

A

Apêndices

A.1 Discretização do laplaciano para o CDS

Suponhamos uma função arbitrária $f(x, y)$, no caso bidimensional. As expansões desta função em série de Taylor em torno de um ponto genérico (x_0, y_0) para pequenos desvios (a, a) serão:

$$f(x_0 + a, y_0) = f(x_0, y_0) + \frac{\partial f}{\partial x} \Big|_{x_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 + \dots \quad (\text{A.1})$$

$$f(x_0 - a, y_0) = f(x_0, y_0) - \frac{\partial f}{\partial x} \Big|_{x_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 - \frac{1}{3!} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 + \dots \quad (\text{A.2})$$

$$f(x_0, y_0 + a) = f(x_0, y_0) + \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2 + \frac{1}{3!} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 + \dots \quad (\text{A.3})$$

$$f(x_0, y_0 - a) = f(x_0, y_0) - \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2 - \frac{1}{3!} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 + \dots \quad (\text{A.4})$$

Estas expansões são as expansões para os primeiros vizinhos do ponto (x_0, y_0) , fazamos agora as expansões para seus segundos vizinhos:

$$\begin{aligned} f(x_0 + a, y_0 + a) = & f(x_0, y_0) + \frac{\partial f}{\partial x} \Big|_{x_0} a + \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2 \\ & + \frac{\partial^2}{\partial x \partial y} \Big|_{y_0} a^2 + \frac{1}{3} \left[\frac{1}{2} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 + \frac{1}{2} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 + \frac{\partial^3}{\partial x^2 \partial y} \Big|_{y_0} a^3 \right. \\ & \left. + \frac{\partial^3}{\partial x \partial y^2} \Big|_{y_0} a^3 \right] + \dots \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} f(x_0 + a, y_0 - a) = & f(x_0, y_0) + \frac{\partial f}{\partial x} \Big|_{x_0} a - \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2 \\ & - \frac{\partial^2}{\partial x \partial y} \Big|_{y_0} a^2 + \frac{1}{3} \left[\frac{1}{2} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 - \frac{1}{2} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 - \frac{\partial^3}{\partial x^2 \partial y} \Big|_{y_0} a^3 \right. \\ & \left. + \frac{\partial^3}{\partial x \partial y^2} \Big|_{y_0} a^3 \right] + \dots \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} f(x_0 - a, y_0 + a) = & f(x_0, y_0) - \frac{\partial f}{\partial x} \Big|_{x_0} a + \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2 \\ & - \frac{\partial^2}{\partial x \partial y} \Big|_{y_0} a^2 + \frac{1}{3} \left[-\frac{1}{2} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 + \frac{1}{2} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 + \frac{\partial^3}{\partial x^2 \partial y} \Big|_{y_0} a^3 \right. \\ & \left. - \frac{\partial^3}{\partial x \partial y^2} \Big|_{y_0} a^3 \right] + \dots \end{aligned} \quad (\text{A.7})$$

$$f(x_0 - a, y_0 - a) = f(x_0, y_0) - \frac{\partial f}{\partial x} \Big|_{x_0} a - \frac{\partial f}{\partial y} \Big|_{y_0} a + \frac{1}{2!} \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} a^2 + \frac{1}{2!} \frac{\partial^2 f}{\partial y^2} \Big|_{y_0} a^2$$

$$\begin{aligned}
 & + \frac{\partial^2}{\partial x \partial y} \Big|_{y_0} a^2 + \frac{1}{3} \left[-\frac{1}{2} \frac{\partial^3 f}{\partial x^3} \Big|_{x_0} a^3 - \frac{1}{2} \frac{\partial^3 f}{\partial y^3} \Big|_{y_0} a^3 - \frac{\partial^3}{\partial x^2 \partial y} \Big|_{y_0} a^3 \right. \\
 & \left. - \frac{\partial^3}{\partial x \partial y^2} \Big|_{y_0} a^3 \right] + \dots
 \end{aligned} \tag{A.8}$$

Somando as expressões (A.1-A.4) temos:

$$\begin{aligned}
 \left[\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right]_{(x_0, y_0)} & = f(x_0 + a, y_0) + f(x_0 - a, y_0) + f(x_0, y_0 + a) \\
 & + f(x_0, y_0 - a) - 4f(x_0, y_0).
 \end{aligned} \tag{A.9}$$

Semelhantemente, a soma das expressões (A.5-A.8) conduz a:

$$\begin{aligned}
 2 \left[\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right]_{(x_0, y_0)} & = f(x_0 + a, y_0 + a) + f(x_0 + a, y_0 - a) \\
 & + f(x_0 - a, y_0 + a) + f(x_0 - a, y_0 - a) \\
 & - 4f(x_0, y_0).
 \end{aligned} \tag{A.10}$$

Por último, dividindo a equação (A.9) por 6 e a (A.10) por 12 e em seguida somando-as obtemos:

$$\begin{aligned}
 \frac{1}{3} \left[\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right]_{(x_0, y_0)} & = \frac{1}{6} \left[f(x_0 + a, y_0) + f(x_0 - a, y_0) + f(x_0, y_0 + a) \right. \\
 & \left. + f(x_0, y_0 - a) \right] + \frac{1}{12} \left[f(x_0 + a, y_0 + a) \right. \\
 & + f(x_0 + a, y_0 - a) + f(x_0 - a, y_0 + a) \\
 & \left. + f(x_0 - a, y_0 - a) \right] - f(x_0, y_0).
 \end{aligned} \tag{A.11}$$

Obtivemos portanto uma expressão para o laplaciano no ponto (x_0, y_0) por uma expansão em série de Taylor até a 4ª ordem. Essa é uma forma isotrópica de discretização do laplaciano. O truncamento das expansões acima fornecem uma boa aproximação sempre que o parâmetro a for muito pequeno, isto é: $(a \ll 1)$. O que fazemos no CDS é considerar válida essas expansões mesmo no caso de $a = 1$. Essa consideração será razoável se fizermos a hipótese de que as derivadas são todas pequenas ($\ll 1$), portanto é nesse regime que temos que trabalhar. É comum escrever a última equação numa forma mais compacta:

$$\nabla_{CDS}^2 f = \langle\langle f \rangle\rangle - f,$$

com

$$\langle\langle f \rangle\rangle \equiv \frac{1}{6} \sum_{NN} f + \frac{1}{12} \sum_{NNN} f. \tag{A.12}$$

A.2 Código do Programa em Fortran

Segue abaixo a listagem do programa por nós desenvolvido em linguagem Fortran. Decidimos listá-lo porque ele será o programa base para novas implementações utilizando o modelo CDS. As subrotinas contidas podem ou não serem utilizadas, dependendo das condições de contorno utilizadas.

A subrotina ran1 é nosso gerador de números aleatórios e foi extraída do Numerical Recipes³⁹.

```

C      *****
C      ESTE PROGRAMA SIMULA A DINAMICA DE UM GAS GRANULAR
C      VIA MODELO CDS
C      *****
C      inicio do programa principal

      PROGRAM GRAINS
      IMPLICIT NONE
      INTEGER L, TIME, ISEED, I, J, L1, Lx, Ly
      INTEGER IT, CONT, CONT1, STEP_FOTO, CONTORNO, START_FOTO
      PARAMETER(L=32)
      DOUBLE PRECISION T(-1:L+2, -1:L+2)
      DOUBLE PRECISION To(-1:L+2, -1:L+2)
      DOUBLE PRECISION Uxo(-1:L+2, -1:L+2)
      DOUBLE PRECISION Uyo(-1:L+2, -1:L+2)
      DOUBLE PRECISION Ux(-1:L+2, -1:L+2)
      DOUBLE PRECISION Uy(-1:L+2, -1:L+2)
      DOUBLE PRECISION N(-1:L+2, -1:L+2)
      DOUBLE PRECISION No(-1:L+2, -1:L+2)
      DOUBLE PRECISION Lbda, EXP, A, D, B, Nmo, St
      DOUBLE PRECISION Eta, Nmedio, Tal, Gama, A_0, v_0
      DOUBLE PRECISION Lambda_max, h_e, h_l, k, g, Eta0, Lbda0, Lr
      CHARACTER*50 Arquivo

C      *****
C      DEFINICAO DOS PRARAMETROS DE ORDEM

C      Uxo : Componente x da velocidade no instante t
C      Ux  : Componente x da velocidade no instante t+1
C      Uyo : Componente y da velocidade no instante t
C      Uy  : Componente y da velocidade no instante t+1
C      To  : Temperatura granular no instante t
C      T   : Temperatura granular no instante t+1
C      No  : Densidade no instante t
C      N   : Densidade no instante t+1

C      PARAMETROS COMPUTACIONAIS

C      Lambda : Coeficiente de condutividade termica
C      Eta    : Coeficiente de viscosidade
C      L      : Tamanho do sistema

```

```

C   Exp      : expoente que caracteriza o coeficiente de restituicao
C             Exp=8/5 --> dependente da velocidade relativa
C             Exp=3/2 --> constante

c   SELECIONA AS CONDICICOES DE CONTORNO

c
c           | 1- caixa fechada;
c   CONTORNO = < 2- tubo continuo;
c           | 3- periodico.
c           -
c   CONTORNO=3

C   *****

      Eta0=0.005
      Lbda0=0.002
5     Lx=16
      Ll=16
      Ly=16
      Lr=L
      START_FOTO=10 000
      STEP_FOTO=1 000
      CONT=START_FOTO
      CONT1=0
      iseed = -432765834
      Gama =1.0d-2
      A = 0.0023d0
      D = .5d0
      EXP = 8.0d0/5.0d0
      B =1.0d0
      A_0=1.6d-2
      v_0=1.0d0
      ta1=(A_0/v_0)*Gama
      TIME = 99 000
      g= 0.0d0
      St=1.0d0
      DO I=1,L
        DO J=1,L
          Uxo(I,J)=0.0
          Uyo(I,J)=0.0
          To(I,J)=0.0
          No(I,J)=0.0
          Ux(I,J)=0.0
          Uy(I,J)=0.0
          T(I,J)=0.0
          N(I,J)=0.0
        ENDDO
      ENDDO
      CALL RANDOM_IC(Nmedio,To,Uxo,Uyo,No,L,ISEED,Nmo)

C   ***** impoe condicao de contorno *****

      IF(CONTORNO.EQ.1) THEN
        CALL COND1(To,Uxo,Uyo,No,L)
      ELSE
        IF(CONTORNO.EQ.2) THEN
          CALL COND2(To,Uxo,Uyo,No,L,Ll,Lx,Ly)
        ELSE
          CALL COND3(To,Uxo,Uyo,No,L)
        END IF
      END IF
      WRITE(Arquivo,('tempMedia&',f5.3,'&',f5.3'.dat'
*      )') ETA0,LBDA0
      OPEN(UNIT=20,FILE=Arquivo,STATUS='UNKNOWN')

```

```

C      ***** inicio da itualizacao dos parametros *****

DO IT = 0,TIME
  Eta=Eta0*sqrt(St)
  Lbda=Lbda0*sqrt(St)

  CALL ATUALIZACAO(T,Ux,Uy,N,To,Uxo,Uyo,No,L,1,L,1,L,g,
*      Gama,Eta,Lbda,Exp,B,Eta0,Lbda0)

  St=0.0d0
  Nmedio=0.0d0
  DO I = 1,L
    DO J = 1,L
      Uxo(I,J)=Ux(I,J)
      Uyo(I,J)=Uy(I,J)
      To(I,J)=T(I,J)
      No(I,J)=N(I,J)
      Nmedio=Nmedio+No(I,J)
      St=St+To(I,J)
    ENDDO
  ENDDO
  St=St/Lr**2.0d0
  Nmedio=Nmedio/(Lr**2.0d0)
  CALL INJECTENERGY(To,No,L,A,Nmo)

C      ***** impoe condicao de contorno *****

  IF(CONTORNO.EQ.1) THEN
    CALL COND1(To,Uxo,Uyo,No,L)
  ELSE
    IF(CONTORNO.EQ.2) THEN
      CALL COND2(To,Uxo,Uyo,No,L,L1,Lx,Ly)
    ELSE
      CALL COND3(To,Uxo,Uyo,No,L)
    END IF
  END IF

C      ***** imprime configuracao do sistema nas arquivos de saida *****

  IF (IT.GT.(START_FOTO-1)) THEN
4    IF (IT.EQ.CONT) THEN
      IF (CONTORNO.EQ.1.OR.CONTORNO.EQ.3) THEN
        CALL SAIDA1(To,No,Nmedio,Uxo,Uyo,St,IT,L,ETA0,LBDA0,
*          L1,Lx,Ly)
        A=0.0d0
      ELSE
        CALL SAIDA(To,No,Nmedio,Uxo,Uyo,St,IT,L,ETA0,LBDA0,
*          L1,Lx,Ly)
      END IF
      CONT=CONT+STEP_FOTO
      PRINT*,'< Eta=',Eta0,' Lambda=',Lbda0,' Iteracao='
*          ,IT,'> Concluida'
      END IF
    END IF
    IF (IT.EQ.CONT1) THEN
      WRITE(20,*) IT,' ',St
      CONT1=CONT1+1000
    END IF

C      ***** fim da atualizacao dos parametros *****

  ENDDO
  CLOSE(20)
  STOP
  END

```

```

C      **** Fim do programa principal ****

C      ***** INICIO DAS SUBROTINAS *****

C      *****
C
C      Esta subrotina efetua a atualizacao dos parametros de ordem no
C      instante t+1 a partir de seus valores no instante t
C
C      *****

SUBROUTINE ATUALIZACAO(T,Ux,Uy,N,To,Uxo,Uyo,No,L,
*      Qxo,Qxf,Qyo,Qyf,g,Gama,Eta,Lbda,Exp,B,Eta0,Lbda0)
IMPLICIT NONE
Integer I,J,L,Qxo,Qxf,Qyo,Qyf
DOUBLE PRECISION T(-1:L+2,-1:L+2)
DOUBLE PRECISION To(-1:L+2,-1:L+2)
DOUBLE PRECISION Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION Uyo(-1:L+2,-1:L+2)
DOUBLE PRECISION Ux(-1:L+2,-1:L+2)
DOUBLE PRECISION Uy(-1:L+2,-1:L+2)
DOUBLE PRECISION N(-1:L+2,-1:L+2)
DOUBLE PRECISION No(-1:L+2,-1:L+2)
DOUBLE PRECISION Lbda,EXP,N1,N2,B
DOUBLE PRECISION Uxdx,Uydy,Uydx,Uydy,Uxdxx,Uxdyy,Uydy,Tdx,Tdy,Tdyy
DOUBLE PRECISION Eta,Ndx,Ndy,Uydx,Uydy,Uydx,Uydy,Gama,Tdxx
DOUBLE PRECISION Ndy,Ndxx,g,Eta0,Lbda0

10 DO I = Qxo,Qxf
    DO J = Qyo,Qyf

C      **** calcula derivadas ****

        CALL DERIVADAS(To,Uxo,Uyo,No,Uxdx,Uydy,Uxdxx,Uydx,
*      Uxdyy,Uydy,Tdx,Tdy,Ndx,Ndy,Uydx,Uydy,Uydx,
*      Uydy,N1,N2,Tdxx,Tdyy,Ndxx,Ndy,I,J,L)

c      **** atualiza a temperatura granular ****

        T(I,J) = To(I,J)+Gama*( -(Uxo(I,J)*Tdx+Uyo(I,J)*Tdy)
*      -To(I,J)*(Uxdx+Uydy)+(3.0d0*Lbda/(No(I,J)))
*      *(2.0d0*N1+N2-12.0d0*To(I,J))/12.0d0
*      +(Eta/No(I,J))*(2.0d0*Uxdx**2.0d0+2.0d0*Uydy**2.0d0
*      +Uxdy**2.0d0+Uydx**2.0d0+2.0d0*Uydx*Uxdy)
*      -Eta/No(I,J)*(Uxdx+Uydy)**2.0d0
*      -B*No(I,J)*(To(I,J)**(EXP)))

C      **** atualiza a componente Ux ****

        Ux(I,J)=Uxo(I,J)+Gama*(-(1.0d0/(No(I,J)))*(No(I,J)*Tdx
*      +To(I,J)*Ndx)+(Eta/No(I,J))*(Uxdxx+Uxdyy)
*      -(Ux(I,J)*Uxdx+Uy(I,J)*Uxdy))

C      **** atualiza a componente Uy ****

        Uy(I,J)=Uyo(I,J)+Gama*(-(1.0d0/(No(I,J)))*(No(I,J)*Tdy
*      +To(I,J)*Ndy)+(Eta/No(I,J))*(Uydy+Uydx)
*      -(Uy(I,J)*Uydy+Ux(I,J)*Uydx)) -Gama*g

C      **** atualiza a densidade ****

        N(I,J) = No(I,J)+Gama*(-(No(I,J)*(Uxdx+Uydy)+Uxo(I,J)

```

```

*          *Ndx+Uyo(I,J)*Ndy)

      ENDDO
      ENDDO
      RETURN
      END

C *****
C
C Esta subrotina inicializa randomicamente todos os
C valores das variaveis
C *****

SUBROUTINE RANDOM_IC(Nmedio,To,Uxo,Uyo,No,L,ISEED,Nmo)
IMPLICIT NONE
Integer L,ISEED,I,J
REAL AMP,RAN1
DOUBLE PRECISION To(-1:L+2,-1:L+2),Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION Uyo(-1:L+2,-1:L+2),No(-1:L+2,-1:L+2)
DOUBLE PRECISION Nmedio,T_O,SUM,Nmo,Lr
SUM=0.0d0
Nmo=0.0d0
T_O=10.0d0
AMP = T_O/10.0d0
Nmedio=0.0d0
Lr=L
DO J = 1,L
  DO I = 1,L
    To(I,J) =0.0d0
    Uxo(I,J)=0.0d0
    Uyo(I,J)=0.0d0
    No(I,J)=0.0d0
  ENDDO
ENDDO
DO J = 1,L
  DO I = 1,L
    To(I,J) = T_O + AMP*(RAN1(ISEED))
    No(I,J)= 0.1d0+0.01d0*(-1.0d0+2.0d0*RAN1(ISEED))
    SUM = SUM + To(I,J)
    Nmo=Nmo+No(I,J)/(L**2)
    Nmedio=Nmedio+No(I,J)
  END DO
ENDDO
Nmedio=Nmedio/L**2.0d0
SUM = SUM/(Lr**2.0d0)
RETURN
END

C *****
C
C Esta subrotina impoe condicao de contorno tipo 'caixa fechada
C *****

SUBROUTINE COND1(To,Uxo,Uyo,No,L)
IMPLICIT NONE
Integer I,J,L
DOUBLE PRECISION To(-1:L+2,-1:L+2),Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION No(-1:L+2,-1:L+2),Uyo(-1:L+2,-1:L+2)
DO J =1,L
  To(0,J) = To(1,J)
  To(-1,J) = To(2,J)
  To(L+1,J) = To(L,J)

```

```

To(L+2,J)= To(L-1,J)
Uxo(0,J)=-Uxo(1,J)
Uxo(-1,J)=-Uxo(2,J)
Uxo(L+1,J)=-Uxo(L,J)
Uxo(L+2,J)=-Uxo(L-1,J)
Uyo(0,J)=Uyo(1,J)
Uyo(-1,J)=Uyo(2,J)
Uyo(L+1,J)=Uyo(L,J)
Uyo(L+2,J)=Uyo(L-1,J)
No(0,J) = No(1,J)
No(-1,J) = No(2,J)
No(L+1,J) = No(L,J)
No(L+2,J)= No(L-1,J)
ENDDO
DO I = 1,L
To(I,0) = To(I,1)
To(I,-1) = To(I,2)
To(I,L+1)=To(I,L)
To(I,L+2)=To(I,L-1)
Uyo(I,0)=-Uyo(I,1)
Uyo(I,-1)=-Uyo(I,2)
Uyo(I,L+1)=-Uyo(I,L)
Uyo(I,L+2)=-Uyo(I,L-1)
Uxo(I,0)=Uxo(I,1)
Uxo(I,-1)=Uxo(I,2)
Uxo(I,L+1)=Uxo(I,L)
Uxo(I,L+2)=Uxo(I,L-1)
No(I,0) = No(I,1)
No(I,-1) = No(I,2)
No(I,L+1)=No(I,L)
No(I,L+2)=No(I,L-1)
ENDDO
DO I = -1,0
To(I,0) = To(I,1)
To(I,-1) = To(I,2)
To(I,L+1)=To(I,L)
To(I,L+2)=To(I,L-1)
Uyo(I,0)=-Uyo(I,1)
Uyo(I,-1)=-Uyo(I,2)
Uyo(I,L+1)=-Uyo(I,L)
Uyo(I,L+2)=-Uyo(I,L-1)
Uxo(I,0)=Uxo(I,1)
Uxo(I,-1)=Uxo(I,2)
Uxo(I,L+1)=Uxo(I,L)
Uxo(I,L+2)=Uxo(I,L-1)
No(I,0) = No(I,1)
No(I,-1) = No(I,2)
No(I,L+1)=No(I,L)
No(I,L+2)=No(I,L-1)
ENDDO
DO I = L+1,L+2
To(I,0) = To(I,1)
To(I,-1) = To(I,2)
To(I,L+1)=To(I,L)
To(I,L+2)=To(I,L-1)
Uyo(I,0)=-Uyo(I,1)
Uyo(I,-1)=-Uyo(I,2)
Uyo(I,L+1)=-Uyo(I,L)
Uyo(I,L+2)=-Uyo(I,L-1)
Uxo(I,0)=Uxo(I,1)
Uxo(I,-1)=Uxo(I,2)
Uxo(I,L+1)=Uxo(I,L)
Uxo(I,L+2)=Uxo(I,L-1)
No(I,0) = No(I,1)
No(I,-1) = No(I,2)
No(I,L+1)=No(I,L)
No(I,L+2)=No(I,L-1)

```



```
ENDDO
RETURN
END
```

```
C *****
C
C Esta subrotina impoe condicao de contorno tipo
C 'tobo continuo'
C
C *****

SUBROUTINE COND2(To,Uxo,Uyo,No,L,L1,Lx,Ly)
IMPLICIT NONE
Integer I,J,L,L1,Lx,Ly
DOUBLE PRECISION To(-1:L+2,-1:L+2),Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION No(-1:L+2,-1:L+2),Uyo(-1:L+2,-1:L+2)
DO I=L1,L1+Lx
    DO J=1,Ly
        Uxo(I,J)=0.0
        Uyo(I,J)=0.0
        To(I,J)=0.0
        No(I,J)=0.0
    ENDDO
ENDDO
DO I=1,L
    DO J=1,L
        To(I,J)=To(I,J)
    ENDDO
ENDDO
DO J=-1,L+2
    To(0,J) = To(L,J)
    To(-1,J) = To(L-1,J)
    To(L+1,J) = To(-1,J)
    To(L+2,J) = To(0,J)
    Uxo(0,J)=Uxo(L,J)
    Uxo(-1,J)=Uxo(L-1,J)
    Uxo(L+1,J)=Uxo(-1,J)
    Uxo(L+2,J)=Uxo(0,J)
    Uyo(0,J)=Uyo(L,J)
    Uyo(-1,J)=Uyo(L-1,J)
    Uyo(L+1,J)=Uyo(-1,J)
    Uyo(L+2,J)=Uyo(0,J)
    No(0,J) = No(L,J)
    No(-1,J) = No(L-1,J)
    No(L+1,J) = No(-1,J)
    No(L+2,J) = No(0,J)
END DO
DO I = -1,L+2
    To(I,0) = To(I,1)
    To(I,-1) = To(I,2)
    To(I,L+1)=To(I,L)
    To(I,L+2)=To(I,L-1)
    Uyo(I,0)=-Uyo(I,1)
    Uyo(I,-1)=-Uyo(I,2)
    Uyo(I,L+1)=-Uyo(I,L)
    Uyo(I,L+2)=-Uyo(I,L-1)
    Uxo(I,0)=Uxo(I,1)
    Uxo(I,-1)=Uxo(I,2)
    Uxo(I,L+1)=Uxo(I,L)
    Uxo(I,L+2)=Uxo(I,L-1)
    No(I,0) = No(I,1)
    No(I,-1) = No(I,2)
    No(I,L+1)=No(I,L)
    No(I,L+2)=No(I,L-1)
END DO
```

```

C      **** impoe condicoes especiais de contorno ****
      CALL SETCURVE(To,Uxo,Uyo,No,L,L1,Lx,Ly)
c      ****
      RETURN
      END

c      ****
c
c      Esta subrotina impoe condicoes de contorno
c      periodico
c
c      ****

      SUBROUTINE COND3(To,Uxo,Uyo,No,L)
      IMPLICIT NONE
      Integer I,J,L
      DOUBLE PRECISION To(-1:L+2,-1:L+2),Uxo(-1:L+2,-1:L+2)
      DOUBLE PRECISION No(-1:L+2,-1:L+2),Uyo(-1:L+2,-1:L+2)
      DO I=1,L
         DO J=1,L
            To(I,J)=To(I,J)
         ENDDO
      ENDDO
      DO J =-1,L+2
         To(0,J) = To(L,J)
         To(-1,J) = To(L-1,J)
         To(L+1,J) = To(1,J)
         To(L+2,J)= To(2,J)
         Uxo(0,J)=Uxo(L,J)
         Uxo(-1,J)=Uxo(L-1,J)
         Uxo(L+1,J)=Uxo(1,J)
         Uxo(L+2,J)=Uxo(2,J)
         Uyo(0,J)=Uyo(L,J)
         Uyo(-1,J)=Uyo(L-1,J)
         Uyo(L+1,J)=Uyo(1,J)
         Uyo(L+2,J)=Uyo(2,J)
         No(0,J) = No(L,J)
         No(-1,J) = No(L-1,J)
         No(L+1,J) = No(1,J)
         No(L+2,J)= No(2,J)
      ENDDO
      DO I = -1,L+2
         To(I,0) = To(I,L)
         To(I,-1) = To(I,L-1)
         To(I,L+1)=To(I,1)
         To(I,L+2)=To(I,2)
         Uyo(I,0)=Uyo(I,L)
         Uyo(I,-1)=Uyo(I,L-1)
         Uyo(I,L+1)=Uyo(I,1)
         Uyo(I,L+2)=Uyo(I,2)
         Uxo(I,0)=Uxo(I,L)
         Uxo(I,-1)=Uxo(I,L-1)
         Uxo(I,L+1)=Uxo(I,1)
         Uxo(I,L+2)=Uxo(I,2)
         No(I,0) = No(I,L)
         No(I,-1) = No(I,L-1)
         No(I,L+1)=No(I,1)
         No(I,L+2)=No(I,2)
      ENDDO
      RETURN
      END

c      ****

```

```

C      Esta subrotina simula uma barreira quadrada dentro do tubo
C
C      *****
SUBROUTINE SETCURVE(To,Uxo,Uyo,No,L,L1,Lx,Ly)
IMPLICIT NONE
INTEGER I,J,L,L1,Ly,Lx
DOUBLE PRECISION To(-1:L+2,-1:L+2),Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION No(-1:L+2,-1:L+2),Uyo(-1:L+2,-1:L+2)
DO J=-1,Ly
  To(L1,J) = To(L1-1,J)
  To(L1+1,J) = To(L1-2,J)
  To(L1+Lx,J) = To(L1+Lx+1,J)
  To(L1+Lx-1,J) = To(L1+Lx+2,J)
  Uxo(L1,J) = -Uxo(L1-1,J)
  Uxo(L1+1,J) = -Uxo(L1-2,J)
  Uxo(L1+Lx,J) = -Uxo(L1+Lx+1,J)
  Uxo(L1+Lx-1,J) = -Uxo(L1+Lx+2,J)
  Uyo(L1,J) = Uyo(L1-1,J)
  Uyo(L1+1,J) = Uyo(L1-2,J)
  Uyo(L1+Lx,J) = Uyo(L1+Lx+1,J)
  Uyo(L1+Lx-1,J) = Uyo(L1+Lx+2,J)
  No(L1,J) = No(L1-1,J)
  No(L1+1,J) = No(L1-2,J)
  No(L1+Lx,J) = No(L1+Lx+1,J)
  No(L1+Lx-1,J) = No(L1+Lx+2,J)
END DO
DO I=L1,L1+Lx
  To(I,Ly) = To(I,Ly+1)
  To(I,Ly-1) = To(I,Ly+2)
  Uxo(I,Ly) = Uxo(I,Ly+1)
  Uxo(I,Ly-1) = Uxo(I,Ly+2)
  Uyo(I,Ly) = -Uyo(I,Ly+1)
  Uyo(I,Ly-1) = -Uyo(I,Ly+2)
  No(I,Ly) = No(I,Ly+1)
  No(I,Ly-1) = No(I,Ly+2)
END DO
RETURN
END
C
C      *****
C      Esta subrotina imprime os valores das variaveis nos arquivos
C      de saida para analise. Esses valores saem a intervalos de tempo
C      iguais, dados pelo parametro STEP_FOTO.
C
C      *****
SUBROUTINE SAIDA(To,No,Nmedio,Uxo,Uyo,St,IT,L,ETA0,LBDA0,L1,Lx,Ly)
IMPLICIT NONE
INTEGER I,J,L,IT,L1,Lx,Ly
DOUBLE PRECISION To(-1:L+2,-1:L+2),No(-1:L+2,-1:L+2)
DOUBLE PRECISION Uxo(-1:L+2,-1:L+2), Uyo(-1:L+2,-1:L+2)
DOUBLE PRECISION St,Nmedio,ETA0,LBDA0
CHARACTER*50 arquivo
WRITE(Arquivo,'('temp&',f4.3,'&',f4.3,'&',I6.0''.dat''
*      )') ETA0,LBDA0,IT
OPEN(UNIT=10,FILE=Arquivo,STATUS='UNKNOWN')
WRITE(Arquivo,'('dens&',f4.3,'&',f4.3,'&',I6.0''.dat''
*      )') ETA0,LBDA0,IT
OPEN(UNIT=11,FILE=Arquivo,STATUS='UNKNOWN')
WRITE(Arquivo,'('vel&',f4.3,'&',f4.3,'&',I6.0''.dat''
*      )') ETA0,LBDA0,IT
OPEN(UNIT=12,FILE=Arquivo,STATUS='UNKNOWN')

```

```

DO I=1,L1-1
  DO J =1,Ly
    WRITE(10,*)I,' ',J, ' ',To(I,J)/St
    WRITE(11,*)I,' ',J,' ',No(I,J)/Nmedio
    WRITE(12,*)I,' ',J,' ',Uxo(I,J)/sqrt(St),
*      ' ',Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO
DO I=L1+Lx+1,L
  DO J =1,Ly
    WRITE(10,*)I,' ',J, ' ',To(I,J)/St
    WRITE(11,*)I,' ',J,' ',No(I,J)/Nmedio
    WRITE(12,*)I,' ',J,' ',Uxo(I,J)/sqrt(St),
*      ' ',Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO
DO I=1,L
  DO J =Ly+1,L
    WRITE(10,*)I,' ',J, ' ',To(I,J)/St
    WRITE(11,*)I,' ',J,' ',No(I,J)/Nmedio
    WRITE(12,*)I,' ',J,' ',Uxo(I,J)/sqrt(St),
*      ' ',Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO
CLOSE(10)
CLOSE(11)
CLOSE(12)
RETURN
END

```

C *****

C Esta subrotina imprime os valores das variaveis nos arquivos
C de saida para analise. Esses valores saem a intervalos de tempo
C iguais, dados pelo parametro STEP_FOTO, para uma suimulacao com
C condicao de contoeno tipo 'caixa fechada'

C *****

```

SUBROUTINE SAIDA1(To,No,Nmedio,Uxo,Uyo,St,IT,L,ETA0,
*   LBDA0,L1,Lx,Ly)
IMPLICIT NONE
INTEGER I,J,L,IT,L1,Lx,Ly,sc
DOUBLE PRECISION To(-1:L+2,-1:L+2),No(-1:L+2,-1:L+2)
DOUBLE PRECISION Uxo(-1:L+2,-1:L+2), Uyo(-1:L+2,-1:L+2)
DOUBLE PRECISION St,Nmedio,ETA0,LBDA0,Uxmax,Uymax
CHARACTER*50 arquivo
Real ITr
ITr=IT
Uxmax=0.0
Uymax=0.0
DO I=1,L
  DO J=1,L
    IF (ABS(Uxo(I,J)).gt.Uxmax) then
      Uxmax=abs(Uxo(I,J))
    END IF
    IF (ABS(Uyo(I,J)).gt.Uymax) then
      Uymax=ABS(Uyo(I,J))
    END IF
  END DO
END DO

WRITE(Arquivo,('temp',f5.3,'&',f5.3,'&',I5'.dat'
*   )) ETA0,LBDA0,IT
OPEN(UNIT=10,FILE=Arquivo,STATUS='UNKNOWN')

```

```

WRITE(Arquivo,('dens&',f5.3,'&',f5.3,'&',I5'.dat'
*   )') ETA0,LBDA0,IT
OPEN(UNIT=11,FILE=Arquivo,STATUS='UNKNOWN')
WRITE(Arquivo,('vel&',f5.3,'&',f5.3,'&',I5'.dat'
*   )') ETA0,LBDA0,IT
OPEN(UNIT=12,FILE=Arquivo,STATUS='UNKNOWN')
sc=1
DO I=1,L
  DO J =1,L
    WRITE(10,*)I,' ',J,          ' ',To(I,J)/St
    WRITE(11,*)I,' ',J,' ',No(I,J)/Nmedio
  ENDDO
ENDDO
DO I=1,L,sc
  DO J =1,L,sc
    WRITE(12,*)I,' ',J,' ',I+1.0d0*Uxo(I,J)/sqrt(St)
*      ,J+1.0d0*Uyo(I,J)/sqrt(St),
*      Uxo(I,J)/sqrt(St),Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO

DO I=-1,2,sc
  DO J = -1,2,sc
    WRITE(13,*)I,' ',J,' ',I+1.0d0*Uxo(I,J)/sqrt(St)
*      ,J+1.0d0*Uyo(I,J)/sqrt(St),
*      Uxo(I,J)/sqrt(St),Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO

DO I=1-1,1+2,sc
  DO J = 1-1,1+2,sc
    WRITE(13,*)I,' ',J,' ',I+1.0d0*Uxo(I,J)/sqrt(St)
*      ,J+1.0d0*Uyo(I,J)/sqrt(St),
*      Uxo(I,J)/sqrt(St),Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO

DO I=-1,2,sc
  DO J = 1-1,1+2,sc
    WRITE(13,*)I,' ',J,' ',I+1.0d0*Uxo(I,J)/sqrt(St)
*      ,J+1.0d0*Uyo(I,J)/sqrt(St),
*      Uxo(I,J)/sqrt(St),Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO

DO I=1-1,1+2
  DO J = -1,2
    WRITE(13,*)I,' ',J,' ',I+1.0d0*Uxo(I,J)/sqrt(St)
*      ,J+1.0d0*Uyo(I,J)/sqrt(St),
*      Uxo(I,J)/sqrt(St),Uyo(I,J)/sqrt(St)
  ENDDO
ENDDO
CLOSE(10)
CLOSE(11)
CLOSE(12)
RETURN
END

```

C *****

C Esta subrotina injeta energia no sistema
c via temperatura granular

C *****

SUBROUTINE INJECTENERGY(To,No,L,A,Nmo)

```

IMPLICIT NONE
INTEGER I,J,L,ISEED
DOUBLE PRECISION To(-1:L+2,-1:L+2),No(-1:L+2,-1:L+2),A,Nmo
REAL RAN1
DO I=1,L
  DO J=1,L
    To(I,J)=To(I,J)+A*(No(I,J)/Nmo)*RAN1(ISEED)
  ENDDO
END DO
RETURN
END

```

C *****

C Esta subrotina e um gerador de numeros aleatorios
C extraidos do Num.Rec.
C ISEED deve ser negativo.

C *****

```

FUNCTION ran1(idum)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INTEGER idum
REAL RAN1
INTEGER IA,IM,IQ,IR,NTAB,NDIV
PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836,
*          NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-9,RNMx=1.-EPS)
INTEGER j,k,iv(NTAB),iy
SAVE iv,iy
DATA iv /NTAB*0/, iy /0/
if (idum.le.0.or.iy.eq.0) then
  idum=max(-idum,1)
  do 11 j=NTAB+8,1,-1
    k=idum/IQ
    idum=IA*(idum-k*IQ)-IR*k
    if (idum.lt.0) idum=idum+IM
    if (j.le.NTAB) iv(j)=idum
11  continue
  iy=iv(1)
endif
k=idum/IQ
idum=IA*(idum-k*IQ)-IR*k
if (idum.lt.0) idum=idum+IM
j=1+iy/NDIV
iy=iv(j)
iv(j)=idum
ran1=min(AM*iy,RNMx)
return
END

```

C *****

C Esta subrotina calcula todas as derivadas em cada passo
C temporal

C *****

```

SUBROUTINE DERIVADAS (To,Uxo,Uyo,No,Uxdx,Uydy,Uxdxx,Uydxx,Uxdyy,
*                    Uydyy,Tdx,Tdy,Ndx,Ndy,Uydx,Uxdy,Uxdxy,Uydx,N1,N2,Tdxx,
*                    Tdyy,Ndxx,Ndyy,I,J,L)
IMPLICIT NONE
INTEGER L,I,J
DOUBLE PRECISION To(-1:L+2,-1:L+2)

```

```

DOUBLE PRECISION Uxo(-1:L+2,-1:L+2)
DOUBLE PRECISION Uyo(-1:L+2,-1:L+2)
DOUBLE PRECISION No(-1:L+2,-1:L+2)
DOUBLE PRECISION Uxdx,Uydy,Uxdxx,Uydxx,Uxdyy,Uydyd,Tdx,Tdy
DOUBLE PRECISION Ndx,Ndy,Uydx,Uxdy,Uxdxy,Uydxxy,N1,N2,Tdxx,Tdyy
DOUBLE PRECISION Ndx,Ndy
Uxdx = (8.0d0*(Uxo(I+1,J)-Uxo(I-1,J))-(Uxo(I+2,J)-
* Uxo(I-2,J)))/12.0d0
Uydy = (8.0d0*(Uyo(I,J+1)-Uyo(I,J-1))-(Uyo(I,J+2)-
* Uyo(I,J-2)))/12.0d0
Uydx = (8.0d0*(Uyo(I+1,J)-Uyo(I-1,J))-(Uyo(I+2,J)-
* Uyo(I-2,J)))/12.0d0
Uxdy = (8.0d0*(Uxo(I,J+1)-Uxo(I,J-1))-(Uxo(I,J+2)-
* Uxo(I,J-2)))/12.0d0
Uxdxx = (Uxo(I+1,J)+Uxo(I-1,J)-2.0d0*Uxo(I,J))
Uydyd = (Uyo(I,J+1)+Uyo(I,J-1)-2.0d0*Uyo(I,J))
Uydxx = (Uyo(I+1,J)+Uyo(I-1,J)-2.0d0*Uyo(I,J))
Uxdyy = (Uxo(I,J+1)+Uxo(I,J-1)-2.0d0*Uxo(I,J))
Uxdxy = (Uxo(I+1,J+1)+Uxo(I-1,J-1)-Uxo(I-1,J+1)
* -Uxo(I+1,J-1))/4.0d0
Uydxxy = (Uyo(I+1,J+1)+Uyo(I-1,J-1)-Uyo(I-1,J+1)
* -Uyo(I+1,J-1))/4.0d0
Tdx = (8.0d0*(To(I+1,J)-To(I-1,J))-(To(I+2,J)-
* To(I-2,J)))/12.0d0
Tdy = (8.0d0*(To(I,J+1)-To(I,J-1))-(To(I,J+2)-
* To(I,J-2)))/12.0d0
Ndx = (8.0d0*(No(I+1,J)-No(I-1,J))-(No(I+2,J)-
* No(I-2,J)))/12.0d0
Ndy = (8.0d0*(No(I,J+1)-No(I,J-1))-(No(I,J+2)-
* No(I,J-2)))/12.0d0
Tdxx = (To(I+1,J)+To(I-1,J)-2.0d0*To(I,J))
Tdyy = (To(I,J+1)+To(I,J-1)-2.0d0*To(I,J))
Ndx = (No(I+1,J)+No(I-1,J)-2.0d0*No(I,J))
Ndy = (No(I,J+1)+No(I,J-1)-2.0d0*No(I,J))
N1 = To(I-1,J)+To(I+1,J)+To(I,J-1)+To(I,J+1)
N2 = To(I-1,J-1)+To(I-1,J+1)+To(I+1,J-1)+To(I+1,J+1)
RETURN
END

```

```

C *****
C
C Fim do programa
C
C *****

```

Referências

- [1] H. M. Jaeger, S. R. Nagel and R. P. Behringer, *Phys. Today* **49** (1996) 32.
- [2] J. Schofield and I. Oppenheim, *Physica A* **196** (1993) 209-240.
- [3] W.A.M Morgado and I. Oppenheim, *Phys. Rev. E* **55** (1997) 1940-1945.
- [4] Sidney R. Nagel et al. *Nature* **414** (2001) : 270-270.
- [5] H. J. Herrmann, *Physica A* **313** (1-2) (2002) 188-210.
- [6] T. Shimbrot and F. J. Mozzio, *Phys. Today* **53**(3) (2000) 25-30.
- [7] Florence Rouyer and Narayanan Menon, *Phys. Rev. Letters* **85** (2000) 3676-3679.
- [8] Hans J. Herrmann, S. Luding and R. Caferio, *Physica A* **295** (2001) 93-100.
- [9] E. Livne, B. Meerson and P. V. Sasorov, *Phys. Rev. E* **65** (2002) 021602.
- [10] B. Meerson *et al* cond-mat/028286 (2002).
- [11] T. P. C. Van Noije and M. H. Ernst cond-mat/9803042 (1998).
- [12] Nikolai V. Brilliantov, Frank Spahn, Jan-Martin Hertzsch and Torsten Pöschel, *Phys. Rev. E* **53** (1996) 5382.
- [13] Rosa Ramírez, Thorten Pöschel, Nikolai V. Brilliantov and Thomas Schwager, *Phys. Rev. E* **60** (1999) 4465.
- [14] T. Schwager and T. Pöschel, *Phys. Rev. E* **57** (1998) 650.
- [15] D. Guban, *Am. J. Phys.* **68** (2000) 920-924.
- [16] W. A. M. Morgado and I. Oppenheim, *Phys. Rev. E* **55** (1997) 1940.
- [17] G. Kuwabara and K. Kono, *Ppn. J. Appl. Phys.* **26** (1987) 1230.
- [18] N. G. Van Kampen, *Stochastic Process in Physics and Chemistry* North-Holand Publishing Company (1981).
- [19] Proleon and W.A.M Morgado, Não publicado.
- [20] Huang, Kerson, "*Statistical Mechanics*", (1987) Jhon Wiley & Sons, (1987).

- [21] W.A.M Morgado and I. Oppenheim, *Physica A* **246** (1997) 547-562.
- [22] W.A.M Morgado and I. Oppenheim, *Physica A* **252** (1998) 308-324.
- [23] J. J. Brey, James W. Dufty, Chang Sub Kim, Andrés Santos, *Phys. Rev. E* **58** (1998) 4638.
- [24] W.A.M Morgado and E. Vernek, Submetido.
- [25] J. J. Brey, F. Moreno and J. W. Dufty, *Phys. Rev. E* **54** (1996) 445.
- [26] I. Goldhirsch and G. Zanetti, *Phys. Rev. Letters* **70** (1993) 1619-1622.
- [27] A. Puglisi, V. Loreto, U. Marini Bettolo Marconi and A. Vulpiani, *Phys. Rev. E* **59** (1999) 5582.
- [28] Alain Barrat and Emmanuel Trizac, *Phys. Rev. E* **65** (2002) 051303.
- [29] W.A.M Morgado and Eduardo R. Mucciolo, *Physica A* **311** (2002) 150-168.
- [30] T. Pöschel N. V. Brilliantov and T. Schwager, cond-mat 0212200 v1 (2002).
- [31] Y. Oono and S. Puri, *Phys. Rev. A* **38** (1988) 434-453.
- [32] S. Martins, W. A. M Morgado, M. S. O Massunaga and M. Bahiana, *Phys.Rev. E* **61** (2000) 4118.
- [33] A.Shinozaki and Y Oono, *Phys. Rev. E* **48** (1993) 2622.
- [34] A. Shinozaki, Thesis, University of Illinois at Urbana-Champaign, 1993.
- [35] J. J. Brey and M.J. Ruiz-Montero *Comp. Phys. Comm.* **121-122** (1999) 278-283.
- [36] S. Chappmam & T. G. Cowling, "*Mathematical Theory of Non-Uniform Gases*", Cambridge (1970).
- [37] A. Kudrolli and J. Henry, *Phys. Rev. E* **62** (2000) 1489-1492.
- [38] Erin C. Rericha Chris Bizon, Mark D. Shattuck and Harry L. Swinney, *Phys. Rev. Letters* **88** (2002) 014301
- [39] H. William *et al*, "*Numerical Recipes in Fortran 77*" - 2nd edition, Cambrigde University Press (1992)