NCPac

Version 1.0.0

Contents

1 Overview

The NCPac code reads in an xyz coordinate file and for each frame calculates,

- Surface particle locations
- Particle cluster populations and optional filtering
- For a single cluster of particles (possibly a nanoparticle), the centre of mass (COM) to surface histogram. From this histogram, the average COM to surface layer distance, standard deviation, skew, kurtosis and maximum and minimum values
- Classification of surface particle packing (100, 111, 110)
- Surface particle curvature
- Visualization output of surface layer, classification and local curvature
- Copy of XYZ input file but with the 5th column designating surface particles

- Average total and particle type Lindemann Index (LI) using a time average over frames around each analysis frame.

- Fractal dimension using the box cutting method.

- Calculates for each frame,
  - Total and average coordinations for each particle type.
  - Coordination histogram for each particle type.
  - Average first nearest neighbour (nn) environment and their populations
  - Average bond lengths, max/min bond length, bond length histogram std. deviation
  - Particle type-type bond fractions
- Outputs,
  - Dynamical quantities as a function of frame number
  - Local particle surface and bulk visualization

- The q6q6 bond coordination histogram

The program reads in a user designated xyz file which can have any number of frames and can have up to 10 different labels. The program removes any overlapping particles ('padding' particles typically used in some visualization packages) in each frame and can remove clusters below a specified cluster size containing less than some specified number of particles (which is useful in messy atomic deposition simulation files). Finally, the code finds the surface layer and output a new xyz file marking the surface atoms.

In the following manual, bold text represents either file names or input variables from the simulation controlling file **NCPac.inp**.

The code along with citation information is located at the CSIRO Data Access Portal:

[FIX]

Please report all errors and issues to Dr George Opletal

george.opletal@data61.csiro.au

1.1 Licence

The NCPac code uses the CSIRO Open Source Software Licence (Based on MIT/BSD Open Source Licence).

This licence releases the Software as 'open source'. It reflects the terms of the BSD/MIT open source software licences, with a specific CSIRO liability disclaimer included. This means that the Software will be free for use, modification and distribution, for both commercial and non commercial purposes, provided it is attributed to CSIRO and subject to the additional terms that exclude CSIRO's liability.

The Software is licensed subject to any additional conditions notified as part of the download process, including under the Supplementary Licence distributed with the Software. The Supplementary Licence may identify third party components that are licensed separately (including where distributed as part of the Software, or required in order to operate the Software). You agree to comply with the separate terms of these third party licences as a condition for accessing the Software.

You are solely responsible for determining if the Software meets your own requirements for use. Subject to law, the Software is provided 'as-is'. CSIRO makes no representations, warranties or undertakings of any kind, including regarding accuracy, defects, title, merchantability, fitness for a purpose, or non-infringement.

If you require additional licence rights, you should contact the relevant area in CSIRO.

1.2 Installation

NCPac is written in Fortran 90 and has been tested for compilation using the GNU Fortran 6.3 (GFortran) compiler and the Intel Fortran Compiler.

NCPac requires compilation of the **NCPac.f90** main program file and the **NCPac_mod.f90** module file. **NCPac.inp** must always be present and contains the simulation parameters. Additionally, a xyz input file which is to be analysed must also be present (see section 2). All the input files need to be in the same directory and the generated output files write out to the same directory. Below are steps to compiling and running the program on a Unix based machine using the Intel Fortran Compiler.

- Type

  ```
  ifort NCPac_mod.f90 NCPac.f90 -o NCPac.exe
  ```

A successful compilation will create the executable file **NCPac.exe**.

2 File I/O

An output file naming convention is employed to order and simplify the output of an analysis. Files names begin either with the prefix ‘ov_’ or ‘od_’ which signifies visualization xyz coordinate files or data files containing numerical analysis values.

Visualization files can also end with the label ‘_VMD’ which signifies that the file is output for the Visual Molecular Dynamics package (these files contain particle padding to keep a constant number of particles of each type). Visualization files also use the .xyz file type.

Data files are often column based outputs which are suitable for reading into spreadsheet programs such as Microsoft Excel. Data files use the .dat file type.

Commas in the descriptions below are spaces in the actual files while bold text represent files and input variable (defined in section 3).

Table 1 Input and output files

FILE	DESCRIPTION
Porosityplus.inp	Input file for simulation variables
Use defined XYZ file	The name of the input xyz file that is to be analysed should be input in the **NCSurf.inp** file. The user defined XYZ input file needs to be in the form, number of particles space or comment elemental symbol, location along x axis, y axis, z axis of particle 1 in frame 1 elemental symbol, location along x axis, y axis, z axis of particle 2 in frame 2 .. .. Below is an example of a two frame XYZ file with the first frame containing 3 hydrogen atoms and the second frame having 2 hydrogen atoms. 3 H 5.456 2.435 4.435 H 3.454 3.667 6.544 H 4.345 1.234 4.326 2 H 3.454 4.343 5.454 H 2.458 5.197 3.693 The total number of particles in each frame does not have to be constant. The element symbols determines the particle types and should be characters of lengths one or two.

ov_CLASSIFY_VMD.xyz	Final configuration of the local surface packing classification using the labels AA, BB and CC to represent 100, 111 and 110 for VMD.
ov_CURVATURE_VMD.xyz	Final configuration of the local average surface curvature classification using the labels AA, BB, CC, DD and EE to signify if the local particle curvature is between 0-10, 10-20, 20-30, 30-40 and 40-50 degrees from the surface normal for VMD.
ov_LAYER.xyz	A copy of the user defined input XYZ file (within the selected analysis frame range) with the addition of a fifth column where each particle has either a 0 or 1 designating if the particle is a surface (1) or bulk particle (0).
ov_LAYER_VMD.xyz	xyz movie file used for quick visualization within Visual Molecular Dynamics (VMD) with same format as user defined input XYZ file except that the particle labels are replaced with either AA or BB designating bulk and surface particle. This file is padded out with particles at the x,y,z location specified by **in_xpad**,**in_ypad**,**in_zpad**. This is required as the VMD program requires the xyz movies to have a constant number of particles for each element.
od_CLASSIFY.dat	Output histograms of the local surface layer classification and curvature. After the frame number, the file contains the number of particles classified as being in 100, 111 and 110 packing environments. The next five values list the number of atoms which are within some curvature range.
od_LAYER.dat	Information on the surface layer geometry using the surface particle to centre of mass distance histogram. After the frame number, the statistics from this histogram are the maximum, minimum distances, difference between these, the average distance, the standard deviation, skew and kurtosis. The final two numbers are the total number of atoms and the number of surface atoms found.
od_CLUSTER.dat	Cluster number and the number of particles within each cluster. This is the cluster population PRIOR to filtering.
od_GR.dat	Partial radial distribution functions for all the unique particle type combinations.

3 Input File

A sample test input file (**NCPac.inp**) is shown below with explanations. The value(s) at the start of the line is the variable that is read in before the code jumps to the next line while the text follow the values are ignored. The number of lines in the input file remains constant. The rest is a description to assist the user including the name of the parameter at the end of each line in square brackets. A detailed explanation of each line follows this example file.

```
test.xyz                          - name of xyz input file               [in_filexyz]
1                                 - write out to screen(0=NO,1=YES)      [in_write_screen]
0 400.00 400.00                   - x,y,z cell length (Angstrom) (0=AUTO) [in_xl,in_yl,in_zl]
1                                 - PBC 0=none, 1=xyz (if cell length…   [in_pbc_option]
0.00 1.20 1.30 1.40 1.50 1.60     - NN cutoffs eg.1-1,1-2,1-3..,…        [in_cutoff(i,j)]
1 7 1                             - First frame,…      [in_frames_start,in_frames_end,in_frames_jump]
20.00 20.00 20.00                 - VMD output padded..               [in_xpad,in_ypad,in_zpad]
1                                 - FILTER CLUSTERS                      [in_option_cluster]
100                               - Cluster size minimum below which removal    [in_mincluster]
1                                 - FIND SURFACE                         [in_option_surf]
70.0                              - Cone angle beyond which…             [in_cone_angle]
300                               - No of points in spherical…           [in_surf_points]
1                                 - CURVATURE/CLASSIFICATION             [in_surf_classify]
```

```
test.xyz                          - name of xyz input file               [in_filexyz]
```

Name of xyz input file.

```
1                                 - write out to screen(0=NO,1=YES)      [in_write_screen]
```

Output program progression to screen with data.

```
0 400.00 400.00                   - x,y,z cell length (Angstrom)  (0=AUTO )   [in_xl,in_yl,in_zl]
```

Cell lengths of xyz movie. If AUTO is used, then a non-periodic cell is setup which has a length in each directions of 1.5 times that separation between the furthest separated particles in each direction.

```
1                                 - PBC 0=none, 1=xyz (if cell length…   [in_pbc_option]
```

Periodic boundary conditions flag.

```
0.00 1.20 1.30 1.40 1.50 1.60     - NN cutoffs eg.1-1,1-2,1-3..,…        [in_cutoff(i,j)]
```

Cutoff distances (Å) between different particle types. The input order is 1-1, 1-2, 1-3, 1-4,…., 2-2, 2-3, 2-4, … , 3-3, 3-4, …, 4-4, … . The cutoff for 2-1, 3-1 etc are not specified as they are equal 1-2 and 1-3.

The AUTO option (set by making the first value equal to 0) attempts to find the minimum between the 1st and 2nd peaks in the partial g(r). These partial g(r) are output to **od_GR.dat** and should be inspected to ensure that AUTO mode worked correctly. The file is also output if AUTO mode is not used and can be used to determine the cutoff values manually.

```
1 7 1                             - First frame,…      [in_frames_start,in_frames_end,in_frames_jump]
```

The frame analysis range used with the first and last frames to be analysed and the number of frames between each analysis.

```
20.00 20.00 20.00                - VMD output padded..                 [in_xpad,in_ypad,in_zpad]
```

Location of padded particles within the VMD output.

```
1                                - FILTER CLUSTERS                             [in_option_cluster]
```

Flag for using the cluster removal option.

```
100                              - Cluster size minimum below which removal    [in_mincluster]
```

The number of particles in a cluster below which the cluster is removed from the output configuration.

```
1                                - FIND SURFACE                                [in_option_surf]
```

Flag for finding surfaces.

```
70.0                             - Cone angle beyond which…                    [in_cone_angle]
```

The cone angle between pairs of first nearest neighbours. If a particle has no cone angles between its neighbours below this cone angle, it is designated a surface particle.

```
300                              - No of points in spherical…                  [in_surf_points]
```

The resolution or number of points in the equidistant spherical point shell around a particle used in sampling to find surface particles. Low values may miss some nearest neighbour particles resulting in incorrect classify.

```
1                                - CURVATURE/CLASSIFICATION                    [in_option_classify]
```

Flag for calculative curvature and surface packing classification

4 Theory/Operation

4.1 Cluster analysis and filtering

A cluster is defined as a set of particles where each particle must have at least one other particle of the cluster as a first nearest neighbour (separation smaller than cut-off distance). The file **ov_CLUSTER.dat** lists for each frame, the population of each cluster.

If the input xyz file contains numerous clusters, the **in_minicluster** parameter can be used to remove all clusters that contain less particles than the **in_minicluster** value. This is useful for example when trying to study the central growing cluster in an atomistic deposition simulation.

4.2 Surface analysis

4.2.1 Surface particle designation

The surface designation of a particle *i* (shown in blue) is according to the following algorithm.

i) An equidistance spherical mesh of points *k* at unit radius around particle *i* is generated using the algorithm detailed by Rakhmanov[1]. The number of points in this mesh or the resolution is specified in the input file. These points are shown in red.

ii) Using the bond length cutoff distances in the input file, the first and second nearest neighbors *j* of particle *i* are found. Shown in green.

iii) For each point *k* around particle *i*, the minimum angle is found between the vector *i*-*k* and *i*-*j*. After looking over all *k* points, if this minimum angle Θ is larger than cone angle (specified in the input file), the particle *i* is designated a surface particle.

[1] Rakhmanov, Saff and Zhou: Minimal Discrete Energy on the Sphere, Mathematical Research Letters, Vol. 1 (1994), pp. 647-662.

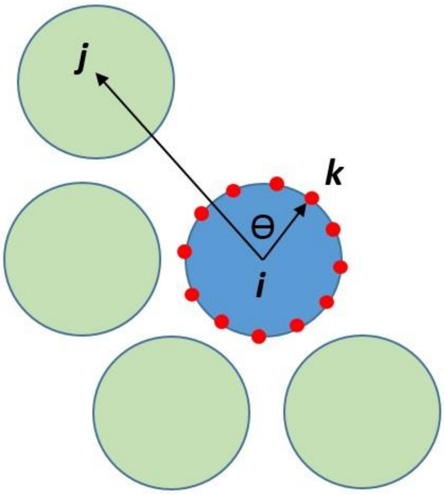


Figure 1 Illustration of the surface particle designation algorithm where designation is based upon an open cone region free of neighbouring particles

4.2.2 Surface layer analysis

If a single cluster remains and the surface layer is determined, then analysis of the surface layer is output to the **od_LAYER.dat** file.

Initially, the mean (μ) surface distance is calculated by averaging the centre of mass (COM) to surface particle distance.

The standard deviation (σ) of the distances *x* from *n* surface particles to COM is given by

$$\sigma = \sqrt{\frac{\sum(x-\mu)^2}{(n-1)}}$$

The standard deviation is a measure of how widely the distances are dispersed from the average value (the mean). A spherical shell of surface particles (spherical cluster) would have a standard deviation near zero.

The skewness (γ) of the surface particle to COM histogram is given by,

$$\gamma = \frac{n}{(n-1)(n-2)}\sum\left(\frac{x-\mu}{\sigma}\right)^3$$

Skewness characterizes the degree of asymmetry of a distribution around its mean. Positive skewness indicates a distribution with an asymmetric tail extending toward more positive values. Negative skewness indicates a distribution with an asymmetric tail extending toward more negative values.

The kurtosis (κ) of the surface particle to COM histogram is given by,

$$\kappa = \frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum\left(\frac{x-\mu}{\sigma}\right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$$

Kurtosis characterizes the relative peakedness or flatness of a distribution compared with the normal distribution. Positive kurtosis indicates a relatively peaked distribution. Negative kurtosis indicates a relatively flat distribution.

4.2.3 Surface curvature and packing classification

The surface curvature for each surface particle is calculated from the displacement vectors from it to its first nearest neighbours. Consider a particle i of coordination four with its nearest neighbours as in figure 2 in a near planar configuration. There exists four angles when loop in a clock-wise or counter clock-wise direction and the three particles defining those angles also define four planes. The surface normal to one such plane is shown by the red arrow for the particles i, j and k in the figure as obtained by the cross product of vectors i-j and i-k.

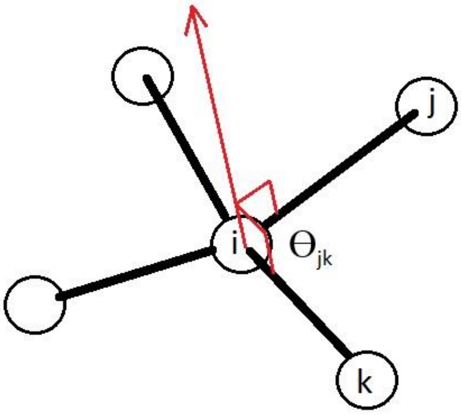


Figure 2 Particle with its first nearest neighbours

The calculation of a particles curvature angle calculates firstly the four surface normal vectors, works out the average surface normal vector from these and then calculates the average angle between this average vector and the four surface normal vectors. Thus, a planar configuration would give a result of zero.

The classification of surface packing uses not only the particle curvature but also the coordination and the bond angle. The ranges used in this code are shown in table 1.

	Curvature (deg.)	Angle (deg.)	Coordination
100	< 15	$70 < \Theta < 110$	4
111	< 15	$40 < \Theta < 80$	6
110	> 15	$40 < \Theta < 80$	6

Table 1 The classification definitions of 100, 111, 110 surface packing environment around a particle.

4.3 Ordering

4.3.1 Lindemann index

The Lindemann Index for particle i is given by

$$l_i = \frac{1}{N-1}\sum_{j(\neq i)} \frac{\sqrt{\langle r_{ij}^2\rangle_T - \langle r_{ij}\rangle_T^2}}{\langle r_{ij}\rangle_T}$$

and the total Lindemann Index for the system is given by

$$L = \frac{1}{N}\sum_i l_i$$

where $\langle \dots \rangle_T$denotes the time average at a constant temperature T (thermal average), r_{ij} is the distance between the *i*th and *j*th particles and N is the total number of particles in the system. This represents the root-mean-square of the relative particle displacement. The time average is usually over a time longer than an particle vibration but shorter than very long times where dynamic evolution of the system occurs or over a significant temperature change.

The Lindemann Criterion states that solids have a value of approximate below 0.1 due to the relatively small displacements from vibrations within solid and liquids are value above 0.1.

The code outputs the average total and particle type Lindemann indexes into **od_LINDEX.dat**. The time average is calculated around each frame. The parameter **in_lindemann_frames** is used to specify how many frames prior and how many frames post each analysis frame the time average is calculated over including the analysis frame.

As an example, if there are 10 frames in the analysis and **in_lindemann_frames** = 3, then the analysis for each frame will be,

First analysis frame = 4 where the time average is over frames 1,2,3,**4**,5,6,7

frame = 5 where the time average is over frames 2,3,4,**5**,6,7,8

frame = 6 where the time average is over frames 3,4,5,**6**,7,8,9

Last analysis frame = 7 where the time average is over frames 4,5,6,**7**,8,9,10

Thus, the output in **od_LINDEX.dat** will have zero values for all frames except 4 to 7. A choice of **in_lindemann_frames** = 0 is undefined as the time average disappears and the local LI equals zero.

The routine also outputs the value of the particle Lindemann Index in xyz format for visualization into the file **ov_LINDEX_VMD.xyz**. The elemental labels are used to visualize different ranges of the particle LI. These are given by,

Lindemann Index	**Particle Label**
Less than 0.001	AA
0.001 – 0.02	BB
0.02 – 0.04	CC
0.04 – 0.06	DD
0.06 – 0.08	EE
0.08 – 0.10	FF
0.10 – 0.12	GG
0.12 – 0.14	HH
0.14 – 0.16	II
0.16 – 0.18	JJ
0.18 – 0.20	KK
0.20 – 0.30	LL
Greater than 0.30	MM

This range of values should be sufficient to cover the transitions from disordered liquid like states (less than 0.1) to ordered solid like states.

The parameter **in_lindem_outxyz** determines if the output xyz file should be produced for visualization within VMD (Visual Molecular Dynamics). Since the file requires padding of particles to work in VMD, it can get quite large for lots of particles and frames.

4.3.2 Fractal Dimension

The fractal dimension is a measure of how a fractal pattern changes with scale. The method implemental in this modules is the Box Counting method adapted from Motofumi T. Suzuki, A Three Dimensional Box Counting Method for Measuring Fractal Dimension of 3D Models, The 11th IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA_2007), Hawaii, USA, 08/2007.

A cell is continuously divided up to produce identical grids of deceasing length and at each scale, the number of occupied grids (containing at least one particle) is counted. The fractal dimension can be estimated from the gradient of the plot of log(N_r) versus log(1/r) where N_r is the number of occupied grids of length r. This module outputs log(N_r) versus log(1/r). The smallest grid is given by **in_xl**/ **in_fradim_maxres**.

4.3.3 q_6 spherical harmonics analysis

The packing environment around a particle can be characterized by the use of the q6 bond-order parameter. For a particle i with n(i) first nearest neighbours (neighbours defined by the cut-off distances **in_cutoff**) , the local orientational structure is characterized by

$$\bar{q}_{lm}(i) = \frac{1}{n(i)}\sum_{j=1}^{n(i)} Y_{lm}(\vec{r}_{ij})$$

where $Y_{lm}(\vec{r}_{ij})$ are the spherical harmonics corresponding to the orientation of the vector $\vec{r}_{ij}$ between particle I and its neighbour j. Restricting l=6, each particle is assigned a vector $\bar{q}_6(i)$ with element fir m = -6,…,6 given by,

$$q_{6m}(i) = \frac{\bar{q}_{6m}(i)}{\left(\sum_{m=-6}^{6}\left|\bar{q}_{6m}(i)\right|^{1/2}\right.}$$

Two neighbouring particles i and j are regarded as 'bonded', their local orientational order added up almost coherently), if the dot product $\bar{q}_6(i) \cdot \bar{q}_6(j) >$ **in_q6order_dotmin**. The q_6q_6 bond coordination is simply the total number of first nearest neighbours where the previously mention dot product criterion is satisfied and represents particles with similar local orientational ordering of their first nearest neighbours. The q6q6 bond coordination has been used previously to find semi-ordered precursor seeds in hard sphere simulations that eventually order into HCP and FCC packing[2,3].

[2] T. Schilling, H. J. Schöpe, M. Oettel, G. Opletal, I. Snook, Phys. Rev. Lett. **105**, 025701 (2010).

[3] T. Schilling, S. Dorosz, H. J. Schöpe, G. Opletal, J. Phys.: Condens. Matter **23**, 194120 (2011).

4.4 Coordination numbers

The coordination of a particle is the number of other particles found within some cut-off distance of that particle. For surface particles, this coordination can be calculated in two ways,

i) The regular coordination which includes counting neighbour below the surface layer.

ii) The surface coordination which only includes other surface designated particles in the count.

For example, an atom on a simple cubic (100) surface has a surface coordination of 4 but a regular coordination of 5 because of the lone atom directly below it.

4.4 Structural units, bond lengths, bond type histogram

4.5 Chain length Analysis

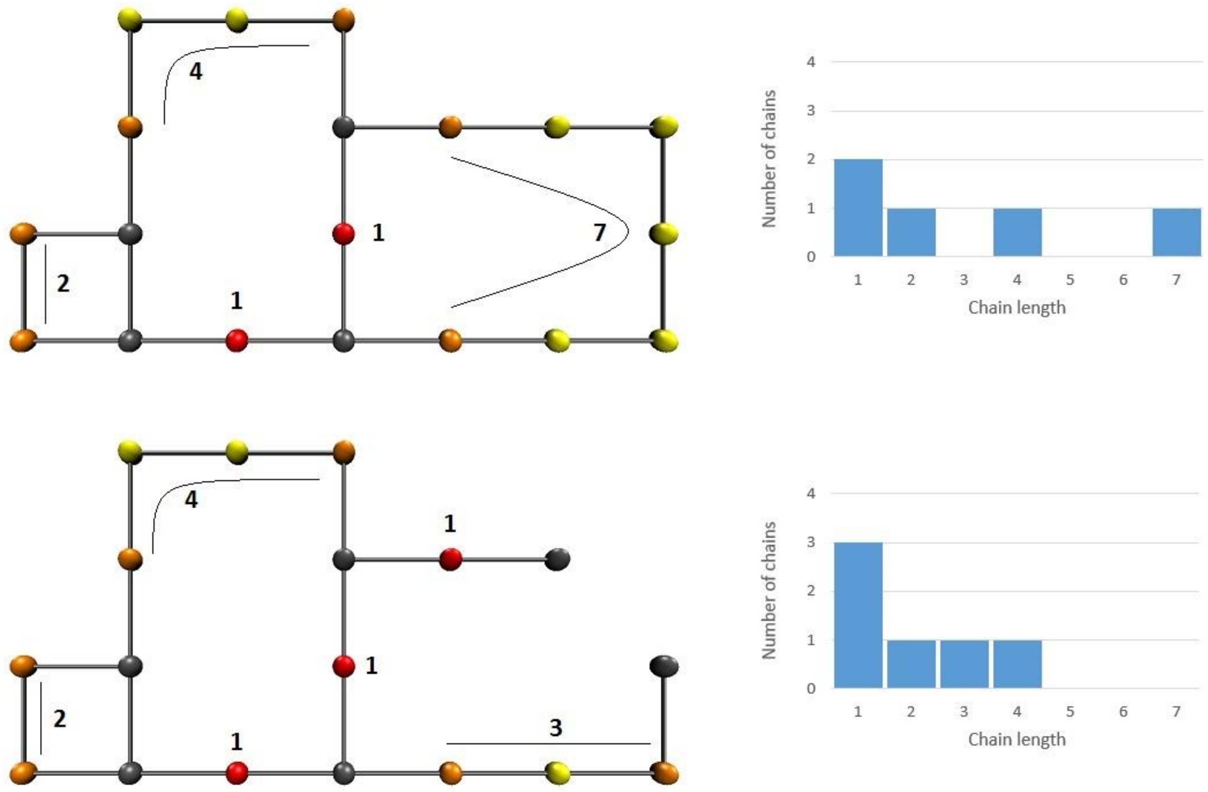


Figure – Two examples illustrating the output of the chain length analysis for a simple test system. These simple structures differ by a single particle in the top right corner. The four particle colours output from the **ov_CHAIN.xyz** file are explained in table X. The corresponding chain length histogram from **od_CHAIN.dat** is also shown along with the locations of the chains overlayed on the network diagrams.

XYZ file label	Colour in fig X	Meaning
AA	grey	Non-coordination two particle
A0	red	Coordination two particle with zero coordination two neighbours
A1	orange	Coordination two particle with one coordination two neighbours
A2	yellow	Coordination two particle with two coordination two neighbours

5 Examples

5.1 Atomic deposition nanoparticle

The **/Example** directory contains the **NCSurf.inp** file and the **test.xyz** file.

A single element (Ru) multi-frame xyz file called **test.xyz** which contains a deposition simulation containing 7 frames is used as an example for surface identification. The xyz file contains 'padded' particles at x=y=z=0 and the code automatically removes these duplicates. Small clusters and lone atoms are also removed in order to filter out only the largest cluster for analysis. The main cluster contains over a 1000 atoms while the second largest cluster was under a 100 atoms (this can be seen by inspecting the **od_CLUSTER.dat** file. Thus, a filtering cluster size **in_mincluster** = 100 was chosen and the code was run.

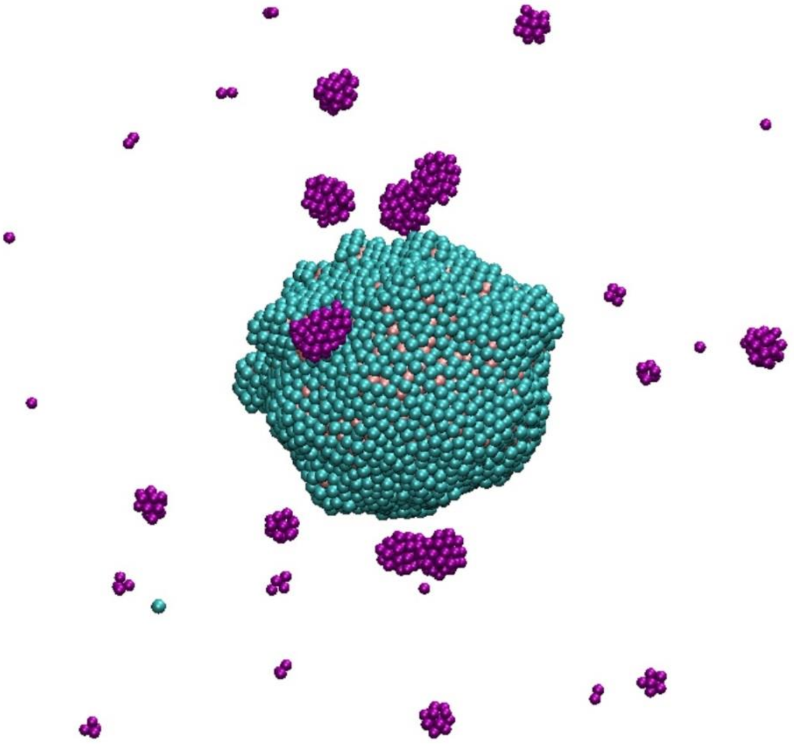


Figure 3 VMD visualization of the large primary cluster and the other small clusters around it in frame 7. The purple atoms are the original configuration (the large cluster overlays the original purple atoms of the large cluster). The blue (surface atoms) and pink (bulk atoms) after filtering are also shown. After filtering, which removed all atoms within clusters smaller than 100 atoms, only the main cluster is left. With only the main cluster left, the centre of mass to surface statistics are now valid as contained in the **od_LAYER.dat** file. The lone blue 'padding' particle at x,y,z = 200Å is in fact many overlapping particles required for multi-frame visualization in VMD.

5.2 Nanoparticle phase changes

The **/Example** directory contains the **NCChaos.inp** file and the **test.xyz** file.

The total Lindemann Index is calculated for a Pt nanoparticle made up of 646 atoms undergoing heating, melting and then cooling and solidification. The initial xyz file consisted of 21009 frames. The initial file was reduced to two files, one consisting of 526 frames (taking 1 frame every 40 frames from the initial) and one file consisting of 2100 frames (taking 1 frame very 10 frames from the initial). The index was calculated at each frame using different values of **in_lindemann_frames** (F value in the figure).

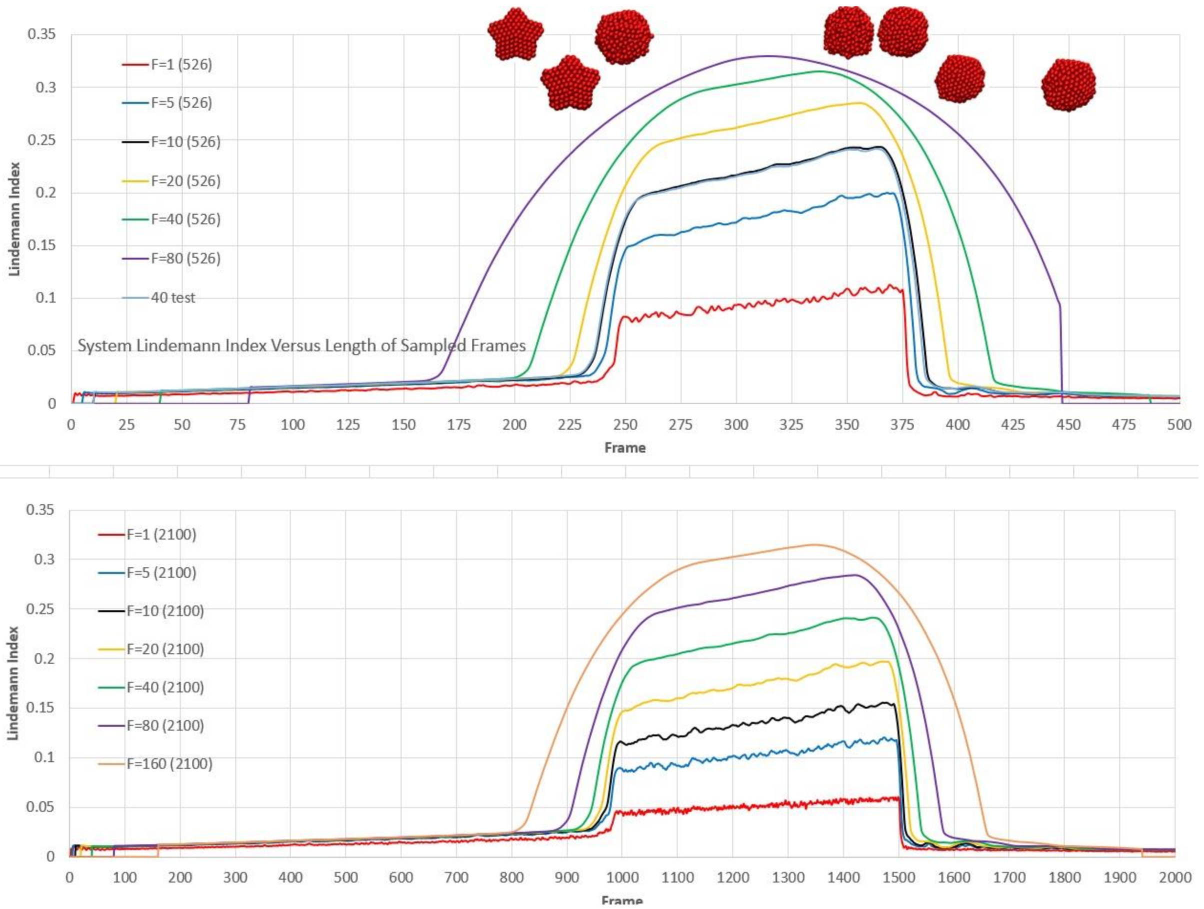


Figure 2 Total Lindemann Index calculated for Pt nanoparticle undergoing melting and recrystallization for different values of the **in_lindemann_frames** parameter using a 526 and 2100 frame xyz datasets.

A few useful observations are made,

- The truncation at the beginning and end of the data are due to the averaging. If **in_lindemann_frames** = 10, then the LI is calculated from frame 11 to the last frame – 11.
- The sampling should ideally be done at equilibration at a constant temperature. Sampling too often (large F values), smooths the index and the predicted transition into the liquid phase occurs too early (index > 0.1). This can be seen above in the F=80 (526) system. Sampling too little result in a small amplitude and the <0.1 criterion becomes inval id. However, the sharp rise in the index, marking the melting time (frame), is consistently at the correct frame.
- Due to the fact that the time average is either side of the central frame, sudden transitions are smoothed more with increasing value of **in_lindemann_frames**.
- In the above, 40 test compares the F=10(526) analysis to that of the F=40(2100) analysis and they are as expected overlapping since 2100/526 is approximately 40/10.
- The **ov_LINDEX_VMD.xyz** file can get large since the number of lines in that file is given by

  total number of analysis frames x number of particles x 13 where the value of 13 comes from the 13 different particle label used to differentiate the different values of the index. This is required for proper visualization in VMD.

The input file and the 526 frame xyz file are found in the example directory.

5.3 Tracking structure units in chalcogenide glasses

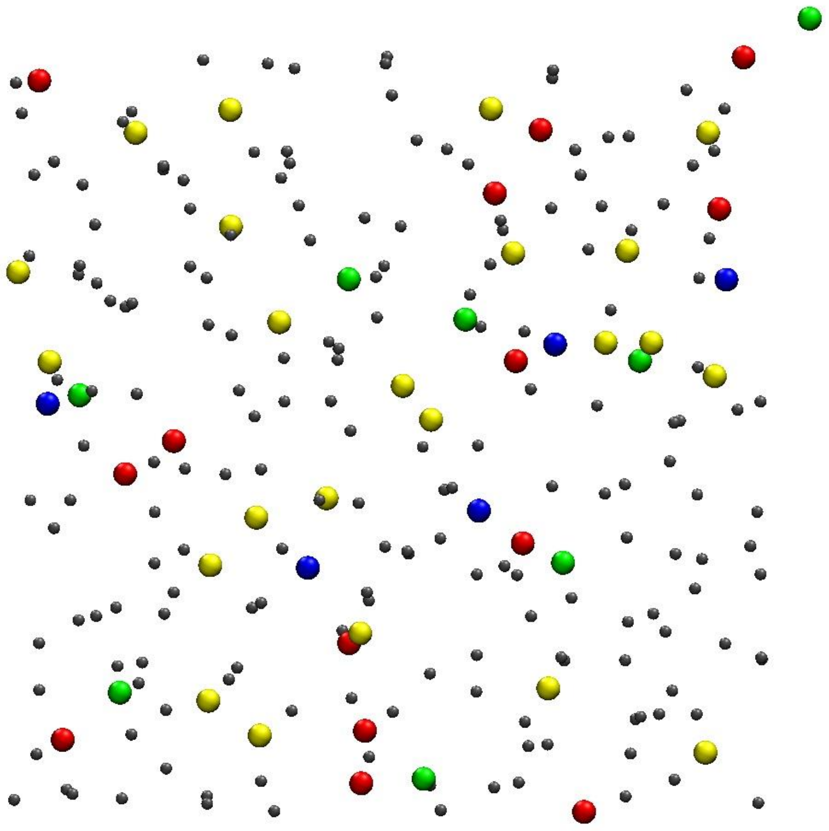
The example xyz movie file is a 240 atom, three component glass (GeAsSe) with periodic boundaries (x=y=z=19.05144Å). The setup input file (**NCstats.inp**) performs analysis from frame 10 to 50 every 2 frames (so 21 frames analysed). The first 5 structural units tracked at given by,

```
3 3 0 0
1 0 0 4
1 0 0 2
3 3 0 1
1 0 0 3
```

which represents Se-Ge$_3$, Ge-Se$_4$, Ge-Se$_2$, Se-Se-Ge$_3$, Ge-Se$_3$. The **outBOND_SUtrack.dat** file contains the number of particles having these first nearest neighbour environments for each analysis frame as shown below,

```
STRUCTURAL UNITS TRACKED
 Frame    SU1    SU2    SU3    SU4    SU5    SU6    SU7    SU8    SU9    SU10
   10       5     14     23      0      7      0      0      0      0      0
   12       5     14     21      0      8      0      0      0      0      0
   14       3     13     18      0     10      0      0      0      0      0
```

These populations are output for visualization in the files **outBOND_SUtrack.xyz** and **outBOND_SUtrack_vmd.xyz**. **outBOND_SUtrack_vmd.xyz** is visualized below via VMD.



The SU1, SU2, SU3, SU4 and SU5 environments are shown in blue, red, yellow, pink and green for frame 10. The top left green atom is the padded atoms at x=y=z=20Å as specified in the input file. This movie is dynamic and can illustrate how the populations of these environments change over time.

5.4 q_6q_6 bond coordination in a icosahedral nanoparticle

The **/Example** directory contains the **NCOrder.inp** file and the **test.xyz** file.

A single element (H), multi-frame xyz file called **test.xyz** which contains a heated, then melted and cooled 646 atom icosahedron nanoparticle over 53 frames is used as an example. The first line of the **od_Q6Q6HIST.dat** file is shown below truncated at the T12 column. No surface q6q6 bond coordination is shown as no surface layer 5th column was used in the input xyz file.

```
q6.q6 > in_q6order_dotmin Similiarty Coordinaton Histogram
Frame Avg_total  Avg_bulk    Avg_surf  T0    T1    T2    T3    T4    T5    T6    T7    T8    T9   T10   T11   T12
1     9.7802     9.7802      0.0000     2     2     6     1    10     0    20    70    59   161    25     0   290
```

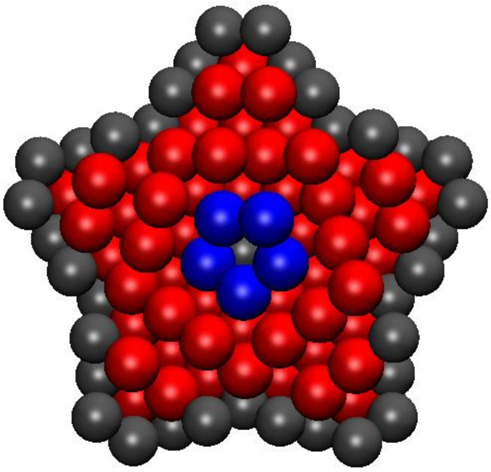


Figure 2 VMD visualization of the q6q6 bond coordination of a cross section of the starting icosahedron nanoparticle (from the file **ov_Q6Q6HIST_VMD.xyz**). The red is labels 'A12', which is the twelve-fold bond coordination, represents the FCC and HCP internal packing.

CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.data61.csiro.au

FOR FURTHER INFORMATION

Dr George Opletal
e george.opletal@data61.csiro.au
w https://research.csiro.au/mmm/

WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today – for our customers, all Australians and the world.

WE IMAGINE
WE COLLABORATE
WE INNOVATE