Molecular dynamics in the microcanonical (NVE) ensemble: the Verlet algorithm

**Javier Junquera** 



# Equations of motion for atomic systems in cartesian coordinates

Classical equation of motion for a system of N molecules interacting via a potential  ${\cal V}$ 

$$m_i \frac{d^2 \vec{r_i}}{dt^2} = \vec{f_i}$$

This equation also aplies to the center of mass motion of a molecule, with the force begin the total external force acting on it

## Hamilton or Lagrange equations of motion

Lagrange's equation

Hamilton's equations



$$\dot{ec{p}_i} = -
abla_{ec{r}_i} \mathcal{V} = ec{f_i}$$

To compute the center of mass trajectories involves solving...

A system of 3*N* second order differential equations

Or an equivalent set of 6*N* first-order differential equations

$$m_i \ddot{\vec{r}_i} = \vec{f_i}$$

## Hamilton or Lagrange equations of motion

Lagrange's equation

Hamilton's equations



$$\dot{ec{p}_i} = -
abla_{ec{r}_i} \mathcal{V} = ec{f_i}$$

To compute the center of mass trajectories involves solving...

A system of 3*N* second order differential equations

Or an equivalent set of 6*N* first-order differential equations

$$m_i \ddot{\vec{r}_i} = \vec{f_i}$$

### **Conservation laws**

Assuming that  $\,\mathcal{K}\,$  and  $\,\mathcal{V}\,$  do not depend explicitly on time, so that

$$\frac{\partial \mathcal{H}}{\partial t} = 0$$

Essential condition: No explicitly time dependent or velocity dependent force acting

Then, the total derivative of the Hamiltonian with respect to time

$$\dot{\mathcal{H}} = \frac{d\mathcal{H}}{dt} = \frac{\partial\mathcal{H}}{\partial t} + \frac{\partial\mathcal{H}}{\partial\vec{p}}\frac{\partial\vec{p}}{\partial t} + \frac{\partial\mathcal{H}}{\partial\vec{q}}\frac{\partial\vec{q}}{\partial t}$$

#### From the Hamilton's equation of motion

$$\dot{q}_{k} = \frac{\partial \mathcal{H}}{\partial p_{k}} \qquad \dot{p}_{k} = -\frac{\partial \mathcal{H}}{\partial q_{k}}$$
$$\dot{\mathcal{H}} = \frac{\partial \mathcal{H}}{\partial t} = \frac{\partial \mathcal{H}}{\partial t} - \frac{\partial \mathcal{H}}{\partial \vec{p}} \frac{\partial \mathcal{H}}{\partial \vec{q}} + \frac{\partial \mathcal{H}}{\partial \vec{q}} \frac{\partial \mathcal{H}}{\partial \vec{p}} = 0$$

The Hamiltonian is a constant of motion

Independent of the existence of an external potential

## The equations are time reversible

By changing the signs of all the velocities and momenta, we will cause the molecules to retrace their trajectories.

If the equations of motion are solved correctly, the computer-generated trajectories will also have this property

## Standard method to solve ordinary differential equations: the finite difference approach

Given molecular positions, velocities, and other dynamic information at a time  $t \$ 

We attempt to obtain the position, velocities, etc. at a later time  $t+\delta t$  , to a sufficient degree of accuracy



Notes:

The equations are solved on a step by step basis

The choice of the time interval  $\delta t$  will depend on the method of solution, but  $\delta t$  will be significantly smaller than the typical time taken for a molecule to travel its own length

### General step of a stepwise Molecular Dynamics simulation



### **Desirable qualities for a successful simulation algorithm**

It should be fast and require little memory

Since the most time consuming part is the evaluation of the force, the raw speed of the integration algorithm is not so important

It should permit the use of long time step  $\,\delta t$ 

Far more important to employ a long time step. In this way, a given period of simulation time can be covered in a modest number of steps

It should duplicate the classical trajectory as closely as possible

It should satisfy the known conservation laws for energy and momentum, and be time reversible

It should be simple in form and easy to program

Involve the storage of only a few coordinates, velocitites,...

### **Desirable qualities for a successful simulation algorithm**

It should be fast and require little memory

Since the most time consuming part is the evaluation of the force, the raw speed of the integration algorithm is not so important

It should permit the use of long time step  $\,\delta t$ 

Far more important to employ a long time step. In this way, a given period of simulation time can be covered in a modest number of steps

It should duplicate the classical trajectory as closely as possible

It should satisfy the known conservation laws for energy and momentum, and be time reversible

It should be simple in form and easy to program

Involve the storage of only a few coordinates, velocitites,...

### Energy conservation is degraded as time step is increased

All simulations involve a trade-off between



A good algorithm permits a large time step to be used while preserving acceptable energy conservation

## Parameters that determine the size of $\delta t$

- Shape of the potential energy curves
- Typical particle velocities

Shorter time steps are used at high-temperatures, for light molecules, and for rapidly varying potential functions

# The Verlet algorithm method of integrating the equations of motion: description of the algorithm

**Direct solution of the second-order equations** 

Method based on:

- the positions  $ec{r}(t)$
- the accelerations  $\vec{a}(t)$

- the positions from the previous step  $\vec{r}(t-\delta t)$ 

A Taylor expansion of the positions around *t* 

$$\vec{r}(t+\delta t) = \vec{r}(t) + \vec{v}(t)\delta t + \frac{1}{2}\vec{a}(t)\delta t^2 + \cdots$$
$$\vec{r}(t-\delta t) = \vec{r}(t) - \vec{v}(t)\delta t + \frac{1}{2}\vec{a}(t)\delta t^2 - \cdots$$

Adding the two equations  $\vec{r}(t + \delta t) + \vec{r}(t - \delta t) = 2\vec{r}(t) + \vec{a}(t)\delta t^{2} + \mathcal{O}(\delta t^{4})$   $\vec{r}(t + \delta t) = 2\vec{r}(t) - \vec{r}(t - \delta t) + \vec{a}(t)\delta t^{2} + \mathcal{O}(\delta t^{4})$ 

# The Verlet algorithm method of integrating the equations of motion: some remarks

 $\vec{r}(t+\delta t) = 2\vec{r}(t) - \vec{r}(t-\delta t) + \vec{a}(t)\delta t^2 + \mathcal{O}(\delta t^4)$ 

#### **Remark 1**

The velocities are not needed to compute the trajectories, but they are useful for estimating the kinetic energy (and the total energy).

They can be computed a posteriori using [ $\vec{v}(t)$  can only be computed once  $\vec{r}(t + \delta t)$  is known]

$$\vec{v}(t) = \frac{\vec{r}(t+\delta t) - \vec{r}(t-\delta t)}{2\delta t} + \mathcal{O}(\delta t^2)$$

#### Remark 2

Whereas the errors to compute the positions are of the order of  $\delta t^4$  . The velocities are subject to errors of the order of  $\delta t^2$ 

# The Verlet algorithm method of integrating the equations of motion: some remarks

 $\vec{r}(t+\delta t) = 2\vec{r}(t) - \vec{r}(t-\delta t) + \vec{a}(t)\delta t^2 + \mathcal{O}(\delta t^4)$ 

#### **Remark 3**

The Verlet algorithm is properly centered:  $\vec{r}(t - \delta t)$  and  $\vec{r}(t + \delta t)$  play symmetrical roles.

The Verlet algorithm is time reversible

#### **Remark 4**

The advancement of positions takes place all in one go, rather than in two stages as in the predictor-corrector algorithm.

# The Verlet algorithm method of integrating the equations of motion: overall scheme



Known the positions at *t*, we compute the forces (and therefore the accelerations at *t*)

> Then, we apply the Verlet algorithm equations to compute the new positions

> > ...and we repeat the process computing the forces (and therefore the accelerations at  $t + \delta t$  )

### **Molecular Dynamics**

- Timestep must be small enough to accurately sample highest frequency motion
- Typical timestep is 1 fs (1 x 10<sup>-15</sup> s)
- Typical simulation length: Depends on the system of study!! (the more complex the PES the longer the simulation time)
- Is this timescale relevant to your process?
- Simulation has two parts
  - equilibration when properties do not depend on time
  - production (record data)
- Results:
  - diffusion coefficients
  - Structural information (RDF's,)
  - Free energies / phase transformations (very hard!)
- Is your result statistically significant?

## How to run a Molecular Dynamic in Siesta: the Verlet algorithm (NVE-microcanonical ensemble)

MD.TypeOfRun		Verlet	# #	Standard Verlet algorithm	
MD.InitialTemperature		600 K	# # # # # # # #	for Molecular Dynamics Initial temperature for the Molecular Dynamics run. The atoms are assigned random velocities drawn from the Maxwell-Bolzmann distribution with the corresponding temperature. The constraint of zero center of mass velocity is imposed.	
MD.Initial.Time.Step		1	#	Initial time step of the MD simulation.	
MD.Final.Time.Step		100	#	Final time step of the MD simulation.	
MD.Length.Time.Step		1.00 fs	#	Length of the time step of the	
			#	Molecular Dynamic simulation.	
WriteCoorStep WriteForces	.true. .true.		# # # # # #	<pre>If .true. it writes the atomic coordinates at every time step of the MD simulation If .true. it writes the atomic forces at every time step of the MD simulation</pre>	
WriteMDHistory .true.		;		If .true. Siesta accumulates the MD	
			#	trajectory in the	
			#	SystemLabel.MD	
			#	SystemLabel.MDE files	
WriteMDXmol	.true.		#	If .true. Siesta it originates the	
			#	writing of an extra file	
			#	SystemLabel.ANI	
			#	containing all the atomic coordinates	
			#	of the simulation in a format directly	
			#	readable by XMOL for animation.	

# Computing the instantaneous temperature, kinetic energy and total energy

$$\sum_{i=1}^{N} \frac{|\vec{p_i}|^2}{m_i} = 2\mathcal{K} = 3Nk_BT$$

## How to run a Molecular Dynamic in Siesta: the Verlet algorithm (NVE-microcanonical ensemble)



Maxwell-Boltzmann

## Output of a Molecular Dynamic in Siesta: the Verlet algorithm (NVE-microcanonical ensemble)

**Conserved quantity** 

### SystemLabel.MDE

#	Step	T (K)	E_KS (eV)	E_tot (eV)	Vol (A^3)	P (kBar)
	1	600.00	-2955.25340	-2954.55542	93.166	166.448
	2	536.29	-2955.18330	-2954.55944	93.166	171.567
	3	639.06	-2955.30406	-2954.56065	93.166	161.570
	4	893.32	-2955.60359	-2954.56439	93.166	136.114
	5	1251.77	-2956.02573	-2954.56955	93.166	104.263
	6	1647.75	-2956.49212	-2954.57530	93.166	68.292
	7	2010.57	-2956.91768	-2954.57879	93.166	29.133
	8	2284.08	-2957.23821	-2954.58114	93.166	-8.519
	9	2431.77	-2957.40992	-2954.58106	93.166	-48.192
	10	2438.54	-2957.41700	-2954.58026	93.166	-86.422
	11	2316.90	-2957.27302	-2954.57779	93.166	-118.168
	12	2095.96	-2957.01291	-2954.57469	93.166	-145.360
	13	1810.62	-2956.67849	-2954.57220	93.166	-166.635
	14	1493.81	-2956.30810	-2954.57036	93.166	-184.229
	15	1173.12	-2955.93339	-2954.56870	93.166	-199.845
	16	873.37	-2955.58295	-2954.56697	93.166	-212.241
	17	615.48	-2955.28182	-2954.56584	93.166	-221.756
	18	414.73	-2955.04759	-2954.56514	93.166	-228.956
	19	281.21	-2954.89192	-2954.56480	93.166	-234.651
	20	220.45	-2954.82120	-2954.56476	93.166	-238.258
	21	233.83	-2954.83693	-2954.56492	93.166	-240.579
	22	319.50	-2954.93689	-2954.56522	93.166	-240.647

#### **Example for MgCoO<sub>3</sub> in the rhombohedral structure**

## **Output of a Molecular Dynamic in Siesta:**

SystemLabel.MD

SystemLabel.MDE

SystemLabel.ANI

Atomic coordinates and velocities (and lattice vectors and their time derivatives if the dynamics implies variable cell). (unformatted; post-process with iomd.F) shorter description of the run, with energy, temperature, etc. per time step (contains the coordinates of every Molecular Dynamics step in xyz format)

These files are accumulative even for different runs. Remember to delete previous ones if you are not interested on them

### Length of time step: 3 fs

\$ gnuplot \$ gnuplot> plot "md\_verlet.MDE" using 1:3 with lines, "md\_verlet.MDE" using 1:4 with lines



Compare: Total energy with KS energy

### Length of time step: 1 fs

\$ gnuplot \$ gnuplot> plot "md\_verlet.MDE" using 1:3 with lines, "md\_verlet.MDE" using 1:4 with lines



Compare: Total energy with KS energy

### Length of time step: 0.5 fs

\$ gnuplot \$ gnuplot> plot "md\_verlet.MDE" using 1:3 with lines, "md\_verlet.MDE" using 1:4 with lines



Compare: Total energy with KS energy

### Length of time step: 0.1 fs

\$ gnuplot \$ gnuplot> plot "md\_verlet.MDE" using 1:3 with lines, "md\_verlet.MDE" using 1:4 with lines



Compare: Total energy with KS energy



### **Visualisation and Analysis**



### Molekel http://www.cscs.ch/molekel

### XCRYSDEN http://www.xcrysden.org/

### GDIS http://gdis.seul.org/