**Indice**

1. Primo quesito, problema inverso 2

2. Discussione analitica 2

3. L’algoritmo risolutivo 9

4. L’output 11

5. Secondo quesito problema diretto 13

6. Discussione analitica 14

7. L’algoritmo risolutivo 15

8. Programma principale per il primo quesito 18

9. Modulo per il primo quesito 23

10. Programma principale per il secondo quesito 28

11. Modulo per il secondo quesito 31

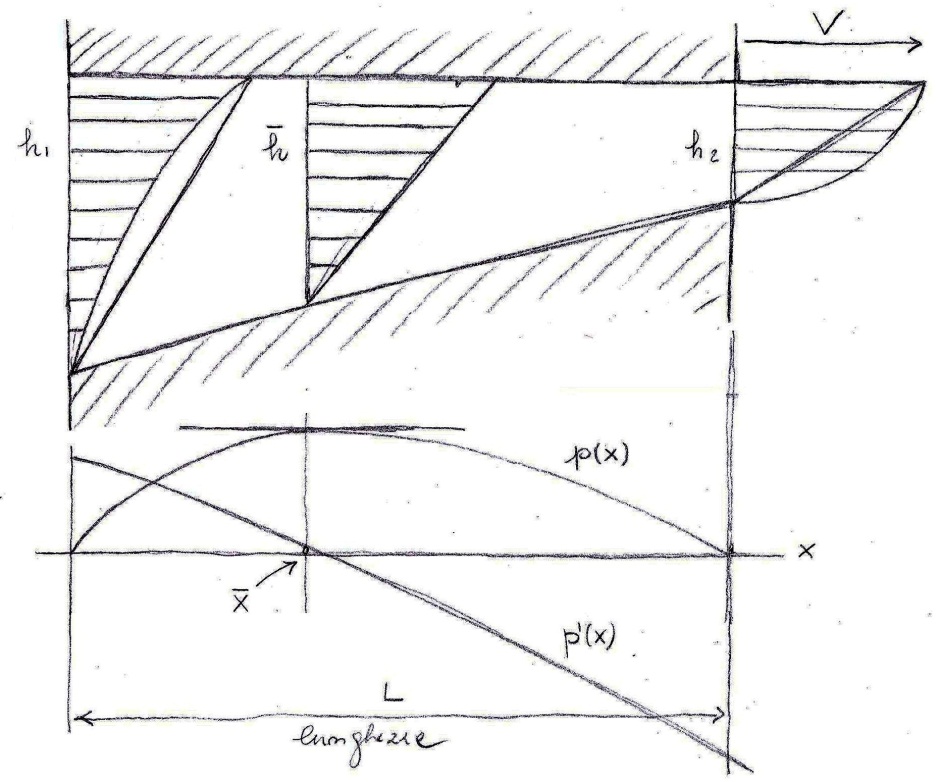
**Lubrificazione idrdinamica**

**1. Primo quesito, problema inverso.** Abbiamo una coppia prismatica lubrificata -con meato a spessore linearmente decrescente- che verifica i seguenti parametri:

1.1)

essendo il significato di ciascuno illustrato in figura. Sia inoltre assegnata la viscosità del lubrificante (alla temperatura di funzionamento):

1.12)



Allora si chiede di determinare:

1.13) il diagramma dei gradienti delle pressioni;

1.14) il diagramma delle pressioni per via analitica;

1.15) il diagramma delle pressioni per via numerica;

1.16) la pressione massima;

1.17) la forza di sostentamento per via analitica;

1.18) la forza di sostentamento per via numerica;

1.19) il punto di applicazione di ;

1.20) il coefficiente di attrito mediato ;

1.21) il diagramma delle velocità in , , .

**2. Discussione analitica.** In un generico meato in condizioni stazionarie si dimostra per la portata (per unità di larghezza del meato) l’espressione

2.1)

Indicando con il valore dell’altezza per la quale si annulla il gradiente della pressione si ha in particolare

2.2)

che sostituito nella **2.1** porge

2.3)

Poiché deve[[1]](#footnote-1) risultare possiamo scrivere:

da cui si deduce per confronto

2.4)

Nel caso particolare di un meato a spessore linearmente decrescente la funzione si scrive poi

2.5)

e dunque, sostituendo nella **2.4** e integrando (sono integrali elementari) si perviene alla espressione

2.6)

che sostituita nella **2.5** fornisce la corrispondente ascissa

2.7)

Sostituendo la **2.6** nella **2.3** abbiamo il gradiente delle pressioni

2.8)

Per integrare il gradiente conviene ricondursi alla variabile , considerando che

2.9)

Ciò posto la **2.8** si scrive

2.10)

L’integrazione a questo punto è piuttosto semplice:

Si ottiene dunque la pressione

2.11)

Introdotto poi il coefficiente caratteristico

2.12)

si ha

2.13)

Per il calcolo della forza di sostentamento integrando la **2.11** si ha

E dunque

2.14)

Introdotto poi il coefficiente caratteristico si ha

2.15)

Calcolo ora, attraverso la **2.11**, il punto di applicazione della risultante delle forze di pressione tale per cui la sollecitazione ridotta sia equivalente a quella data.

Introducendo poi abbiamo

Sostituendo la **2.14** si ha allora

Quindi ho trovato

2.16)

Per procedere con l’analisi e pervenire al coefficiente di attrito mediato abbiamo bisogno del campo delle velocità. Dalla teoria generale dei meati lubrificati abbiamo

2.17)

che inserendo la **2.8** si scrive

2.18)

Considerando la legge di Petroff possiamo allora scrivere

ovvero

2.19)

Gli sforzi che il fluido esercita sulla parete superiore allora si ottengono ponendo e sono

2.20)

Per ricavare la forza che complessivamente il fluido esercita sulla parete mobile, per unità di larghezza, dobbiamo integrare sulla lunghezza del meato:

Sostituendo poi il valore **2.6** si conclude

2.21)

Introducendo poi il parametro caratteristico abbiamo

2.22)

È immediato allora il calcolo del coefficiente di attrito mediato, che indichiamo , come rapporto . In realtà però

Dunque

2.23)

Posso tuttavia scrivere anche

Per cui si ha

2.24)

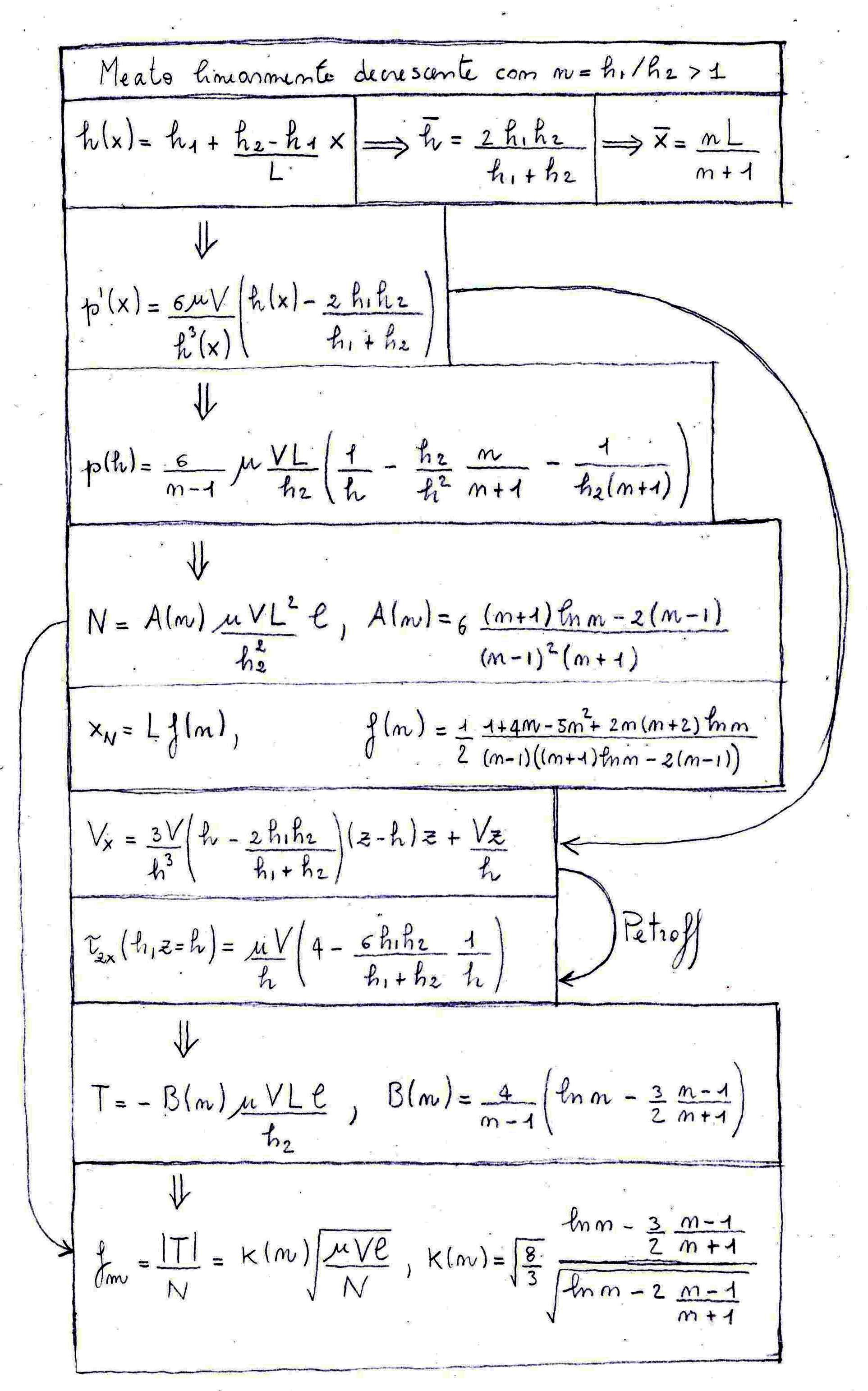
avendo posto

2.25)

Riassumo in tabella l’insieme di equazioni che risolvono analiticamente il problema del meato linearmente decrescente.

|  |  |  |
| --- | --- | --- |
| 2.3) |  | gradiente della pressione |
| 2.13) |  | pressione |
| 2.7) |  | punto di massima pressione |
| 2.6) |  | altezza di massima pressione |
| 2.17) |  | velocità |
| 2.15) |  | sostentamento |
| 2.16) |  | punto di applicazione del sostentamento |
| 2.24) |  | coef. di attrito mediato |

Nella seguente figura è invece riassunto il flusso delle deduzioni.



**3. L’algoritmo risolutivo.** Il programma che risolve questo quesito è costituito dalle seguenti unità:

* l’unità chiamante *main\_ese\_sette*, la quale si occupa di risolvere la parte analitica, ovvero di calcolare per via analitica; per il resto utilizza le subroutine di modulo;
* il modulo *mod\_ese\_sette*, il quale contiene le costanti fisiche e geometriche del problema, oltre alle seguenti subroutine:
  + *diagramma\_p\_primo*, il quale si occupa di disegnare il grafico del gradiente della pressione;
  + *diagramma\_p*, il quale si occupa di disegnare il diagramma della pressione, tanto quella esatta che quella calcolata per via numerica;
  + *diagramma\_v*, che disegna i grafici delle velocità in funzione di , nei tre punti del meato di ascisse ;
  + *cavalieri* che integra col metodo Cavalieri-Simpson le funzioni inviategli, ed è utilizzata per calcolare a partire da , per calcolare a partire da e per calcolare a partire da .

Segue il diagramma di flusso.

invoca *diagramma\_p\_primo* per disegnare il grafico del gradiente esatto

*diagramma\_p\_primo*

Inizio

L’unità chiamante (*main\_ese\_sette*) calcola

* il gradiente della pressione esatto
* la pressione esatta
* il sostentamento esatto
* il punto esatto

invoca *diagramma\_p* per disegnare il grafico della pressione esatta

*diagramma\_p*

invoca *cavalieri* per calcolare la pressione approssimata

*cavalieri*

invoca *diagramma\_p* per disegnare il grafico della pressione approssimata

*diagramma\_p*

invoca *cavalieri* per calcolare il sostentamento approssimato, lo mette in un array di cui ci interessa l’ultimo valore

*cavalieri*

invoca *cavalieri* per calcolare approssimato, integrando la funzione in ; risolvendo poi l’equazione otteniamo ; utilizza approssimata e esatta

*cavalieri*

calcola la velocità nei punti e la mette in tre array monodimensionali

invoca *diagramma\_v* per ciascuno dei tre array delle velocità

*diagramma\_v*

calcola il massimo della pressione esatta e il massimo della pressione approssimata utilizzando MAXVA(array), funzione intrinseca del Fortran

scrive sullo schermo

* la pressione massima esatta;
* la pressione massima approssimata;
* il sostentamento esatto
* il suo punto di applicazione
* il sostentamento approssimato
* il suo punto di applicazione
* il coefficiente di attrito mediato

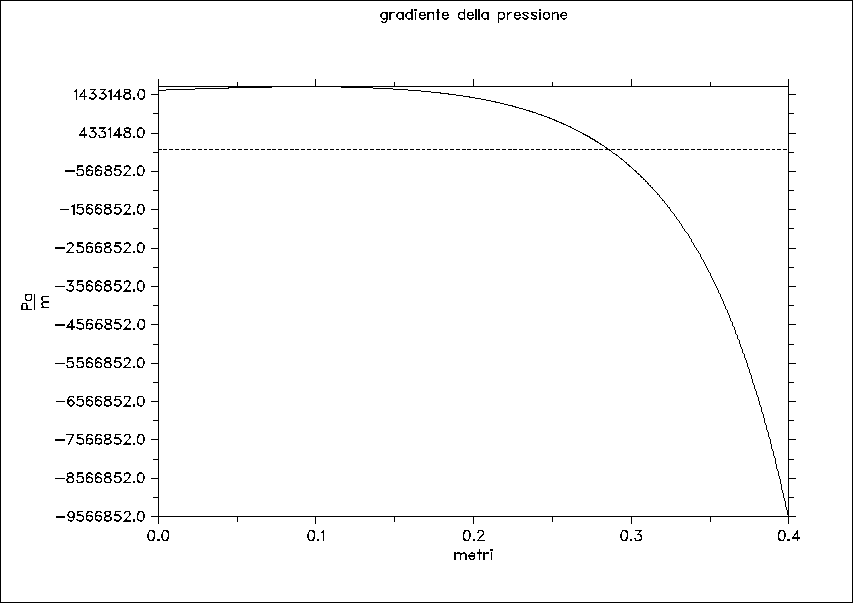
calcola il coefficiente di attrito mediato con la formula analitica fornita dalla teoria

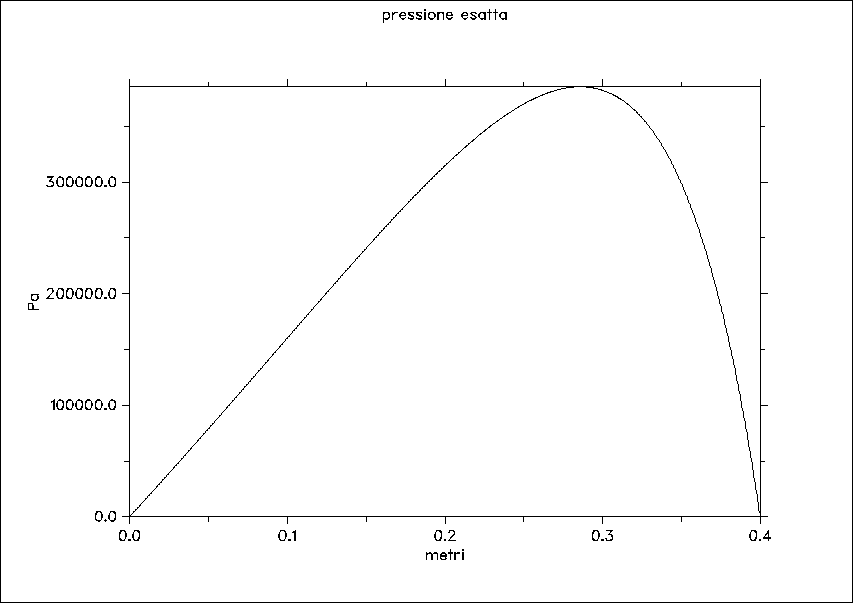
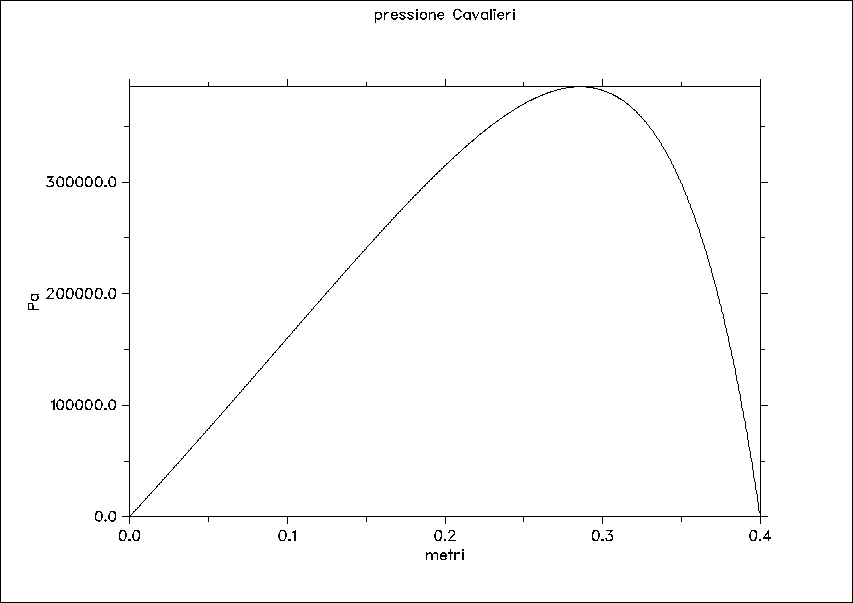
fine

**4. L’output.** Riporto gli output numerici e grafici del programma. I primi sono riassunti nella tabella seguente.

|  |  |  |
| --- | --- | --- |
|  | soluzione esatta | soluzione numerica |
| pressione massima | 385714.13 Pa | 385711.34 Pa |
| sostentamento | 94636.664 N/m | 94634.51 N/m |
| punto di applicazione | 0.23607785 m | 0.2360758 m |
| coef. di attrito mediato | 0.0018491517 | |

Il diagramma del gradiente della pressione è riportato nel seguito, insieme ai diagrammi della pressione calcolata in modo analitico e alla sua approssimazione ottenuta per integrazione numerica.



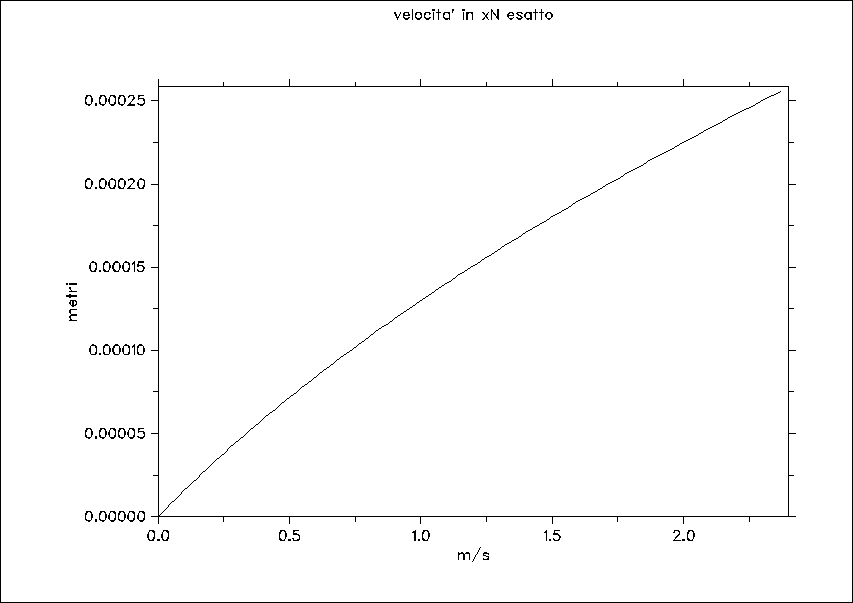


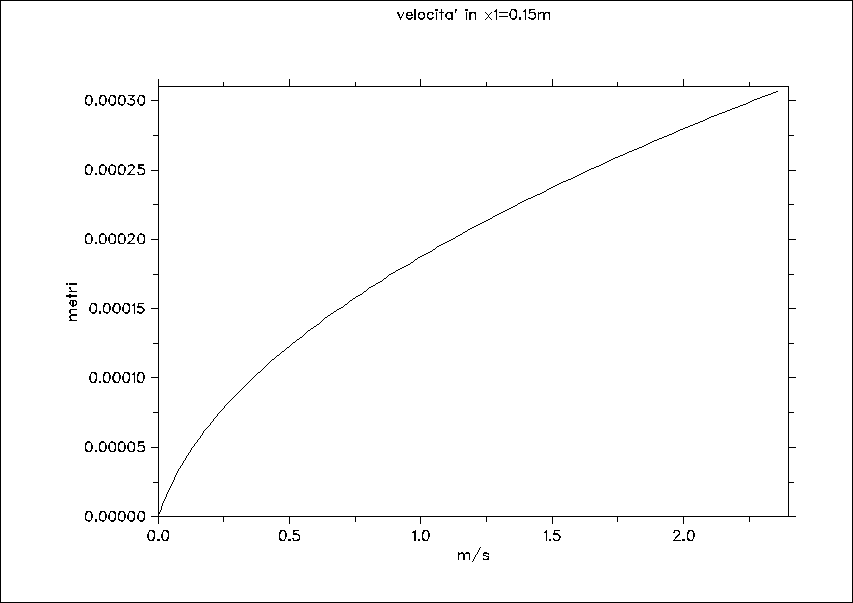
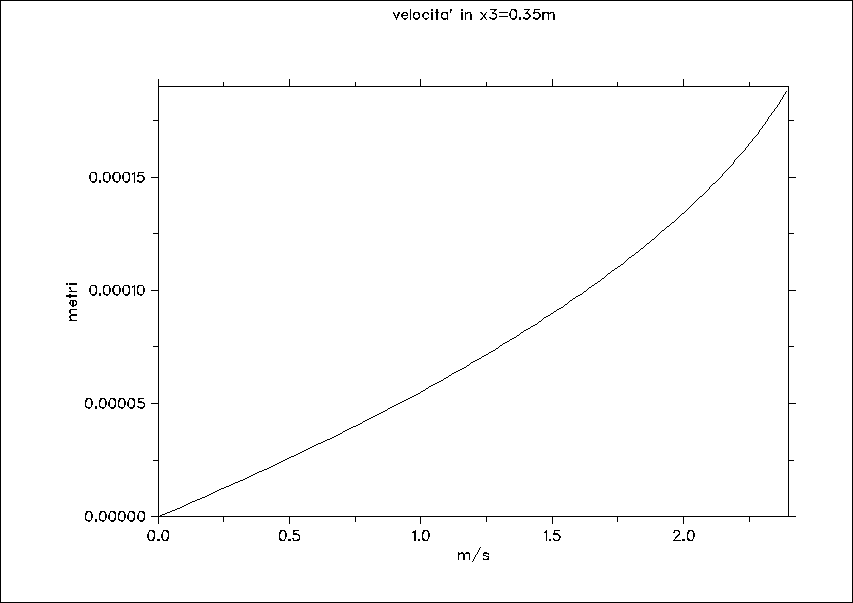
Riporto inoltre i diagrammi delle velocità in funzione di in corrispondenza dei tre punti richiesti

4.1)

segnalando che le tre altezze del meato ad essi relative sono rispettivamente

4.2)

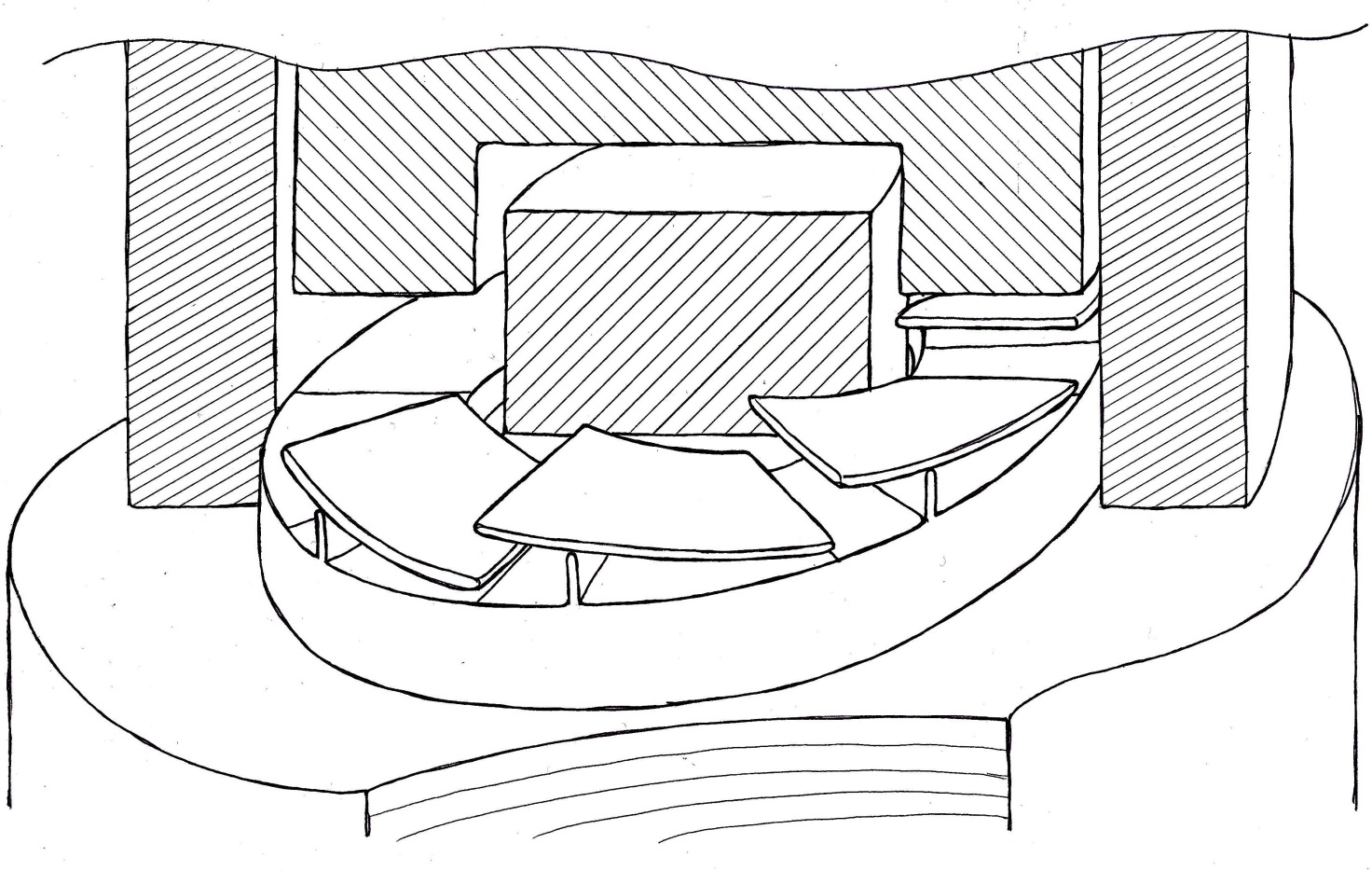




**5. Secondo quesito, problema diretto.** Abbiamo un cuscinetto Michell a pattini auto-orientabili per il quale si abbiano i seguenti dati

5.1)

essendo il il carico spingente, la velocità di rotazione dell’albero e la viscosità (alla temperatura di funzionamento). Il significato degli altri parametri è spiegato attraverso la figura.



Si chiede dunque di risolvere il problema diretto ovvero di ricavare

5.2) le altezze dei pattini;

5.3) il coefficiente di attrito mediato .

5.4) la pressione media.

Si precisa che il cuscinetto presenta otto pattini uguali.

**6. Discussione analitica.** Il cuscinetto Michell a pattini auto-orientabili si può considerare come una successione di meati ad altezza decrescente, tanti quanti sono i pattini. Dunque per ottenere la soluzione del problema si procede come segue.

* il punto di applicazione di ciascuna forza di sostentamento è noto, cadendo sul perno[[2]](#footnote-2); dunque la **2.16** porge , come soluzione dell’equazione trascendente

6.1)

* la forza di sostentamento offerta da ciascun pattino è nota e vale , sostituendola nella **2.15** ricavo attraverso la

6.2)

* da e si ricava ;
* si ricava il coefficiente di attrito mediato usando direttamente la **2.14**.

**7. L’algoritmo risolutivo.** L’unica difficoltà risiede nella soluzione della equazione non lineare **6.1** per la quale si può adottare il metodo di Newton-Raphson. Per applicare questo metodo definisco la funzione

7.1)

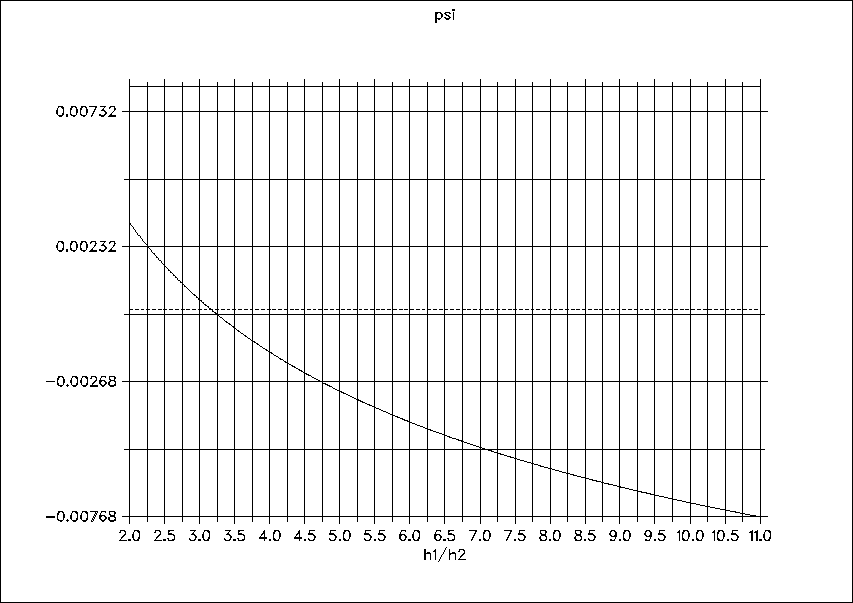
e ne calcolo la derivata in . Allora l’algoritmo di Newton-Raphson assume la forma

7.4)

e un possibile innesco, dato il significato fisico di , potrebbe essere . Poiché la derivata risulta macchinosa da calcolare e da scrivere nel codice, si può approssimare con il rapporto incrementale, ponendo ad esempio

7.5)

Nel codice ho richiesto di plottare la curva di , così da avere una verifica al risultato ottenuto col metodo di Newton-Raphson; la curva che si ottiene è riportata in figura, e dalla sua lettura è possibile avere una buona stima del valore di .



Il diagramma di flusso relativo al metodo di Newton-Raphson è indicato in figura.

*resto <=0.001*

*falso*

*vero*

la funzione *psi* calcola

Descrivo ora il codice nel suo complesso. Esso è costituito dalle seguenti unità:

* l’unità chiamante *main\_ese\_sette\_due*, la quale si occupa di calcolare per via analitica; calcola inoltre attraverso il metodo di Newton-Raphson;
* il modulo *mod\_ese\_sette\_due*, il quale contiene le costanti fisiche e geometriche del problema, oltre alle seguenti subroutine:
  + *diagramma\_p\_primo*, il quale si occupa di disegnare il grafico del gradiente della pressione;
  + *diagramma\_p*, il quale si occupa di disegnare il diagramma della pressione, tanto quella esatta che quella calcolata per via numerica;
  + *diagramma\_psi*, che disegna il grafico della funzione *psi* la cui radice è ;
  + *cavalieri* che integra col metodo Cavalieri-Simpson la pressione, calcolando così il sostentamento dovuto a un pattino, da confrontare con il carico assegnato, per verifica;
  + *psi*, funzione che calcola il valore della funzione *psi* in corrispondenza del valore *n* inviatogli.

Il diagramma di flusso complessivo è indicato in figura.

Inizio

invoca *diagramma\_p* per disegnare il grafico della pressione

*diagramma\_p*

invoca *diagramma\_p\_primo* per disegnare il grafico del gradiente

*diagramma\_p\_primo*

invoca *cavalieri* per calcolare il sostentamento approssimato, lo mette in un array di cui ci interessa l’ultimo valore

*cavalieri*

calcola il coefficiente caratteristico cercando lo zero (con Newton-Raphson) della funzione **7.1**.

* calcola utilizzando la **6.2**
* calcola

L’unità chiamante (*main\_ese\_sette\_due*) calcola

* il gradiente della pressione
* la pressione
* la pressione massima
* la pressione media
* il coefficiente di attrito mediato

invoca *diagramma\_psi* per disegnare il grafico della funzione *psi*

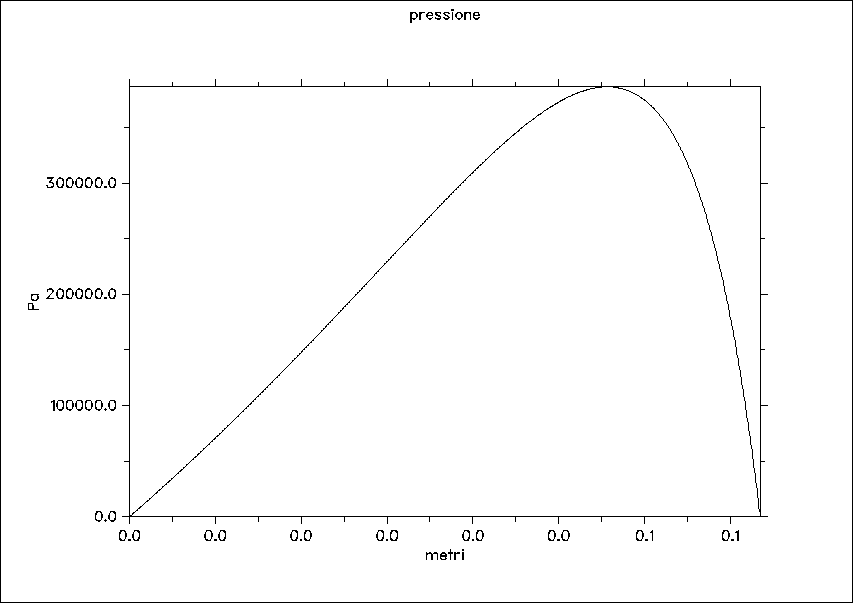
*diagramma\_psi*

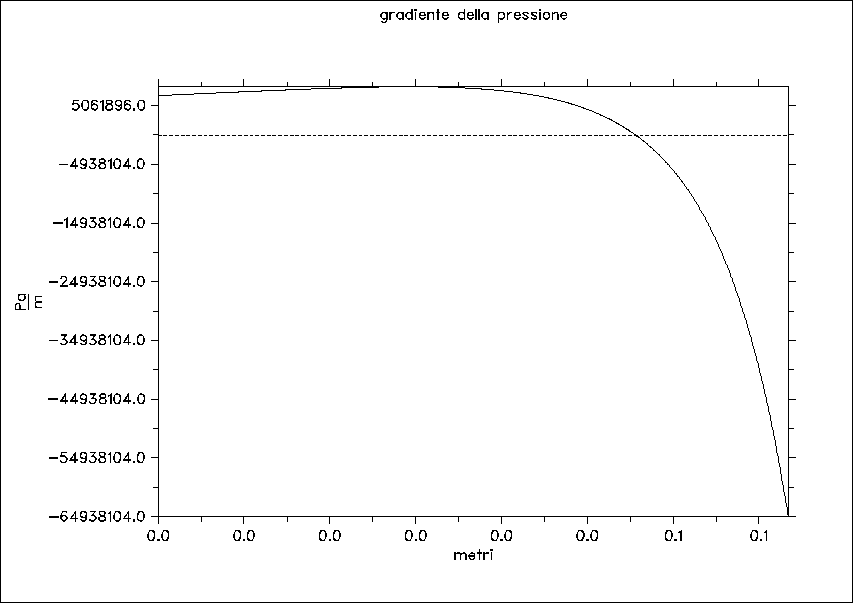
scrive sullo schermo

* la pressione massima
* la pressione media
* il sostentamento dato
* il sostentamento approssimato
* il coefficiente di attrito mediato

fine

Seguono i grafici della pressione e del gradiente per il singolo pattino:





Gli output numerici del programma sono riassunti nella tabella seguente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 386532.94 | 226757.39 | 16666.668 | 3.119554 | 0.007603118 |
|  | |  |  |  |
| 16666.5 | 0.00036638076 |
|  |  |
| 0.00011744652 |

**8. Unità chiamante per il primo quesito.** Riporto il codice in Fortran.

!main program per la settima esercitazione di meccanica apl.

!utilizza il modulo mod\_ese\_sette

!25/11/2012

PROGRAM main\_ese\_sette

USE DISLIN !libreria grafica

USE mod\_ese\_sette !modulo con costanti e procedure

!sezione dichiarativa

IMPLICIT NONE

!dichiaro alcune variabili utilizzate

REAL::x1=0.15 !punto del meato dove si calcola v

REAL::x2 !idem

REAL::x3=0.35 !idem

REAL::h\_x1, h\_x2, h\_x3

INTEGER::i,j !indici dei cicli

CHARACTER(len=10):: chiusura !serve per chiudere il programma

CHARACTER(len=25):: nome !si usa per il titolo dei grafici

CHARACTER(len=15):: unita !si usa per le unita' di misura dei grafici

!dichiaro gli array di tutte le funzioni coinvolte

REAL,DIMENSION(1:inter):: h !altezza del meato

REAL,DIMENSION(1:inter):: x !array delle ascisse

REAL,DIMENSION(1:inter):: p\_primo !derivata della pressione

REAL,DIMENSION(1:inter):: p\_esatta !pressione esatta

REAL,DIMENSION(1:inter):: p\_cavalieri !pressione ottenuta per via numerica

REAL,DIMENSION(1:inter):: integrando !serve a calcolare xN-cavalieri

REAL,DIMENSION(1:inter):: primit !idem

!dichiaro le variabili relative al sostentamento

REAL:: N\_esatta !N esatta

REAL,DIMENSION(1:inter)::N\_cavalieri!N ottenuta per via numerica

REAL::xN\_esatto !punto di applicazione di N esatto

REAL::xN\_cavalieri !punto di applicazione di N per via numerica

REAL::cost\_N\_esatta !costante moltiplicativa di N\_esatta

!dichiaro le variabili relative alla pressione

REAL::p\_max\_esatta !la pressione esatta massima

REAL::p\_max\_cavalieri !la pressione numerica massima

REAL::cost\_p\_esatta !costante moltiplicativa di p\_esatta

REAL::p\_primo\_x1

REAL::p\_primo\_x2

REAL::p\_primo\_x3

!dichiaro le variabili relative all'asse z e alla velocita'

REAL,DIMENSION(1:inter2)::z !array delle z

REAL:: delta\_z\_x1, delta\_z\_x2, delta\_z\_x3 !incrementi

REAL,DIMENSION(1:inter2):: v\_x1 !velocita' in x\_1

REAL,DIMENSION(1:inter2):: v\_x2 !velocita' in x\_2

REAL,DIMENSION(1:inter2):: v\_x3 !velocita' in x\_3

!dichiaro le variabili relative al coefficiente di attrito

REAL::fm !coefficiente di attrito mediato

REAL::kn !parte di fm

REAL::cost\_fm !idem

!altre variabili

REAL::delta\_x !incremento dell ascisse

REAL::h\_max !altezza in cui p è max

REAL::x\_max !ascissa in cui p è max

REAL::n !coefficente caratteristico

!sezione esecutiva

!alcune operazioni preliminari

delta\_x = L/inter !incremento dell ascisse

h\_max = (2.\*h1\*h2)/(h1+h2) !altezza in cui p e' max

x\_max = L\*h1/(h1+h2) !ascissa in cui p e' max

n = h1/h2 !coefficente caratteristico

!dichiaro l'array delle ascisse

x(1) = 0.

ciclo\_x: DO i=2,inter,1

x(i) = x(i-1) + delta\_x

END DO ciclo\_x

!dichiaro l'array delle z

z(1) = 0.

ciclo\_z: DO i=2,inter2,1

z(i) = z(i-1) + delta\_x

END DO ciclo\_z

!calcolo l'altezza del meato

ciclo\_h: DO i=1,inter,1

h(i) = h1 + (h2-h1)\*x(i)/L

END DO ciclo\_h

!calcolo la derivata della pressione

ciclo\_p\_primo: DO i=1,inter,1

p\_primo(i) = (6\*mu\*V)\*(h(i)-h\_max)/(h(i)\*\*3)

END DO ciclo\_p\_primo

!calcolo la pressione esatta

cost\_p\_esatta = (6\*mu\*V\*L)/((n-1)\*h2)

ciclo\_p: DO i=1,inter,1

p\_esatta(i) = cost\_p\_esatta\*((1/h(i))-((h2/(h(i)\*\*2))\*(n/(n+1)))-(1/(h2\*(n+1))))

END DO ciclo\_p

!calcolo la pressione massima esatta

p\_max\_esatta = MAXVAL(p\_esatta)

!calcolo il sostentamento esatto

cost\_N\_esatta = (6\*mu\*V\*(L\*\*2))/((h2\*\*2)\*((n-1)\*\*2)\*(n+1))

N\_esatta = cost\_N\_esatta\*(((n+1)\*LOG(n))-2\*(n-1))

!calcolo xN esatto

xN\_esatto = L\*((2\*n\*(n+2)\*LOG(n))-(5\*(n\*\*2))+4\*n+1)/(2\*(n-1)\*((n+1)\*LOG(n)-2\*(n-1)))

CALL BMPMOD (300,'inch','resolution') !fisso risoluzione dell'immagine in formato .bmp

!traccio il diagramma del gradiente della pressione

nome = "gradiente della pressione"

unita = "\frac{Pa}{m} "

CALL diagramma\_p\_primo (p\_primo, x, nome, unita)

!traccio il diagramma della pressione esatta

nome = "pressione esatta "

unita= "Pa "

CALL diagramma\_p (p\_esatta, x, nome, unita)

!calcola la pressione col metodo cavalieri

CALL cavalieri (p\_primo, p\_cavalieri, p\_esatta(1), p\_esatta(2))

!traccio il diagramma della pressione secondo cavalieri

nome = "pressione Cavalieri "

unita= "Pa "

CALL diagramma\_p (p\_cavalieri, x, nome, unita)

!calcolo la pressione numerica massima

p\_max\_cavalieri = MAXVAL(p\_cavalieri)

!calcola il sostentamento col metodo cavalieri

CALL cavalieri (p\_esatta, N\_cavalieri, 0., p\_esatta(2)\*delta\_x/2)

!calcola il punto di applicazione col metodo cavalieri

ciclo\_x\_N: DO i=1,inter,1

integrando(i) = p\_esatta(i)\*x(i)

END DO ciclo\_x\_N

CALL cavalieri (integrando, primit, 0., p\_esatta(2)\*x(2)\*x(2))

xN\_cavalieri = primit(inter)/N\_cavalieri(inter)

!calcolo la velocità nei punti richiesti

h\_x1 = h1 + (h2-h1)\*x1/L

h\_x2 = h1 + (h2-h1)\*xN\_esatto/L

h\_x3 = h1 + (h2-h1)\*x3/L

p\_primo\_x1 = (6\*mu\*V)\*(h\_x1-h\_max)/(h\_x1\*\*3)

p\_primo\_x2 = (6\*mu\*V)\*(h\_x2-h\_max)/(h\_x2\*\*3)

p\_primo\_x3 = (6\*mu\*V)\*(h\_x3-h\_max)/(h\_x3\*\*3)

delta\_z\_x1 = h\_x1/inter2

delta\_z\_x2 = h\_x2/inter2

delta\_z\_x3 = h\_x3/inter2

z=0

ciclo\_z\_x1: DO j=2,inter2,1

z(j)=z(j-1)+delta\_z\_x1

END DO ciclo\_z\_x1

ciclo\_v\_x1: DO j=1,inter2,1

v\_x1(j)=(p\_primo\_x1\*(z(j)-h\_x1)\*z(j)/(2.\*mu)) + (V\*z(j)/h\_x1)

END DO ciclo\_v\_x1

!plotto la velocità in x1

nome = "velocita' in x1=0.15m "

unita= "m/s "

CALL diagramma\_v (v\_x1,z ,h\_x1 ,nome, unita)

z=0

ciclo\_z\_x2: DO j=2,inter2,1

z(j)=z(j-1)+delta\_z\_x2

END DO ciclo\_z\_x2

ciclo\_v\_x2: DO j=1,inter2,1

v\_x2(j)=((1./(2.\*mu))\*p\_primo\_x2\*(z(j)-h\_x2)\*z(j))+(V\*z(j)/h\_x2)

END DO ciclo\_v\_x2

!plotto la velocità in x2

nome = "velocita' in xN esatto "

unita= "m/s "

CALL diagramma\_v (v\_x2,z ,h\_x2 ,nome ,unita)

z=0

ciclo\_z\_x3: DO j=2,inter2,1

z(j)=z(j-1)+delta\_z\_x3

END DO ciclo\_z\_x3

ciclo\_v\_x3: DO j=1,inter2,1

v\_x3(j)=((1./(2.\*mu))\*p\_primo\_x3\*(z(j)-h\_x3)\*z(j))+(V\*z(j)/h\_x3)

END DO ciclo\_v\_x3

!plotto la velocità in x3

nome = "velocita' in x3=0.35m "

unita= "m/s "

CALL diagramma\_v (v\_x3, z, h\_x3, nome, unita)

!calcolo il coefficiente di attrito mediato

kn = SQRT(8./3.)\*(LOG(n)-((3\*(n-1))/(2\*(n+1))))/SQRT(LOG(n)-2\*((n-1)/(n+1)))

cost\_fm = SQRT(mu\*V/N\_esatta)

fm = kn\*cost\_fm

!scrivo sullo schermo i dati calcolati

WRITE (\*,\*) " "

WRITE (\*,\*) "La pressione massima esatta vale", p\_max\_esatta, "Pa"

WRITE (\*,\*) " "

WRITE (\*,\*) "La pressione massima secondo Cavalieri vale", p\_max\_cavalieri, "Pa"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il sostentamento esatto vale", N\_esatta, "N/m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il suo punto di applicazione e'", xN\_esatto, "m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il sostentamento secondo Cavalieri vale", N\_cavalieri (inter), "N/m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il suo punto di applicazione e'", xN\_cavalieri, "m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il coefficiente di attrito mediato e'", fm

WRITE (\*,\*) " "

WRITE (\*,\*) "le tre altezze sono", h\_x1, h\_x2, h\_x3

WRITE (\*,\*) "l'altezze h-max e'", h\_max

WRITE (\*,\*) "le tre velocità massime sono", v\_x1(inter2), v\_x2(inter2), v\_x3(inter2)

WRITE (\*,\*) "variabilità della terza z", z(1), z(inter2)

!chiusura del programma

WRITE (\*,\*)"Per chiudere il programma premi una lettera qualunque."

WRITE (\*,\*)"Tutti i dati andranno persi."

READ (\*,\*) chiusura

STOP

END PROGRAM main\_ese\_sette

**9. Modulo per il primo quesito.** Riporto il codice in Fortran.

!modulo per la settima esercitazione di meccanica apl.

!invocato dal programma main\_ese\_uno

!25/11/2012

MODULE mod\_ese\_sette

!sezione dichiarativa

IMPLICIT NONE

!fisso i valori delle costanti fisiche del problema

REAL, PARAMETER:: L = 0.4 !lunghezza del meato in m

REAL, PARAMETER:: V = 2.4 !velocita' della slitta in m/s

REAL, PARAMETER:: h1 = 0.00040 !altezza maggiore in m

REAL, PARAMETER:: h2 = 0.00016 !altezza minore in m

REAL, PARAMETER:: mu = 0.040 !viscosità in Ns/m^2

INTEGER, PARAMETER:: inter = 1000 !lunghezza degli array

INTEGER, PARAMETER:: inter2 = 100 !lunghezza array z

!fisso il formato di output per i diagrammi

CHARACTER(len=4):: formato ='bmp '

!scrivo le subroutine

CONTAINS

!-----------------------------------------------------------------------------------

SUBROUTINE diagramma\_p\_primo (funzione, x, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter):: x !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter):: funzione !funzione da plottare

CHARACTER (len=25), INTENT(IN):: nome !titolo !

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL AXSPOS (550,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME ('metri','x') !nome delle ascisse

CALL NAME (unita,'y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (0.,L,0.,0.1,min,max,min,1000000.)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (x,funzione,inter) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_p\_primo

!-----------------------------------------------------------------------------------

SUBROUTINE diagramma\_p (funzione, x, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter):: x !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter):: funzione !funzione da plottare

CHARACTER (len=25), INTENT(IN):: nome !titolo !

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL AXSPOS (450,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME ('metri','x') !nome delle ascisse

CALL NAME (unita,'y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (0.,L,0.,0.1,min,max,min,100000.)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (x,funzione,inter) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_p

!-----------------------------------------------------------------------------------

SUBROUTINE diagramma\_v (funzione, z, h, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter2):: z !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter2):: funzione !funzione da plottare

REAL, INTENT(IN):: h !altezza del meato

CHARACTER (len=25), INTENT(IN):: nome !titolo

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL LABDIG (5,'y') !5 cifre decimali per le ordinate

CALL AXSPOS (550,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME (unita,'x') !nome delle ascisse

CALL NAME ('metri','y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (0.,V,0.,0.5,0.,h,0.,0.00005)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (funzione,z,inter2) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_v

!---------------------------------------------------------------------------------------

SUBROUTINE Cavalieri (funzione, primitiva, inizio\_1, inizio\_2)

!integra la funzione inviata con il metodo di Cavalieri

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(OUT),DIMENSION(1:inter)::primitiva !primitiva

REAL, INTENT(IN), DIMENSION(1:inter)::funzione !integrando

REAL, INTENT(IN):: inizio\_1 !il valore esatto di primitiva(1)

REAL, INTENT(IN):: inizio\_2 !approssimazione di primitiva(2)

!dichiaro le variabili locali

INTEGER::i !indice del ciclo

REAL::delta\_x = L/inter !incremento dell ascisse

!sezione esecutiva

primitiva(1)= inizio\_1 !inizializza il primo valore

primitiva(2)= inizio\_2 !inizializza il secondo valore

!calcolo la primitiva (elementi dispari dell'array)

ciclo\_integra: DO i=1,inter-2,2

primitiva(i+2)=primitiva(i)+((funzione(i+2)+funzione(i+1)\*4+funzione(i))\*(delta\_x/3))

END DO ciclo\_integra

!calcolo la primitiva (elementi pari dell'array)

ciclo\_integra2: DO i=2,inter-2,2

primitiva(i+2)=primitiva(i)+((funzione(i+2)+funzione(i+1)\*4+funzione(i))\*(delta\_x/3))

END DO ciclo\_integra2

RETURN

END SUBROUTINE Cavalieri

!-----------------------------------------------------------------------------------

END MODULE mod\_ese\_sette

**10. Unità chiamante per il secondo quesito.** Riporto il codice in Fortran.

!main program per la settima esercitazione (secondo quesito) di meccanica apl.

!utilizza il modulo mod\_ese\_sette\_due

!26/11/2012

PROGRAM main\_ese\_sette\_due

USE DISLIN !libreria grafica

USE mod\_ese\_sette\_due !modulo con costanti e procedure

!sezione dichiarativa

IMPLICIT NONE

!dichiaro le variabili coinvolte nel calcolo di n

REAL:: n !il coefficiente caratteristico

REAL, DIMENSION(1:inter)::na !l'array dei valori di n

REAL:: resto !si deve annullare

REAL:: h1,h2! le due altezze

!dichiaro le variabili coinvolte col calcolo della pressione

REAL,DIMENSION(1:inter):: h !altezza del meato

REAL,DIMENSION(1:inter):: x !array delle ascisse

REAL,DIMENSION(1:inter):: p\_primo !derivata della pressione

REAL,DIMENSION(1:inter):: p\_esatta !pressione esatta

REAL::p\_max\_esatta !la pressione esatta massima

REAL::p\_max\_cavalieri !la pressione numerica massima

REAL::cost\_p\_esatta !costante moltiplicativa di p\_esatta

REAL::p\_media !la pressione media

!dichiaro altre variabili utilizzate

CHARACTER(len=10):: chiusura !serve per chiudere il programma

CHARACTER(len=25):: nome !si usa per il titolo dei grafici

CHARACTER(len=15):: unita !si usa per le unita' di misura dei grafici

INTEGER::i,j !indici di ciclo

!dichiaro le variabili relative al coefficiente di attrito

REAL::fm !coefficiente di attrito mediato

REAL::kn !parte di fm

REAL::cost\_fm !idem

!dichiaro il sostentamento e la velocita'

REAL::N\_dato

REAL, DIMENSION (1:inter)::N\_cavalieri

REAL::V

!altre variabili

REAL::delta\_x !incremento dell ascisse

REAL::h\_max !altezza in cui p è max

REAL::x\_max !ascissa in cui p è max

!dichiaro le variabili utilizzate per tracciare il diagramma di psi

REAL, DIMENSION (inter)::psia

REAL::u

!sezione esecutiva

!applico il metodo di newton-raphson per ricavare n

i=1 !valore iniziale dell'indice

na(i)=4.5 !initial guess

ciclo: DO

na(i+1) = na(i) - ( psi(na(i))/((psi(na(i)+0.0001) - psi(na(i)))/0.0001))

resto = psi(na(i)) / ((psi(na(i)+0.0001) - psi(na(i)))/0.0001)

n=na(i+1)

IF (resto<0.001) EXIT ciclo

i=i+1

END DO ciclo

!calcolo il sostentamento e la velocita'

N\_dato = (P/8)/b

V = rm\*omega

!calcolo h2 e h1

h2 = ( L/(n-1) )\*SQRT( 6\*mu\*V\*( (n+1)\*LOG(n)-2\*(n-1) )/(N\_dato\*(n+1)) )

h1 = n\*h2

!alcune operazioni preliminari

delta\_x = L/inter !incremento dell ascisse

h\_max = (2.\*h1\*h2)/(h1+h2) !altezza in cui p e' max

x\_max = L\*h1/(h1+h2) !ascissa in cui p e' max

!dichiaro l'array delle ascisse

x(1) = 0.

ciclo\_x: DO i=2,inter,1

x(i) = x(i-1) + delta\_x

END DO ciclo\_x

!calcolo l'altezza del meato

ciclo\_h: DO i=1,inter,1

h(i) = h1 + (h2-h1)\*x(i)/L

END DO ciclo\_h

!calcolo la derivata della pressione

ciclo\_p\_primo: DO i=1,inter,1

p\_primo(i) = (6\*mu\*V)\*(h(i)-h\_max)/(h(i)\*\*3)

END DO ciclo\_p\_primo

!calcolo la pressione esatta

cost\_p\_esatta = (6\*mu\*V\*L)/((n-1)\*h2)

ciclo\_p: DO i=1,inter,1

p\_esatta(i) = cost\_p\_esatta\*((1/h(i))-((h2/(h(i)\*\*2))\*(n/(n+1)))-(1/(h2\*(n+1))))

END DO ciclo\_p

!calcolo la pressione massima esatta

p\_max\_esatta = MAXVAL(p\_esatta)

!calcolo la pressione media

p\_media = N\_dato/L

CALL BMPMOD (300,'inch','resolution') !fisso risoluzione dell'immagine in formato .bmp

!traccio il diagramma del gradiente della pressione

nome = "gradiente della pressione"

unita = "\frac{Pa}{m} "

CALL diagramma\_p\_primo (p\_primo, x, nome, unita)

!traccio il diagramma della pressione esatta

nome = "pressione "

unita= "Pa "

CALL diagramma\_p (p\_esatta, x, nome, unita)

!calcolo il sostentamento integrando la pressione come verifica

CALL cavalieri (p\_esatta, N\_cavalieri, 0., p\_esatta(2)\*delta\_x/2)

!calcolo il coefficiente di attrito mediato

kn = SQRT(8./3.)\*(LOG(n)-((3\*(n-1))/(2\*(n+1))))/SQRT(LOG(n)-2\*((n-1)/(n+1)))

cost\_fm = SQRT(mu\*V/N\_dato)

fm = kn\*cost\_fm

!traccio il diagramma di psi

nome = "psi "

unita= " "

na(1)=1

psia(1)=0

ciclo\_psi: DO i=1, inter-1, 1

na(i+1)=na(i) + 0.01

u = na(i+1)

psia(i+1)= xN - L\*((2\*u\*(u+2)\*LOG(u))-(5\*(u\*\*2))+4\*u+1)/(2\*(u-1)\*((u+1)\*LOG(u)-2\*(u-1)))

END DO ciclo\_psi

CALL diagramma\_psi (psia, na, nome, unita)

!scrivo sullo schermo i dati calcolati

WRITE (\*,\*) " "

WRITE (\*,\*) "La pressione massima vale", p\_max\_esatta, "Pa"

WRITE (\*,\*) " "

WRITE (\*,\*) "La pressione media vale", p\_media, "Pa"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il sostentamento assegnato vale", N\_dato, "N/m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il sostentamento ricalcolato dalla pressione vale", N\_cavalieri (inter), "N/m"

WRITE (\*,\*) " "

WRITE (\*,\*) "Il coefficiente di attrito mediato e'", fm

WRITE (\*,\*) " "

WRITE (\*,\*) "l'altezze h-max e'", h\_max

WRITE (\*,\*) " "

WRITE (\*,\*) "l'altezze h1 e'", h1

WRITE (\*,\*) " "

WRITE (\*,\*) "l'altezze h2 e'", h2

WRITE (\*,\*) " "

WRITE (\*,\*) "il coefficente caratteristico del meato e'", n

!chiusura del programma

WRITE (\*,\*)"Per chiudere il programma premi una lettera qualunque."

WRITE (\*,\*)"Tutti i dati andranno persi."

READ (\*,\*) chiusura

STOP

END PROGRAM main\_ese\_sette\_due

**11. Modulo per il secondo quesito.** Riporto il codice in Fortran.

!modulo per la settima esercitazione di meccanica apl.

!invocato dal programma main\_ese\_sette\_due

!26/11/2012

MODULE mod\_ese\_sette\_due

!sezione dichiarativa

IMPLICIT NONE

!fisso i valori delle costanti fisiche del problema

REAL, PARAMETER:: L = 0.0735 !lunghezza del meato in m

REAL, PARAMETER:: omega = 83.7758 !velocita' angolare dell'albero in rad/s

REAL, PARAMETER:: mu = 0.035 !viscosità in Ns/m^2

REAL, PARAMETER:: rm = 0.1 !raggio medio del cuscinetto in m

REAL, PARAMETER:: b = 0.06 !larghezza della slitta in m

REAL, PARAMETER:: xN = 0.045 !punto di applicazione del carico in m

REAL, PARAMETER:: P =8000.0 !carico spingente complessivo

!fisso la lunghezza degli array utilizzati

INTEGER, PARAMETER:: inter = 1000

!fisso il formato di output per i diagrammi

CHARACTER(len=4):: formato ='bmp '

!scrivo le subroutine

CONTAINS

!-----------------------------------------------------------------------------------

SUBROUTINE diagramma\_p\_primo (funzione, x, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter):: x !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter):: funzione !funzione da plottare

CHARACTER (len=25), INTENT(IN):: nome !titolo !

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL AXSPOS (550,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME ('metri','x') !nome delle ascisse

CALL NAME (unita,'y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (0.,L,0.,0.01,min,max,min,10000000.)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (x,funzione,inter) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_p\_primo

!-----------------------------------------------------------------------------------

SUBROUTINE diagramma\_p (funzione, x, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter):: x !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter):: funzione !funzione da plottare

CHARACTER (len=25), INTENT(IN):: nome !titolo !

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL AXSPOS (450,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME ('metri','x') !nome delle ascisse

CALL NAME (unita,'y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (0.,L,0.,0.01,min,max,min,100000.)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (x,funzione,inter) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_p

!---------------------------------------------------------------------------------------

SUBROUTINE diagramma\_psi (funzione, x, nome, unita)

!traccia il diagramma della funzione passatagli dal main program

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(IN), DIMENSION (1:inter):: x !contiene i valori delle ascisse

REAL, INTENT(IN), DIMENSION (1:inter):: funzione !funzione da plottare

CHARACTER (len=25), INTENT(IN):: nome !titolo !

CHARACTER (len=15), INTENT(IN):: unita !ascisse

!dichiaro le variabili locali

INTEGER:: i !indice del ciclo

REAL:: max !valore massimo della funzione

REAL:: min !valore minimo della funzione

!sezione esecutiva

CALL METAFL (formato) !indico il formato dell'output

CALL SCRMOD ('revers') !scritta nera su fondo bianco

CALL DISINI

CALL TEXMOD ('ON') !chiedo di poter scrivere le formule in maniera grafica

CALL PAGERA !traccio un bordo per il piano xy

CALL DUPLX !font a doppio spessore

CALL LABDIG (5,'y') !5 cifre decimali per le ordinate

CALL AXSPOS (450,1800) !coordinate angolo basso sinistra

CALL AXSLEN (2200,1500)!lunghezza dei due assi in pixel

CALL NAME ('h1/h2','x') !nome delle ascisse

CALL NAME (unita,'y') !nome delle ordinate

CALL TITLIN (nome,1) !prima riga del titolo

max=MAXVAL(funzione(5:inter)) !il massimo della funzione

min=MINVAL(funzione) !il minimo della funzione

CALL GRAF (2.,11.,2.,0.5,min,max,min,0.005)

CALL GRID (2,2)

CALL TITLE !stampa il titolo di cui sopra

CALL CURVE (x,funzione,inter) !plotto la funzione

CALL DASH !tratteggio per gli assi coordinati

CALL XAXGIT !traccio la retta y=0

CALL YAxGIT !traccio la retta x=0

CALL DISFIN

RETURN

END SUBROUTINE diagramma\_psi

!---------------------------------------------------------------------------------------

SUBROUTINE Cavalieri (funzione, primitiva, inizio\_1, inizio\_2)

!integra la funzione inviata con il metodo di Cavalieri

IMPLICIT NONE

!dichiaro gli argomenti fittizi

REAL, INTENT(OUT),DIMENSION(1:inter)::primitiva !primitiva

REAL, INTENT(IN), DIMENSION(1:inter)::funzione !integrando

REAL, INTENT(IN):: inizio\_1 !il valore esatto di primitiva(1)

REAL, INTENT(IN):: inizio\_2 !approssimazione di primitiva(2)

!dichiaro le variabili locali

INTEGER::i !indice del ciclo

REAL::delta\_x = L/inter !incremento dell ascisse

!sezione esecutiva

primitiva(1)= inizio\_1 !inizializza il primo valore

primitiva(2)= inizio\_2 !inizializza il secondo valore

!calcolo la primitiva (elementi dispari dell'array)

ciclo\_integra: DO i=1,inter-2,2

primitiva(i+2)=primitiva(i)+((funzione(i+2)+funzione(i+1)\*4+funzione(i))\*(delta\_x/3))

END DO ciclo\_integra

!calcolo la primitiva (elementi pari dell'array)

ciclo\_integra2: DO i=2,inter-2,2

primitiva(i+2)=primitiva(i)+((funzione(i+2)+funzione(i+1)\*4+funzione(i))\*(delta\_x/3))

END DO ciclo\_integra2

RETURN

END SUBROUTINE Cavalieri

!-----------------------------------------------------------------------------------

FUNCTION psi (u)

!calcola la funzione psi in u

!sezione dichiarativa

REAL:: psi

REAL:: u

!sezione esecutiva

psi = xN - L\*((2\*u\*(u+2)\*LOG(u))-(5\*(u\*\*2))+4\*u+1)/(2\*(u-1)\*((u+1)\*LOG(u)-2\*(u-1)))

RETURN

END FUNCTION psi

!-----------------------------------------------------------------------------------

END MODULE mod\_ese\_sette\_due

1. In questa trattazione si pone uguale a zero la pressione esterna, per cui la funzione delle pressioni porge quella che viene detta ‘sovrapressione’, che è poi ciò che serve per calcolare la forza di sostentamento. [↑](#footnote-ref-1)
2. Infatti all’equilibrio il momento di ciascuna forza deve essere nullo rispetto al perno stesso. [↑](#footnote-ref-2)