

Proyecto III: Dinámica Molecular

Desarrollo Experimental II

Integrantes:

Paredes Sosa Martín Alejandro

Orcí Fernández Luisa Fernanda

Mayo, 2018

Introducción

El objetivo principal de este proyecto fue el identificar las partes esenciales de un código de simulación de Dinámica Molecular NVT básico utilizando el algoritmo equivalente de Verlet para un sistema de átomos iguales, los cuales interactúan entre sí a través de un modelo de potencial central, en este caso se utilizó el de Lennard-Jones, el cual tiene la forma

$$u(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right] \quad (1)$$

donde σ es el diámetro efectivo y ϵ es el mínimo del potencial atractivo, usualmente se identifica como energía de enlace.

Metodología

La simulación Molecular se enfoca en resolver las ecuaciones de movimiento del sistema con N partículas, se pueden tomar dos enfoques: el de Newton, el cual consiste de N ecuaciones diferenciales de segundo orden o del Hamilton, el cual consiste en $2N$ ecuaciones diferenciales de primer orden.

El algoritmo comúnmente utilizado es el de Verlet (1967), este algoritmo es un método de solución numérica para la ecuación de Newton, tiene la siguiente forma

$$\vec{r}(t + \delta t) = 2\vec{r}(t) - \vec{r}(t - \delta t) + \vec{a}(t)(\delta t)^2 \quad (2)$$

Para calcular la energía cinética es necesario contar con una ecuación para el cálculo de la velocidad, la cual tiene la forma

$$\vec{v}(t) = \frac{1}{2}[\vec{r}(t' - \delta t) - \vec{r}(t + \delta t)] \quad (3)$$

Para facilitar la implementación del potencial al código se definen los siguientes parámetros reducidos

- Longitud:

$$r^* \equiv \frac{r}{\sigma}$$

- Energía:

$$E^* \equiv \frac{E}{\epsilon}$$

- Temperatura:

$$T^* \equiv \frac{k_b T}{\epsilon}$$

- Presión:

$$P^* \equiv \left(\frac{\sigma^3}{\epsilon}\right) P$$

- Fuerza:

$$F^* \equiv \frac{\sigma F}{\epsilon}$$

- Velocidad:

$$v^* \equiv \sqrt{\frac{m}{\epsilon}} v$$

- Tiempo:

$$t^* \equiv \sqrt{\frac{\epsilon}{m\sigma^2}} t$$

- Concentración:

$$n^* \equiv n\sigma^3$$

finalmente, el potencial de Lennard-Jones reducido nos queda de la forma

$$u^* = 4\epsilon^* \left[\left(\frac{1}{r^*}\right)^{12} - \left(\frac{1}{r^*}\right)^6 \right] \quad (4)$$

Código

```
1 ! Curso: Desarrollo Experimental II (2018-1)
2 ! Proyecto III:
3
4 PROGRAM DMLJ
5   IMPLICIT REAL*8 (a-h,o-z)
```

```

6  PARAMETER ( N=250)           !NUMERO DE PARTICULAS
7  PARAMETER ( NN2=5000)        !CONFIGURACIOS DE EQUILIBRIO A GUARDAR
8  PARAMETER ( NENER=50000)     !CONFIGURACIONES ANTES DE LA DE EQUILIBRIO
9  PARAMETER ( FREE=3.0)        !GRADOS DE LIBERTAD
10 PARAMETER ( PI=3.1415927)
11 REAL, EXTERNAL ::ZRAN        !GENERADOR DE NUMEROS ALEATORIOS
12
13 !BLOQUE CONTENEDORES (MODULO)
14 COMMON /BLOCK1/ RX, RY, RZ, VX, VY, VZ, FX, FY, FZ
15 COMMON /VALORES/ DENS,RCUT,BOX,NSTEP
16 COMMON /BLOCK2/ RXC, RYC, RZC
17
18 !INIZIALIZANDO ARREGLOS DE:
19 DIMENSION CX(n,nn2),CY(n,nn2),CZ(n,nn2) !ARREGLOS DE CONFIG G(r)
20 DIMENSION CXR(n,nn2),CYR(n,nn2),CZR(n,nn2) !ARREGLOS DE CONFGIF W(t) D(t)
21 DIMENSION FX(N), FY(N), FZ(N), VX(N), VY(N), VZ(N) !FUERZA Y VELOCIDADES
22 DIMENSION RX(N), RY(N), RZ(N), RXC(N), RYC(N), RZC(N) !POSICIONES
23
24 NSTEP=150000 !NUMERO DE PASOS
25 IPRINT=10000 !FRECUENCIA DE IMPRESION EN PANTALLA
26 NFREC=20     !FRECUENCIA DE GUARDADO DE CONFIGS
27 DT=0.0001    !PASO DE TIEMPO
28
29 DENS=0.6      !CONCENTRACION REDUCIDA
30 TEMP=1.5     !TEMPERATURA REDUCIDA
31
32 XM=1.0       !MASA PARTICULA
33 SIGMA=1.0    !DIAMETRO ||
34
35 !CALCULOS PREVIOS
36 A=1.0/3.0
37 BOX=(N/DENS)**A !Longitud DE CELDA
38 RCUT=BOX/2.0   !RADIO DE CORTE
39 TEMPI=TEMP     !TEMPERATURA INICIAL
40 KI2=0          !CONTADOR DE CONFIG GUARDADAS
41
42 !IMPRESION DE PARAMETROS DE SIMULACION
43 WRITE(*,*)'LENNARD-JONES'
44 WRITE(*, '(' NUMBER OF ATOMS = ',I10)') N
45 WRITE(*, '(' NUMBER OF STEPS = ',I10)') NSTEP
46 WRITE(*, '(' OUTPUT FREQUENCY = ',I10)') IPRINT
47 WRITE(*, '(' POTENTIAL CUTOFF = ',F10.4)') RCUT
48 WRITE(*, '(' DENSITY = ',F10.4)') DENS
49 WRITE(*, '(' RED. TEMPERATURE = ',F10.4)') TEMP
50 WRITE(*, '(' MASS = ',F10.4)') XM
51 WRITE(*, '(' TIME STEP = ',F10.6)') DT
52
53 !ABRIENDO ARCHIVOS
54 OPEN(15,FILE='cfdm0.dat',STATUS='UNKNOWN') !CONFIG FINAL

```

```

55 OPEN(12,FILE='vfdm0.dat',STATUS='UNKNOWN') !VELOCIDAD FINAL
56 OPEN(13,FILE='vidm0.dat',STATUS='UNKNOWN') !VELOCIDAD INICIAL
57 OPEN(14,FILE='tedm0.dat',STATUS='UNKNOWN') !TERMALIZACION
58
59 CALL CONFIGINI (BOX,RX,RY,RZ) !GENERA LA CONFIGURACION INICIAL
60 CALL COMVEL (TEMP)           !GENERA VELOCIDADES ALEATORIAS
61
62 DO INIV=1 , N
63
64     WRITE(13,*)VX(INIV),VY(INIV),VZ(INIV)
65
66 End Do
67
68 !INICIANDO EN CERO
69 ACV = 0.0
70 ACE = 0.0
71 ACP = 0.0
72 ACT = 0.0
73 ACVSQ = 0.0
74 ACESQ = 0.0
75 ACPSQ = 0.0
76 ACTSQ = 0.0
77 FLV = 0.0
78 FLE = 0.0
79 FLP = 0.0
80 FLT = 0.0
81
82 SR3 = ( SIGMA / RCUT ) ** 3
83 SR9 = SR3 ** 3
84 BOXCUB = 1.0/BOX** 3 !INVERSO DEL VOLUMEN DE LA CELDA
85 VLRC = ( 8.0 /9.0 ) * PI * DENS * REAL ( N )* ( SR9 - 3.0 * SR3) !CORRECCION LARGO ALCANCE
    ENERGIA
86 WLRC = ( 16.0 / 9.0 ) * PI * DENS * REAL ( N )* ( 2.0 * SR9 -3.0 * SR3 )!CORRECCION LARGO
    ALCANCE TERMINO VIRIAL PARA LA PRESION
87
88 CALL FORCE (SIGMA, RCUT, BOX, V, W )!CALCULO DE FUERZAS
89 WRITE(*,*)' STEP ', EN-MEC ', EN-CIN ', EN-POT ', PRES ', TEMP '
90
91 !CALCULANDO LAS CONFIG
92 DO ISTEP = 1, NSTEP
93     CALL MOVEA ( DT, XM ) !IMPLEMETA PARTE 1 USA CONDICIONES DE IMAGEN MIN
94     CALL FORCE ( SIGMA, RCUT, BOX, V, W )
95     CALL MOVEB ( DT, XM, XK )!SEGUNDA ETAPA DEL ALGORITMO
96
97     V = V + VLRC !AGREGANDO LARGO ALCANCE ENERGIA POTENCIAL
98     W = (W + WLRC)*BOXCUB !PRESION (DEL VIRIAL)
99     E = XK + V !ENERGIA MECANICA
100
101     VN = V / REAL ( N ) !ENEGIA POT POR PARTICULA

```

```

102  XKN = XK / REAL ( N ) !ENERGIA CINETICA PORPARTICULA
103  EN = E / REAL ( N ) !ENERGIA TOTAL POR PARTICULA
104  TEMP = 2.0 * XKN / FREE !TEMPERATURA
105  PRES = DENS * TEMP + W !PRESION
106
107  ! COMENTARIO LYR: TERMOSTATO PARA MANTENER LA TEMPERATURA DEL
108  ! SISTEMA CONSISTENTE CON LA TEMPERATURA DEL BANO TERMICO PARA
109  ! UNA DESCRIPCION NVT.
110  ALFA=SQRT(TEMPI/TEMP)
111
112  Do IS= 1, N , 1
113
114      VX(IS)=ALFA*VX(IS)
115      VY(IS)=ALFA*VY(IS)
116      VZ(IS)=ALFA*VZ(IS)
117
118  End Do
119
120  ! CONCLUYE COMDENTARIO LYR.
121
122  !MONITOREO EN PANTALLA
123  IF ( MOD( ISTEP, IPRINT ) .EQ. 0 ) THEN
124
125      WRITE(*,'(1X,I8,6(2X,F10.4))') ISTEP, EN, XKN, VN, PRES, TEMP
126
127  End If
128  !GUARDAR CONFIG FINAL Y VELOCIDADES FINAL
129  IF(ISTEP.EQ.NSTEP)THEN
130
131      DO JFIN=1,N
132
133          WRITE(15,*)RX(JFIN), RY(JFIN), RZ(JFIN)
134
135      End DO
136
137      DO JFIN=1,N
138
139          WRITE(12,*)VX(JFIN), VY(JFIN), VZ(JFIN)
140
141      End DO
142
143  ENDIF
144
145  !SALIDA A ARCHIVO DE TERMALIZACION
146  WRITE(14,*)ISTEP,EN,XKN,VN,PRES,TEMP
147
148  xmod=mod(ISTEP,nfrec)
149  if(xmod.eq.0.0 .and. ISTEP.GT.NENER)then
150      if(ISTEP.LE.NSTEP)then

```

```

151      ki2=ki2+1 !CONTADOR DE CONFIGURACIONES
152
153      ACE = ACE + EN
154      ACK = ACK + XKN
155      ACV = ACV + VN
156      ACP = ACP + PRES
157
158      ACESQ = ACESQ + EN ** 2
159      ACKSQ = ACKSQ + XKN ** 2
160      ACVSQ = ACVSQ + VN ** 2
161      ACPSQ = ACPSQ + PRES ** 2
162
163      !GUARDAR CONFIGURACIONES DE EQUILIBRIO
164      do i=1,n
165          CX(I,KI2)=RX(I)
166          CY(I,KI2)=RY(I)
167          CZ(I,KI2)=RZ(I)
168
169          CXR(I,KI2)=RXC(I)
170          CYR(I,KI2)=RYC(I)
171          CZR(I,KI2)=RZC(I)
172      End do
173  End If
174 End IF
175 End DO
176
177 XNORM = REAL ( KI2 )
178
179 AVE = ACE / XNORM
180 AVK = ACK / XNORM
181 AVV = ACV / XNORM
182 AVP = ACP / XNORM
183 ACESQ = ( ACESQ / XNORM ) - AVE ** 2
184 ACKSQ = ( ACKSQ / XNORM ) - AVK ** 2
185 ACVSQ = ( ACVSQ / XNORM ) - AVV ** 2
186 ACPSQ = ( ACPSQ / XNORM ) - AVP ** 2
187
188 IF ( ACESQ .GT. 0.0 ) FLE = SQRT ( ACESQ )
189 IF ( ACKSQ .GT. 0.0 ) FLK = SQRT ( ACKSQ )
190 IF ( ACVSQ .GT. 0.0 ) FLV = SQRT ( ACVSQ )
191 IF ( ACPSQ .GT. 0.0 ) FLP = SQRT ( ACPSQ )
192 AVT = AVK * 2.0 / FREE
193 FLT = FLK * 2.0 / FREE
194
195 WRITE(*, '(' AVE = ',F10.4)') AVE
196 WRITE(*, '(' FLE = ',F10.4)') FLE
197 WRITE(*, '(' AVV = ',F10.4)') AVV
198 WRITE(*, '(' FLV = ',F10.4)') FLV
199 WRITE(*, '(' AVP = ',F10.4)') AVP

```

```

200 WRITE(*, '(' FLP = ',F10.4)') FLP
201 WRITE(*, '(' AVT = ',F10.4)') AVT
202 WRITE(*, '(' FLT = ',F10.4)') FLT
203
204 !CALCULO DE PROPIEDADES
205 CALL GDR(CX,CY,CZ,KI2)
206 CALL WDT(CXR,CYR,CZR,KI2,DT,NFREC)
207
208 END program DMLJ
209 !
=====
210 !
=====
211 !
=====
212 SUBROUTINE FORCE (SIGMA, RCUT, BOX, V, W )
213   IMPLICIT REAL*8 (a-h,o-z)
214   PARAMETER ( N = 250 )
215   COMMON /BLOCK1/ RX, RY, RZ, VX, VY, VZ, FX, FY, FZ
216   DIMENSION RX(N), RY(N), RZ(N), VX(N), VY(N), VZ(N), FX(N),FY(N), FZ(N)
217
218   BOXINV = 1.0 / BOX !INVERSO DE LA LONGITUD DE LA CELDA
219   RCUTSQ = RCUT ** 2 !EL CUADRADO DEL RADIO DE CORTE
220   SIGSQ = SIGMA ** 2 !TAMBIEN EL DIAMETRO
221   EPS4 = 4.0
222   EPS24 = 24.0
223
224   !TODO A CEROS
225   Do I = 1, N
226     FX(I) = 0.0
227     FY(I) = 0.0
228     FZ(I) = 0.0
229   End Do
230
231   V = 0.0
232   W = 0.0
233
234   DO I = 1, N - 1
235
236     RXI = RX(I)
237     RYI = RY(I)
238     RZI = RZ(I)
239     FXI = FX(I)
240     FYI = FY(I)
241     FZI = FZ(I)
242

```

```

243 DO J = I + 1, N
244     !CALCULO DE DISTANCIAS CON IMAGEN MINIMA
245     RXIJ = RXI - RX(J)
246     RYIJ = RYI - RY(J)
247     RZIJ = RZI - RZ(J)
248     RXIJ = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
249     RYIJ = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
250     RZIJ = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX
251     RIJSQ = RXIJ ** 2 + RYIJ ** 2 + RZIJ ** 2
252
253     !CHECANDO INTERACION
254     IF ( RIJSQ .LT. RCUTSQ ) THEN
255
256         !POTENCIAL Y INFO PARA EL CALCULO DE FUERZAS
257         SR2 = SIGSQ / RIJSQ
258         SR6 = SR2 * SR2 * SR2
259         SR12 = SR6 ** 2
260
261         VIJ = SR12 - SR6
262         V = V + VIJ
263         WIJ = VIJ + SR12
264         W = W + WIJ
265         FIJ = WIJ / RIJSQ
266         FXIJ = FIJ * RXIJ
267         FYIJ = FIJ * RYIJ
268         FZIJ = FIJ * RZIJ
269
270         FXI = FXI + FXIJ
271         FYI = FYI + FYIJ
272         FZI = FZI + FZIJ
273         FX(J) = FX(J) - FXIJ
274         FY(J) = FY(J) - FYIJ
275         FZ(J) = FZ(J) - FZIJ
276     End If
277 End Do
278 FX(I) = FXI
279 FY(I) = FYI
280 FZ(I) = FZI
281 End Do
282
283 Do I = 1, N
284     FX(I) = FX(I) * EPS24
285     FY(I) = FY(I) * EPS24
286     FZ(I) = FZ(I) * EPS24
287 End Do
288
289 V = V * EPS4
290 W = W * EPS24 / 3.0
291

```



```

292 !RETURN
293 END SUBROUTINE FORCE
294 !
=====
295 !
=====
296 !
=====
297 SUBROUTINE MOVEA ( DT, XM )
298   IMPLICIT REAL*8 (a-h,o-z)
299   PARAMETER ( N = 250 )
300   COMMON /BLOCK1/ RX, RY, RZ, VX, VY, VZ, FX, FY, FZ
301   COMMON /BLOCK2/ RXC, RYC, RZC
302   COMMON /VALORES/ DENS,RCUT,BOX,NSTEP
303
304   DIMENSION RX(N), RY(N), RZ(N), RXC(N), RYC(N), RZC(N)
305   DIMENSION VX(N), VY(N), VZ(N), FX(N), FY(N), FZ(N)
306
307   DT2 = DT / 2.0
308   DTSQ2 = DT * DT2
309
310   DO I = 1, N
311
312     !PROPIEDADES ESTRUCTURALES
313     RX(I) = RX(I) + DT * VX(I) + DTSQ2 * FX(I) / XM
314     RY(I) = RY(I) + DT * VY(I) + DTSQ2 * FY(I) / XM
315     RZ(I) = RZ(I) + DT * VZ(I) + DTSQ2 * FZ(I) / XM
316
317     !PROPIEDADES DINAMICAS
318     RXC(I) = RXC(I) + DT * VX(I) + DTSQ2 * FX(I) / XM
319     RYC(I) = RYC(I) + DT * VY(I) + DTSQ2 * FY(I) / XM
320     RZC(I) = RZC(I) + DT * VZ(I) + DTSQ2 * FZ(I) / XM
321
322     !IMAGEN MINIMA
323     RX(I)=RX(I)-BOX*ANINT(RX(I)/BOX)
324     RY(I)=RY(I)-BOX*ANINT(RY(I)/BOX)
325     RZ(I)=RZ(I)-BOX*ANINT(RZ(I)/BOX)
326
327     !PRIMERA ETAPA DEL ALGORITMO
328     VX(I) = VX(I) + DT2 * FX(I) / XM
329     VY(I) = VY(I) + DT2 * FY(I) / XM
330     VZ(I) = VZ(I) + DT2 * FZ(I) / XM
331   End Do
332
333 END SUBROUTINE MOVEA

```

```

334 !
=====
335 !
=====
336 !
=====
337 SUBROUTINE MOVEB ( DT, XM, XK )
338   IMPLICIT REAL*8 (a-h,o-z)
339   PARAMETER ( N = 250 )
340   COMMON /BLOCK1/ RX, RY, RZ, VX, VY, VZ, FX, FY, FZ
341   DIMENSION RX(N), RY(N), RZ(N)
342   DIMENSION VX(N), VY(N), VZ(N), FX(N), FY(N), FZ(N)
343
344   DT2 = DT / 2.0
345   XK = 0.0
346
347   Do I = 1, N
348
349       !SEGUNDA ETAPA DEL ALGORITMO
350       VX(I) = VX(I) + DT2 * FX(I) / XM
351       VY(I) = VY(I) + DT2 * FY(I) / XM
352       VZ(I) = VZ(I) + DT2 * FZ(I) / XM
353
354       XK = XK + VX(I) ** 2 + VY(I) ** 2 + VZ(I) ** 2 !CUADRADO DE LA VELOCIDAD
355   End Do
356
357   XK = 0.5 * XM * XK !ENERGIA CINETICA
358
359 END SUBROUTINE MOVEB
360 !
=====
361 !
=====
362 !
=====
363 SUBROUTINE COMVEL ( TEMP )
364   IMPLICIT REAL*8 (a-h,o-z)
365   PARAMETER ( N = 250 )
366   COMMON /BLOCK1/ RX, RY, RZ, VX, VY, VZ, FX, FY, FZ
367   COMMON /semillas/iseed3,iseed2,iseed1
368   DIMENSION RX(N), RY(N), RZ(N)
369   DIMENSION VX(N), VY(N), VZ(N), FX(N), FY(N), FZ(N)
370

```

```

371 !SEMILLAS
372 ISEED = 43560
373 ISEED1= 39467
374 ISEED2= 148420
375 ISEED3= 7845901
376
377 !NUMERO ALEATORIO GAUSSIANO
378 CALL AZARG(iseed,AX)
379 CALL AZARG(iseed,AY)
380 CALL AZARG(iseed,AZ)
381
382 RTEMP = SQRT ( TEMP ) !RAIZ DE INTERVALO TEMPORAL
383
384 !FIJA VEL ALEATORIAS
385 DO I = 1, N
386     VX(I) = RTEMP * AX
387     VY(I) = RTEMP * AY
388     VZ(I) = RTEMP * AZ
389 End DO
390
391 SUMX = 0.0
392 SUMY = 0.0
393 SUMZ = 0.0
394
395 !SUMA DE LAS VELOCIDADES
396 DO I = 1, N
397     SUMX = SUMX + VX(I)
398     SUMY = SUMY + VY(I)
399     SUMZ = SUMZ + VZ(I)
400 End DO
401
402 !PROMEDIO DE LAS VELOCIDADES
403 SUMX = SUMX / REAL ( N )
404 SUMY = SUMY / REAL ( N )
405 SUMZ = SUMZ / REAL ( N )
406
407 !DESVIACION DE LA MEDIA
408 DO 300 I = 1, N
409     VX(I) = VX(I) - SUMX
410     VY(I) = VY(I) - SUMY
411     VZ(I) = VZ(I) - SUMZ
412 End DO
413
414 END SUBROUTINE COMVEL
415 !
=====
416 !
=====

```

```

417 !
=====
418 SUBROUTINE CONFIGINI (BOX,RX,RY,RZ)
419   IMPLICIT REAL*8 (a-h,o-z)
420   PARAMETER ( N = 250 )
421   COMMON /BLOCK2/ RXC, RYC, RZC
422
423   DIMENSION RX(N), RY(N), RZ(N), RXC(N), RYC(N), RZC(N)
424   DIMENSION X(N), Y(N), Z(N)
425
426   OPEN(11,FILE='cidm0.dat',STATUS='UNKNOWN')
427
428   NP=N !PARTICULAS (NO LO CONSIDERO NECESARIO)
429   A=1.0/3.0
430   ISEED1=456808
431   ISEED2=780
432   ISEED3=7598
433
434   DO I=1,NP
435
436     !POSICION ALEATORIA DENTRO DE LA CELDA
437 2    R=zran(iseed1)-0.5d0
438     S=zran(iseed2)-0.5D0
439     T=zran(iseed3)-0.5D0
440
441     !GUARDANDO POSICION
442     X(I)=R*BOX
443     Y(I)=S*BOX
444     Z(I)=T*BOX
445
446     !CHECANDO POR TRASLAPES
447     DO J=1 , I-1
448
449         xij=X(I)-X(J)
450         yij=Y(I)-Y(J)
451         zij=Z(I)-Z(J)
452
453         RO=(xij)**2+(yij)**2+(zij)**2
454
455         IF (RO.LE.1.0) THEN
456
457             WRITE(*,*)'traslape',I,J
458             GO TO 2
459
460         ENDIF
461
462     End DO

```

```

463      !GUARDANDO POSICIONES EN LOS BLOQUES
464      RX(I)=X(I)
465      RY(I)=Y(I)
466      RZ(I)=Z(I)
467      RXC(I)=X(I)
468      RYC(I)=Y(I)
469      RZC(I)=Z(I)
470      WRITE(11,*)SNGL(RX(I)),SNGL(RY(I)),SNGL(RZ(I))
471
472      End DO
473
474  END SUBROUTINE CONFIGINI
475  !
476  !
477  !
478  SUBROUTINE GDR(CX,CY,CZ,KI)
479      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
480      PARAMETER ( N = 250 )
481      PARAMETER(NN2=5000)
482      PARAMETER(NN3=3500)
483      INTEGER NHIST(NN3)
484      COMMON /VALORES/ DENS,RCUT,BOX,NSTEP
485      DIMENSION CX(n,nn2),CY(n,nn2),CZ(n,nn2)
486
487      NP=N
488
489      NHIST=0
490
491      DELTAR=0.01E0 !TAMANO DE LA CINTA
492      MAXBIN=INT(RCUT/DELTAR) !NUMERO DE CINTAS
493      PI=3.141592
494      NTMAX=KI !NUMERO DE CONFIGURACIONES
495
496      DO 20 L=1,NP
497
498          DO 25 M=1,NP
499
500              IF (M.EQ.L) GOTO 25
501
502              DO 40 J=1,NTMAX
503
504                  !CONFIGURACIONES
505                  XLO=CX(L,J)

```

```

506      XLT=CX(M,J)
507      XLOT=XLO-XLT
508
509      YLO=CY(L,J)
510      YLT=CY(M,J)
511      YLOT=YLO-YLT
512
513      ZLO=CZ(L,J)
514      ZLT=CZ(M,J)
515      ZLOT=ZLO-ZLT
516
517      !IMAGEN MINIMA
518      XLOT=XLOT-BOX*ANINT(XLOT/BOX)
519      YLOT=YLOT-BOX*ANINT(YLOT/BOX)
520      ZLOT=ZLOT-BOX*ANINT(ZLOT/BOX)
521      ROT=SQRT(XLOT**2+YLOT**2+ZLOT**2)
522      NBIN=INT(ROT/DELTAR)+1
523
524      IF (NBIN.LE.MAXBIN) THEN
525          NHIST(NBIN)=NHIST(NBIN)+1
526      ENDIF
527
528      End Do
529  End Do
530 End DO
531
532 C1=(4.0/3.0)*(PI*DENS) !PRECALCULO PARA G(r)
533
534 OPEN(60,FILE='grdm0.dat',STATUS='UNKNOWN')
535
536 DO 30 NBIN=1,MAXBIN
537
538     RL=REAL(NBIN-1)*DELTAR !CINTA INFERIOR
539     RU=RL+DELTAR           !SUPERIOR
540     RT=RL+DELTAR/2.0       !MEDIA
541     C2=C1*(RU**3-RL**3)
542     GDRTA=REAL(NHIST(NBIN))/REAL(NTMAX)/REAL(NP)/C2 !CALCULO DE GDR
543     WRITE(60,*)SNGL(RT),SNGL(GDRTA)
544
545 End DO
546
547 CLOSE(60)
548 RETURN
549 END SUBROUTINE GDR
550
551 !
=====

```

```

552 !
=====
553 !
=====
554
555 SUBROUTINE WDT(CX,CY,CZ,KI,DT,NFREC)
556   PARAMETER (N=250)
557   PARAMETER (NN2=5000)
558   IMPLICIT REAL*8 (A-H,O-Z)
559   DIMENSION CX(N,NN2),CY(N,NN2),CZ(N,NN2)
560   COMMON /VALORES/ DENS,RCUT,BOX,NSTEP
561
562   TIM=REAL(NFREC)*DT
563
564   open(50,file='wtm0.dat',STATUS='UNKNOWN')
565
566   DO I=1, KI-1
567     NTMAX=KI-I
568     WTX=0.d0
569     WTY=0.d0
570     WTZ=0.d0
571     WT= 0.d0
572
573     DO L=1,N
574       DO J=1,NTMAX
575         !CONFIGS
576         WTX=WTX+( CX(L,I+J)-CX(L,J) )**2
577         WTY=WTY+( CY(L,I+J)-CY(L,J) )**2
578         WTZ=WTZ+( CZ(L,I+J)-CZ(L,J) )**2
579
580       END DO
581     END DO
582     !CALCULO PROP DINAMICAS
583     TIME=TIM*REAL(I)
584     WT=(WTX+WTY+WTZ)/REAL(NTMAX)/REAL(N)/6.DO
585     DIF=WT/TIME
586     WRITE(50,*)TIME,WT,DIF
587     if(time.gt.10)goto 11
588
589   END DO
590 11 CLOSE(50)
591
592 END SUBROUTINE WDT
593 !
=====

```

```

594 !
=====
595 !
=====
596
597 ! GENERADOR DE NUMEROS ALEATORIOS CON DISTRIBUCION UNIFORME
598 ! computers in physics
599 ! vol. 8, No. 1 (1994) pag.117
600
601 FUNCTION ZRAN(ISEED)
602   implicit real*8 (a-h,o-z)
603   common/semillas/iseed3,iseed2,iseed1
604   mzzran=iseed3-iseed1
605
606   if(mzzran.lt.0) mzzran=mzzran+2147483579
607   iseed3=iseed2
608   iseed2=iseed1
609   iseed1=mzzran
610   iseed=ishft(3533*ishft(iseed,-16)+iand(iseed,125535),16)+3533*iand(iseed,65535)
611   mzzran=mzzran+iseed
612   zran=.5+.2328306d-9*mzzran
613
614   RETURN
615 END FUNCTION ZRAN
616 !
=====
617 !
=====
618 !
=====
619 ! GENERADOR DE NUMEROS ALEATORIOS CON DISTRIBUCION GAUSSIANA
620 SUBROUTINE AZARG( ISEED,X )
621   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
622   external zran
623   common/semillas/iseed3,iseed2,iseed1
624
625   pi=4.0*atan(1.0)
626   R=zran(iseed)
627   S=zran(iseed)
628   X=SQRT(-2.0*LOG(R))*COS(2.0*PI*S)
629
630   RETURN
631 END SUBROUTINE AZARG

```


Resultados

Referencias

- [1] Yeomans, Laura *Guía del curso sobre simulación de dinámica browniana y algoritmo de Verlet (1982)*, Desarrollo experimental, semestre 2018-1.