KALYPSO TUTORIAL: 0.5 keV Ar+-Cu(100) collision

Contents

1. Introduction

In this elementary project, named "copper", we shall simulate a single 0.5 keV Ar+ inelastic collision with a 343 atom Cu(100) cluster. (This is not a realistic simulation of a Cu crystal, because the target is too small to contain the collision even approximately.) The projectile will be directed towards the surface Cu atom at normal incidence, with an impact parameter of 0.1 Å (corresponding to an *x*-displacement of 0.1 Å, and a zero *y*-displacement). Our objective will be to generate data which can be used to plot atomic trajectories for any atoms that have kinetic energies above 2.0 eV after 100 fs have elapsed.

If you want to know more about any of the steps, try hitting the F1 key to see the Help relevant to that context.

2. Creating input files (with Spider)

For this project, we need to create 6 input files. These are already supplied in the `\tutorial` directory as `copper.trg`, `copper.prj`, `copper.run`, `copper.mdl`, `copper.imp`, `copper.inl`. However, we will summarise the steps needed to create these files from the beginning in this section.

The first thing to do is create a new directory on your hard disk for storing the various files to be generated by this project. Give the directory a name of your own choice (e.g. 'copper'). Then start the Spider program (click the icon in the `\bin` directory)

2.1 Target file (`copper.trg`)

Select Target|New target|Face-centred cubic|[100] surface from the Spider menus. In the data dialog box that comes up, enter the following values:

```
Element: Cu   Lattice constant: 3.6147
Atomic number: 29   Atomic mass: 63.546
x-width: 3   x-origin: 0.0
y-width: 3   y-origin: 0.0
z-depth: 7   z-origin: 0.0
Flags: 0
```

Then click Ok and save the file as `copper.trg`. These inputs will create a target file `copper.trg` which contains the coordinates of your lattice. It will consist of 343 atoms: 7 atomic <110> rows in the x- and y-directions (surface plane), and 7 layers in the z-direction (surface normal). The Flags parameter should be set at 0 for an elemental target (in a binary target, the Flags parameter = 1 for atoms of the second element).

Optional: if you wish, you can examine this target file in an edit window (Edit|Open option), or as a graphical display (Target|Display/Orient).

2.2. Projectile file (`copper.prj`)

Select the Projectile|New projectile menu option from the menu.

Except for the projectile energy, use the default parameters in the dialog box that comes up, since these are correct for our purposes:

(`Ar, 0.5 keV, Z = 18, mass = 39.948, Flags = 0`)

Then click Ok and enter `copper.prj` in the file dialog box.

2.3. Run file (`copper.run`)

Select Run|New run from the menu. Move systematically through the dialog box, entering the following data:

`Sampling frequency`: check the 'periodic' box, and enter '10' for the period. Leave the 'At termination of each run' box unchecked. Explanation: This will provide a 'snapshot' of the system after every 10 timesteps during the simulation.

`Output dynamical variables info for...`: Check the box KE > 10 eV. Explanation: Output data will be recorded only for those particles having KE > 10.0 eV.

`Ion incident angles`: Use the default settings (polar = 90º, azimuthal = 0º). Explanation: these angles correspond to a normally incident projectile.

`Periodic and misc. parameters`: Set the following values:
`Initial timestep` = 2.0E-16 s;
`Neighbour update` =10 timesteps;
`Termination time (min.)` = 0.0 fs;
`Termination time (max.)` = 100 fs;
`Maximum number of partners` = 50.

Explanation: (a) The simulation will start off with a time step of 0.2 fs: this value is mainly determined by the speed of the projectile, or the speed that could be attained by a displaced target atom (whichever is the larger). The idea is that no atom should move a great distance in any one timestep. (b) The neighbour update refers to the number of timesteps which elapse before the neighbour lists are updated. No error will result from setting this parameter too high or low. However, if too high, your simulation will waste memory; if too low, your simulation will waste time updating the neighbour lists unnecessarily; (c) Your simulation will terminate unconditionally after 100 fs have elapsed. The maximum number of partners determines how much memory will be allocated for the neighbours list. (here we assume that no atom will have more than 50 neighbours within its sphere of influence). The sphere of influence (computed by the program) is larger than the value specified for the interaction range (in the MDL file) because we have to take the range of the particle between updates into account (particles may move into interaction range between updates). The value of 50 partners is quite conservative, reflecting the heuristic nature of the parameter. When you run the simulation, you will find that Kalypso reports that only about 30 of these partner slots were in fact needed.

Save the file (click Ok, enter `copper.run` in the file dialog box that comes up).

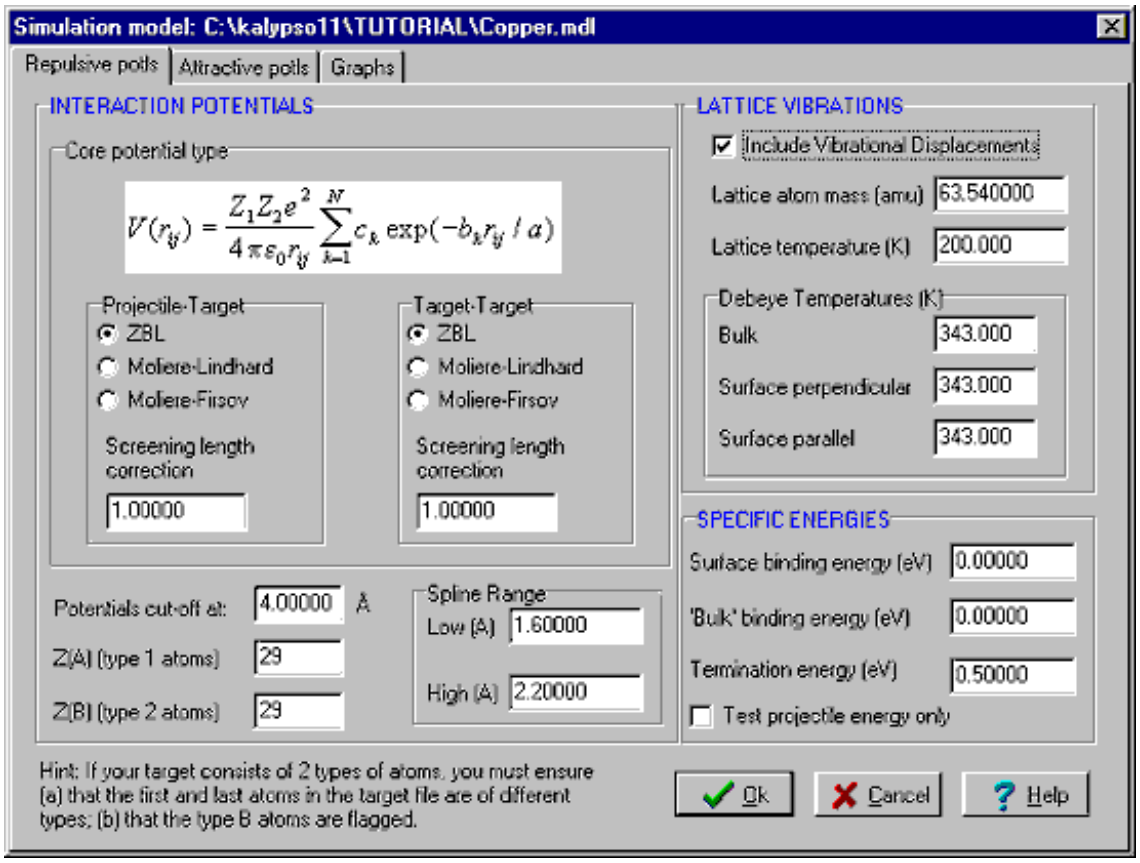
2.4. Model file (`copper.mdl`)

Select Model | New Model from Spider's menu. Fill out the input data as shown on the screen shots below.

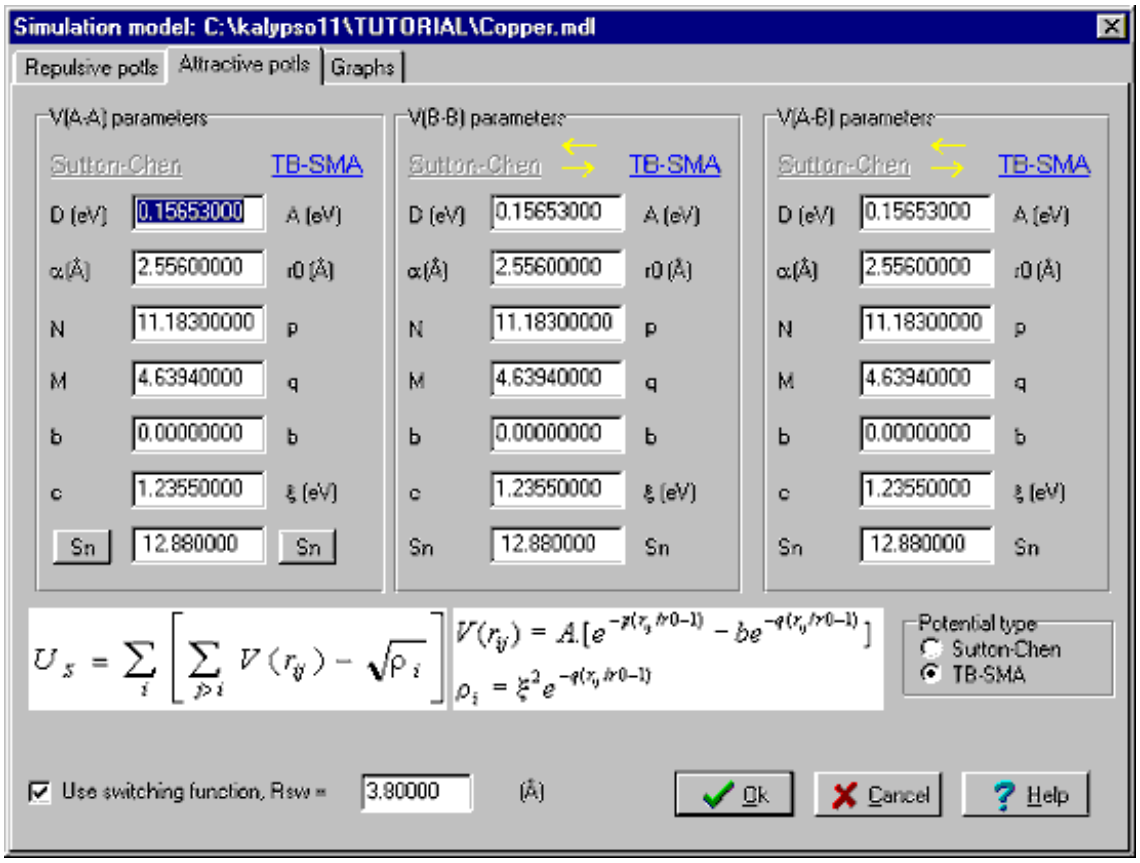
The simulation model will use a ZBL screened Coulombic potential for both the Ar-Cu interaction, and for the short range part of the Cu-Cu interaction. The Cu-Cu interaction at long range (r > 2.2 Å) will be described by a tight-binding (TB) potential. In the intermediate region, the potential will be described by a cubic spline function, which is computed automatically once its range (1.6-2.2 Å) is specified. A switching function will be applied above 3.8 Å to smoothly terminate the potential at the cut-off distance (4 Å).

Vibrational displacements will be added to Cu atoms in the lattice, but we make no distinction in this example between bulk and surface atoms (the bulk Debeye temperature is used for both). The termination energy will prove to be irrelevant because the simulation will terminate before the atoms have slowed down to the value specified (0.5 eV).

The attractive potential parameters are obtained from chapter 10 of the Simulation primer (in the `\docs` directory). The V(B-B) and V(A-B) potential parameters are irrelevant, since the target is elemental. You can leave them at the default values.



Since the target is an element, the setting for Z(B) is irrelevant in this example. Normally, the surface and bulk binding energy parameters will remain at 0.0.



Note: to calculate the lattice sum parameter yourself, S_n, you can click on one of the buttons marked 'Sn'. Select 'copper.trg' for the lattice file, and enter 179 for the lattice atom index (this is a bulk-like atom located in the middle of the target). After entering these data, press Enter or click OK to save (as `copper.mdl`).

2.5. Impact file (`copper.imp`)

Select the Impact|New menu in Spider.

We are going to simulate a single trajectory, normally incident on Cu(100), aimed at a point between 2 atoms in a <110> row. The central Cu atom in the surface is located at (0, 0, 0) (examine the first line in the TRG file you created). We will examine a trajectory in which the Ar ion is aimed initially at the (x, y) surface point (0.1, 0.0) (in Å). In other words, b_x = 0.1 and b_y = 0.0. The impact parameter (b) is therefore (($b_x^2 + b_y^2$) = 0.1 Å).

- First, specify the displacements: b_x(min) = 0.1; b_x(max) = 0.11; number of impacts = 1. The reason we use 0.11 for b_x(max) is because the logic of the algorithm dictates that the maximum value must be greater than or equal to the minimum value. Sometimes if they are equal the logical comparison can fail because of conversion problems, so we increase the maximum value by an arbitrary amount: if the number of impacts is 1, then only the minimum value will be used.
- Do the same for the b_y (min,max) values: 0.0 (min), 0.0 (max.) and 1 (number of impacts). Keep the (x, y) angle at 90°, and projectile z_0 at 3 Å (= vertical start position for projectile).

- Click Ok or hit Enter to save the file.

You can use the Impact|Orient or Editor|Open menu commands to examine the coordinates in the file you created.

2.6. Inelastic file (`copper.inl`)

The Inelastic file is optional: simulations can be run using purely elastic models. If you do wish to use an Inelastic file, it is also necessary to activate the relevant inelastic loss options in Kalypso itself (via the Simulation Options menu command). In this example, we shall assume that the electronic loss can be calculated as an equipartition (i.e. equal scale factors of 0.5) between the (Lindhardt-Scharff-Schiott) LSS and (Oen-Robinson) OR models.

- Select Inelastic|New: a 4-tab dialog box comes up.
- Select the LSS tab.
- Click the Add Atom button, and fill out the grid as shown below. The K_{LSS} values were computed using the 3rd tab: 'Compute K(LSS), K(OR)' (you can try to reproduce the values given in the table).

	Z1	K(LSS)	Scale
atom 1	18	9.51	0.5
atom 2	29	13.32	0.5

- Enter 0.0 for velocity threshold (the threshold velocity of the inelastic loss mechanism).
- Now click the OR tab.
- Click the add pair button twice
- Fill out the grid as follows (for example, pair 1 represents Ar-Ar interactions: 18*18 = 324):

	Z1*Z2	K(OR)	Scale
pair 1	324	0.00090	0.5
pair 2	841	0.00157	0.5
pair 3	522	0.00117	0.5

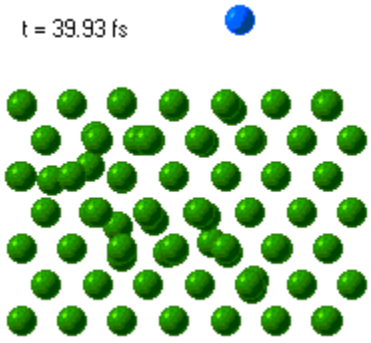
- Enter 1.5 (Å) for the maximum apsidal distance (assumed range of the inelastic interaction), and 0.3 for the exponential parameter (a parameter of the model, normally fixed at 0.3).
- You can ignore the Shapiro-Tombrello and Thermal tabs, as we won't be using those energy transfer models in this example.
- Click the Ok button, and save as `copper.inl`.

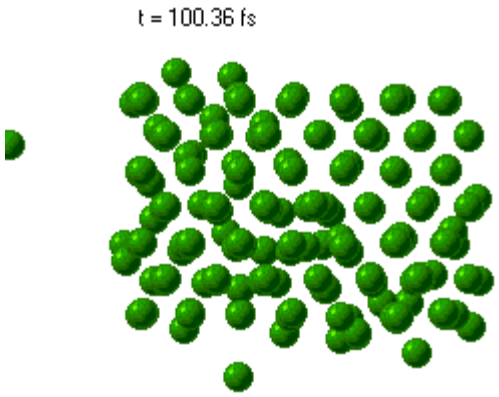
The simulation project is now ready to run, so you can close Spider if you wish.

3. Running the simulation (with Kalypso)

- Start Kalypso.

- Select Run|Specify input files; navigate to the directory which contains your input files (e.g. `c:\kalypso\tutorial`).
- Click button marked ‘projectile’ and select `copper.prj` from the file dialog box, then click ok. You will see that all the other files (`copper.trg`, `copper.imp` etc.) also appear in the main dialog box. Notice that the Inelastic file name appears dimmed - this is because you haven't yet specified which inelastic algorithms you want to use. Click OK.
- The words ‘ready to run’ appear in the screen window. If you run the simulation at this point, the simulation will run, but no inelastic loss effects will be included. To include the inelastic losses, click on the Options|Simulation menu item. Look for the Inelastic Loss Models box, and select (check) the 2 boxes corresponding to OR and LSS losses (i.e., the models you set up in Spider).
- There are a number of other options available in this dialog box. You can read about them in the online help. For now, leave them at their default values.
- Select File|Run (or click the Run speed-button). The simulation begins by initialising variables and outputting information to the screen. To proceed, hit the Enter key, or click Ok with the mouse.
- The simulation runs to completion (in a few seconds on a Pentium).
- See the Annexe below for an explanation of the verbose output written by Kalypso to the screen.
- If you look on disk after running the simulation, you will find a file named `DYNVARS.SNK`, which contains the output data. The Win 95 version of Kalypso will store this file in the directory named in the “File|Specify output file” option (`\kalypso\bin` by default). A better place to store the file would be in the `tutorial` directory (try to do it).
- The images below are *xz* views of the collision as copied from the Graph window (click on the speed-button to open it during or before the simulation) at 40 fs and 100 fs elapsed time respectively (the projectile is indicated by the blue icon):





4. Processing output (with Winnow)

We will now run Winnow from within KALYPSO using the Winnow menu command. (You can run Winnow independently, if you prefer).

- Select Winnow's Process | Filter option.
- Enter `DYNVARS.SNK` as the source file in the file dialog box (use the Browse button to make sure you get the file path correct), then click OK.
- Enter `2ev.snk` as the destination file in the next dialog box. Click Ok.
- Enter [`ke/ep > 2.0`] as the conditional expression in the Filter input line. What this means is that only records satisfying this condition (particle kinetic energy above 2 eV) will be written to your output file (2ev.snk). Here, `ke` refers to kinetic energy in SI units, while `ep` is the proton charge. This is an example of Winnow's query language. Click Ok: the file `2ev.snk` is created. It contains records for 28 atoms. Winnow reports this as follows:
  Filter; [ke/ep > 2.0]
  dynvars.snk (344 recs.) --> 2 eV.snk (28 recs. = 8.1395%)
  There are 344 records in all (343 atoms in target, plus the projectile).
- Now select Process|Convert.
- Enter `2ev.snk` as the file name, and click Ok. This will create a file named `2ev.dat` which contains a listing of the dynamical variables recorded in `20ev.dat`. In particular, the (*x*, *y*, *z*, *px*, *py*, *pz*) coordinates are listed in columns 1-6 respectively (click the Help button on the Convert dialog box for information about the units). The remaining columns refer to the time, the particle mass, the particle 'row number' (`rw`), and the simulation 'run number' (`rn`) respectively. The first two lines (of 28) are shown below.

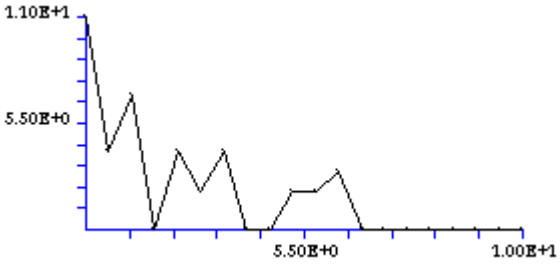
```
8.517761  -0.000005  11.321669   5.777756  -0.000004   7.798811 100.36  6.633E-0026 0  1
-5.824169  3.413858 1.269061  -0.938064   1.628044   2.828470 100.36  1.055E-0025  38  1
```

- If you wish to examine the trajectories, you can do this with either the SNK file or with the DAT file (the one you just created) using the trajectory plotter (found on Winnow's Utilities menu). Alternatively, you can use your own plotting software to display the trajectories (the *x*, *y*, *z* coordinates respectively in Å are found in the first 3 columns of the 20eV.dat file).

The trajectory plotter lets you step through the data on a frame-by-frame basis (among other options). The xz view on the left (obtained by loading 2eV.snk trajectory plotter with the frame refresh option off) shows that only a few atoms in the system acquire energies of 2 eV or greater. The projectile is shown in pink (top right).

That concludes the main tutorial.

As a further exercise you could try creating an energy spectrum based on the data written to `dynvars.snk`. (Select the Process|Spectrum option, and enter `dynvars.snk` as the input (source) file and "`ke/ep`" as the independent variable. Set a range of 1.0-10.0 (eV) and 20 channels. The resulting 'spectrum' - which is not very interesting - is shown below (energy in eV on horizontal axis, counts on vertical axis).

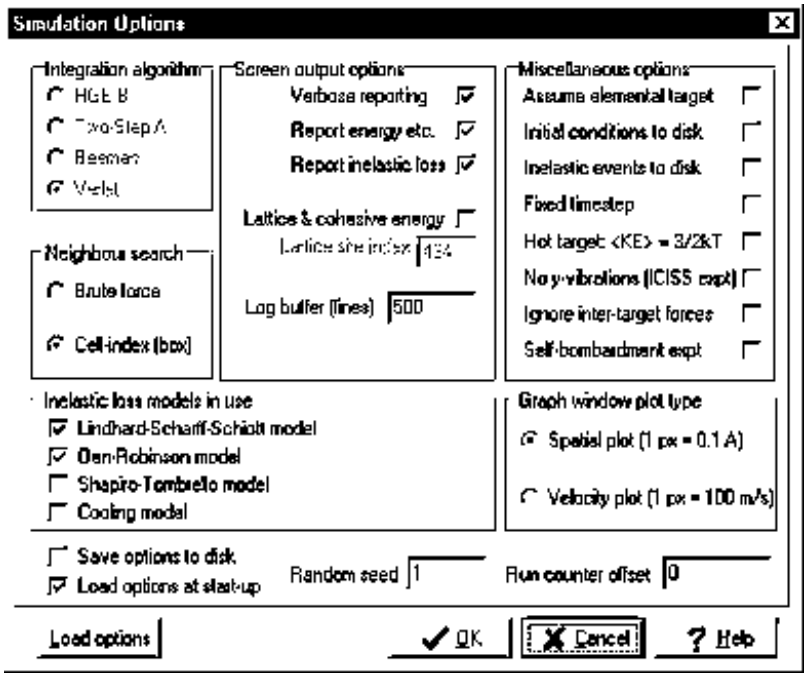


5. Further development of the tutorial

- In a real simulation you would not only use a larger target, but you would also study hundreds of trajectories directed into a symmetrically unique zone of the target lattice. The Impact file **xcopper.imp** which is found in your tutorial, defines a triangular zone containing just 48 such trajectories for illustration. The file was created in Spider using the following parameters: b_x(min) = b_y(min) = 0.0; b_x(max) = b_y(max) = 1.28Å; No. of impacts (both axes) = 10; *xy* angle = 45; proj. z_0 = 3Å. You can run a scaled-down version of a real simulation by substituting `xcopper.imp` for `copper.imp` in Kalypso's list of input files (File|Specify input files, then click the impact file Select button).
- Study the effects of varying the timestep (defined in the Run file) on the energy conservation reported by Kalypso.

6. ANNEXE: Kalypso Screen Output

The following screen output was generated by the Copper project described in this tutorial. The following diagram shows how the simulation options were set in Kalypso (Set-Up menu):



You can load the same options yourself by opening the file 'tutorial.cfg' found in the `tutorial` directory (use the Load options button seen at bottom left in the image above).

Most of the screen output is routine, but it is sometimes useful for diagnostic purposes. Exceptional conditions (errors, warnings) are signalled by output in bright red. These are not always fatal conditions, but they always merit investigation. For example, you may have indicated that you would like to use the LSS inelastic model, but forgot to enter any parameters into the inelastic file. Kalypso will flag this with a warning about 'null parameters', but will allow you to continue.

```
-------------------------------------------------------------------------------------------
SIMULATION PARAMETERS
Output: for all particles
Integration algorithm: Verlet
Sampling at termination
Projectile: 0.500 keV Ar+ (Z=18, mass=39.94800, altitude = 90.00, azimuth = 0.00)
Number of atoms in target: 343
General parameters:
 - Calculations will start with 0.2000 fs timestep
 - Time before termination tests begin: 0.00 fs
 - Time before compulsory termination: 100.00 fs
 - Max. No. of collision partners = 50
 - Including inelastic effects: LSS, OR, model(s)
 - Ignoring image potential effects

=> Ready to run calculations in earnest.
```

- The preceding information summarises the main features of the simulation, as deduced from the input files. A simulation consists of one or more runs. The same output format is repeated for every run. Each run is uniquely identified by a run number, which is counted from 1, and which gets written alongside any simulation output written to disk.

```
Run count: 1
Target : Cu
Tight-Binding potential
 - Potential cuts off at 4.000 A
 - Using switching function for r > 3.800 A
Proj-Targ screening lengths (A): 0.11389, 0.11389
Targ-Targ screening lengths (A): 0.10798, 0.10798, 0.10798
Detected elemental target
```

- Kalypso detected that the target was an element. Kalypso can and must determine whether the target is elemental or complex, since it affects the way forces are calculated. However, you can instruct Kalypso explicitly to treat the target as elemental (even if it is not!) via the Simulations Options dialog box.

```
* INITIALISED *
```

- The following lines give a summary of the system dynamical variables, in the form of energies, momenta and angular momenta. The locations of particles #0 (projectile) and #1 (anchor target atom) respectively are also reported. Lastly, the (*x*, *y*, *z*) physical dimensions of the lattice region plus a bounding region equal in width to the cut-off distance are reported. These numbers refer to the coordinates of 2 lattice corners. The lattice region extends from z = 6.7 Å (= $z1 + z0$ + cut-off)[1] down to z = -14.8 Å.

```
Total System Energy: -557.744 eV
 Potential: -1057.74 eV; Kinetic : 500.0000 eV
 Px:-0.00000 Py:0.000000 Pz:-3.26007 (*1.0E+21 kg m/s)
 Lx:0.000000 Ly:0.326007 Lz:0.000000 (*1.0E+31 Js)
 Projectile (x,y,z) coordinates (A): 0.100000 0.000000 3.000000
 Projectile (x,y,z) velocities (Mm/s): -0.00000 -0.00000 -0.04914
 Particle 1 (x,y,z) coordinates (A): 0.000000 0.000000 0.000000
 Particle 1 (x,y,z) velocities (Mm/s): 0.000000 0.000000 0.000000
 Lattice zone lower coordinates (A): -11.6679 -11.6679 -14.8441
 Lattice zone upper coordinates (A): 12.94593 12.94593 7.000000
```

- NB. inelastic energy loss models do not conserve linear and angular momentum: see the Simulation Primer, chapter 8. The potential energy reported here takes into account the potential at the cut-off distance, so it cannot be directly compared with the cohesive energy of the solid. Use Kalypso's Lattice and Cohesive Energy command for this purpose.

- Discrete Oen-Robinson inelastic events are reported below as they occur. The particle indices are given, along with the associated energy loss. The distance of closest approach, before and after the energy subtraction, are also reported.

```
OR event, particles: 0 and 1, dE = -4.53632 eV, R(old) = 0.742703 A, R(new) = 0.746106 A
OR event, particles: 1 and 123, dE = -2.33834 eV, R(old) = 0.973293 A, R(new) = 0.977121 A
OR event, particles: 1 and 130, dE = -0.67009 eV, R(old) = 1.263120 A, R(new) = 1.266922 A
OR event, particles: 123 and 172, dE = -0.48046 eV, R(old) = 1.401170 A, R(new) = 1.405842 A
OR event, particles: 123 and 173, dE = -0.48101 eV, R(old) = 1.398997 A, R(new) = 1.403636 A
OR event, particles: 123 and 214, dE = -0.95042 eV, R(old) = 1.149587 A, R(new) = 1.152967 A
OR event, particles: 130 and 137, dE = -0.26487 eV, R(old) = 1.418217 A, R(new) = 1.420953 A
OR event, particles: 1 and 228, dE = -0.21835 eV, R(old) = 1.465136 A, R(new) = 1.467822 A
```

[1] $z1$ is the anchor atom vertical position, while $z0$ is the vertical position of the projectile relative to the anchor atom.

```
OR event, particles: 214 and 263, dE = -0.30391 eV, R(old) = 1.476573 A, R(new) = 1.480480 A
OR event, particles: 214 and 264, dE = -0.30112 eV, R(old) = 1.483732 A, R(new) = 1.487707 A
OR event, particles: 137 and 144, dE = -0.19872 eV, R(old) = 1.476513 A, R(new) = 1.479062 A
OR event, particles: 214 and 305, dE = -0.32280 eV, R(old) = 1.398856 A, R(new) = 1.401959 A
          * TERMINATED *

Maximum no. of collision partners required was: 38
```

1. The above number shows that only 38 partners needed to be tracked in the interaction neighbour lists, rather than the 50 specified/anticipated in the Run file.

```
Projectile energy (outside lattice zone) fell to 44.31825 eV
At last check, KE of fastest particle inside lattice zone was 16.83524 eV (for particle #175)
```

- Reports projectile energy, fastest lattice atom kinetic energy

```
 No. of timesteps executed = 198
Timestep at termination = 1.302460 fs
```

- The timestep is reported: it adapts during the simulation.

```
Time elapsed = 100.36 fs.
Getting final configuration...
```

- The dynamical variables are again reported, as at the start of the simulation. The energy loss dE reflects both errors in the integration process, and errors in the OR inelastic energy book-keeping. To isolate the former, you can temporarily disable the inclusion of OR inelastic effects dE is also reported as a fraction of the total system energy (some authors use a different convention). The various inelastic energy losses for this run are also reported.

```
Total System Energy: -627.203 eV
Potential: -969.294 eV; Kinetic : 342.0910 eV
Px:0.005310 Py:0.000352 Pz:-2.84996 (*1.0E+21 kg m/s)
Lx:-0.00017 Ly:0.754178 Lz:-0.00050 (*1.0E+31 Js)
Projectile (x,y,z) coordinates (A): 8.517761 -0.00000 11.32166
Projectile (x,y,z) velocities (Mm/s): 0.008709 -0.00000 0.011756
Particle 1 (x,y,z) coordinates (A): -3.34433 -0.00083 -6.66657
Particle 1 (x,y,z) velocities (Mm/s): -0.00056 -0.00000 -0.00014
Lattice zone lower coordinates (A): -11.6679 -11.6679 -14.8441
Lattice zone upper coordinates (A): 12.94593 12.94593 7.000000
```

- dE is the energy error = integration error + inelastic loss book-keeping error. This is an important parameter, since it essentially gauges the accuracy of the simulation.

```
dE = -2.81842 eV; dE/(KE+|PE|) = -0.2149 %;  (Cumulative: -0.2149 %)
```

- The inelastic energy losses from the various channels are also summarised.

```
LSS loss (eV): This run: -55.5736, cumulative: -55.5736
OR loss  (eV): This run: -11.0664, cumulative: -11.0664
ST loss  (eV): This run: 0.000000, cumulative: 0.000000
```