

```

program BUOYAN
-----
c   PROGRAMMA PER IL CALCOLO DEL ' 'BUOYANCY-DRIVEN CAVITY FLOW' '
c   -GEOMETRIA CARTESIANA
c   -GRIGLIA CON SPAZIATURA UNIFORME
c   -EQUAZIONI RISCritte IN TERMINI DI VORTICITA-FUNZIONE DI CORRENTE
c
c   autore: Francesco Zonta
-----

implicit real*8(a-h,o-z)

C-----NX,NY=NUMERO DI NODI SU X E Y
C#####
parameter(NX=40,NY=40)

DIMENSION X(NX),Y(NY)

DIMENSION U(NX,NY),V(NX,NY)
DIMENSION T(NX,NY),T2(NX,NY)
DIMENSION PSI(NX,NY)
DIMENSION VOR(NX,NY),VOR2(NX,NY)

DIMENSION DIFF(NX,NY),CONV(NX,NY)
DIMENSION DIFFT(NX,NY),CONVT(NX,NY)
DIMENSION BUOY(NX,NY)

C-----PARAMETRI DELLA GRIGLIA-----
LX=1.
LY=1.

DX=FLOAT(LX)/FLOAT(NX-1)
DY=FLOAT(LY)/FLOAT(NY-1)
C-----

C-----PARAMETRI SIMULAZIONE--
PR=1.
RA=1.E4
UWALL=0.
DT=1.E-3
TIMEMAX=50.E+0

TC=-0.5
TH=0.5

C-----
C   GRIGLIA COMPUTAZIONALE

DO I=1,NX
  X(I)=X(I-1)+DX
ENDDO

DO J=1,NY
  Y(J)=Y(J-1)+DY
ENDDO

C-----INIZIALIZZAZIONE VETTORI

DO I=1,NX
  DO J=1,NY
    U(I,J)=0.
    V(I,J)=0.
    T(I,J)=0.
    T2(I,J)=0.
    PSI(I,J)=0.
    VOR(I,J)=0.
    VOR2(I,J)=0.
  ENDDO
ENDDO

C-----

c-----CONDIZIONI AL CONTORNO--
C   TEMPERATURA
DO J=1,NY
  T2(NX,J)=TH
  T2(1,J)=TC
ENDDO
C-----
C-----

```

```

C      CONTATORI
      EPS=1.E-6
      TIME=0.
      NODI=(NX)*(NY)
      NUMERO=0      ! NUMERO NODI CONVERGENTI
      ITERTOT=0.

C-----SOLVER-----

10  CONTINUE

      TIME=TIME+DT
      ITERTOT=ITERTOT+1

      WRITE(*,*) 'ITERAZIONE=',ITERTOT

C-----
C      POISSON EQUATION FOR STREAM-FUNCTION

      W=1.9 !WEIGHT
      EPSILP=1.E-6
      ITERMAX=1000
      ITER=0
      NUM=0
      NUMEQ=(NX-2)*(NY-2)      !numero di equazioni considerate

20  CONTINUE

      NUM=0

      DO I=2,NX-1
      DO J=2,NY-1

      PSI2=(1./(2.*(DX**2+DY**2)))*
&      ((PSI(I+1,J)+PSI(I-1,J))*DY**2+
&      (PSI(I,J+1)+PSI(I,J-1))*DX**2-VOR2(I,J)*DX**2*DY**2)

c      PSI2=(PSI(I+1,J)+PSI(I-1,J)+
c      &      PSI(I,J+1)+PSI(I,J-1)-VOR2(I,J)*DX**2)/4.

      PSI2=(1.-W)*PSI(I,J)+W*PSI2

      ERROR1=abs(PSI2-PSI(I,J))

      PSI(I,J)=PSI2;

      IF (ERROR1.LT.EPSILP) THEN
          NUM=NUM+1;
      ENDIF

      ENDDO
      ENDDO

      ITER=ITER+1

C-----

      IF (NUM.LT.NUMEQ.AND.ITER.LT.ITERMAX) THEN

          GOTO 20

      ELSE

          GOTO 21

      ENDIF
C#####
21  CONTINUE

c-----iterazioni solver poisson:
c
c      write(*,*) 'Iterazioni Poisson:',ITER !nota: deve essere <1000
c
c-----

c--Calcolo della vorticita al bordo espandendo in serie di Taylor la funzione
c      di corrente

c bordi orizzontali
      DO I=1,NX
          VOR2(I,1)=(0.5/DY**2)*(8.*PSI(I,2)-PSI(I,3))      !bordo basso

```

```

      VOR2(I,NY)=(0.5/DY**2)*(8.*PSI(I,NY-1)-PSI(I,NY-2))+
&      (3.*UWALL)/DY      !bordo alto
      ENDDO

```

C bordi verticali

```

      DO J=1,NY
      VOR2(1,J)=(0.5/DX**2)*(8.*PSI(2,J)-PSI(3,J))      !bordo sinistro
      VOR2(NX,J)=(0.5/DX**2)*(8.*PSI(NX-1,J)-PSI(NX-2,J)) !bordo destro
      ENDDO

```

C Calcolo del campo di moto

```

      DO I=2,NX-1
      DO J=2,NY-1
      U(I,J)=(PSI(I,J+1)-PSI(I,J-1))/(2.*DY)
      V(I,J)=- (PSI(I+1,J)-PSI(I-1,J))/(2.*DX)
      ENDDO
      ENDDO

```

c-----AGGIORNAMENTO CAMPI PER AVANZAMENTO TEMPORALE

```

      DO I=1,NX
      DO J=1,NY
      VOR(I,J)=VOR2(I,J)
      T(I,J)=T2(I,J)
      ENDDO
      ENDDO

```

C Calcolo della vorticita nel nucleo del dominio

C all'istante successivo attraverso l'equazione di trasporto della vorticita

```

      DO I=2,NX-1
      DO J=2,NY-1

      CONV(I,J)=(0.5/DX)*(U(I+1,J)*VOR(I+1,J)-U(I-1,J)*VOR(I-1,J))
&      +(0.5/DY)*(V(I,J+1)*VOR(I,J+1)-V(I,J-1)*VOR(I,J-1))

      DIFF(I,J)=(VOR(I+1,J)-2.*VOR(I,J)+VOR(I-1,J))/(DX**2)+
&      (VOR(I,J+1)-2.*VOR(I,J)+VOR(I,J-1))/(DY**2)

      BUOY(I,J)=(T(I+1,J)-T(I-1,J))/(DX)

      RDIFF=PR/(RA*(0.5))
      RBUOY=PR

```

```

      VOR2(I,J)=VOR(I,J) + DT*(      -CONV(I,J) +
&      RDIFF*DIFF(I,J)
&      RBUOY*BUOY(I,J)
      ENDDO
      ENDDO

```

c#####

c calcolo del campo termico

c- condizioni al contorno campo termico

c bordi orizzontali- dt/dy=0

```

      DO I=1,NX
      T2(I,1)=T(I,2)      !bordo basso
      T2(I,NY)= T(I,NY-1) !bordo alto
      ENDDO

```

C bordi verticali T=Tfix

```

      DO J=1,NY
      T2(1,J)=TC      !bordo sinistro
      T2(NX,J)=TH      !bordo destro
      ENDDO

```

C Calcolo della temperatura nel nucleo del dominio

C all'istante successivo attraverso l'equazione dell' energia

```

      DO I=2,NX-1
      DO J=2,NY-1

```

```

CONVT(I,J)=(0.5/DX)*(U(I+1,J)*T(I+1,J)-U(I-1,J)*T(I-1,J))
&          +(0.5/DY)*(V(I,J+1)*T(I,J+1)-V(I,J-1)*T(I,J-1))

DIFFFT(I,J)=(T(I+1,J)-2.*T(I,J)+T(I-1,J))/(DX**2)+
&          (T(I,J+1)-2.*T(I,J)+T(I,J-1))/(DY**2)

T2(I,J)=T(I,J) + DT*(
&          -CONVT(I,J) +
          (1./(RA**(0.5)))*DIFFFT(I,J) )

ENDDO
ENDDO

```

C-----

C Calcolo dell'errore sulla vorticita

```

NUMERO=0
ERRTOT=0.
DO I=1,NX
DO J=1,NY

ERROR=ABS(VOR2(I,J)-VOR(I,J))
ERRTOT=ERRTOT+ERROR

IF (ERROR.LT.EPS) THEN      !Condizione per incrementare
NUMERO=NUMERO+1           !il numero di nodi che raggiungono
ENDIF                     !la stabilita' (nel LOOP TEMPORALE)

ENDDO
ENDDO

MEANERR=ERRTOT/FLOAT((NX)*(NY))

IF(NUMERO.LT.NODI.AND.TIME.LT.TIMEMAX) THEN
GOTO 10
ELSE
GOTO 900
ENDIF

```

C-----

C-----USCITA LOOP TEMPORALE

900 CONTINUE

C-----

```

DO I=1,NX
DO J=1,NY
WRITE(53,153) X(I),Y(J),U(I,J),V(I,J),T(I,J),PSI(I,J),VOR2(I,J)
ENDDO
write(53,*)
ENDDO
153 FORMAT(7(G16.9,1X))

I=NX/2+1
DO J=1,NY
WRITE(54,*) Y(J),U(I,J)
ENDDO

J=NY/2+1
DO I=1,NX
WRITE(55,*) X(I),V(I,J)
ENDDO

```

C-----

end