, NP
      NCOD(I) = 0
 1    CONTINUE
C
C Essential b.c's for velocity at the entrance
C
      DO 2 J           = 1, NY+2*2*NEX1A
      L                = NODNOR(1,J)
      NCOD(NOPP(L))    = 1
      NCOD(NOPP(L)+1)  = 1
 2    CONTINUE
C
C Essential b.c's for velocity along the top
C
      DO 3 I           = 1, 1+2*NEX1A
      LT               = NODNOR(I,NY+2*(2*NEX1A-(I-1)))
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 3    CONTINUE
C
      DO 31 I          = 1+2*NEX1A+1, NX1
      LT               = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 31   CONTINUE
C
      LT               = NODNOR(NX1,NY) + NY-2
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
      LT               = NODNOR(NX1,NY)+NY-2 + NY-1
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
C
      DO 32 I          = NX1+1, NX2+2+2*NEX3
      LT               = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 32   CONTINUE
C
      DO 33 I          = NX3+1, NX
      LT               = NODNOR(I,NY)
      NCOD(NOPP(LT))   = 1
      NCOD(NOPP(LT)+1) = 1
 33   CONTINUE
C
C Essential b.c's for velocity along the bottom
C
      DO 4 I           = 1, NX1
      LB               = NODNOR(I,1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
 4    CONTINUE
C
      DO 5 I           = NX1+1, NX2
      LB               = NODNOR(I,1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
 5    CONTINUE
C
      DO 6 J           = 1, NY1
      LH               = NODNOR(NX2,J)
      NCOD(NOPP(LH))   = 1
      NCOD(NOPP(LH)+1) = 1
 6    CONTINUE
C
      LB               = NODNOR(NX2+1,NY1)
      NCOD(NOPP(LB))   = 1
      NCOD(NOPP(LB)+1) = 1
C
      DO 7 J           = 1, NY1
      LH               = NODNOR(NX2+2,J)
      NCOD(NOPP(LH))   = 1
      NCOD(NOPP(LH)+1) = 1
 7    CONTINUE
C
```

```
      DO 9 I            = NX2+2, NX2+2+2*NEX3
      LB                = NODNOR(I,1)
      NCOD(NOPP(LB))    = 1
      NCOD(NOPP(LB)+1)  = 1
    9 CONTINUE
C
      LB                = NODNOR(NX2+2+2*NEX3,NY) + 1
      NCOD(NOPP(LB))    = 1
      NCOD(NOPP(LB)+1)  = 1
      LB                = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + 1
      NCOD(NOPP(LB))    = 1
      NCOD(NOPP(LB)+1)  = 1
C
      DO 8 I            = NX3+1, NX
      LB                = NODNOR(I,1)
      NCOD(NOPP(LB))    = 1
      NCOD(NOPP(LB)+1)  = 1
    8 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE MSHCOO
C
      PARAMETER (NXNP=173000,ND=65000,NG=501,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /PAR1/   ALPHA,DX,XL
C
      YT        =  YY1+YWGHT(NY)
      YB        = -YY1+YWGHT(1)
      X1        =  XWGHT(1)
      X0        =  XWGHT(1+2*NEX1A)
      NX1A      =  1+2*NEX1A
C
      DO 11 I   = 1, 1+2*NEX1A
C
      DO 18 J   = 1, NY
      L         = NODNOR(I,J+2*NEX1A-(I-1))
      X(L)      = XWGHT(I)
      Y(L)      = YWGHT(J)
   18 CONTINUE
C
      IF (I .LT. 1+2*NEX1A) THEN
      DO 17 J1  = 1,2*NEX1A-(I-1)
      LB        = NODNOR(I,J1)
      X(LB)     = XWGHT(I)
      Y(LB)     = (YB-YWGHT(1))*(XWGHT((I-1)+J1)-X0)/(X1-X0)+YWGHT(1)
      LT        = NODNOR(I,NY+2*NEX1A-(I-1)+J1)
      X(LT)     = XWGHT(I)
      Y(LT)     = (YT-YWGHT(NY))*(XWGHT(NX1A-J1)-X0)/(X1-X0)+YWGHT(NY)
   17 CONTINUE
      END IF
C
   11 CONTINUE
C
      DO 19 I   = 1+2*NEX1A+1, NX1
      DO 19 J   = 1, NY
      L         = NODNOR(I,J)
      X(L)      = XWGHT(I)
      Y(L)      = YWGHT(J)
   19 CONTINUE
C
      DO 21 J   = 1, NY-1
      L         = NODNOR(NX1,NY)+NY-2 + J
      Y(L)      = YWGHT(J+1)
      IF (J .LE. NY1-1)  THEN
```

```
      X(L)      = XWGHT(NX1) + YWGHT(J+1)/TAN(ALPHA)
      ELSE
      X(L)      = XWGHT(NX1) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   21 CONTINUE
C
      DO 22 J   = 3, NY
      L1        = NODNOR(NX1,J)
      L2        = NODNOR(NX1,NY)+NY-2 + J-1
      L         = NODNOR(NX1,NY) + J-2
      X(L)      = (X(L1)+X(L2))/2.
      Y(L)      = YWGHT(J)
   22 CONTINUE
C
      DO 23 I   = NX1+1, NX2
      DO 23 J   = 1, NY
      L         = NODNOR(I,J)
      Y(L)      = YWGHT(J)
      IF (J .LE. NY1)  THEN
      X(L)      = XWGHT(I) + YWGHT(J)/TAN(ALPHA)
      ELSE
      X(L)      = XWGHT(I) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   23 CONTINUE
C
      DO 24 J   = NY1, NY
      L         = NODNOR(NX2+1,J)
      Y(L)      = YWGHT(J)
      X(L)      = XWGHT(NX2+1) + YWGHT(NY1)/TAN(ALPHA)
   24 CONTINUE
C
      DO 25 I   = NX2+2, NX2+2+2*NEX3
      DO 25 J   = 1, NY
      L         = NODNOR(I,J)
      Y(L)      = YWGHT(J)
      IF (J .LE. NY1)  THEN
      X(L)      = XWGHT(I) + YWGHT(J)/TAN(ALPHA)
      ELSE
      X(L)      = XWGHT(I) + YWGHT(NY1)/TAN(ALPHA)
      END IF
   25 CONTINUE
C
      DO 26 J   = 1, NY1-1
      L         = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      Y(L)      = YWGHT(J)
      X(L)      = XWGHT(NX2+2*NEX3+4)
   26 CONTINUE
C
      DO 27 J   = 1, NY1-2
      L         = NODNOR(NX2+2+2*NEX3,NY) + J
      L1        = NODNOR(NX2+2+2*NEX3,J)
      L2        = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      Y(L)      = YWGHT(J)
      X(L)      = (X(L1)+X(L2))/2.
   27 CONTINUE
C
      DO 13 I   = NX3+1, NX
      DO 13 J   = 1, NY
      L         = NODNOR(I,J)
      X(L)      = XWGHT(I)
      Y(L)      = YWGHT(J)
   13 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE INIGES
C
      PARAMETER (ND=65000,NG=501,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /FRON1/   NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /MESH/    NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                 NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
```

```
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /MESOLD/ NX1OLD, NXXOLD, NXOLD, NY1OLD, NYOLD, NHOLD
C
C Remember that the initial guess must always satisfy the conservation
C of mass! Linear interpolation may be used in order to avoid steep
C gradients which give high values to the momentum residuals.
C
      DO 7 I  = 1, NH
      V(I)    = 0.
      U(I)    = 0.
      P(I)    = 0.
    7 CONTINUE
C
      DO 8 I  = 1, 2*NEX1A
      DO 8 J  = 1, NY+2*2*NEX1A-2*(I-1)
      L       = NODNOR(I,J)
      IF (J .GT. 2*NEX1A-(I-1) .AND. J .LE. NY+2*NEX1A-(I-1)) THEN
      U(L)    = 1.
      ELSE
      U(L)    = 0.
      END IF
    8 CONTINUE
C
      XL      = YWGHT(NY)
C
C Parabolic velocity profile upstream the obstacle
C
      DO 9 I  = 1+2*NEX1A, NX1
      DO 9 J  = 1, NY
      L       = NODNOR(I,J)
      A       = -6./(XL**2.)
      B       = 6./XL
      U(L)    = A*(Y(L)**2.) + B*Y(L)
      IF (J .EQ. 1  .OR. J .EQ. NY)
     *U(L)    = 0.
    9 CONTINUE
C
C Parabolic velocity profile above the obstacle
C
      DO 21 I = NX1+1, NX2+2+2*NEX3
      DO 21 J = NY1, NY
      A1      = Y(NODNOR(I,NY1))
      L       = NODNOR(I,J)
      A       = -6.*XL/((XL-A1)**3)
      B       = 6.*XL*(XL+A1)/((XL-A1)**3)
      C       = -6.*A1*XL*XL/((XL-A1)**3)
      U(L)    = A*Y(L)**2 + B*Y(L) + C
      IF (J .EQ. NY1  .OR.  J .EQ. NY)
     *U(L)    = 0.
   21 CONTINUE
C
C Parabolic velocity profile downstream the obstacle
C
      DO 11 I = NX3+1, NX
      DO 11 J = 1, NY
      L       = NODNOR(I,J)
      A       = -6./(XL**2)
      B       = 6./XL
      U(L)    = A*Y(L)**2 + B*Y(L)
      IF (J .EQ. 1  .OR.  J .EQ. NY)
     *U(L)    = 0.
   11 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE SVSOL
C
      PARAMETER (ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
```

```
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /PAR1/   ALPHA,DX,XL
C
      REWIND NCO
C
      WRITE(NCO,101) RE, ALPHA
C
      DO 103 I = 1, NH
      WRITE(NCO,104) I, X(I), Y(I), U(I), V(I), P(I)
 103  CONTINUE
C
 101  FORMAT ('RE : ', F12.5, ', ALPHA : ', F5.1)
 104  FORMAT (I5,4F12.6,1PE19.10)
C
 999  RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE GAUSCT
C
      PARAMETER (NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      IF (NET(NELL) .EQ. 1)  THEN
      GPX(1)          = 1./3.
      GPX(2)          = 1./2.
      GPX(3)          = 1./2.
      GPX(4)          = 0.
      GPX(5)          = 0.
      GPX(6)          = 1.
      GPX(7)          = 0.
      GPY(1)          = GPX(1)
      GPY(5)          = GPX(5)
      DO 13  I        = 2, 4
      GPY(I)          = GPX(6-I)
 13   CONTINUE
      DO 11  I        = 6, 7
      GPY(I)          = GPX(13-I)
 11   CONTINUE
      GWEI(1)         = 27./120.
      DO 12  I        = 2, 4
      GWEI(I)         = 8./120.
 12   CONTINUE
      DO 15  I        = 5, 7
      GWEI(I)         = 3./120.
 15   CONTINUE
      ELSE
      R               = SQRT(.6)
      VAL             = -2.*R
      DO 10 I         = 1, 3
      VAL             = VAL+R
      GPX((I-1)*3+1)  = VAL
      GPX((I-1)*3+2)  = VAL
      GPX((I-1)*3+3)  = VAL
      GPY((I-1)*3+1)  = -R
      GPY((I-1)*3+2)  = 0.
      GPY((I-1)*3+3)  = R
      GPT(I)          = GPY(I)
   10 CONTINUE
      GWEI(1)         = 25./81.
      GWEI(2)         = 40./81.
      GWEI(3)         = GWEI(1)
      GWEI(4)         = GWEI(2)
      GWEI(5)         = 64./81.
      GWEI(6)         = GWEI(2)
      GWEI(7)         = GWEI(1)
```

```
      GWEI(8)          = GWEI(2)
      GWEI(9)          = GWEI(1)
      GWE(1)           = 5./9.
      GWE(2)           = 8./9.
      GWE(3)           = GWE(1)
      END IF
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE TFUNC(C,E)
C
      PARAMETER (NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      IF (NET(NELL) .EQ. 1)  THEN
      PHI(1) = 1. -3.*C - 3.*E + 4.*C*E + 2.*C*C + 2.*E*E
      PHI(2) = 2.*C*C - C
      PHI(3) = 2.*E*E - E
      PHI(4) = 4.*C - 4.*C*E - 4.*C*C
      PHI(5) = 4.*C*E
      PHI(6) = 4.*E - 4.*C*E - 4.*E*E
      ELSE
      PHI(1) = .25*C*(C-1.)*E*(E-1.)
      PHI(2) = .5*C*(C-1.)*(1.-E*E)
      PHI(3) = .25*C*(C-1.)*E*(E+1.)
      PHI(4) = .5*(1.-C*C)*E*(E-1.)
      PHI(5) = (1.-C*C)*(1.-E*E)
      PHI(6) = .5*(1.-C*C)*E*(E+1.)
      PHI(7) = .25*C*(C+1.)*E*(E-1.)
      PHI(8) = .5*C*(C+1.)*(1.-E*E)
      PHI(9) = .25*C*(C+1.)*E*(E+1.)
      END IF
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE DER(C,E)
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
      COMMON /JACOB/  XC,XEDA,YC,YEDA,Y1,AY1,CX,CY,EX,EY
      COMMON /NODCOO/ XE(9),YE(9)
C
      IF (NET(NELL) .EQ. 1)  THEN
      PHIC(1) = -3. + 4.*E + 4.*C
      PHIE(1) = -3. + 4.*C + 4.*E
      PHIC(2) = 4.*C - 1.
      PHIE(2) = 0.
      PHIC(3) = 0.
      PHIE(3) = 4.*E - 1.
      PHIC(4) = 4. - 4.*E - 8.*C
      PHIE(4) = -4.*C
      PHIC(5) = 4.*E
      PHIE(5) = 4.*C
      PHIC(6) = -4.*E
      PHIE(6) = 4. - 4.*C - 8.*E
      ELSE
      PHIC(1) = E*(E-1.)*(.5*C-.25)
      PHIC(2) = (1.-E*E)*(C-.5)
      PHIC(3) = E*(E+1.)*(.5*C-.25)
      PHIC(4) = -C*E*(E-1.)
      PHIC(5) = -2.*C*(1.-E*E)
```

```
      PHIC(6) = -C*E*(E+1.)
      PHIC(7) = E*(E-1.)*(.5*C+.25)
      PHIC(8) = (1.-E*E)*(C+.5)
      PHIC(9) = E*(E+1.)*(.5*C+.25)
      PHIE(1) = C*(C-1.)*(.5*E-.25)
      PHIE(4) = (1.-C*C)*(E-.5)
      PHIE(7) = C*(C+1.)*(.5*E-.25)
      PHIE(2) = -E*C*(C-1.)
      PHIE(5) = -2.*E*(1.-C*C)
      PHIE(8) = -E*C*(C+1.)
      PHIE(3) = C*(C-1.)*(.5*E+.25)
      PHIE(6) = (1.-C*C)*(E+.5)
      PHIE(9) = C*(C+1.)*(.5*E+.25)
      END IF
C
      XC      = 0.
      XEDA    = 0.
      YC      = 0.
      YEDA    = 0.
      DO 3 I  = 1, IQ
      XC      = XC+XE(I)*PHIC(I)
      XEDA    = XEDA+XE(I)*PHIE(I)
      YC      = YC+YE(I)*PHIC(I)
      YEDA    = YEDA+YE(I)*PHIE(I)
    3 CONTINUE
      Y1      = XC*YEDA-YC*XEDA
C
      IF (Y1 .LE. 0.) THEN
      PRINT*, 'NELL : ', NELL
      PRINT*, 'PROBLEMS IN ISOPARAMETRIC MAPPING'
      DO 7 I  = 1, IQ
      PRINT*, 'XE, YE : ', XE(I), YE(I)
    7 CONTINUE
      STOP
      END IF
C
      AY1     = ABS(Y1)
      CX      = YEDA/Y1
      CY      = -XEDA/Y1
      EX      = -YC/Y1
      EY      = XC/Y1
      DO 5 I  = 1,IQ
      PHIX(I) = PHIC(I)*CX+PHIE(I)*EX
    5 PHIY(I) = PHIC(I)*CY+PHIE(I)*EY
C
      RETURN
      END
C
C*********************************************************************
C
      FUNCTION FEV(VAR,I,EEF)
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION VAR(ND)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /OLDVAR/ UOLD(ND),VOLD(ND),POLD(ND)
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      FEV     = 0.
      DO 10 J = 1, IQ
      LN      = IABS(NOP(I,J))
      EEE     = EEF
      IF (EEE .NE. 0.  .AND.  LB .NE. LN)  EEE = 0.
      FEV     = FEV + (VAR(LN)+EEE)*PHI(J)
   10 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
```

```
      FUNCTION FEVX(VAR,I,EEF)
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION VAR(ND)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      FEVX    = 0.
      DO 10 J = 1, IQ
      LN      = IABS(NOP(I,J))
      EEE     = EEF
      IF (EEE .NE. 0.  .AND.  LB .NE. LN)  EEE = 0.
      FEVX    = FEVX + (VAR(LN)+EEE)*PHIX(J)
   10 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      FUNCTION FEVY(VAR,I,EEF)
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION VAR(ND)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      FEVY    = 0.
      DO 10 J = 1, IQ
      LN      = IABS(NOP(I,J))
      EEE     = EEF
      IF (EEE .NE. 0.  .AND.  LB .NE. LN)  EEE = 0.
      FEVY    = FEVY + (VAR(LN)+EEE)*PHIY(J)
   10 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE TPRES(C,E)
C
      PARAMETER (NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
      PHIP(1) = .25*(1.-C)*(1.-E)
      PHIP(2) = .25*(1.-C)*(1.+E)
      PHIP(3) = .25*(1.+C)*(1.-E)
      PHIP(4) = .25*(1.+C)*(1.+E)
C
      RETURN
      END
C
C*********************************************************************
C
      FUNCTION FEVP(VAR,I,EEF)
C
      PARAMETER  (NXNP=173000,ND=65000,NL=17000)
C
```

```
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION VAR(ND)
C
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
      COMMON /PRES/   NNPRES(4),IL
C
      FEVP    = 0.
      DO 10 K = 1, IL
      LN      = IABS(NOP(I,NNPRES(K)))
      EEE     = EEF
      IF (EEE .NE. 0.  .AND.  LB .NE. LN)  EEE = 0.
      FEVP    = FEVP + (VAR(LN)+EEE)*PHIP(K)
   10 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE ABFIND
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000,NCRIT1=939)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /JACOB/  XC,XEDA,YC,YEDA,Y1,AY1,CX,CY,EX,EY
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/  SK(NXNP),AA(25,25)
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /OLDVAR/ UOLD(ND),VOLD(ND),POLD(ND)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
      COMMON /NODCOO/ XE(9),YE(9)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /PRES/   NNPRES(4),IL
C
      DIMENSION R(22,22)
      DIMENSION IU(9),IV(9),IP(4),IUQ(9),IVQ(9),IPQ(4),IUT(6),IVT(6),
     *          IPT(3),NNPRQ(4),NNPRT(3)
C
      DATA IUT   /1,4,7,10,12,14/
      DATA IVT   /2,5,8,11,13,15/
      DATA IPT   /3,6,9/
      DATA IUQ   /1,4,6,9,11,13,15,18,20/
      DATA IVQ   /2,5,7,10,12,14,16,19,21/
      DATA IPQ   /3,8,17,22/
      DATA NNPRT /1,2,3/
      DATA NNPRQ /1,3,7,9/
C
C Distinguish between triangular and quadrilateral elements
C
      IF (NET(NELL) .EQ. 1) THEN
      NGP         = 7
      IQ          = 6
      IL          = 3
      DO 391 I    = 1, IQ
      IU(I)       = IUT(I)
      IV(I)       = IVT(I)
  391 CONTINUE
      DO 393 I    = 1, IL
      IP(I)       = IPT(I)
  393 CONTINUE
      DO 395 I    = 1, IL
      NNPRES(I)   = NNPRT(I)
  395 CONTINUE
      ELSE
```

```
      NGP          = 9
      IQ           = 9
      IL           = 4
      DO 392 I     = 1, IQ
      IU(I)        = IUQ(I)
      IV(I)        = IVQ(I)
  392 CONTINUE
      DO 394 I     = 1, IL
      IP(I)        = IPQ(I)
  394 CONTINUE
      DO 396 I     = 1, IL
      NNPRES(I)    = NNPRQ(I)
  396 CONTINUE
      END IF
C
      EEU        = 0.
      EEV        = 0.
      EET        = 0.
      EEP        = 0.
      EEOLD      = 0.
C
C Initialize Jacobian ,R, and remember residual ,R1, is initialized in FRONT
C
      DO 101 I = 1, 22
      DO 101 J = 1, 22
      R(I,J)     = 0.
  101 CONTINUE
C
C Give global coordinates to the master element for isoparametric mapping
C
      DO 102 J = 1, IQ
      LN         = IABS(NOP(NELL,J))
      XE(J)      = X(LN)
      YE(J)      = Y(LN)
  102 CONTINUE
C
C Assembly of R and R1
C
      DO 103 K    = 1, NGP
      CALL GAUSCT
      C           = GPX(K)
      E           = GPY(K)
      CALL TFUNC(C,E)
      CALL TPRES(C,E)
      CALL DER(C,E)
      UEVX        = FEVX(U,NELL,EEU)
      UEVY        = FEVY(U,NELL,EEU)
      VEVX        = FEVX(V,NELL,EEV)
      VEVY        = FEVY(V,NELL,EEV)
      UEV         = FEV(U,NELL,EEU)
      VEV         = FEV(V,NELL,EEV)
      PEV         = FEVP(P,NELL,EEP)
      WE          = GWEI(K)*Y1
C
C Compute -Ru, -Rv
C
      DO 104 KI   = 1, IQ
      IX          = NOPP(IABS(NOP(NELL,KI)))
      IY          = IX+1
      R1(IX)      = R1(IX) - WE*((UEV*UEVX+VEV*UEVY)*PHI(KI)          +
     *                           (-PEV+(2./RE)*UEVX)*PHIX(KI)         +
     *                           (1./RE)*(UEVY+VEVX)*PHIY(KI))
C
      R1(IY)      = R1(IY)  - WE*((UEV*VEVX+VEV*VEVY)*PHI(KI)         +
     *                            (-PEV+(2./RE)*VEVY)*PHIY(KI)        +
     *                            (1./RE)*(UEVY+VEVX)*PHIX(KI))
C
      DO 105 KJ   = 1, IQ
      KKI         = IV(KI)
      KKJ         = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*(PHI(KJ)*VEVX*PHI(KI)              +
     *                               (1./RE)*PHIY(KJ)*PHIX(KI))
      KKJ         = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*((UEV*PHIX(KJ)+VEV*PHIY(KJ)        +
     *                               VEVY*PHI(KJ))*PHI(KI)            +
     *                               (2./RE)*PHIY(KJ)*PHIY(KI)        +
     *                               (1./RE)*PHIX(KJ)*PHIX(KI))
C
```

```
      KKI         = IU(KI)
      KKJ         = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*((VEV*PHIY(KJ)+UEV*PHIX(KJ)      +
     *                              UEVX*PHI(KJ))*PHI(KI)           +
     *                              (2./RE)*PHIX(KJ)*PHIX(KI)       +
     *                              (1./RE)*PHIY(KJ)*PHIY(KI))
      KKJ         = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*(PHI(KJ)*UEVY*PHI(KI)            +
     *                             (1./RE)*PHIX(KJ)*PHIY(KI))
 105  CONTINUE
C                                                                    C
      DO 106 KJ   = 1, IL
      KKI         = IU(KI)
      KKJ         = IP(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WE*PHIP(KJ)*PHIX(KI)
C
      KKI         = IV(KI)
      R(KKI,KKJ) = R(KKI,KKJ) - WE*PHIY(KI)*PHIP(KJ)
 106  CONTINUE
C
 104  CONTINUE
C
C Compute -Rc
C
      DO 107 KI   = 1, IL
      IC          = NOPP(IABS(NOP(NELL,NNPRES(KI)))) + 2
      R1(IC)      = R1(IC) - WE*PHIP(KI)*(UEVX+VEVY)
      DO 107 KJ   = 1, IQ
      KKI         = IP(KI)
      KKJ         = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*PHIP(KI)*PHIX(KJ)
      KKJ         = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WE*PHIP(KI)*PHIY(KJ)
  107 CONTINUE
C
  103 CONTINUE
C
C Free boundary condition at the outflow
C
      IF (NEO(NELL) .EQ. 1) THEN
C
      DO 108 K    = 1, 3
      CALL GAUSCT
      C           = 1.
      E           = GPT(K)
      CALL TFUNC(C,E)
      CALL TPRES(C,E)
      CALL DER(C,E)
      UEVX        = FEVX(U,NELL,EEU)
      UEVY        = FEVY(U,NELL,EEU)
      VEVX        = FEVX(V,NELL,EEV)
      PEV         = FEVP(P,NELL,EEP)
      WT          = GWE(K)*YEDA
C                                                                    C
      DO 109 KI   = 7, 9
      IX          = NOPP(IABS(NOP(NELL,KI)))
      IY          = IX+1
      R1(IX)      = R1(IX) + WT*(-PEV+(2./RE)*UEVX)*PHI(KI)
      R1(IY)      = R1(IY) + WT*(1./RE)*(UEVY+VEVX)*PHI(KI)
C
      DO 110 KJ   = 1, IQ
      KKI         = IU(KI)
      KKJ         = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(2./RE)*PHIX(KJ)*PHI(KI)
C
      KKI         = IV(KI)
      KKJ         = IV(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(1./RE)*PHIX(KJ)*PHI(KI)
      KKJ         = IU(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) - WT*(1./RE)*PHIY(KJ)*PHI(KI)
C
 110  CONTINUE
C
      DO 111 KJ   = 1, IL
      KKI         = IU(KI)
      KKJ         = IP(KJ)
      R(KKI,KKJ) = R(KKI,KKJ) + WT*PHIP(KJ)*PHI(KI)
 111  CONTINUE
```

```
C
 109  CONTINUE
C
 108  CONTINUE
C
      END IF
C
C Transfer the Jacobian to the Frontal solver
C
      DO 112 I    = 1, 22
      DO 112 J    = 1, 22
      AA(J,I)     = R(J,I)
 112  CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE POSTPR
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      PARAMETER (NXNP=173000,NG=501,ND=65000,NL=17000)
C
      COMMON /XYWGT/  XWGHT(NG),YWGHT(NG),XAGHT(NG)
      COMMON /JACOB/  XC,XEDA,YC,YEDA,Y1,AY1,CX,CY,EX,EY
      COMMON /MESH/   NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /INOUT/  NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INOT/   TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /GSPNT/  GPX(9),GPY(9),GWEI(9),GPT(3),GWE(3)
      COMMON /FRON1/  NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/  R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /SHPFCT/ PHI(9),PHIX(9),PHIY(9),PHIC(9),PHIE(9),PHIP(4)
      COMMON /NODCOO/ XE(9),YE(9)
      COMMON /PAR/    RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /BDYEL/  NET(NL),NEO(NL),IQ,LB
C
C Compute streamlines for tecplot!
C
C At the entrance!
C
      IF (ICONT .GE. 2)
     *WRITE(NPP,393) TIME, NEY+1, NEX1A+1
      IF (ICONT .LE. 1)
     *WRITE(NPP,391) RE,   NEY+1, NEX1A+1
      DO 46 I = 1, 2*NEX1A+1, 2
      DO 46 J = 1+2*NEX1A-(I-1), NY+2*NEX1A-(I-1), 2
      LL      = NODNOR(I,J)
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
 46   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEX1A+1, NEX1A+1
      DO 47 I = 1, 1+2*NEX1A, 2
      DO 47 J = 1, 2*NEX1A+1, 2
      JJ      = 1+2*NEX1A-(I-1)-(J-1)
      IF (JJ .LE. 1  .AND.  I .GT. 1)  THEN
      L1      = NODNOR(I,1)
      WRITE(NPP,'(4F12.6)')  X(L1), Y(L1), U(L1), V(L1)
      ELSE
      LL      = NODNOR(I,JJ)
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
      END IF
 47   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEX1A+1, NEX1A+1
      DO 48 I = 1, 2*NEX1A+1, 2
      DO 48 J = 1, 2*NEX1A+1, 2
      IF (J .GT. 2*NEX1A-(I-1))  THEN
      L1      = NODNOR(I,NY+2*(2*NEX1A-(I-1)))
      WRITE(NPP,'(4F12.6)')  X(L1), Y(L1), U(L1), V(L1)
      ELSE
      LL      = NODNOR(I,J-1+NY+2*NEX1A-(I-1))
      WRITE(NPP,'(4F12.6)')  X(LL), Y(LL), U(LL), V(LL)
```

```
      END IF
 48   CONTINUE
C
C Upstream the obstacle!
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX1B+1
      DO 49 I = 1+2*NEX1A, NX1, 2
      DO 49 J = 1, NY, 2
      LL      = NODNOR(I,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 49   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX2+2
      DO 59 J = 1, NY, 2
      LL      = NODNOR(NX1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 59   CONTINUE
      DO 61 I = 1, 1+2*NEX2, 2
      IF (I .EQ. 1)  THEN
      DO 62 J = 1, NY, 2
      IF (J .EQ. 1)  THEN
      LL      = NODNOR(NX1,1)
      ELSE
      LL      = NODNOR(NX1,NY)+NY-2 + J-1
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 62   CONTINUE
      ELSE
      DO 63 J = 1, NY, 2
      LL      = NODNOR(NX1+I-1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 63   CONTINUE
      END IF
 61   CONTINUE
C
C Over the obstacle!
C
      NNN     = 2
      WRITE(NPP,*)
      WRITE(NPP,392) NEY2+1, NNN
      DO 50 I = 1, NNN
      IF (I .EQ. 1)  THEN
      DO 51 J = NY1, NY, 2
      LL      = NODNOR(NX2,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 51   CONTINUE
      ELSE
      DO 52 J = NY1, NY, 2
      LL      = NODNOR(NX2+2,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 52   CONTINUE
      END IF
 50   CONTINUE
C
C Downstream the obstacle!
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX3+2
      DO 53 I = 1, 1+2*NEX3, 2
      DO 53 J = 1, NY, 2
      LL      = NODNOR(NX2+1+I,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 53   CONTINUE
      DO 57 J = 1, NY, 2
      IF (J .LE. NY1-2)  THEN
      LL      = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      ELSE
      LL      = NODNOR(NX2+2+2*NEX3,J)
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 57   CONTINUE
C
      WRITE(NPP,*)
      WRITE(NPP,392) NEY+1, NEX4+1
      DO 71 I = 1, 1+2*NEX4, 2
      IF (I .EQ. 1)  THEN
```

```
      DO 72 J = 1, NY, 2
      IF (J .LE. NY1-2)  THEN
      LL       = NODNOR(NX2+2+2*NEX3,NY)+NY1-2 + J
      ELSE
      LL       = NODNOR(NX2+2+2*NEX3,J)
      END IF
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 72   CONTINUE
      ELSE
      DO 73 J = 1, NY, 2
      LL       = NODNOR(NX3+I-1,J)
      WRITE(NPP,'(4F12.6)') X(LL), Y(LL), U(LL), V(LL)
 73   CONTINUE
      END IF
 71   CONTINUE
C
      CLOSE(UNIT=NPP,STATUS='KEEP')
C
 392  FORMAT('ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
 391  FORMAT('TITLE = "Re=', F9.5, '", VARIABLES="XE","YE","UEV","VEV",'
     *  / 'ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
 393  FORMAT('TITLE = "t =', F9.2, '", VARIABLES="XE","YE","UEV","VEV",'
     *  / 'ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
C
      IQ           = 9
      EEE          = 0.
C
      WRITE(NLP,999)  RE, NEY1+NEY2+1, NEX2+NEX3+2
      DO 102 NELL = 1, NEY
      NEL1         = NELL + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 103 KI    = 1, 9
      LN           = IABS(NOP(NEL1,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
 103  CONTINUE
      PSIE         = 0.
      IF (NELL .EQ. 1) THEN
      C            = -1.
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX         = FEVX(V,NEL1,EEE)
      UEVY         = FEVY(U,NEL1,EEE)
      ZETA1        = VEVX-UEVY
      LN1          = IABS(NOP(NEL1,1))
      END IF
      C            = -1.
      E            =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX         = FEVX(V,NEL1,EEE)
      UEVY         = FEVY(U,NEL1,EEE)
      ZETA         = VEVX-UEVY
      LN           = IABS(NOP(NEL1,3))
      IF (NELL .EQ. 1)
     *WRITE(NLP,901) XE(1),YWGHT(1),ZETA1,P(LN1)
      WRITE(NLP,901) XE(3),YE(3),ZETA,P(LN)
 102  CONTINUE
C
      DO 101 I     = 1, NEX2
      DO 101 J     = 1, NEY
      NELL         = (I-1)*NEY+J + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 106 KI    = 1, 9
      LN           = IABS(NOP(NELL,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
 106  CONTINUE
      IF (J .EQ. 1) THEN
      C            =  1.
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX         = FEVX(V,NELL,EEE)
      UEVY         = FEVY(U,NELL,EEE)
      ZETA1        = VEVX-UEVY
      END IF
      C            =  1.
      E            =  1.
```

```
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX          = FEVX(V,NELL,EEE)
      UEVY          = FEVY(U,NELL,EEE)
      ZETA          = VEVX-UEVY
      L1            = IABS(NOP(NELL,7))
      L             = IABS(NOP(NELL,9))
      IF (J .EQ. 1)
     *WRITE(NLP,901) XE(7),YWGHT(1),ZETA1,P(L1)
      WRITE(NLP,901) XE(9),YE(9),ZETA,P(L)
  101 CONTINUE
C
      DO 302 NELL = 1, NEY
      NEL1          = NELL + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                     + NEY2+NEX2*NEY
      DO 303 KI     = 1, 9
      LN            = IABS(NOP(NEL1,KI))
      XE(KI)        = X(LN)
      YE(KI)        = Y(LN)
  303 CONTINUE
      PSIE          = 0.
      IF (NELL .EQ. 1) THEN
      C             = -1.
      E             = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX          = FEVX(V,NEL1,EEE)
      UEVY          = FEVY(U,NEL1,EEE)
      ZETA1         = VEVX-UEVY
      LN1           = IABS(NOP(NEL1,1))
      END IF
      C             = -1.
      E             =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX          = FEVX(V,NEL1,EEE)
      UEVY          = FEVY(U,NEL1,EEE)
      ZETA          = VEVX-UEVY
      LN            = IABS(NOP(NEL1,3))
      IF (NELL .EQ. 1)
     *WRITE(NLP,901) XE(1),YWGHT(1),ZETA1,P(LN1)
      WRITE(NLP,901) XE(3),YE(3),ZETA,P(LN)
  302 CONTINUE
C
      DO 301 I      = 1, NEX3
      DO 301 J      = 1, NEY
      NELL          = (I-1)*NEY+J + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                            + NEY2+NEX2*NEY
      DO 306 KI     = 1, 9
      LN            = IABS(NOP(NELL,KI))
      XE(KI)        = X(LN)
      YE(KI)        = Y(LN)
  306 CONTINUE
      IF (J .EQ. 1) THEN
      C             =  1.
      E             = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX          = FEVX(V,NELL,EEE)
      UEVY          = FEVY(U,NELL,EEE)
      ZETA1         = VEVX-UEVY
      END IF
      C             =  1.
      E             =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      VEVX          = FEVX(V,NELL,EEE)
      UEVY          = FEVY(U,NELL,EEE)
      ZETA          = VEVX-UEVY
      L1            = IABS(NOP(NELL,7))
      L             = IABS(NOP(NELL,9))
      IF (J .EQ. 1)
     *WRITE(NLP,901) XE(7),YWGHT(1),ZETA1,P(L1)
      WRITE(NLP,901) XE(9),YE(9),ZETA,P(L)
  301 CONTINUE
C
      DO 701 I      = 1, NEX1B
      NELL          = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+NEX1A*NEY
```

```
      DO 706 KI    = 1, 9
      LN           = IABS(NOP(NELL,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
  706 CONTINUE
      DO 716 J     = 1, 11
      C            = -1. + (J-1)*0.2
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY         = FEVY(U,NELL,EEE)
      XX           = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  716 CONTINUE
  701 CONTINUE
C
      NEL1         = 1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 503 KI    = 1, 9
      LN           = IABS(NOP(NEL1,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
  503 CONTINUE
      DO 513 J     = 1, 11
      C            = -1. + (J-1)*0.2
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY         = FEVY(U,NEL1,EEE)
      XX           = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  513 CONTINUE
C
      DO 501 I     = 1, NEX2
      NELL         = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
      DO 506 KI    = 1, 9
      LN           = IABS(NOP(NELL,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
  506 CONTINUE
      DO 516 J     = 1, 11
      C            = -1. + (J-1)*0.2
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY         = FEVY(U,NELL,EEE)
      XX           = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  516 CONTINUE
  501 CONTINUE
C
      NEL1         = 1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *               + NEY2+NEX2*NEY
      DO 603 KI    = 1, 9
      LN           = IABS(NOP(NEL1,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
  603 CONTINUE
      DO 613 J     = 1, 11
      C            = -1. + (J-1)*0.2
      E            = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY         = FEVY(U,NEL1,EEE)
      XX           = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  613 CONTINUE
C
      DO 601 I     = 1, NEX3
      NELL         = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                           + NEY2+NEX2*NEY
      DO 606 KI    = 1, 9
      LN           = IABS(NOP(NELL,KI))
      XE(KI)       = X(LN)
      YE(KI)       = Y(LN)
  606 CONTINUE
      DO 616 J     = 1, 11
      C            = -1. + (J-1)*0.2
      E            = -1.
```

```
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY          = FEVY(U,NELL,EEE)
      XX            = XE(1) + (C+1.)*(XE(7)-XE(1))/2.
      WRITE(NLI,901) XX, UEVY
  616 CONTINUE
  601 CONTINUE
C
      DO 801 I      = 1, NEX4
      NELL          = (I-1)*NEY+1 + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY
     *                            + (NEX2+NEX3+1)*NEY
      DO 806 KI     = 1, 9
      LN            = IABS(NOP(NELL,KI))
      XE(KI)        = X(LN)
      YE(KI)        = Y(LN)
  806 CONTINUE
      C             = -1.
      E             = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY          = FEVY(U,NELL,EEE)
      WRITE(NLI,901) XE(1), UEVY
  801 CONTINUE
C
      WRITE(NLI,'(A50)')  'Shear stress along upstream wall'
      DO 671 K      = 1, NEY1
      NELL          = K + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+1)*NEY+(NEX2-1)*NEY
      DO 676 KI     = 1, 9
      LN            = IABS(NOP(NELL,KI))
      XE(KI)        = X(LN)
      YE(KI)        = Y(LN)
  676 CONTINUE
      DO 686 J      = 1, 11
      E             = -1. + (J-1)*0.2
      C             =  1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY          = FEVY(U,NELL,EEE)
      UEVX          = FEVX(U,NELL,EEE)
      VEVX          = FEVX(V,NELL,EEE)
      VEVY          = FEVY(V,NELL,EEE)
      A             = 2.*UEVX
      B             = UEVY+VEVX
      D             = 2.*VEVY
      TAU           = B*(COS(ALPHA)**2.-SIN(ALPHA)**2.)+
     *                (D-A)*SIN(ALPHA)*COS(ALPHA)
      YY            = YE(7) + (E+1.)*(YE(9)-YE(7))/2.
      WRITE(NLI,901) YY/SIN(ALPHA), TAU
  686 CONTINUE
  671 CONTINUE
C
      WRITE(NLI,'(A50)')  'Shear stress along downstream wall'
      DO 771 K      = 1, NEY1
      NELL          = K + NEX1A*(NEX1A+1)+(NEX1A+NEX1B+NEX2+1)*NEY+NEY2
      DO 776 KI     = 1, 9
      LN            = IABS(NOP(NELL,KI))
      XE(KI)        = X(LN)
      YE(KI)        = Y(LN)
  776 CONTINUE
      DO 786 J      = 1, 11
      E             = -1. + (J-1)*0.2
      C             = -1.
      CALL TFUNC(C,E)
      CALL DER(C,E)
      UEVY          = FEVY(U,NELL,EEE)
      UEVX          = FEVX(U,NELL,EEE)
      VEVX          = FEVX(V,NELL,EEE)
      VEVY          = FEVY(V,NELL,EEE)
      A             = 2.*UEVX
      B             = UEVY+VEVX
      D             = 2.*VEVY
      TAU           = B*(COS(ALPHA)**2.-SIN(ALPHA)**2.)+
     *                (D-A)*SIN(ALPHA)*COS(ALPHA)
      YY            = YE(1) + (E+1.)*(YE(3)-YE(1))/2.
      WRITE(NLI,901) YY/SIN(ALPHA), TAU
  786 CONTINUE
  771 CONTINUE
C
```

```
 901   FORMAT(4F12.6)
 999   FORMAT('TITLE = "Re=', F9.5, '", VARIABLES="XE","YE","UEV","VEV",'
     *  / 'ZONE T="ZONE 1", Z=0, I=', I5, ', J=', I5, ', F=POINT')
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE UPDATE
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000,NCRIT1=939)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INOT/    TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /INOUT/   NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /MESH/    NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *                 NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
      COMMON /FRON1/   NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/   R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/   SK(NXNP),AA(25,25)
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
      COMMON /ER/      ERR
      COMMON /PAR/     RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /ITIT/    ITER
C
      ERR      = 0.
      DO 50 I = 1, NH
      CAN      = SK(NOPP(I))
      CAN2     = CAN*CAN
      U(I)     = U(I)+CAN
      IF (CAN2 .GT. ERR) ERR = CAN2
      CAN      = SK(NOPP(I)+1)
      CAN2     = CAN*CAN
      V(I)     = V(I)+CAN
      IF (CAN2 .GT. ERR) ERR = CAN2
      IF (MDF(I) .EQ. 3) THEN
      CAN      = SK(NOPP(I)+2)
      CAN2     = CAN*CAN
      IF (CAN2 .GT. ERR) ERR = CAN2
      P(I)     = P(I)+CAN
      END IF
   50 CONTINUE
      ERR      = SQRT(ERR)
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE ITRATE
C
      PARAMETER (NXNP=173000,ND=65000,NL=17000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /INDATB/ ERRMAX,IFPRNT,MAXITE
      COMMON /INOUT/   NCR,NLP,NCO,NSA,NPP,NLI
      COMMON /INOT/    TSTEP,TIME,TMAX,TOL,ICONT,ICHECK,NSTEP,IPRD,IPERT
      COMMON /PAR/     RE,YY1,PE1,FR1,UO,YD,PL
      COMMON /ER/      ERR
      COMMON /IITT/    ICOUNT
      COMMON /FRON2/   R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON1/   NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /ITIT/    ITER
      COMMON /XNDVAR/ X(ND),Y(ND),U(ND),V(ND),P(ND)
C
C Correct coding of NOP array before first entrance to FRONT
C
      NTRA    = 1
      ITER    = 0
C
      IF (ICOUNT .EQ. 1)  NTRA=0
C
 2    DO 5 I = 1, NP
      IF (NCOD(I) .EQ. 2) NCOD(I)=1
```

```
    5 CONTINUE
C
      CALL FRONT
C
      NTRA   = 0
      ITER   = ITER+1
C
      CALL UPDATE
      WRITE(*,'(A13,I5,1PE11.4)')  'ITER, ERR : ', ITER, ERR
C
      IF (ERR .GT. ERRMAX) THEN
      REWIND NLP
      WRITE(NLP,'(A10,I5)') 'ITER : ', ITER
      DO 51 I = 1, NH
      WRITE(NLP,'(I5,4F12.6,1PE19.10)')  I,X(I),Y(I),U(I),V(I),P(I)
   51 CONTINUE
      GO TO 50
      ELSE IF (ERR .LE. ERRMAX) THEN
      WRITE(*,*)   'CONVERGENCE'
      GO TO 200
      END IF
C
   50 IF(ITER .LT. MAXITE) THEN
      GO TO 2
      ELSE
      WRITE(*,*) 'MAXIMUM ITERATION NUMBER'
      STOP
      END IF
C
  200 CONTINUE
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE FRONT
C
      PARAMETER (NXNP=173000,NL=17000,ND=65000,NCRIT1=939,
     *           NMAX1=NCRIT1+22,NCRAY=13000000)
C
C IGAME1=2*NMAX+3
C
      IMPLICIT REAL*8 (A-H,O-Z)
C                                                                      C
      DIMENSION EQ(NMAX1,NMAX1),QQ(NMAX1),PVKOL(NMAX1),JMOD(NMAX1),
     *          LDEST(NCRIT1),KDEST(NCRIT1),NK(NCRIT1),LHED(NMAX1),
     *          KHED(NMAX1),KPIV(NMAX1),LPIV(NMAX1)
C
      COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/ SK(NXNP),AA(25,25)
      COMMON /FRON4/ CRAY(NCRAY)
      COMMON /MESH/  NEX1A,NEX1B,NEX1,NEX2,NEX3,NEX4,NEX,NEY1,NEY2,
     *               NEY,NX1A,NX1B,NX1,NX2,NX3,NXX1,NX,NY1,NY2,NY
C
      ICE  =  0
      ICE1 =  0
      ICE5 =  0
      MWGA =  0
      ND1  = 30
C
      OPEN(UNIT=ND1,FILE='front',FORM='UNFORMATTED',ACCESS='SEQUENTIAL',
     *     STATUS='NEW')
C
C     PREFRONT
C     FIND LAST APPEARANCE OF EACH NODE
C
      IF (NTRA .EQ. 1)  THEN
      NLAST          = 0
      DO 12 I        = 1, NHADD
      DO 8 N         = 1, NE
      LEND           = NBN(N)
      DO 4 L         = 1, LEND
      IF (NOP(N,L) .EQ. I)  THEN
      NLAST1         = N
      IF (NLAST .EQ. NLAST1)  THEN
      NERROR         = 1
```

```
      WRITE(*,416) NERROR, N
      STOP
      END IF
      NLAST          = N
      L1             = L
      END IF
    4 CONTINUE
    8 CONTINUE
      NOP(NLAST,L1) = -NOP(NLAST,L1)
      NLAST          = 0
   12 CONTINUE
      END IF
C
C     ASSEMBLY
C
      NTRA     = 0
      LCOL     = 0
      KROW     = 0
      DO 16 I = 1, NMAX1
      DO 16 J = 1, NMAX1
      EQ(J,I) = 0.
   16 CONTINUE
C
C     *****      INITIALIZING RHS     *****
C
      II       = NP+2
      DO 17 I = 1, II
      R1(I)    = 0.
   17 CONTINUE
C
      NEPR1    = NEX1A*(NEX1A+1)+(NEX1A+NEX1B)*NEY
      NEPR2    = NEPR1 + 4*NEY + 1
      NEPR3    = NEPR1 + NEY2*NEY + NEY2 + NEX3*NEY
      NEPR4    = NEPR3 + NEY1 + 3*NEY
C
      NELL     = 0
   18 NELL     = NELL+1
C
C      IF (NELL .GT. NEPR1  .AND.  NELL .LE. NEPR2  .OR.
C     *    NELL .GT. NEPR3  .AND.  NELL .LE. NEPR4)   THEN
      NCRIT    = 10*NEY+8
C      ELSE
C      NCRIT    = 5*(NEY+10)+8
C      END IF
      NMAX     = NCRIT+22
C
C      WRITE(*,'(A10,I5)') 'NELL = ', NELL
C      IF (NELL .EQ. NE) PRINT*, 'ASSEMBLY HAS FINISHED'
C
      CALL ABFIND
C
      N                  = NELL
      KC                 = 0
      IF (MWGA .NE. 0)  THEN
      NEND               = NBN(N)
      DO 20 I            = 1, NEND
      NK(I)              = NOP(N,I)
   20 CONTINUE
      ELSE
      JEND               = NBN(N)
      DO 22 J            = 1, JEND
      NN                 = NOP(N,J)
      M                  = IABS(NN)
      K                  = NOPP(M)
      IDF                = MDF(M)
      DO 22 L            = 1, IDF
      KC                 = KC+1
      II                 = K+L-1
      IF (NN .LT. 0) II = -II
      NK(KC)             = II
   22 CONTINUE
      END IF
C
C     SET UP HEADING VECTORS
C
      LEND        = NCN(N)
      DO 52 LK    = 1, LEND
      NODK        = NK(LK)
```

```
      IF (LCOL .EQ. 0)  GO TO 28
      DO 24 L     = 1, LCOL
      LL          = L
      IF (IABS(NODK) .EQ. IABS(LHED(L)))  THEN
      LDEST(LK)   = LL
      LHED(LL)    = NODK
      GO TO 36
      END IF
   24 CONTINUE
   28 LCOL        = LCOL+1
      LDEST(LK)   = LCOL
      LHED(LCOL)  = NODK
      GO TO 36
   36 IF (KROW .EQ. 0)  GO TO 44
      DO 42 K     = 1, KROW
      KK          = K
      IF (IABS(NODK) .EQ. IABS(KHED(K)))  THEN
      KDEST(LK)   = KK
      KHED(KK)    = NODK
      GO TO 52
      END IF
   42 CONTINUE
   44 KROW        = KROW+1
      KDEST(LK)   = KROW
      KHED(KROW)  = NODK
   52 CONTINUE
C
      IF (KROW .GT. NMAX  .AND.  LCOL .GT. NMAX)  THEN
      NERROR = 2
      WRITE (*,417)  NERROR
      STOP
      END IF
C
      LEND    = NCN(N)
      KEND    = NCN(N)
      DO 56 L = 1, LEND
      DO 56 K = 1, KEND
      EQ(KDEST(K),LDEST(L)) = EQ(KDEST(K),LDEST(L))+AA(K,L)
   56 CONTINUE
C
      IF (KROW .LT. NCRIT  .AND.  NELL .LT. NE)  GO TO 18
C
C     FIND OUT WHICH MATRIX ELEMENTS ARE FULLY SUMMED
C
   60 LC        = 0
      DO 64 L   = 1, LCOL
      IF (LHED(L) .LT. 0)  THEN
      LC        = LC+1
      LPIV(LC)  = L
      END IF
   64 CONTINUE
      IR        = 0
      KR        = 0
      DO 68 K   = 1, KROW
      KT        = KHED(K)
      IF (KT .LT. 0)  THEN
      KR        = KR+1
      KPIV(KR)  = K
      KRO       = IABS(KT)
      IF (NCOD(KRO) .EQ. 1)  THEN
      IR        = IR+1
      JMOD(IR)  = K
      NCOD(KRO) = 2
      R1(KRO)   = 0.
      END IF
      END IF
   68 CONTINUE
C
C     MODIFY EQUATIONS WITH APPLIED BOUNDARY CONDITIONS
C
      IF (IR .NE. 0)  THEN
      DO 70 IRR = 1, IR
      K         = JMOD(IRR)
      DO 69 L   = 1, LCOL
      IF(IABS(LHED(L)) .EQ. IABS(KHED(K)))  THEN
      EQ(K,L)   = 1.
      ELSE
      EQ(K,L)   = 0.
```

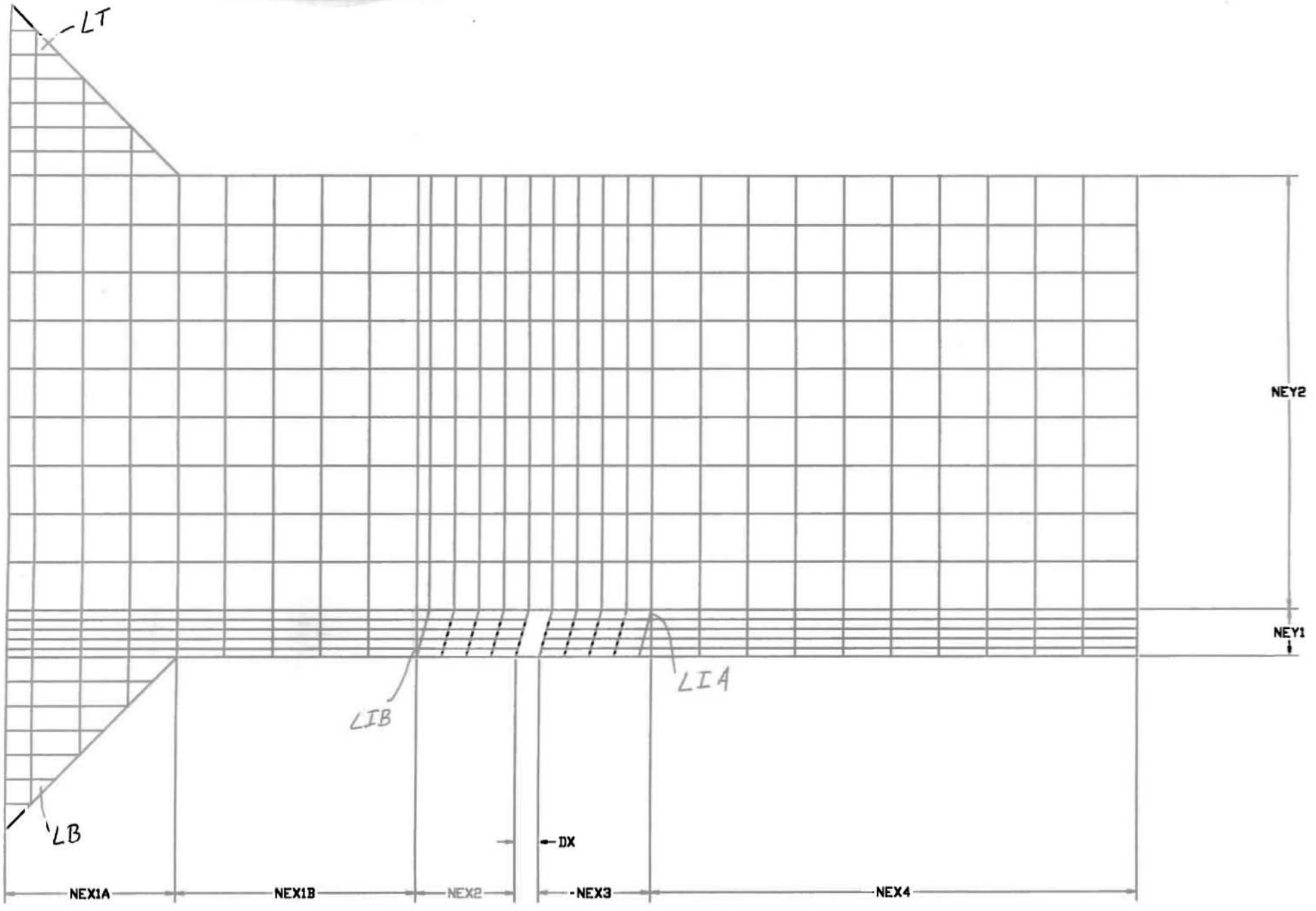
```
      END IF
   69 CONTINUE
   70 CONTINUE
      END IF
C
      IF (KR .LE. 0  .AND.  LC .LE. 0)  THEN
      NERROR = 3
      WRITE(*,418) NERROR, NELL
      STOP
      END IF
C
C     SEARCH FOR ABSOLUTE PIVOT
C
      KPIVRO  = KPIV(1)
      LPIVCO  = LPIV(1)
      PIVOT   = EQ(KPIVRO,LPIVCO)
      IF(ABS(PIVOT) .LE. 1.E-4)  THEN
      PIVOT   = 0.
      DO 76 L = 1, LC
      LPIVC   = LPIV(L)
      DO 74 K = 1, KR
      KPIVR   = KPIV(K)
      PIVA    = EQ(KPIVR,LPIVC)
      IF(ABS(PIVA) .GE. ABS(PIVOT))  THEN
      PIVOT   = PIVA
      LPIVCO  = LPIVC
      KPIVRO  = KPIVR
      END IF
   74 CONTINUE
   76 CONTINUE
      END IF
C
C     NORMALIZE PIVOT ARM
C
      KRO           =IABS(KHED(KPIVRO))
      IF (ABS(PIVOT) .LT. 1.E-12)  THEN
      WRITE(*,'(A10,1PE11.4)') 'PIVOT : ', PIVOT
      WRITE(*,476)
      END IF
      DO 80 L       = 1, LCOL
      QQ(L)         = EQ(KPIVRO,L)/PIVOT
   80 CONTINUE
      RHS           = R1(KRO)/PIVOT
      R1(KRO)       = RHS
      PVKOL(KPIVRO) = PIVOT
C
C     ELIMINATE THEN DELETE THE PIVOTAL ROW AND COLUMN
C
      KPIVP             = KPIVRO+1
      KPIVM             = KPIVRO-1
      LPIVP             = LPIVCO+1
      LPIVM             = LPIVCO-1
C
      DO 102 K          = 1, KPIVM
      PVKOL(K)          = EQ(K,LPIVCO)
  102 CONTINUE
      DO 103 K          = 1, KPIVM
      R1(IABS(KHED(K))) = R1(IABS(KHED(K)))-PVKOL(K)*RHS
  103 CONTINUE
C
      DO 101 K          = KPIVP, KROW
      PVKOL(K)          = EQ(K,LPIVCO)
  101 CONTINUE
      DO 191 K          = KPIVP, KROW
      R1(IABS(KHED(K))) = R1(IABS(KHED(K)))-PVKOL(K)*RHS
  191 CONTINUE
C
      DO 994 L          = 1, LPIVM
      DO 995 K          = 1, KPIVM
      EQ(K,L)           = EQ(K,L)-PVKOL(K)*QQ(L)
  995 CONTINUE
  994 CONTINUE
      DO 976 L          = 1, LPIVM
      DO 997 K          = KPIVP, KROW
      EQ(K-1,L)         = EQ(K,L)-PVKOL(K)*QQ(L)
  997 CONTINUE
  976 CONTINUE
C
```

```
      DO 194 L          = LPIVP, LCOL
      DO 195 K          = 1, KPIVM
      EQ(K,L-1)         = EQ(K,L)-PVKOL(K)*QQ(L)
 195  CONTINUE
 194  CONTINUE
      LMAX1             = LCOL
      KMAX1             = KROW
      DO 192 L          = LPIVP, LMAX1, 2
      EQ(KPIVRO,L)      = EQ(KPIVP,L+1)    - PVKOL(KPIVP)*QQ(L+1)
      DO 193 K          = KPIVP, KMAX1
      EQ(K-1,L-1)       = EQ(K,L)      - PVKOL(K)*QQ(L)
      EQ(K,L)           = EQ(K+1,L+1)  - PVKOL(K+1)*QQ(L+1)
 193  CONTINUE
 192  CONTINUE
C
C     WRITE PIVOTAL EQUATION ON DISC
C
      IF (ICE .LT. NCRAY  .AND.  ICE5 .EQ. 0)  THEN
      ICE2              = ICE
      DO 701 IG         = 1, LCOL
      CRAY(ICE+IG)      = LHED(IG)
 701  CONTINUE
      ICE               = ICE+LCOL
      DO 702 IG         = 1, LCOL
      CRAY(ICE+IG)      = QQ(IG)
 702  CONTINUE
      ICE               = ICE+LCOL
      CRAY(ICE+1)       = KRO
      CRAY(ICE+2)       = LPIVCO
      CRAY(ICE+3)       = LCOL
      ICE               = ICE+3
      IF (ICE .GT. NCRAY)  THEN
      ICE               = ICE2
      ICE5              = 1
      GO TO 771
      END IF
      END IF
C
 771  IF (ICE .GE. NCRAY  .OR.  ICE5 .EQ. 1)  THEN
      ICE1              = ICE1+1
      WRITE(ND1) KRO,LCOL,LPIVCO,(LHED(L),QQ(L),L=1,LCOL)
      END IF
C
      DO 129 K    = 1, KROW
      EQ(K,LCOL)  = 0.
  129 CONTINUE
      DO 130 L    = 1, LCOL
      EQ(KROW,L)  = 0.
  130 CONTINUE
C
C     REARRANGE THE HEADING VECTORS
C
      LCOL      = LCOL-1
      IF (LPIVCO .NE. LCOL+1)  THEN
      DO 132 L = LPIVCO, LCOL
      LHED(L)   = LHED(L+1)
  132 CONTINUE
      END IF
      KROW      = KROW-1
      IF (KPIVRO .NE. KROW+1)  THEN
      DO 140 K = KPIVRO,KROW
      KHED(K)   = KHED(K+1)
  140 CONTINUE
      END IF
C
C     DETERMINE WHETHER TO ASSEMBLE,ELIMINATE OR BACKSUBSTITUTE
C
      IF (KROW .GT. NCRIT) GO TO 60
      IF (NELL .LT. NE)    GO TO 18
      IF (KROW .GT. 1)     GO TO 60
      KPIVRO   = 1
      PIVOT    = EQ(1,1)
      KRO      = IABS(KHED(1))
      LPIVCO   = 1
      QQ(1)    = 1.
      R1(KRO) = R1(KRO)/PIVOT
C
      IF (ICE .GE. NCRAY  .OR.  ICE5 .EQ. 1)  THEN
```

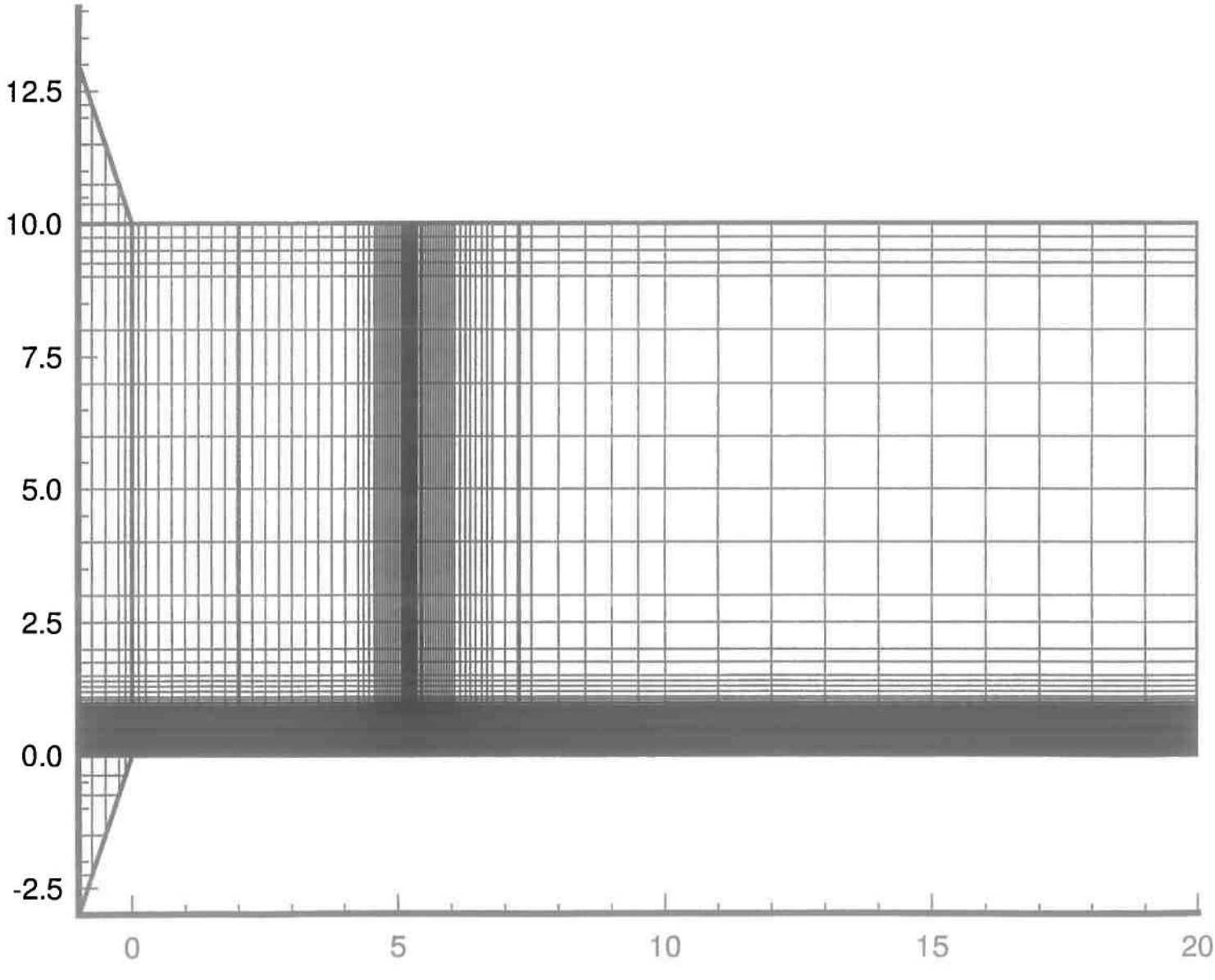
```
      ICE1              = ICE1+1
      WRITE(ND1) KRO,LCOL,LPIVCO,LHED(1),QQ(1)
      ELSE
      CRAY(ICE+1)       = LHED(1)
      CRAY(ICE+2)       = QQ(1)
      CRAY(ICE+3)       = KRO
      CRAY(ICE+4)       = LPIVCO
      CRAY(ICE+5)       = LCOL
      ICE               = ICE+5
      END IF
C
      CALL BACSUB(ICE,ICE1)
C
  416 FORMAT(/ ' NERROR = ' , I5 // I5,
     1' TH. ELEMENT HAS MORE THAN ONE NODE WITH THE',
     2' SAME NODAL NUMBER'
     3/)
  417 FORMAT(/ ' NERROR = ', I5 //
     1' THE DIFFERENCE NMAX-NCRIT IS NOT SUFFICIENTLY LARGE',
     2' TO PERMIT ASSEMBLY OF THE NEXT ELEMENT---',
     3' EITHER INCREASE NMAX OR LOWER NCRIT'
     4/)
  418 FORMAT(' NERROR = ', I5, ' NELL = ', I5 //
     1' THERE ARE NO MORE ROWS FULLY SUMMED, THIS WAY BE DUE TO---'
     2/ ' (1) INCORRECT CODING OF NOP OR NK ARRAYS'
     3/ ' (2) INCORRECT VALUE OF NCRIT. INCREASE NCRIT TO PERMIT'
     4/ ' WHOLE FRONT TO BE ASSEMBLED'
     5/)
  476 FORMAT(' WARNING-MATRIX SINGULAR OR ILL CONDITIONED')
C
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE BACSUB(ICE,ICE1)
C
C     BACK-SUBSTITUTION FOR FULL PIVOTING
C     BACK SUBSTITUTION
C
      PARAMETER (NXNP=173000,NL=17000,ND=65000,NCRIT1=939,
     *           NMAX1=NCRIT1+22,NCRAY=13000000)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION QQ(NMAX1),LHED(NMAX1)
C
      COMMON /FRON1/ NP,NH,NHADD,NE,NBN(NL),NCN(NL),ND1,NWGA,NELL,NTRA
      COMMON /FRON2/ R1(NXNP),NOP(NL,9),NOPP(ND),MDF(ND),NCOD(NXNP)
      COMMON /FRON3/ SK(NXNP),AA(25,25)
      COMMON /FRON4/ CRAY(NCRAY)
C
      ICE1        = ICE1+1
C
      DO 4 I      = 1, NP
      SK(I)       = 0.
    4 CONTINUE
C
      DO 32 IV    = 1, NP
C
      ICE1        = ICE1-1
      IF (ICE1 .GE. 1) THEN
      BACKSPACE ND1
      READ(ND1) KRO,LCOL,LPIVCO,(LHED(L),QQ(L),L=1,LCOL)
      BACKSPACE ND1
      ELSE
      LCOL        = CRAY(ICE)
      LPIVCO      = CRAY(ICE-1)
      KRO         = CRAY(ICE-2)
      ICE         = ICE-3
      DO 701 I1   = ICE, ICE-(LCOL-1), -1
      IG          = LCOL-(ICE-I1)
      QQ(IG)      = CRAY(I1)
  701 CONTINUE
      ICE         = ICE-LCOL
      DO 702 I1   = ICE, ICE-(LCOL-1), -1
      IG          = LCOL-(ICE-I1)
      LHED(IG)    = CRAY(I1)
```

```
 702  CONTINUE
      ICE         = ICE-LCOL
      END IF
C
      LCO         = IABS(LHED(LPIVCO))
C
      IF (NCOD(LCO) .LE. 0)  THEN
      GASH        = 0.
      QQ(LPIVCO)  = 0.
      DO 16 L     = 1, LCOL
      GASH        = GASH-QQ(L)*SK(IABS(LHED(L)))
   16 CONTINUE
      SK(LCO)     = R1(KRO)+GASH
      ELSE
      NCOD(LCO)   = 1
      END IF
C
   32 CONTINUE
C
      CLOSE(UNIT=ND1,STATUS='DELETE')
C
      RETURN
      END
```

```
MAXITE  ERRMAX  IFPRNT  RE    YY1
20      1.E-5   0       0.02  3.
NEX1A   NEX1B   NEX2    NEX3 NEX4   NEY1   NEY2
5       9       44      38   16     72     21
ALPHA   DX      XL
75.     0.01    1.
XWGHT(I)
-1. -0.75  -0.5 -0.25 -0.125 0.
0.125 0.25 0.5 0.75 1. 1.25 1.5 1.75 2.
2.25 2.5 2.75  3. 3.25 3.5 3.75 4. 4.1 4.2 4.3 4.35 4.4 4.45 4.5 4.55 4.6 4.65
4.7 4.75 4.8 4.825 4.85 4.875 4.9 4.91 4.92 4.93 4.94 4.95 4.96 4.97 4.98 4.99
4.991 4.992 4.993 4.994 4.995 4.996 4.997 4.998 4.999 5.
5.01 5.011 5.012 5.013 5.014 5.015 5.016 5.017 5.018 5.019 5.02
5.045 5.06 5.08 5.1 5.15 5.175 5.2 5.25 5.3 5.35 5.4 5.45 5.5 5.55 5.6 5.65 5.7
5.75 5.8 5.9 6. 6.1 6.2 6.3 6.4 6.5 6.75 7. 7.2588
7.5 8. 8.5 9. 9.5 10.
11. 12. 13. 14. 15. 16. 17. 18. 19. 20.
YWGHT(J)
0. 0.001   0.002 0.003  0.004 0.005  0.006 0.007 0.008 0.009 0.01 0.0125 0.015
   0.0175  0.02  0.0225 0.025 0.0275 0.03  0.035 0.04  0.045 0.05 0.055  0.06
   0.065   0.07  0.075  0.08  0.085  0.09  0.095 0.1   0.11  0.12 0.13   0.14
   0.15    0.16  0.17   0.18  0.19   0.2   0.225 0.25  0.275 0.3  0.325  0.35
   0.375   0.4   0.425  0.45  0.475  0.5   0.525 0.55  0.575 0.6  0.625  0.65
   0.675   0.7   0.725  0.75  0.775  0.8   0.825 0.85  0.875 0.9  0.925  0.9659
   1.      1.05  1.1    1.2   1.3    1.4   1.5   1.75  2.    2.5  3.     4.
   5.      6.    7.     8.    9.     9.25  9.5   9.75 10.
```


LT
LIB
LIA
LB
NEY2
NEY1
DX
NEX1A
NEX1B
NEX2
NEX3
NEX4

ΔΟΜΗ ΠΛΕΓΜΑΤΟΣ



ΛΕΠΤΟΜΕΡΕΙΑ ΠΛΕΓΜΑΤΟΣ ΓΥΡΩ ΑΠΟ ΤΟΝ ΚΕΚΛΙΜΕΝΟ ΦΡΑΚΤΗ