A short user guide for the Velocity Störmer-Verlet program

R.O. Ocaya

Contents

Chapter 1

Background reading

1.1 Introduction

The Velocity Störmer-Verlet software, henceforth abbreviated VSV, was written to carry out fast molecular dynamics (MD) calculations on the mechanical evolutions of large collections of well-defined particles with masses, positions and velocities in a known potential energy field. Such a particle collection is typically referred to as an "ensemble".

Loosely speaking an ensemble is a group of items, such as a troupe of musicians, whose actions and interactions are considered holistically rather than through its individual components. In mechanics, particle ensembles may be defined uniquely using parameters that convey size, energy, and so on. Such parameters may be volume, pressure, force, area, particle number, temperature, etc.

1.1.1 The canonical ensemble

Specifically, statistical mechanics defines the *canonical* ensemble as a statistical arrangement that represents possible states of a mechanical system that is in thermal equilibrium with a heat bath at some fixed temperature, T_b. The equilibrium is not necessarily static, but can be dynamic, implying steady energy exchange between the heat bath and the system. Figure 1.1 illustrates the canonical ensemble, defined as a system having N particles, volume V and temperature T. The bath, although drawn bounded, is considered to be infinite energy source.

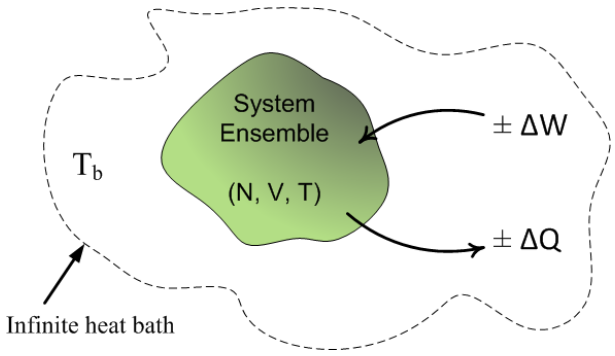

Figure 1.1: The canonical ensemble.

1.2 Defining a particle system

In dynamical simulations of mechanical systems, the main, essential ingredient is a model for the physical system. Choosing a model amounts to *defining* a potential energy function that best describes the system. Therefore, it is not enough to merely define an ensemble, which in reality only specifies the boundary conditions under which the model operates. The model specifies the physical laws that govern the evolution of states between boundary conditions.

In the currently followed approach in MD, which is the main aim of the VSV code, the potential energy function V is defined in terms of the position vector set $\{\bar{r}_i\}$ for all N particles, i.e. i={1, 2, ..., N}. Thus

$$V = V(\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_N). \tag{1.1}$$

The potential must be invariant to translation and rotation.

1.2.1 Force in atomic particle systems

Force ($\bar{F}$) is derived as the gradient of potential energy with respect to the particle displacements, i.e.

$$\bar{F} = -\nabla V(\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_N). \tag{1.2}$$

This form of force specification implies energy conservation i.e. the existence of the Hamiltonian, E:

$$\mathrm{E} = \mathrm{K} + \mathrm{T}, \tag{1.3}$$

where total kinetic and potential energies are respectively K and T. The simplest choice of V relies on pairwise interactions between any two system particles i and j and their relative positions, i.e.

$$V(\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_N) = \sum_{i}^{N} \sum_{j>1}^{N-1} \phi(|\bar{r}_i - \bar{r}_j|). \tag{1.4}$$

While intuitive and relatively easy to implement, there are two fundamental problems with this two-body modeling approach, namely:

1. It is computationally demanding as system size increases.
2. It does not properly describe important particle systems such as metals, semiconductors, new materials.

1.3 Embedded Atom Modeling method

Two-body potentials that are still either in use or have historical significance are the Lennard-Jones (LJ), Stillinger-Weber, Buckingham, Long-range Coulomb, Tersoff, Brenner, Sutton-Chen, Morse potentials, and many more. These potentials mostly describe particle interactions through the interplay of attractive and repulsive particle-particle forces. For instance, when the particles get to within a specified distance of each other repulsion dominates; beyond a certain separation, attraction dominates. No one potential models all systems equally well and the best one for a particular system is arguable. The LJ potential, a.k.a. the 12-6 potential is, for instance, known to correctly model atomic systems having closed-shell atom interactions (e.g. rare gases such as argon). However, it fails dismally for closed-shell systems where the localized bonds are strong (e.g. for covalent bonds), or even where there is a strong electron delocalization "sea" (e.g. metals). Therefore in metals, the LJ potential describes surface relaxation poorly.

1.3.1 The Finnis-Sinclair approach

To address the limitations of the two-body approach, the *embedded atom model* (EAM) method was developed. The Finnis-Sinclair (FS) potential is an EAM method potential that is a composite of a two-body potential to model repulsion, but with the advancement of a *density of state* function ρ_i to model attraction. The particles are presumed to be in the *face-centered cubic* (FCC) crystalline arrangement. This additional term represents the energy needed to embed an atom i (hence the term "embedded") into the array in terms of the receiving host electron density. The total system potential energy, U_{tot} (or T in the Hamiltonian) is then

$$U_{tot} = \frac{1}{2}\varepsilon \sum_{i}^{N} u_i, \tag{1.5}$$

where for n>m,

$$u_i = \frac{1}{2}\varepsilon \sum_{j\neq i}^{N} \left[V(r_{ij}) - c\rho_i^{1/2}\right], \quad V(r_{ij}) = \left(\frac{a}{r_{ij}}\right)^n, \quad \rho_i = \sum_{j\neq i}^{N} \left(\frac{a}{r_{ij}}\right)^m. \tag{1.6}$$

The constant ε is a dimensionless energy scaling parameter, a is the lattice constant and c is a fitting parameter.

1.3.2 The Sutton-Chen form of the FS potential

The VSV code implements the Sutton-Chen (SC) form of the FS method:

$$E_{tot} = \frac{1}{2}\sum_{ij}^{N} V(r_{ij}) + \sum_{i}^{N} F(\bar{\rho}_i), \tag{1.7}$$

where $V{=}V(r_{ij})$ is the pairwise interaction potential. It introduces a term for the embedding energy, $F{=}F(\bar{\rho}_i)$. According to SC,

$$V = \varepsilon \sum_{i=1}^{N} \left(\sum_{j=i+1}^{N} \left(\frac{\sigma}{r_{ij}}\right)^n - c\sqrt{S_i} \right), \tag{1.8}$$

where

$$S_i = \sum_{j=1, j\neq i}^{N} \left(\frac{\sigma}{r_{ij}}\right)^m. \tag{1.9}$$

All other symbols have their usual meanings. Hence the force on the i-th particle is shown to be

$$\overline{F}_i = -\overline{\nabla} V(r) = \varepsilon \sum_{j=1,\ j\neq i}^{N} \left[n\Big(\frac{\sigma}{r_{ij}}\Big)^n - \frac{cm}{2}\left(\frac{1}{\sqrt{S_i}} + \frac{1}{\sqrt{S_j}}\right)\Big(\frac{\sigma}{r_{ij}}\Big)^m \right] \frac{\bar{r}_{ij}}{r_{ij}^2}, \tag{1.10}$$

where $\bar{r}_{ij}$ is the vector between particles i and j, i.e.

$$\bar{r}_{ij} = \bar{x}_j - \bar{x}_i. \tag{1.11}$$

1.3.3 The 'adatom' concept in MD simulation

The EAM approach, because it fundamentally involves particle embedding, presents the unique opportunity to add or subtract atoms arbitrarily to existing arrangement of atoms and therefore to investigate additive/subtractive perturbations on the global system. The added or subtracted atom or atoms are called 'adatoms'. In this way, the effects of missing atoms, substitution by different atoms, arbitrary configurations may be examined in simulations, something that is impossible to do experimentally. In essence, this tantamount to postulating defects in the crystalline array. The study of such defects then leads to enhanced understanding of many macro properties of the material i.e. its thermal properties by phonon coupling, melting point, mechanical strength, and so on. These results may then be interpreted in the context of established theories.

The next chapter shows how the VSV program can be used to calculate the global minimum in transition metal clusters using the foregoing Sutton-Chen potential in the EAM method.

1.4 Velocity Störmer-Verlet integration

A version of this time evolution technique is implemented in the program. It is commonly used to discretize using Newtonian equations of motion [4], and has $\mathrm{O}(\delta t^2)$ discretization error. In the discretized notation $\bar{p}_i^n := \bar{p}_i(t_n)$ which describes the vector $\bar{p}$ sampled at the discrete n-th time interval, where $\bar{p}{=}\{\bar{x},\ \bar{v},\ \bar{F}\}$, it follows that the approximation of $\bar{p}_i^{n+1}$, i.e. on the next or $(n+1)$-st time interval or $(t+\delta t)$ gives the new positions

$$\overline{x}_i^{n+1} = \overline{x}^n + \delta t \overline{v}_i^{n+1} + \frac{\overline{F}_i^n}{2m_i}\delta t^2. \tag{1.12}$$

Similarly, the next velocities are

$$\overline{v}_i^{n+1} = \overline{v}_i^n + \left(\frac{\overline{F}_i^n + \overline{F}_i^{n+1}}{2m_i}\right)\delta t. \tag{1.13}$$

Chapter 2

Illustrative example

2.1 Specifying a particle array

The example described below is based on a simple arrangement of three copper atoms in a known, room-temperature stable matrix as shown in Figure 2.1. The VSV code calculates the energies in the system and evaluates the forces using the foregoing formulations. The program has been tested with up to about ten thousand atoms. Table 2.1 summarizes the simulation input data.

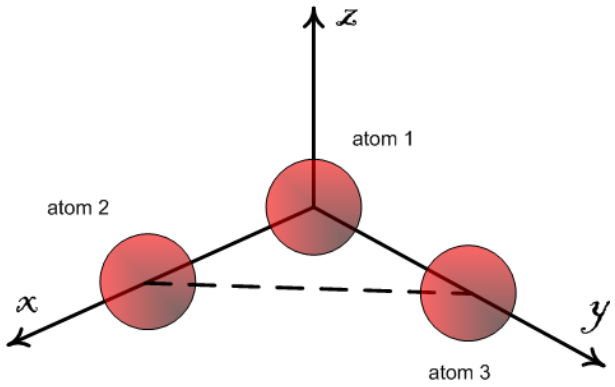

Figure 2.1: Atomic arrangement of copper atoms used.

Table 2.1: Sample data for FCC copper obtained from [3].

Atom	m	x	y	z	v_x	v_y	v_z
1	63	-0.5264206369	0.1019469049	0.0252070114	0	0	0
2	63	-0.8377547075	-0.1828574677	-0.4214018482	0	0	0
3	63	-0.2345929501	-0.0832085752	-0.4827850722	0	0	0

The VSV code requires that the input particle data be in an ASCII text file (`data.txt`) in the space-delimited 3-dimensional format specifying mass, initial positions (x,y,z) and initial velocities (vx, vy, vz), i.e.

```
mass  x_pos  y_pos  z_pos  v_x  v_y  v_z
```

The code enumerates the file contents to determine the number of particles, N, and builds a RAM array of particles to speed up computations. The above particle data would then appear in `data.txt` as:

```
63  -0.5264206369   0.1019469049    0.0252070114  0  0  0
63  -0.8377547075  -0.1828574677   -0.4214018482  0  0  0
63  -0.2345929501  -0.0832085752   -0.4827850722  0  0  0
```

This data specifies that all three particles are at rest. However, a dimensionless initial velocity can be introduced to study the effect of an atom moving in a specific direction within the lattice. This could be done to study heat propagation by phonon stimulation and coupling.

For this simulation, the simulation constants are defined in a separate ASCII file `sim_constants.h` in terms of scaled energy ε, lattice parameter and the exponents m and n of the LJ pairwise potential. The contents of this file are

```
// define the simulation constants
#define eps         1     // an energy parameter
#define lat_const   3.61 // lattice constant of copper, in angstrom
#define cn 39.432 // density of state constant
#define nint 9 // potential function steepness
#define mint 6 // potential function range

// define macros for index mapping of cells in 3 dimensions
#define index(ic,nc) ((ic)[0] + (nc)[0]*((ic)[1] + (nc)[1]*(ic)[2]))
```

2.2 Running the VSV code

The program is platform independent and can therefore be run many different environments. This guide details two environments i.e. Microsoft Windows and Ubuntu Linux LTS 12.04 run as a virtual machine using VM Ware.

2.2.1 Running in Windows 7.0

There are two ways to run the program on a Windows computer, namely

1. Running the built executable file included in the debug folder release i.e. `velocity-stormer-verlet_02.exe`
2. Running it from the coding environment in `debug` mode.

These ways are discussed separately below.

Running the executable

The only requirement for this method is to have the executable, the header file, simulation constants and the data file in one folder, say, `fcc`, to be accessed by the executable. Execute the system run command `cmd` as administrator to open the DOS box. Navigate to the folder and type the name of the executable to start the program, e.g.

```
cd \fcc
velocity-stormer-verlet_02.exe
```

For all purposes this brings the user to the same point of execution described for MS Windows, see **Note A** below.

Running on Windows debug

The program was developed using Microsoft Visual C++ Express edition 2008. The integrated development environment which has a debug mode to troubleshoot the code. This environment allows the program and its parameters to be modified and tested instantly before deployment.

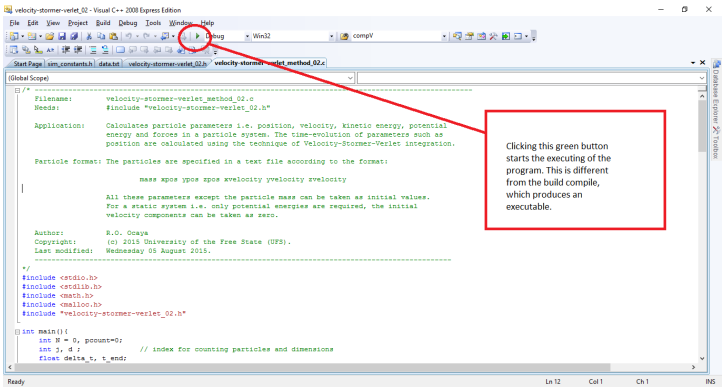

Figure 2.2: The debug environment in MS Visual C++ compiler.

Note A: After clicking the green button the program execution is at the same stage as the DOS method above. The resulting execution is shown in Figure 2.3. In this simple execution, the force and its components on each of the three particles have been calculated are in μN. The program correctly enumerates the particles as being 3 in total. Examining the components of the force, which for particle i has the form:

$$F_i = (F_{ix},\ F_{iy},\ F_{iz}), \tag{2.1}$$

gives the results in Table 2.2. Interestingly, the net forces are not zero on each particle but *equal*. This suggests that there is a tendency of each particle to fly apart in the directions determined by the largest force components, although the interplay of the components considered together is such that the arrangement remains in place. This is an illustration of mechanical stability of the array.

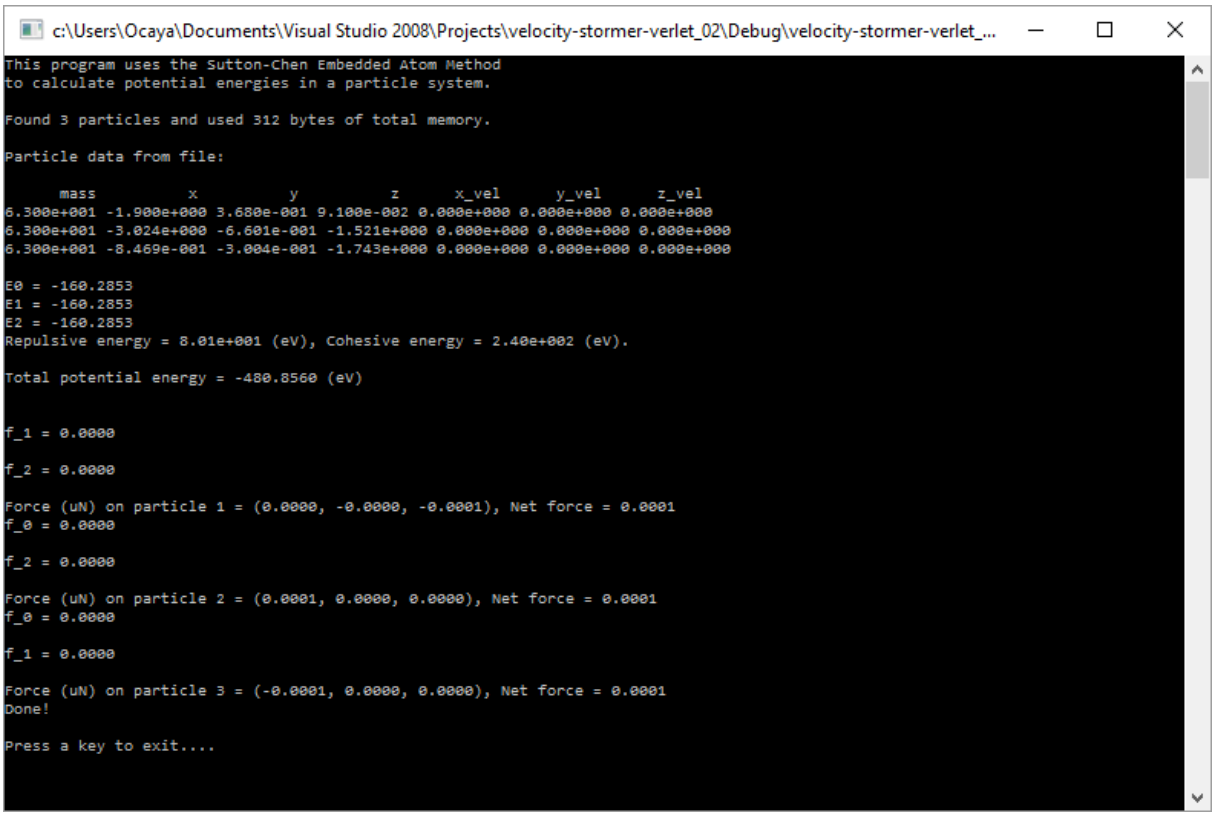

Figure 2.3: The debug environment in MS Visual C++ compiler.

Table 2.2: Summary of force calculations on the simple 3-atom array in Cu from [3]. All forces are in μN.

Atom	F_{ix}	F_{iy}	F_{ix}	Net force (F)
1	0.0000	0.0000	0.0001	0.0001
2	0.0001	0.0000	0.0000	0.0001
3	-0.0001	0.0000	0.0000	0.0001

Table 2.3: Summary of potential energies in the system of 3-atom array in Cu from [3].

Energy	eV
per atom	-160.2853
repulsive	80.1
cohesive	240.0
total	-480.856

The calculated potential energies in electron volts (eV) of the individual atoms and of the total system are also shown in Figure 2.3. The extent to which these energies are distributed between repulsion and cohesion are also

shown. Table summarizes these energies. The results show that each atom has the same potential energy (-160.3 eV) in the cluster, which is another indication of mechanical stability. The total potential energy of the system is -480.9 eV, which is in exact agreement with the results of Doye et al [3]. The additional information obtained by the VSV code is that the predominant energy in the cluster is cohesive than repulsive, suggesting that the atoms in fact try to condense onto each other rather than fly apart. Interestingly, although copper forms an FCC structure whose conventional unit cell contains 4 atoms and has a coordination number of 8, using only three atoms shows that such a system can exist without any problems. It will at the same time be noted that simulation is only as good as the interpretation of results go. Simulation is not a substitute for theory. The former has to be interpreted in the context of the latter.

Running in Ubuntu Linux

The program should run on most `gcc` systems without any problems. The executable can be created as follows (assuming all the above files are included in the same directory):

```
 gcc velocity-stormer-verlet_02.c -o VSV_prog -lm
./VSV_prog
```

The first command compiles the code `velocity-stormer-verlet_02.c` into an executable called `VSV_prog.o` for the sake of brevity. The second command (alternatively `./VSV_prog.o`) executes the program. The results are identical.

Bibliography

[1] Finnis MW, Sinclair JE, Philos. Mag. A 50 (1984) 45.

[2] Sutton AP, Chen J. Long-range Finnis-Sinclair potentials. Philosophical Magazine Letters, 1990 Vol. 61 (3), 139-156.

[3] Doye JPK, Wales DJ. Global minima for transition metal clusters described by the Sutton-Chen potentials. New J. Chem. 1998, 733-744.

[4] Verlet L. Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, Phys. Rev., 159 (1967), pp. 98103.

[5] Wales DJ, Doye JPK. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms, J. Phys. Chem. A., 101, 28 (1997), pp. 5111-5116.