

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/354224535>

Introducción a Lammmps

Presentation · August 2021

DOI: 10.13140/RG.2.2.35111.27042

CITATIONS

0

READS

197

1 author:



[Gabriel Oscar Ibáñez-García](#)

31 PUBLICATIONS **31 CITATIONS**

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Coated Nanoparticles and Grafted colloids self-assembly [View project](#)

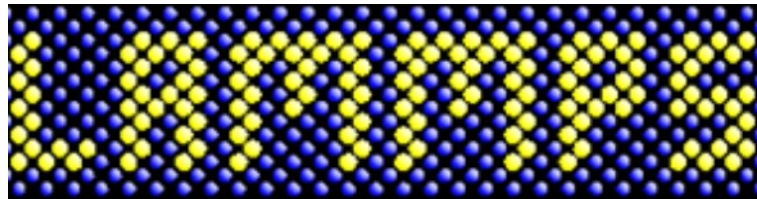


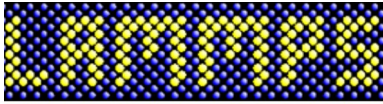
Non-Fickian penetrant diffusion in polymer membranes [View project](#)

Introducción a LAMMPS

Gabriel Oscar Ibáñez-García

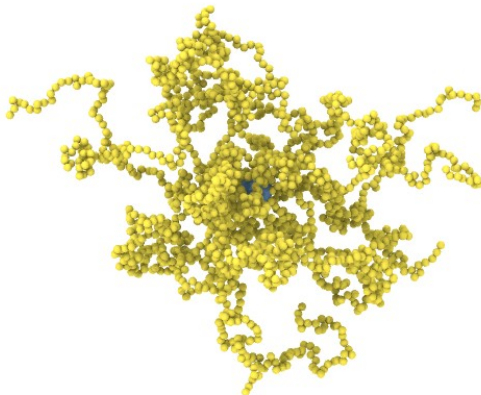
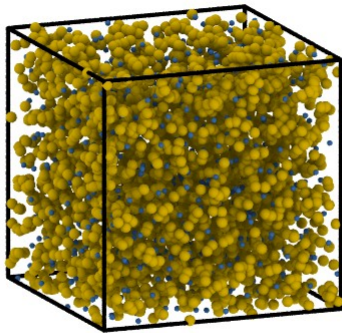
Agosto de 2021



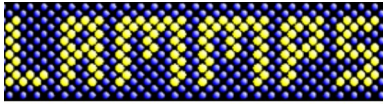


LAMMPS

Large-scale Atomic/Molecular Massively Parallel Simulator

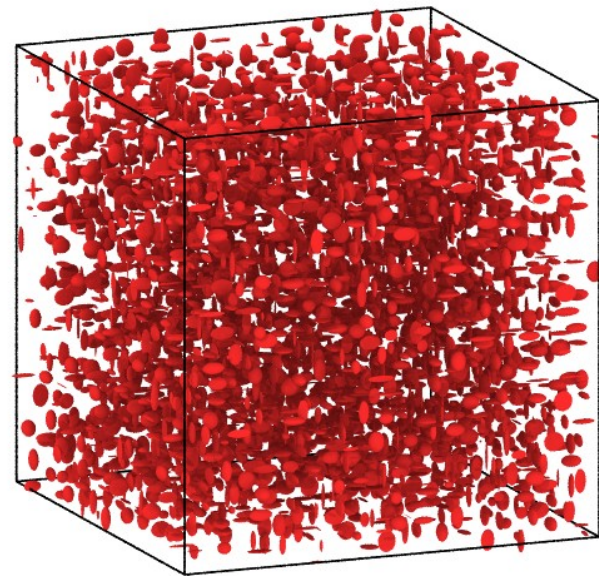
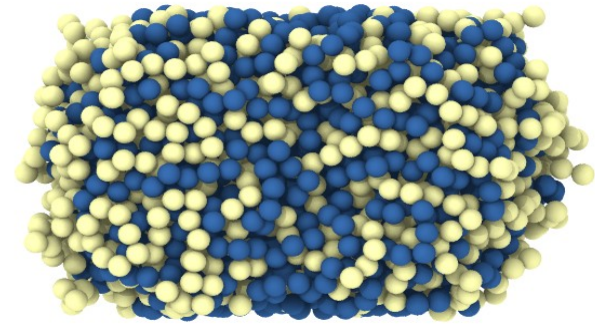


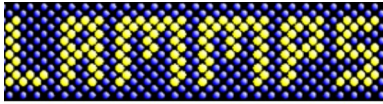
- Fue diseñado en los Lab. Sandia, por Steve Plimpton, Aidan Thompson, Stan Moore y otros autores.
- Simulación atomística de la Materia Condensada, Simulaciones de Granos Grueso, Métodos Mesoscópicos, Monte Carlo, Elemento finito entre otros.
- *Materia Suave en Métodos Mesoscópicos: Brownian Dynamics, Lattice Boltzmann, Dissipative Particle Dynamics, Stochastic Rotational dynamics.*
- “Paquete simula casi todo sistema”.
- Escrito en C++.
- Basado en scripts/comandos en línea/librería dinámica de python.



Características generales de LAMMPS

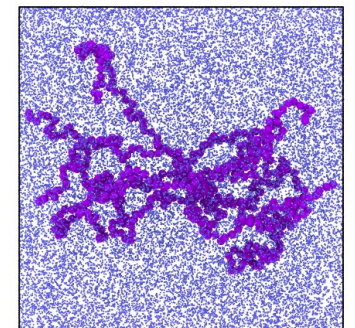
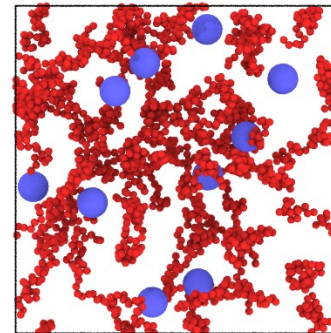
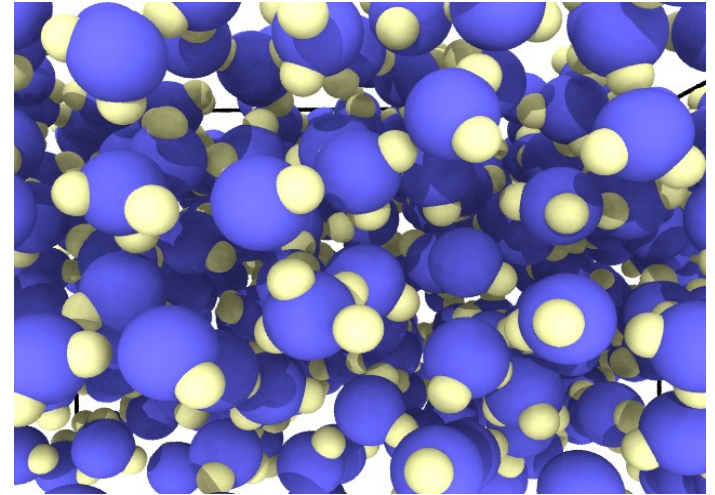
- Tiene una gran variedad de clases de átomos de diferente naturaleza: átomos semi/ cuánticos, clásicos, granogrueso y modelos híbridos .
- Potenciales Atomísticos OPLS, CHARMM, AMBER.
- Potenciales de Grano grueso, Tabulados.
- Dinámica molecular, MonteCarlo, Optimización, Elemento finito, dinámica granular.
- Diferentes potenciales.
- Ensamblajes termodinámicos: NVE, NVT, NPT, NHT, Langevin, DPD.
- Dinámica mesoscópica SRD, LB.
- Minimización.
- Nuevos métodos y características incluidos por lo usuarios.

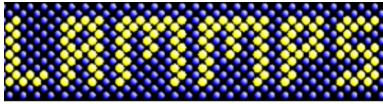




Tipos de “átomos” en LAMMPS

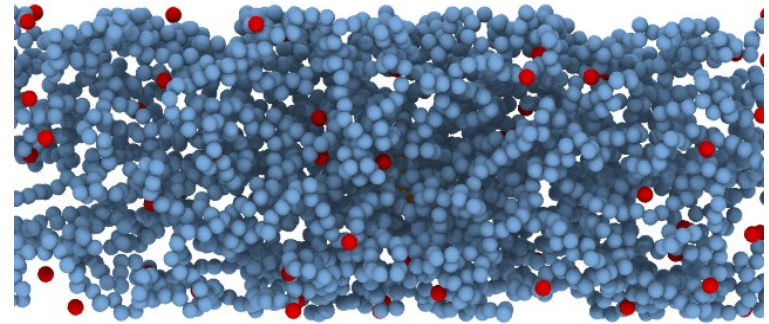
- Modelos de átomos reales y átomos de grano grueso.
- Moléculas orgánicas e inorgánicas.
- Dinámica molecular atomística de moléculas orgánicas, proteínas, DNA.
- Modelos de grano grueso mesoescala (SRD, DPD, LB).
- Partículas puntuales, esféricas y no esféricas (i. e. elipsoidales) .
- Segmentos lineales finitos y triangulares en 2D y 3D
- Partículas cargadas, dipolares.
- Colecciones rígidas de partículas:
- Modelos híbridos combinando las anteriores.

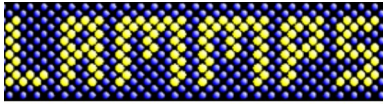




Estructura básica del software

- El código es libre, precompilado en varias plataformas ([UNIX](#), [Windows](#), [MAC](#)).
- Se puede también compilar el código haciendo diferentes “máquinas” para arquitecturas según el tipo de supercomputadora, procesadores o tarjetas gráficas. Formado por varios paquetes no preinstalados con diferentes propósitos: [molecule](#), [manybody](#), [misc](#), [python](#), [kspace](#), [dpd](#), [colloid](#), [kspace](#), [asphere](#), [python](#), [dpd](#), [gpu](#), [omp](#), etc..
- También puede funcionar en computadoras con un solo procesador o multinúcleos, estaciones de trabajo, pcs y macs.
- Mayor desempeño en clusters y supercomputadoras: cálculo en paralelo utilizando librerías de [MPI](#), [OPENMPI](#) en procesadores multinúcleos y gpus.
- Puede utilizar paquetes aceleradores para el cálculo de interacciones y listas de vecinos con los paquetes : [GPU](#), [CUDA](#), [OPENMPI](#), [KOKKOS](#), [IBM](#), etc.
- Emplea librerías opcionales tales como [FFT](#), [JPEG](#), [Vor++](#).
- Se vincula con otros códigos como [Quantum Espresso](#), [MSCG](#) por su función como librería dinámica de python.
- Cálculos AB initio/ LAMMPS cálculos clásicos.

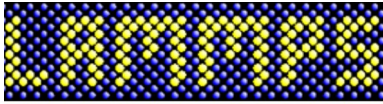




LAMMPS no fue diseñado para:



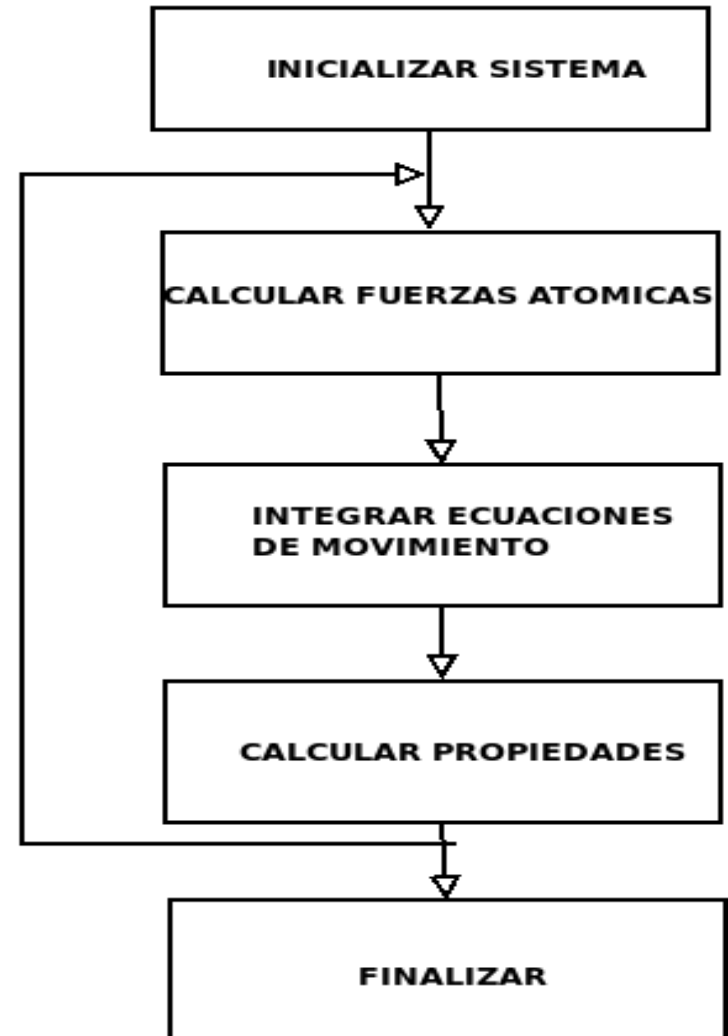
-
- Funcionar a través de una GUI.
 - Construir o diseñar sistemas atómicos.
 - Visualizar simulaciones de forma interactiva.
 - Graficar datos de salida.
 - Realizar análisis de datos post-simulación.
 - Pizza.py (python)
 - LAMMPS tools, Packmol, Moltemplate, códigos en python, C, Matlab etc...
 - Ovito, VMD, Topotools, Paraview, OpenEye, etc..
 - MDAnalysis, python Matplotlib, Matlab, Octave, Gnuplot etc..
 - Python, C, C++, etc..
 - PyLAMMPS, PyLat-Master otros paquetes.

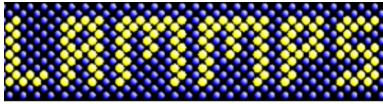


Dinámica molecular

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -\nabla_{\mathbf{r}_i} V(\mathbf{r}_i)$$

$$\begin{aligned} V(\mathbf{r}_1 \dots \mathbf{r}_N) = & \sum_{ij} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \\ & + \sum_{ij} \frac{q_i q_j}{4\pi\epsilon_{ij} r_{ij}} \\ & + \sum_{bond} K_b (r_{ij} - r_0)^2 \\ & + \sum_{angle} K_\theta (\theta_{ij} - \theta_0)^2 \\ & + \sum_{torsion} K_\phi [1 + \cos(n\phi_{ijkl})] \\ & + \sum_{dihedral} K_\omega (\omega_{ij} - \omega_0)^2 \end{aligned}$$





Potenciales de LAMMPS

- Atomísticos: Campos de fuerza clásicos:
OPLS, AMBER, CHARMM, Dreiding

- De Grano grueso.
- Tabulados/AB initio.

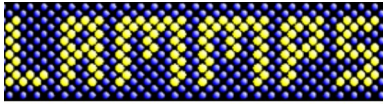
Interacciones intramoleculares:

- Enlaces: Armónicos
- Angulo entre enlaces: Armónicos coseno.
- Torsión: Armónicos seno y coseno.

Interacciones intermoleculares

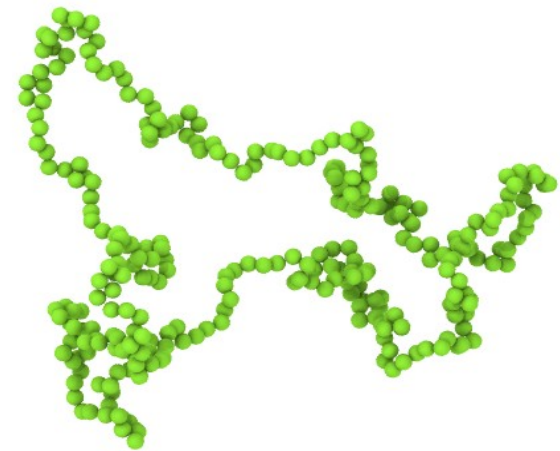
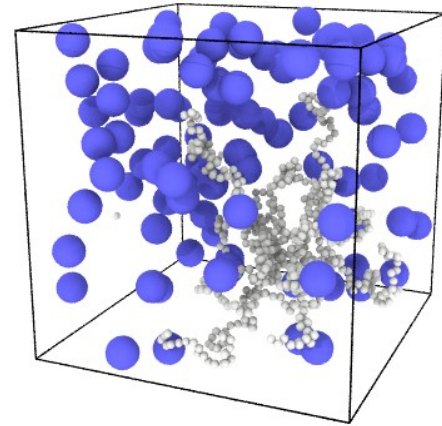
- Atracción/repulsión: Lennard-Jones o Coulomb.

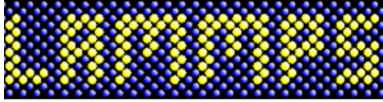
$$\begin{aligned} V(\mathbf{r}_1 \dots \mathbf{r}_N) = & \sum_{ij} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \\ & + \sum_{ij} \frac{q_i q_j}{4\pi\epsilon_{ij} r_{ij}} \\ & + \sum_{bond} K_b (r_{ij} - r_0)^2 \\ & + \sum_{angle} K_\theta (\theta_{ij} - \theta_0)^2 \\ & + \sum_{torsion} K_\phi [1 + \cos(n\phi_{ijkl})] \\ & + \sum_{dihedral} K_\omega (\omega_{ij} - \omega_0)^2 \end{aligned}$$



Instrucciones de LAMMPS

- Atom_Styles
- Boundary
- Create atoms /read_data file
- Potentials: Pair_style,Pair_Coeff
- Bond, Angle, Dihedral
- Minimization
- fix, compute, create, mass, lattice, minimize, velocity, region, set, group..
- thermodynamic output
- temp, rdf, msd, vacf, com, gyration
- variables/print
- thermo, dump
- run
- restart/write_data





Configuración inicial

- Configuración inicial simple: usando arreglos de partículas con instrucciones de LAMMPS

(sistemas simples en latiz cubica, hexagonal, sq etc..)

`lattice fcc/hexa/sq`

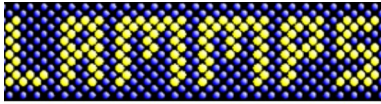
`create: box, atoms`

`mass, velocity`

- Archivos iniciales escritos por el usuario utilizando herramientas externas (la gran mayoría de los casos 99%). Python, C, C++, Moltemplate.

`read_data name.data`

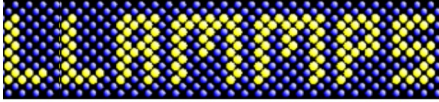
- Formato del archivo.
- 1) Encabezado: numero y tipos de átomos, numero de enlaces, ángulos dihedral, tipos de átomos, enlaces, ángulos dihedrals.
- Dimension de la caja de simulación.
- Masas/Cargas.
- Coeficientes de los Potenciales
- 2)Encabezado/Tipo de Atomo
- Coordenadas/Velocidades/Enlaces/Angulos Dihedrals/Impropios (siguiendo un formato específico para cada tipo de atomo de LAMMPS).



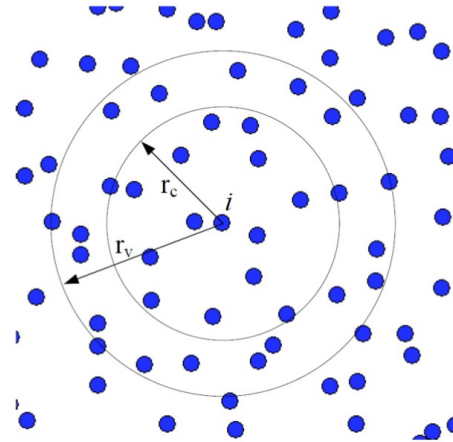
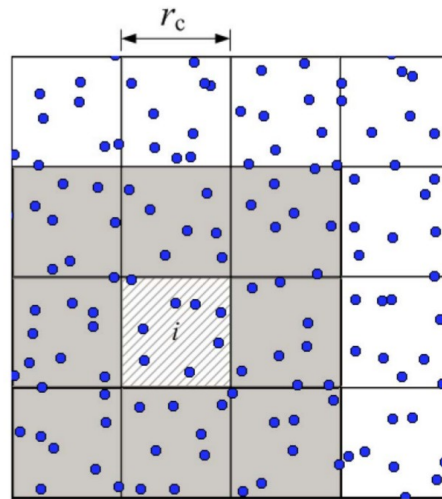
Tipos de Atomos en LAMMPS

- Atomic
- Charge
- Full
- Bond
- Angle
- Sphere
- Ellipsoid
- Asphere
- Body
- Dipole
- Line
- Meso
- Tri
- Hybrid

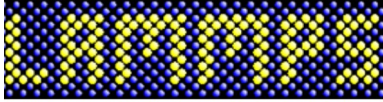
| | |
|------------|--|
| angle | atom-ID molecule-ID atom-type x y z |
| atomic | atom-ID atom-type x y z |
| body | atom-ID atom-type bodyflag mass x y z |
| bond | atom-ID molecule-ID atom-type x y z |
| charge | atom-ID atom-type q x y z |
| dipole | atom-ID atom-type q x y z mux muy muz |
| electron | atom-ID atom-type q spin eradius x y z |
| ellipsoid | atom-ID atom-type ellipsoidflag density x y z |
| full | atom-ID molecule-ID atom-type q x y z |
| line | atom-ID molecule-ID atom-type lineflag density x y z |
| meso | atom-ID atom-type rho e cv x y z |
| molecular | atom-ID molecule-ID atom-type x y z |
| peri | atom-ID atom-type volume density x y z |
| sphere | atom-ID atom-type diameter density x y z |
| template | atom-ID molecule-ID template-index template-at |
| tri | atom-ID molecule-ID atom-type triangleflag dens |
| wavepacket | atom-ID atom-type charge spin eradius etag cs_r |
| hybrid | atom-ID atom-type x y z sub-style1 sub-style2 ... |



Listas de Verlet y celdas de vecinos

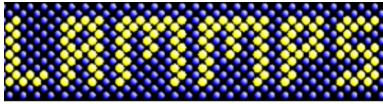


- Comando *neighbor skin style (bin,nsq)*
- Comando *Neigh_modify every n delay m yes*



Warming up

- Ajusta las coordenadas de los átomos para evitar traslapes entre estas, minimizando la energía de interacción molecular.
- Comando *minimize*.
- *minimize* Tolerancia de Energía, Tolerancia de Fuerza, No. max itera , No. Max Eval de la fuerza.
- Comando *displace_atoms* mueve las partículas inicialmente.
- *displace_atoms* id group move/random xo yo zo random/seed.
- Utilizar algún potencial de partículas suaves como *pair_style gauss* para evitar la ruptura de enlaces y generar una configuración inicial sin traslapes.



Fix

Es cualquier operación realizada sobre un grupo de átomos que se aplica sobre el sistema durante todo el proceso de simulación.

- Ensamblajes termodinámicos: NVE, NV, NPT, Langevin.

fix id nve

fix id group nvt temp Tstart Tfin Tdamp

*fix id group npt temp Tstart Tfin Tdamp & iso
Pstar Pstop Tpdamp*

fix id group Langevin Tstar Tfin Friction Seed

- Termostatos Mesoscópicos

fix id group dpd/energy

*fix id group srd BIG freq SMALL temp cellsize
seed*

Acciones

fix id group addforce fx fy fz args

fix id group print N \$var1 \$var2 file.output

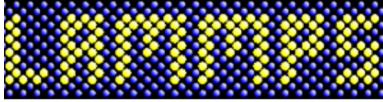
fix id group ave/time Nev Nrep Nfreq args

fix id group ave/correlate Nev Nrep Nfreq args

fix id group momentum N args

fix id group wall/lj126 args

fix id group vector every args



Ensambls Termodinámicos

Definen las variables termodinámicas del sistema N,V,T,P así como las condiciones de integración de las ecuaciones de Newton: termostátos/barostátos

Ensambls termodinámicos: NVE, NPT, Langevin.

Integrar numericamente las ecuaciones de Newton Algoritmo de Verlet (default).

NVE (Sin termostato)

fix id group nve

NVT Termostato de Nose/Hoover

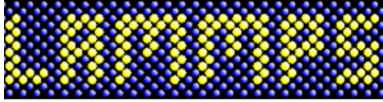
fix id group nvt temp Tstart Tfin Tdamp

NVT Termostato de Langevin

fix id group Langevin Tstar Tfin Friction Seed

NPT Barostato de Nose/Hoover

fix id group npt temp Tstart Tfin Tdamp iso Pstart Pstop Tpdamp



Compute

Valores calculados a partir de información obtenida de los átomos: propiedades termodinámicas, estructurales y de transporte.

Propiedades termodinámicas

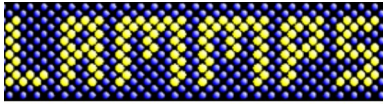
- `compute id group pe` (*energía potencial*)
- `compute id group ke` (*energía cinética traslacional*)
- `compute id group temp` (*temperatura*)
- `compute id group Press args` (*presión y tensor de esfuerzos*)

Propiedades estructurales

- `compute id group rdf args` (*Función radial de distribución*)
- `compute id group gyration args` (*Radio de giro*)

Propiedades de transporte

- `compute id group msd args` (*Recorrido libre medio*)
- `compute id group heat/flux ke id Pe id stress id`



Cálculo de propiedades I

- Función de distribución radial.

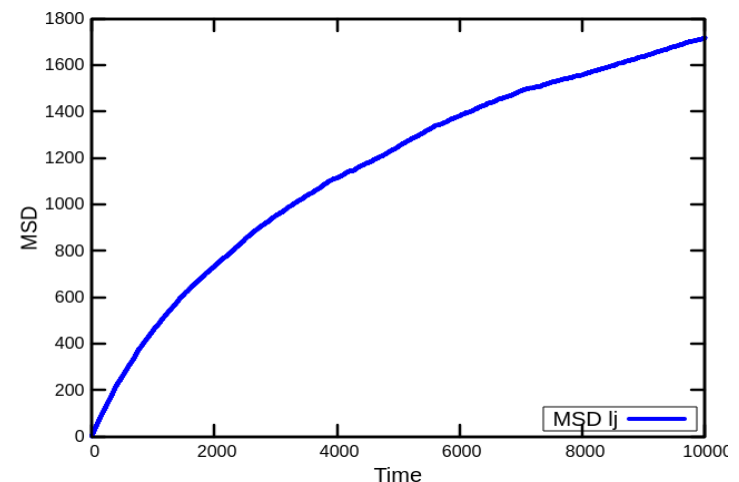
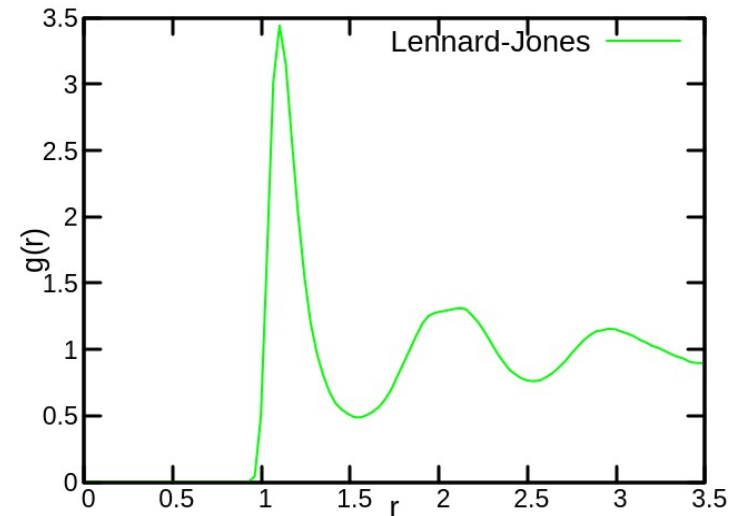
`compute` RDF *rdf* group bin part part

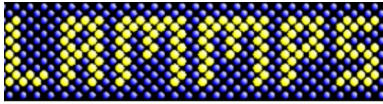
- Recorrido libre medio (MSD).

`compute` MSD *msd* group com yes/no

- Función de autocorrelación de velocidades.

`compute` VACF *vacf* group





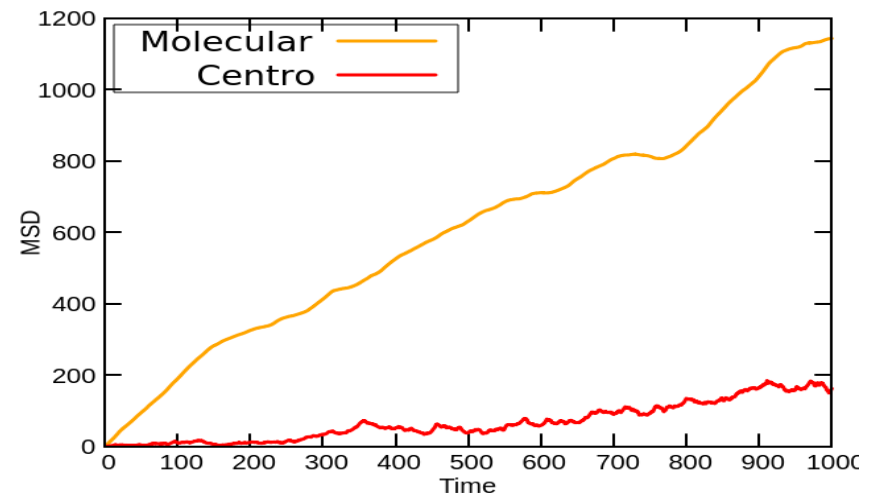
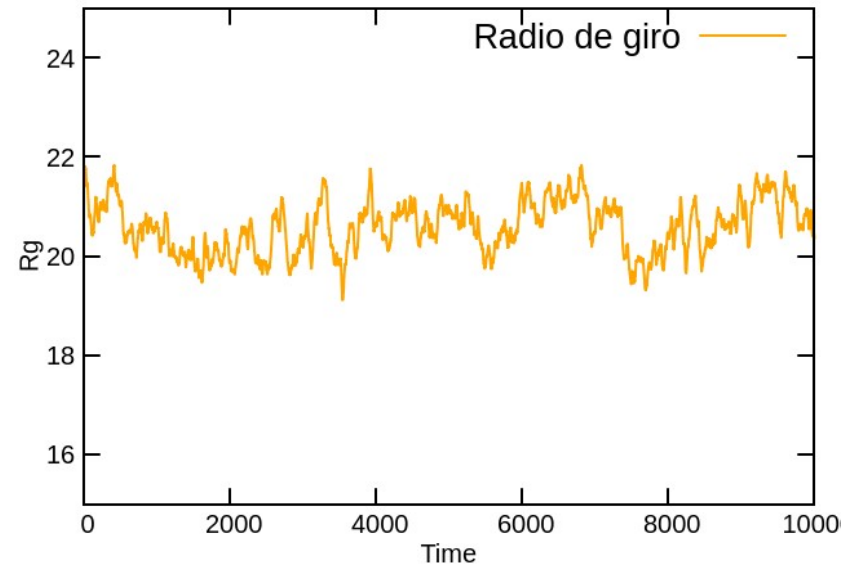
Cálculo de propiedades II

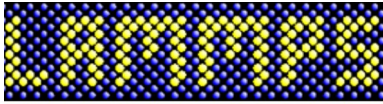
- Radio de giro total

compute id gyration group

- Recorrido libre medio MSD

compute id msd group





Ejecutar LAMMPS/paquetes

- SERIAL

```
lmp_serial -in in.lj
```

```
lmp_serial < in.lj
```

```
lmp_linux<in.fene
```

- PARALELO

```
mpirun -np 4 lmp_mpi -in in.lj
```

```
mpirun -np 16 lmp_ibm -in in.fene
```

```
mpirun -n 64 lmp_linux<in.comb
```

- SERIAL

```
lmp_serial -opk -8 -sf -opt -in in.lj
```

```
lmp_serial -sf opt -in in.lj
```

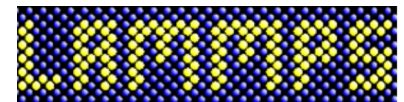
```
lmp_linux<in.fene
```

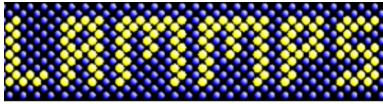
- PARALELO

```
mpirun -np 4 lmp_mpi -l my.log< in.lj
```

```
mpirun -np 16 lmp_ibm -sf gpu 4 -in  
in.fene
```

```
mpirun -n 64 lmp_linux<in.fene
```





Información en pantalla:log.lammps

- LAMMPS realiza varias inicializaciones
- Descripción del sistema (no. y tipo de átomos, enlaces, angulos, cargas.
- Imprime la cantidad de memoria que la simulación requiere.
- Estado termodinámico inicial, Energía Cintica, Potencial, Volumen etc..
- Estadísticas referentes a la construcción de celdas de vecinos .
- Durante la corrida se imprime la información termodinámica periodicamente determinado numero de pasos.
- Al concluir la simulación, LAMMPS imprime tambien el estado termodinámico final, estadísticas del numero acerca de las listas de vecinos, performace computacional y el tiempo total de simulación.
- La información que aparece en la pantalla se imprime en un archivo default denominado **log.lammps**.

Script code

(Code)

```
Neighbor list info ...
update every 1 steps, delay 10 steps, check yes
max neighbors/atom: 2000, page size: 100000
master list distance cutoff = 1.72
ghost atom cutoff = 1.72
binsize = 0.86, bins = 163 163 163
1 neighbor lists, perpetual/occasional/extra = 1 0 0
(1) pair lj/cut, perpetual
    attributes: half, newton on
    pair build: half/bin/newton
    stencil: half/bin/3d/newton
    bin: standard
Per MPI rank memory allocation (min/avg/max) = 28.14 | 28.14 | 28.14 Mbytes
```

#Run information

```
Step Temp E_pair E_mol TotEng Press
0 1.0199419 0.053426848 20.87182 22.454735 6.11902e-05.....
```

Loop time of 148.396 on 1 procs for 100000 steps with 3601 atoms

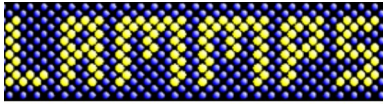
Performance: 582224.247 tau/day, 673.871 timesteps/s
99.6% CPU use with 1 MPI tasks x no OpenMP threads

MPI task timing breakdown:

| Section | min time | avg time | max time | %varavg | %total |
|---------|-----------|-----------|-----------|---------|--------|
| Pair | 8.8205 | 8.8205 | 8.8205 | 0.0 | 5.94 |
| Bond | 14.982 | 14.982 | 14.982 | 0.0 | 10.10 |
| Neigh | 114.25 | 114.25 | 114.25 | 0.0 | 76.99 |
| Comm | 2.5851 | 2.5851 | 2.5851 | 0.0 | 1.74 |
| Output | 0.0056954 | 0.0056954 | 0.0056954 | 0.0 | 0.00 |
| Modify | 6.1123 | 6.1123 | 6.1123 | 0.0 | 4.12 |
| Other | | 1.644 | | | 1.11 |

Nlocal: 3601 ave 3601 max 3601 min
Histogram: 1 0 0 0 0 0 0 0
Nghost: 42 ave 42 max 42 min
Histogram: 1 0 0 0 0 0 0 0
Neighs: 4290 ave 4290 max 4290 min
Histogram: 1 0 0 0 0 0 0 0

Total # of neighbors = 4290
Ave neighs/atom = 1.19134
Ave special neighs/atom = 1.99944



LAMMPS: archivos dump.trajectory

- Escribe en un archivo `dump.nombre` las trayectorias de los átomos.

Información de la simulación: tipo de átomo, id, tipo posiciones, velocidades...

- Varios formatos de salida
- Formato estándar: `dump 1 all atom 100 dump.ejemplo`

- Posible modificar usando “custom”

`dump 1 group custom 100 dump.file x y z, vx ,vy, vz, fx, fy, fz, ix, iy, iz`

- **Custom** permite incluir propiedades calculadas por compute o fix o variables definidas por el usuario
- `dump 1 all custom 100 dump.file x y z, vx, vy, vz, c_1, f_1, v_2.`

ITEM: TIMESTEP

0

ITEM: NUMBER OF ATOMS

3601

ITEM: BOX BOUNDS pp pp pp

-7.000000000000000e+01 7.000000000000000e+01

-7.000000000000000e+01 7.000000000000000e+01

-7.000000000000000e+01 7.000000000000000e+01

ITEM: ATOMS id type xs ys zs

3121 1 0.608998 0.456066 0.370946

3120 1 0.610497 0.463118 0.371157

3119 1 0.611586 0.46982 0.374726

3103 1 0.620015 0.426709 0.38137

3105 1 0.610512 0.433542 0.378166

3104 1 0.617068 0.432572 0.379083

3117 1 0.602263 0.463827 0.382713

3118 1 0.606093 0.46614 0.376982

3102 1 0.619868 0.424551 0.387667

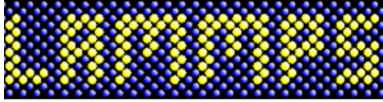
3107 1 0.604177 0.436073 0.384341

3106 1 0.611113 0.437416 0.384045

3110 1 0.597269 0.446361 0.387111

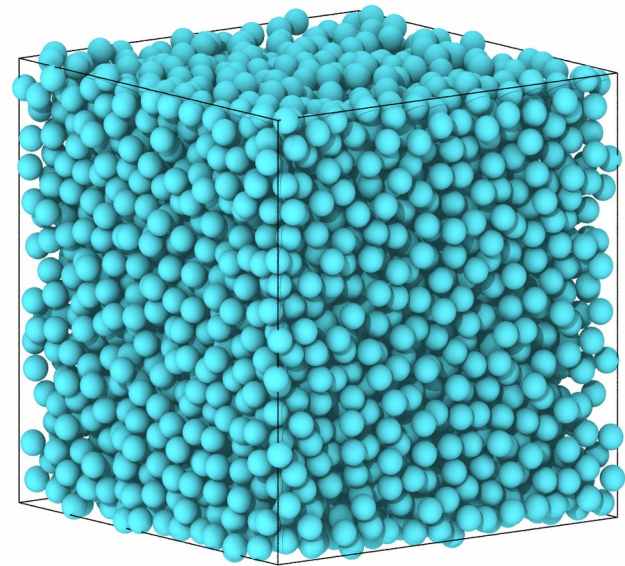
3116 1 0.60796 0.462832 0.386588

3101 1 0.61931 0.418086 0.390217

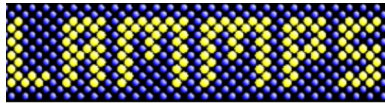


Ejemplo I. Lennard-Jones fluid en LAMMPS

- Partículas puntuales que interaccionan a través del potencial de Lennard-Jones.
- Sistema de unidades lj/ (adimensional).
- sigma, epsilon, masa.
- Se simulan 4000 partículas .
- En una caja de dimensiones 20x20x20.
- Condiciones de frontera periódicas.
- Paso de integración $t=0.01$ en unidades lj.



$$U_{LJ}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - 0.5 \left(\frac{\sigma}{r_{ij}} \right)^6 \right]$$



Ejemplo I.

Tipo de átomos: Atomic

1.-Iniciación: se designa las unidades, tipo de átomos, condiciones de frontera y se crean los átomos en una región.

```
lattice fcc, hex, sq
```

```
region id block xi xh yi yh zi zh
```

```
create atoms id region
```

```
create box id region
```

```
mass id masa
```

2.-Definición de los parámetros: listas de vecinos y potenciales de interacción.

```
neighbor y neigh_modify
```

```
pair_style
```

```
pair_coeff
```

3d Lennard-Jones Melt

#1.- INICIALIZAR

#Paquete de aceleración

```
package          gpu 1
```

```
units           lj
```

```
atom_style      atomic
```

```
lattice         fcc 0.8442
```

```
region          box block 0 10 0 10 0 10
```

```
create_box      1 box
```

```
create_atom     1 box
```

```
mass            1 1.0
```

```
velocity        all create 3.0 87287
```

#2.-PARAMETROS DE POTENCIAL

#Lista de vecinos

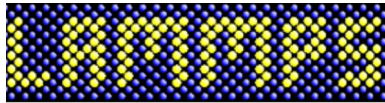
```
neighbor        0.3 bin
```

```
neigh_modify    every 20 delay 0 check no
```

#Interacciones de no-enlace

```
pair_style      lj/cut/gpu 2.5
```

```
pair_coeff      1 1 1.0 1.0 2.5
```



Ejemplo I.

Tipo de átomos: Atomic

3.-Procedimientos y parámetros de la simulación:

fix define el ensamble termodinámico.

variable define variables de observables usando variables internas de LAMMPS.

compute calcula propiedades estructurales, termodinámicas o de transporte/[Manual](#): **rdf**, **msd**.

fix ave/time: calcula promedios temporales.

fix vector: acumula variables.

fix print imprime propiedades en un archivo para analisis posterior.

3. PROCEDIMIENTOS, DEFINICION DE VARIABLES Y CALCULO DE PROPIEDADES

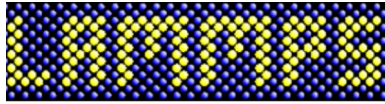
```
fix          1 all nve
variable    dt equal 0.01
variable    time equal step*dt
variable    P equal   pe
variable    K equal   ke
variable    Tot equal etotal
```

#RDF(Función de distribución radial)

```
compute     rdf all rdf 100 1 1 cutoff 2.5
fix         rdf all ave/time 1 100 10000 c_rdf[*] file&
rdf.dat mode vector
```

#MSD(Recorrido libre medio)

```
compute     MSD all msd
variable    RMS equal c_MSD[4]
fix        2 all vector 1 c_MSD[4]
fix        extra1 all print 100 "${time} ${RMS}"&
Difusion.dat screen no
```



Ejemplo I.

Tipo de átomos: Atomic

4.-Ejecución del programa se corre la simulación.

thermo define el formato de salida.

dump guarda la trayectorias de los átomos en un cierto intervalo de tiempo.

run indica el numero de pasos en que la simulación se realiza.

write_data escribe la configuración final de la simulación.

4.-EJECUCION DEL PROGRAMA

#Formato para el reporte de propiedades termodinámicas

thermo 10000

thermo_style custom step v_P v_K c_MSD[4]

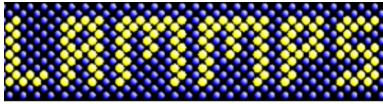
#Trayectorias

dump 1 all atom 10000 dump.melt

timestep 0.01

run 1000000

write_data melt.data



Ejemplo II. polimero

1.-Macromolécula definida a partir de partículas tipo Lennard-Jones lj/cut+ algún potencial de enlace.

Modelo de cuentas y resortes de Kremer/Grest.

Interacciones de no/enlace: potencial de FENE +
Interacciones de no enlace potencial de Weeks-Chandler-Andersen (WCA).

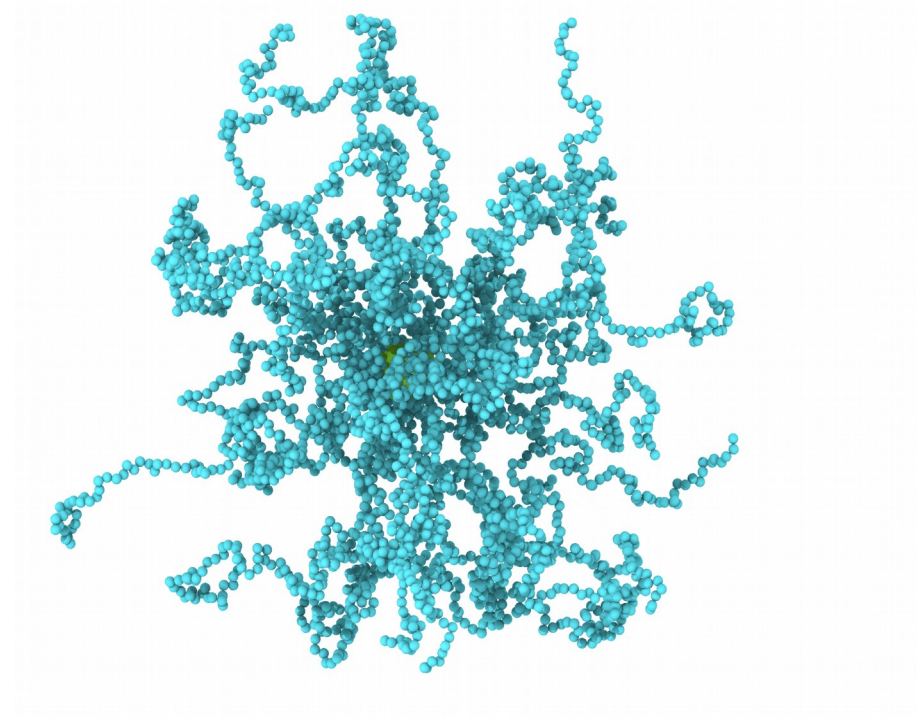
Unidades de lj.

Se simulan un polimero ramificado de estrella compuesto de un centro, 30 brazos con 120 monómeros.

Caja de simulación de 30x30x30.

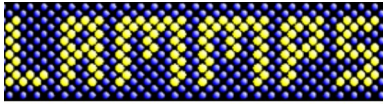
Condiciones de frontera periódicas.

Temperatura $T=1.0$, paso de simulación de 0.01.



$$V_{ch}(r) = -\frac{k}{2}R_0^2 \ln \left(1 - \frac{r^2}{R_0^2} \right)$$

$$V_{LJ}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 + \frac{1}{4} \right] & r \leq 2^{1/6}\sigma, \\ 0 & r \geq 2^{1/6}\sigma, \end{cases}$$



Ejemplo II. polimero lectura de archivo.data

1.-Iniciación usando `archivo_data`: se designa unidades, tipo de átomos, unidades, condiciones de frontera y el nombre del archivo de configuración inicial de coordenadas, velocidades y estructura: enlaces, angulos, dihedrals, etc..

2.-Definición de los parámetros: masa, carga, posiciones y velocidades iniciales, listas de vecinos, potenciales de interacción enlace y no-enlace.

neighbor y neigh_modify

bond_style: bond

bond_coeff

pair_style:lj/cut/gpu

pair_coeff

Branched star polymer

#1.-INICIALIZAR

package **gpu 1**

units **lj**

atom_style **bond**

special_bonds **fene**

boundary **p p p**

read_data **equilibrio.data**

#2.-PARAMETROS DE POTENCIAL

neighbor 0.6 bin

bond_style **fene**

bond_coeff **1 30.0 1.5 1.0 1.0**

bond_coeff **2 30.0 2.5 1.0 1.0**

pair_style **lj/cut/gpu 1.122**

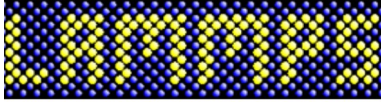
pair_modify **shift yes**

pair_coeff **1 1 1.0 1.0 1.122**

pair_coeff **2 2 1.0 1.0 1.122**

pair_coeff **1 2 1.0 1.0 1.122**

group **core type 2**



Tipos de Atomos: Bond

Definido para partículas puntuales con enlaces interacciones de no-enlace.

(Modelo de Grano-Grueso)

- Cabecera: numero y tipo de átomos, numero y tipo de enlace.
- Dimensiones de la caja de simulación
- Masas, coeficientes de no-enlace y de enlace.
- Encabezado #bond
- Posiciones
- Velocidades
- Bonds

id idatom id mol x y z

id idatom id mol vx vy vz

id idbond id atom1 id atom2

Branched star polymer

LAMMPS data file via write_data, version 3 Mar 2020, timestep = 1000000

3601 atoms
2 atom types
3600 bonds
2 bond types

```
-7.000000000000000e+01 7.000000000000000e+01 xlo xhi
-7.000000000000000e+01 7.000000000000000e+01 ylo yhi
-7.000000000000000e+01 7.000000000000000e+01 zlo zhi
```

Masses

```
1 1
2 2
```

Atoms # bond

```
2881 1 1 3.7808467001080743e+01 -2.2861731027536923e+01 -5.6954357632857020e+00 0 0 0
2880 1 1 3.8356876551072226e+01 -2.2246556832263831e+01 -5.2560836467157399e+00 0 0 0
2036 1 1 1.7122609436729952e+01 -1.1655930725334628e+01 -5.2656074342652168e+00 0 0 0
2033 1 1 1.6189084754462439e+01 -9.5893568400133127e+00 -5.1918349399940045e+00 0 0 0
2032 1 1 1.5649403195868267e+01 -8.8151488301643841e+00 -5.2266807314929720e+00 0 0 0
```

.

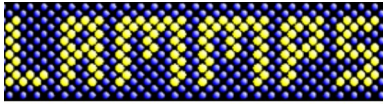
Velocities

```
2881 -1.1972894713731090e+00 1.5354568395920789e+00 -1.1380795486329611e+00
2880 5.3571738770085059e-02 5.0205310228195499e-01 -7.6383816982731101e-01
2036 6.7830465393084394e-01 -5.4970825792315026e-01 -5.8183513453749458e-01
```

.

Bonds

```
1 1 2880 2881
2 1 2036 2037
3 1 2033 2034
4 1 2032 2033
5 1 2879 2880
```



Ejemplo II de Script

3.-Procedimientos y parámetros de la simulación:

Modulo de minimización.

Incluyen **fix** con el termostato de **Langevin**,
compute define las cantidades a calcularse,
(**gyration**) radio de giro.

compute id group msd

variable MSD all “Recorrido libre medio”

variable Rg “Radio de giro”

fix print: imprime Rg vs tiempo,

MSD vs tiempo en archivos.

3. PROCEDIMIENTOS, DEFINICION DE VARIABLES Y CALCULO DE PROPIEDADES

minimize 1e-6 1e-6 1000 1000

fix 1 all nve

fix 2 all langevin 1.0 1.0 10.0 456701

compute 1 all gyration

fix 3 all vector 100 c_1

compute MSD all msd

compute MSDcore core msd

variable msdall equal c_MSD[4]

variable msdcore equal c_MSDcore[4]

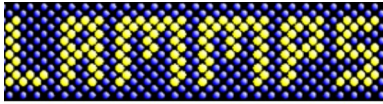
variable Rg equal c_1

variable dt equal 0.01

variable time equal step*dt

fix Extra1 all print 100 "\${time} \${Rg} " file L.dat &
screen no

fix Extra2 all print 100 "\${time} \${msdall} \${
msdcore} " file diff.dat screen no



Ejemplo II de Script

4.-Ejecución del programa: se corre o ejecuta la simulación.

Formato de trayectorias en el archivo `dump`, define el reporte de propiedades termodinámicas `thermo/thermo_style` custom.

Se define el paso de tiempo y el numero de iteraciones.

Se ejecuta el script con run.

Se puede guardar la configuración final.

`write_data`

4.-EJECUCION DEL PROGRAMA

```
dump          1 all atom 1000 dump.$i
thermo        1000
thermo_style   custom step c_1 v_length c_1[*]

timestep      0.01
run           100000
write_data    equilibrio.data
```