Molecular Dynamics Simulation of a Van der Waals Gas

Generated by Doxygen 1.10.0

Chapter 1

Project-III

Manel Serrano Visualització i estadística

Albert Plazas Forces

Anna Monclús Cond. inicials i de contorn (Coordinació e integració de tot el codi)

Aina Gaya Integració

Chapter 2

Modules Index

2.1 Modules List

Here is a list of all documented modules with brief descriptions:

Chapter 3

Module Documentation

3.1 integrators Module Reference

Module containing various integrators for molecular dynamics simulations.

Functions/Subroutines

- subroutine time_step_vverlet (r, vel, pot, n, l, cutoff, dt)

  Perform a time step using the velocity Verlet integration method. Calculates new positions and velocities for particles based on forces and previous positions/velocities.
- subroutine time_step_euler_pbc (r_in, r_out, vel, n, l, cutoff, dt, pot)

  Perform a time step using the Euler method with periodic boundary conditions. Calculates new positions and velocities for particles based on forces and previous positions/velocities.
- subroutine bm (ndat, xnums, sigma)

  Generate random numbers following a Box-Muller transformation. Generates normally distributed random numbers using the Box-Muller transformation.
- subroutine time_step_verlet (r, rold, vel, n, l, cutoff, dt, pot)

  Perform a time step using the Verlet integration method. Calculates new positions and velocities for particles based on forces and previous positions/velocities.

3.1.1 Detailed Description

Module containing various integrators for molecular dynamics simulations.

3.1.2 Function/Subroutine Documentation

3.1.2.1 bm()

```
subroutine integrators::bm (
             integer, intent(in) ndat,
             real(8), dimension(ndat), intent(out) xnums,
             real(8), intent(in) sigma )
```

Generate random numbers following a Box-Muller transformation. Generates normally distributed random numbers using the Box-Muller transformation.

Parameters

	ndat	Number of data points (must be even).
	xnums	Output array containing the generated random numbers.
	sigma	Standard deviation of the normal distribution.
in	*ndat*	Number of data points (must be even)
out	*xnums*	Output array containing generated random numbers
in	*sigma*	Standard deviation of the normal distribution

3.1.2.2 time_step_euler_pbc()

```
subroutine integrators::time_step_euler_pbc (
             real(8), dimension(n, 3), intent(in) r_in,
             real(8), dimension(n, 3), intent(out) r_out,
             real(8), dimension(n, 3) vel,
             integer, intent(in) n,
             real(8), intent(in) l,
             real(8), intent(in) cutoff,
             real(8), intent(in) dt,
             real(8) pot )
```

Perform a time step using the Euler method with periodic boundary conditions. Calculates new positions and velocities for particles based on forces and previous positions/velocities.

Parameters

	r_in	Input array containing initial particle positions.
	r_out	Output array containing updated particle positions.
	vel	Input/Output array containing particle velocities.
	N	Number of particles.
	L	Box size.
	cutoff	Cutoff distance for LJ potential.
	dt	Time step size.
	pot	Output potential energy.
in	*n*	Number of particles
in	*r_in*	Initial particle positions
out	*r_out*	Updated particle positions
	vel	Particle velocities
in	*cutoff*	Box size, time step size, cutoff distance
	pot	Potential energy

3.1.2.3 time_step_verlet()

```
subroutine integrators::time_step_verlet (
             real(8), dimension(n, 3), intent(inout) r,
             real(8), dimension(n, 3), intent(inout) rold,
             real(8), dimension(n, 3), intent(inout) vel,
             integer, intent(in) n,
```

```
            real(8), intent(in) l,
            real(8), intent(in) cutoff,
            real(8), intent(in) dt,
            real(8) pot )
```

Perform a time step using the Verlet integration method. Calculates new positions and velocities for particles based on forces and previous positions/velocities.

Parameters

	r	Input/Output array containing current particle positions.
	rold	Input/Output array containing previous particle positions.
	vel	Input/Output array containing particle velocities.
	N	Number of particles.
	L	Box size.
	cutoff	Cutoff distance for LJ potential.
	dt	Time step size.
	pot	Output potential energy.
in	*n*	Number of particles
in,out	*r*	Current particle positions
in,out	*rold*	Previous particle positions
in,out	*vel*	Particle velocities
in	*l*	Time step size, cutoff distance, box size
	pot	Potential energy

3.1.2.4 time_step_vverlet()

```
subroutine integrators::time_step_vverlet (
            real(8), dimension(n, 3), intent(inout) r,
            real(8), dimension(n, 3), intent(inout) vel,
            real(8), intent(out) pot,
            integer, intent(in) n,
            real(8), intent(in) l,
            real(8), intent(in) cutoff,
            real(8), intent(in) dt )
```

Perform a time step using the velocity Verlet integration method. Calculates new positions and velocities for particles based on forces and previous positions/velocities.

Parameters

	r	Input/Output array containing particle positions.
	vel	Input/Output array containing particle velocities.
	pot	Output potential energy.
	N	Number of particles.
	L	Box size.
	cutoff	Cutoff distance for LJ potential.
	dt	Time step size.
in	*n*	Number of particles
in,out	*r*	Particle positions
in,out	*vel*	Particle velocities
out	*pot*	Potential energy
in	*cutoff*	Time step size, box size, cutoff distance

Index