Programação estruturada no Fortran 90 - 3

Alexandre Diehl

Departamento de Física – UFPel

Subprograma do tipo SUBROTINA

Considere um sistema com N partículas de diâmetro $\sigma = 1.0$. Construa um programa em FORTRAN 90 para inserir as N partículas numa caixa quadrada de lado $L = 10.0$ de forma aleatória.

As seguintes condições devem ser verificadas no programa: a) durante a inserção das partículas, a separação entre quaisquer duas partículas não pode ser menor do que σ. b) Caso não seja possível inserir as N partículas, o programa deve ser interrompido e uma mensagem de erro deve ser informada ao usuário. c) Caso seja possível inserir as N partículas, o programa deve escrever as coordenadas x e y num arquivo de saída.

Subprograma do tipo SUBROTINA

```
program insere_particulas
  implicit none
  integer :: i
  integer :: N, iseed
  integer :: tentativa_maxima
  real :: L, sigma
  real, allocatable :: x(:), y(:)

  read*,N, sigma
  read*,L
  read*,tentativa_maxima
  read*,iseed
  allocate(x(N),y(N))
  call inicializar()
  do i = 1,N
     print*,x(i),y(i)
  end do

CONTAINS

 [...]
```

- Programa principal claramente identificado.
- Usar Subprogramas (Subrotinas e Funções) para cálculos específicos.
- Cuidado com o uso de variáveis globais e locais.

Subprograma do tipo SUBROTINA

```
SUBROUTINE inicializar()
   integer :: i = 1, j, tentativa = 1
   real :: distancia
   logical :: overlap
   do while (i <= N)
      x(i) = ran2(iseed)*L;    y(i) = ran2(iseed)*L
      j = 1
      overlap = .false.
      do while (j < i .and. (.not.overlap))
         overlap = look_overlap(x(i),x(j),y(i),y(j))
         j = j + 1
      enddo
      if (overlap) then
         tentativa = tentativa + 1
         if (tentativa > tentativa_maxima) then
            write (*,'(A)') ' tentativas excedidas. Pare!'
            stop
         endif
      else
         tentativa = 1;       i = i + 1
      endif
   enddo
 end SUBROUTINE inicializar
```

Subprograma do tipo SUBROTINA

```
logical function look_overlap(x1,x2,y1,y2)
  implicit none
  real :: x1, x2, y1, y2
  real :: distancia

  look_overlap = .false.
  distancia = sqrt((x1-2)**2+(y1-y2)**2)

  if (distancia < sigma) then
     look_overlap = .true.
  endif

end function look_overlap
```

- A Função ran2 deve ser inserida dentro do bloco CONTAINS, pois é usada dentro da Subrotina de inicialização das coordenadas das partículas.