Supplementary Information: Manual

NanoMaterialsCAD: Flexible Software for the Design of Nanostructures

Giorgos Nikoulis, Panagiotis Grammatikopoulos, Stephan Steinhauer[1]*, Joseph Kioseoglou**

G. Nikoulis, Prof. J. Kioseoglou

Department of Physics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece

Dr. P. Grammatikopoulos, Dr. S. Steinhauer

Okinawa Institute of Science and Technology Graduate University, 1919-1 Tancha, Onna-Son, Okinawa 904-0495 Japan

*E-mail: sifisl@auth.gr

NanoMaterialsCAD is a new open source tool created for the production, manipulation, and 3D visualization of crystalline structures at the nanoscale. It is designed for pre-processing atomistic systems to be used as input for atomistic (e.g., molecular dynamics or Monte Carlo) or *ab initio* (e.g., density functional theory) computer simulations. Below, we further analyze the tools of the program, such as the command line and the main menu. Furthermore, we introduce some utilities that are not mentioned in the main body of this work, namely the script implementation and the right-click menu.

Commands and descriptions

At the very bottom of the window, we implemented a command line for various utility commands. We also offer an "auto-filler prediction" while the user can use, to add a quality of life experience. The user can hover through the predictions by using the arrow keys and auto-complete the command by pressing the enter key. All the commands are listed below:

- cut(h, k, l, D)
  Cut a parallel surface to the (hkl) facet with D Angstrom distance from the center.
- clean
  Delete all the surfaces that were previously created by the command "cut(h, k, l, D)"
- undo
  Delete the previous surface that was created by the command "cut(h, k, l, D)"
- supercell(x,y,z)
  Periodically duplicate the initial system by x*y*z times along the three axes.

[1] Current address: Department of Applied Physics, KTH Royal Institute of Technology, SE 106-91, Stockholm, Sweden

- script(“C:\Path\to\the\script”)
  Opens and executes the script from the according path.
- sphere
  Render the atoms as spheres.
- points
  Render the atoms as points.
- rndDisplacement(r)
  Randomly displaces all atoms by r Angstrom from their position.
- render on/off
  Turns on and off the rendering process of the atoms
- randSelection%(p)
  Select N amount of atoms. Every atom has a chance of p to be selected with p = 0-1
- randSelection(N)
  Randomly select N atoms.
- changeElement(x)
  Change the currently selected atoms to the element x. For example x could be Fe, Si, Se etc.
- save(“C:\Path\to\save\the\POSCAR”)
  Saves the current system as a POSCAR format to the according path.
- loadOrigin
  Deletes every change that has been made by the user and loads the initial system.
- rotate(axes (x, y or z), theta)
  Rotates the selected atoms along the x, y or z axis by theta amount.
- delete
  Delete selected atoms
- restart
  Delete every data and start fresh.
- fopen(“C:\Path\to\the\data\file”)
  Instead of drag and drop a POSCAR or .lmp file to the program, we can also open the file by using this command. The command is mainly intended to be used when writing a script.
- mkfacet(h, k, l)
  Rotate the current unit-cell such as the (h,k,l) surface is perpendicular to the z axis. The periodic conditions are also taken into account.
- randstress
  Randomly adds stress to the system.
- icosahedron(lattice constant, noshells)
  Creates an icosahedron NP (see Defected NPs paragraph).
- decahedron(p, q, r, lattice constant)
  Creates a decahedron NP (see Defected NPs paragraph).

Menu-bar

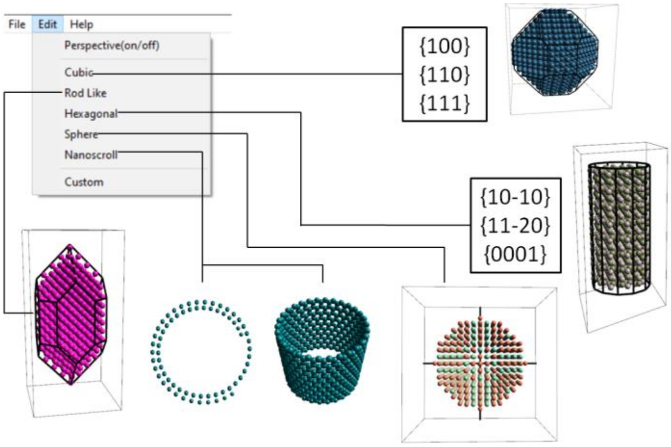


Figure 1: Edit menu options

After creating a bulk system using the command “supercell(x,y,z)”, someone can directly create NPs with predefined surfaces with the options “Cubic”, “Rod Like”, “Hexagonal” and “Sphere”, which are located at the top menu bar of the program. The surfaces of the NPs can be changed as described in the below examples. Here we demonstrate some NPs that we can create by using those options and a brief explanation of their implementations.

- Sphere

  The code calculates the distance of every atom of the bulk system, from the center of the system. By comparing those distances with a fixed radius R, we cut the bulk as a sphere based on that radius. We can increase/decrease the radius by using the “q” /“w” keys respectively.

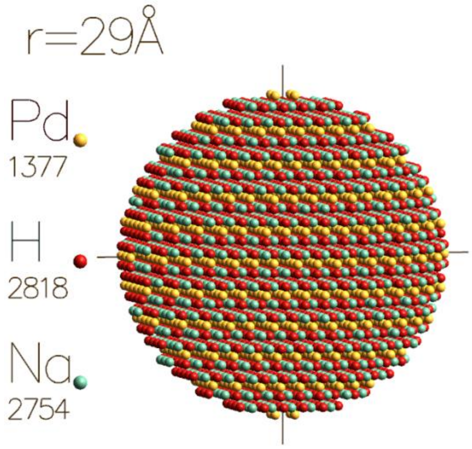
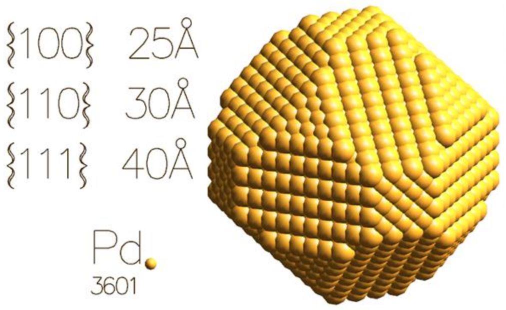
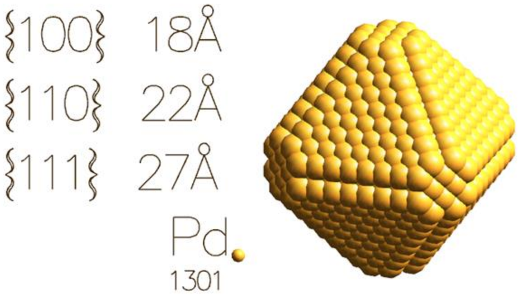


Figure 2: Spherical nanoparticle composed by the tetragonal I4/mmm Na2H2Pd crystal

- Cubic

The cubic option is composed by the {100}, {110} and {111} cubic surface families. The code calculates the distance of every atom from the center of the crystal and compares it with the 3-D plane equation $Ax + By + Cz = D$ for every plane family. For example if we want to compare it with the {110} family we have $|A| = 1$, $|B| = 1$, $C = 0$ and $D = d > 0$. The variable "d" is displayed on the top left corner of the program for every family and can be adjusted by pressing "q", "w" for the {100}, "a", "s" for the {110} and "z", "x" for the {111} families to decrease or increase it respectively. The facets that don't participate to the formation of the particle show in gray (if some surfaces are too far away to intersect with the rest of the NP's surfaces).





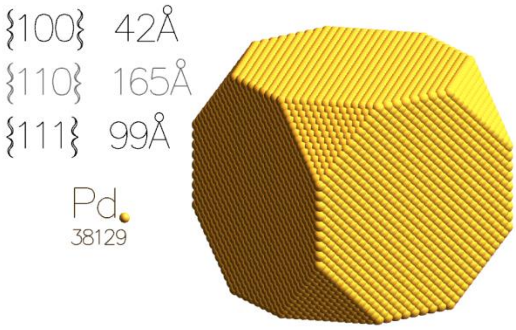


Figure 3: Pd $Fm\overline{3}m$ crystal with various plane parameters [1][2][3]

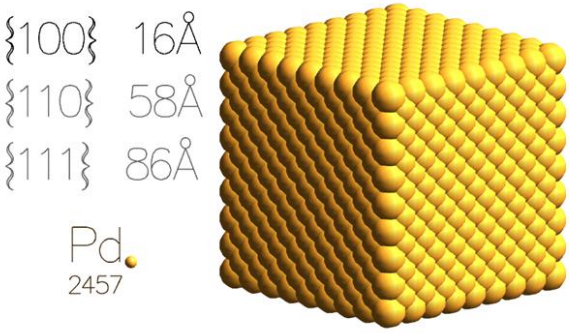


Figure 4: Cubic crystal from Pd $Fm\overline{3}m$ [4]

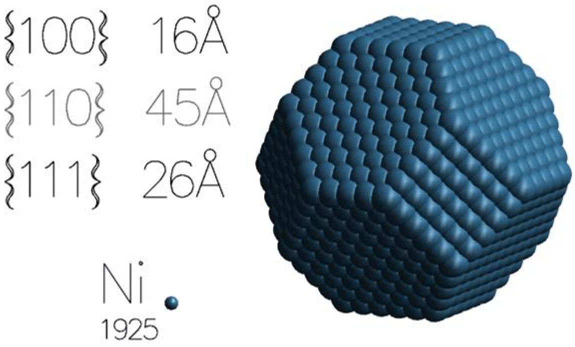


Figure 5:Ni Fm3m crystal composed of {100} and {111} facets [5]

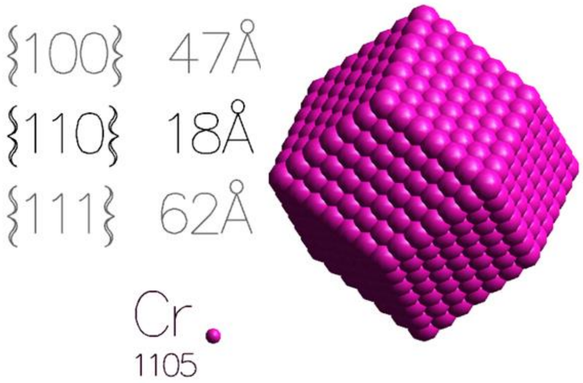


Figure 6: Rhombic Dodecahedron, Cr $Im\bar{3}m$ crystal [6]

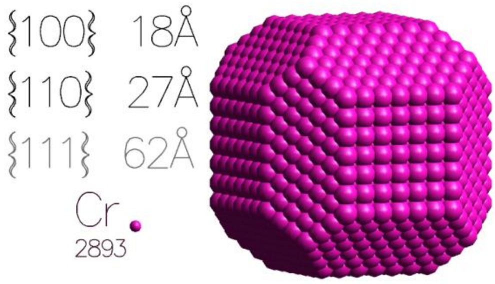


Figure 7: Cr $Im\bar{3}m$ crystal composed of {100} and {110} facets [1]

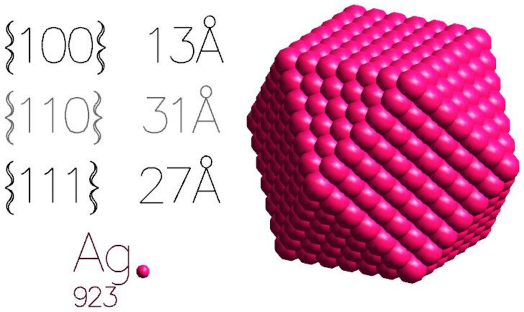


Figure 8: Cuboctahedron, Ag $Fm\bar{3}m$ crystal [7]

- Rod-Like

Rod-Like crystals [1] are made from the cubic mold by eliminating the influence of the (001), (00-1), (101), (011), (-101), (01-1), (10-1), (-10-1), (0-11), (0-1-1) surfaces.

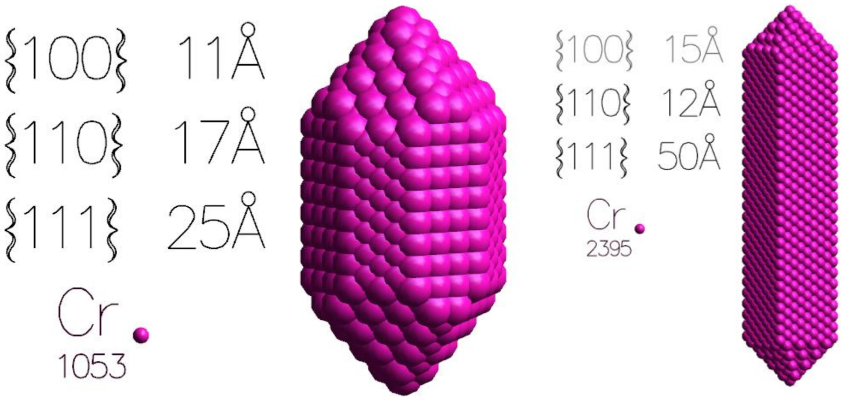


Figure 9: Rod-Like crystals, Cr $Im\bar{3}m$ crystal [1]

- Hexagon
  The “Hexagon” option works by using surfaces from the hexagon symmetry. More specifically, the facets are the $\{11\bar{2}0\}$ family which creates a hexagon and the $\{10\bar{1}0\}$ family which also creates a hexagon, rotated thirty degrees along the z axes in relation to the previous one, and finally the {0001} family. For code-consistence purposes, the Miller indices are translated into Cartesian planes. By doing so, we can use the same source code to create the hexagonal and the cubic NPs.

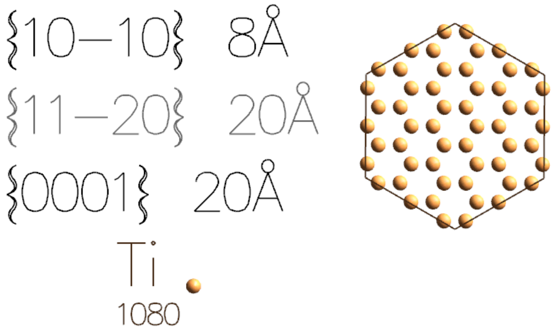


Figure 10: Ti $P6_3/mmc$ constructed with the $\{10\bar{1}0\}$ and {0001} families looking at the (1000) plane [8]

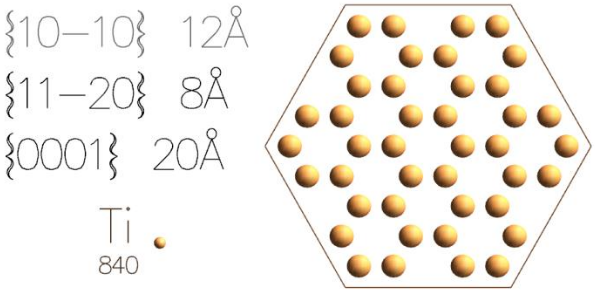


Figure 11: Ti $P6_3/mmc$ constructed with the $\{11\bar{2}0\}$ and {0001} families looking at the (0001) plane [8]

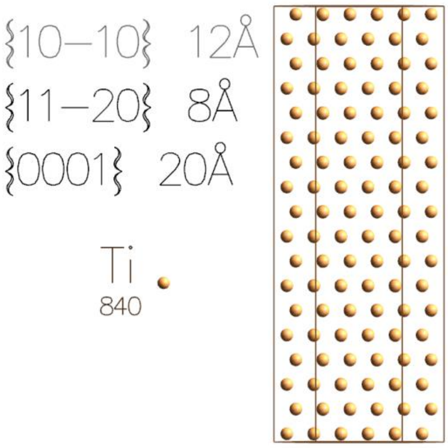


Figure 12: Ti $P6_3/mmc$ constructed with the $\{11\bar{2}0\}$ and {0001} families looking at the $(11\bar{2}0)$ plane [8]

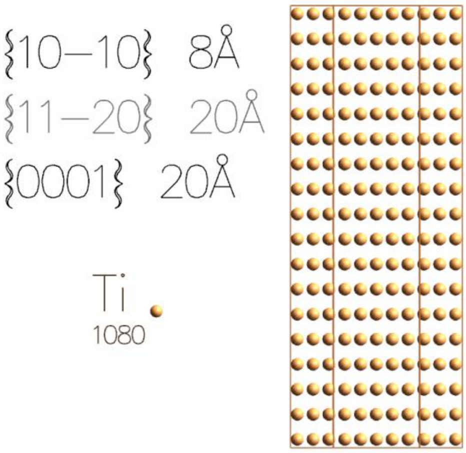


Figure 13: Ti $P6_3/mmc$ constructed with the $\{10\bar{1}0\}$ and {0001} families looking at the $(10\bar{1}0)$ plane [8]

- Hollow cluster

While creating a NP using any of the above options, except the “Nanoscroll”, we can press “ctrl+v” to create a hollow cluster inside a bulk system. The shape of the cluster is identical with the shape of the corresponding NP.

1.

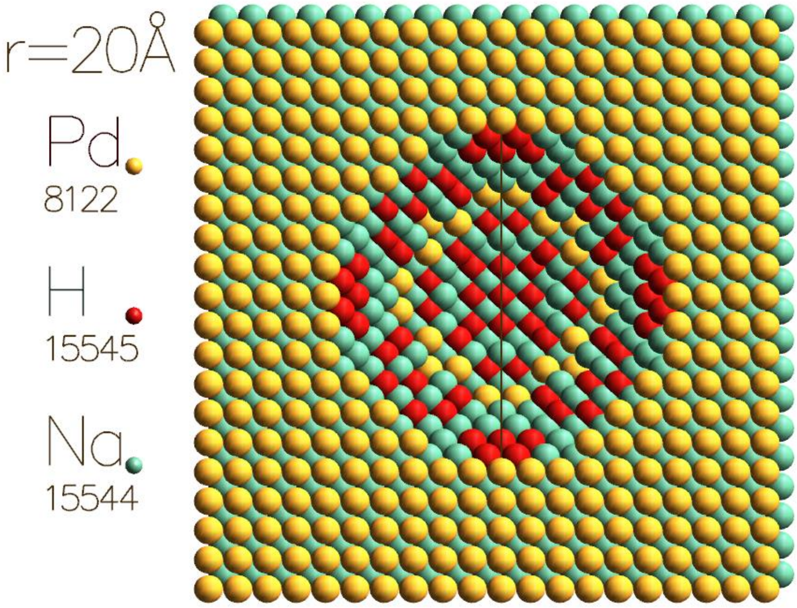


Figure 14: Semi-spherical vacuum on the (001) surface of a tetragonal I4/mmm Na2H2Pd crystal

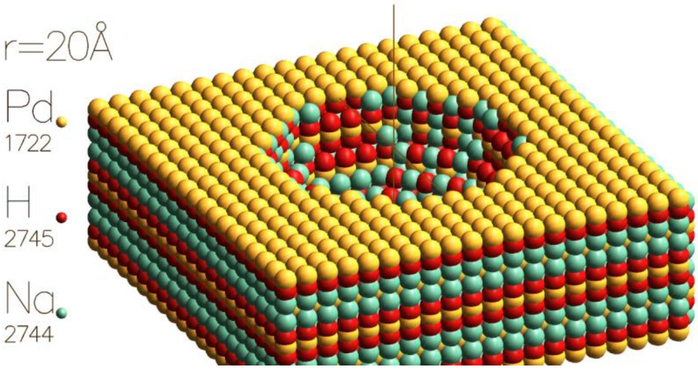


Figure 15: Semi-spherical vacuum on the (001) surface of a tetragonal I4/mmm Na2H2Pd crystal

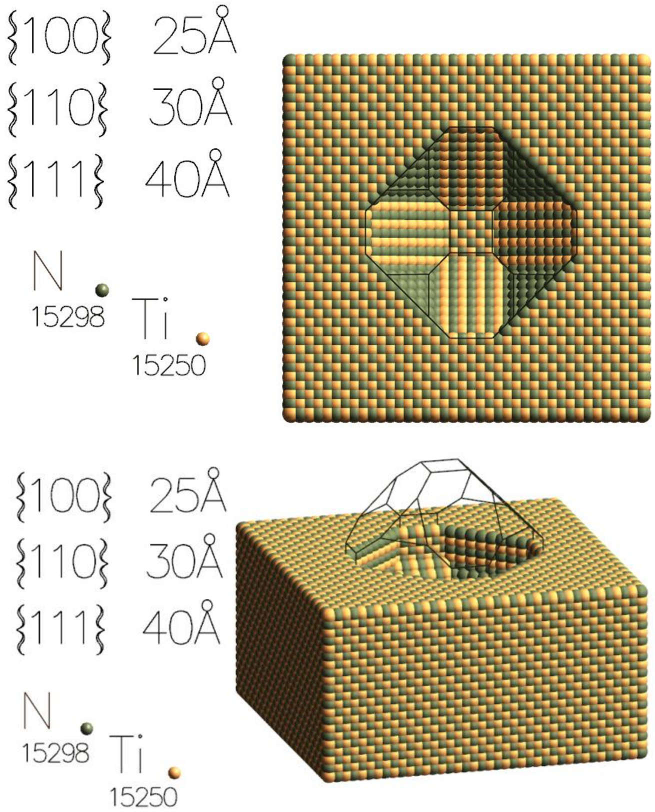


Figure 16: Vacuum on the (001) surface of the TiN $Fm\bar{3}m$ crystal

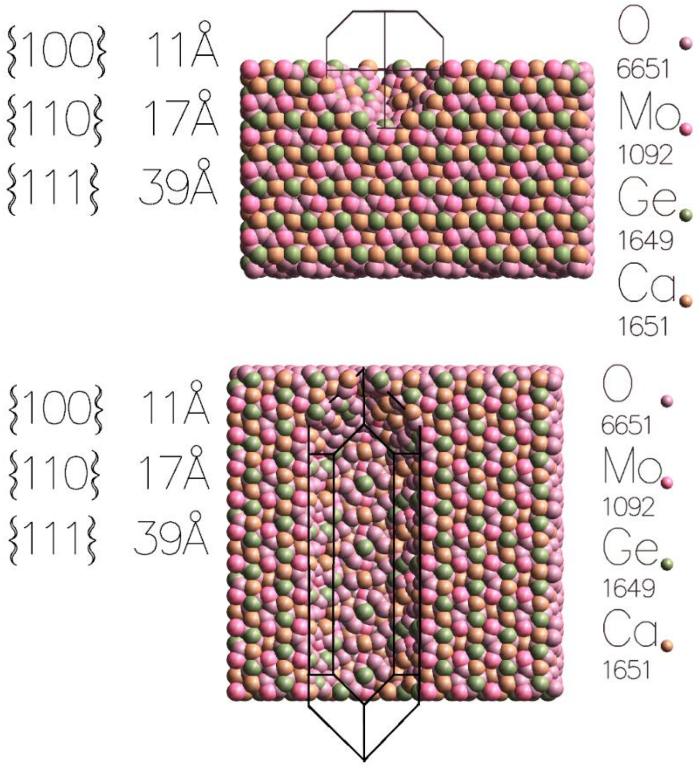


Figure 17: Vacuum on the (100) surface of the Ca3Ge3(MoO6)2, $Ia\bar{3}d$ crystal

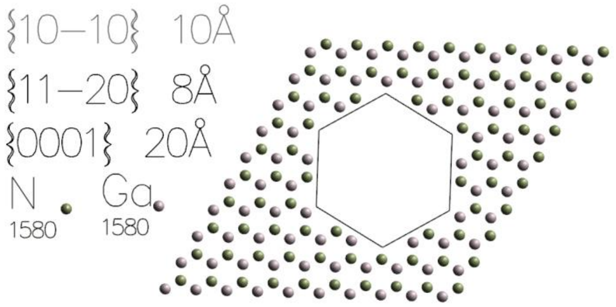


Figure 18: GaN $P6_3mc$ a vacuum using GaN Crystal.

Right click menu

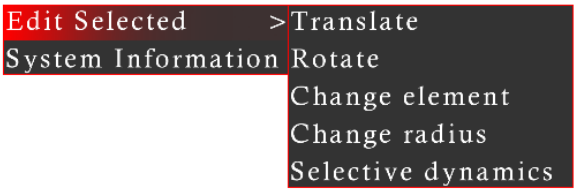


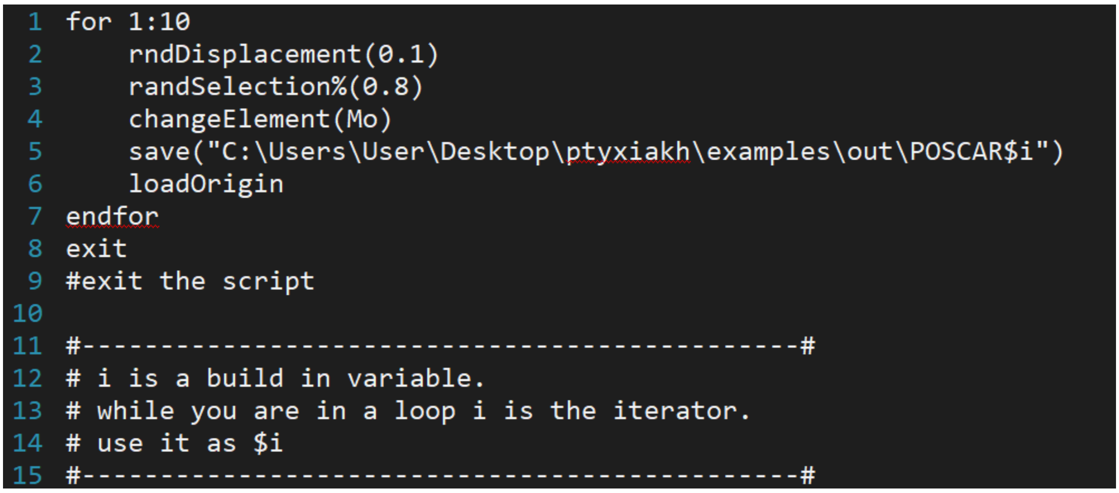
Figure 19: Right clicking anywhere on the GUI, pops out a number of options.

This menu handles and edits the currently selected atoms of the system. In order to select atoms of the system, press “s” to change to the select tool, or click on the select tool icon at the left of the program. Then click and drag to select atoms, or press shift and click individual atoms to select them.

From the menu, select “Translate” or “Rotate” to translate the selected atoms, or rotate them along their centroid, or geometric center. The “Change element” option, pops a GUI window in which the user can insert the elements to which he wants to change the selected atoms (e.g. Mo). The “Change element” option also pops a GUI window which the user can change the rendered radius of the atoms. The default value is 1 and the units are arbitrary. The “Selective dynamics” option changes the selective dynamics of the POSCAR output. For example we can type FFT to allow forces only along the z direction.

- **Script example for the script command**

As mentioned, someone can write their own scripts and run them using the “script” command. The script can contain any of the commands mentioned. This works exactly as if the user entered the commands one by one in the command line. The script must always terminate with the line “exit”. We also offer a bash-like for loop with a $i variable which can be used to output multiple POSCAR$i files using one script. Below is a demonstration of such a script.



```
for 1:10
    rndDisplacement(0.1)
    randSelection%(0.8)
    changeElement(Mo)
    save("C:\Users\User\Desktop\ptyxiakh\examples\out\POSCAR$i")
    loadOrigin
endfor
exit
#exit the script

#----------------------------------------------#
# i is a build in variable.
# while you are in a loop i is the iterator.
# use it as $i
#----------------------------------------------#
```

Figure 20: Example of a script

The script loops 10 times all command between the “for 1:10” and “endfor” lines. rndDisplacement(0.1) randomly displaces all the atoms by 0.1 Angstrom. randSelection%(0.8) randomly selects 80% of the atoms in the system. changeElement(Mo) changes all selected atoms to Mo. The save command saves the current system as a POSCAR file to the corresponding path. The $i variable takes values from 1 to 10 for each loop. Hence, the name of the files will be POSCAR1, POSCAR2 etc. loadOrigin discards all changes made in the system, and loads the original system.

- **Defected NPs**

The program offers two defected NP shapes. The icosahedron and the decahedron [9]. The program also adds 15 Å vacuum by default. It defaults used Hydrogen atoms to construct the NP which can then be changed by the change element option.

The command *icosahedron(a, n)* creates an icosahedron NP with *a* (in Angstrom) lattice constant and *n* (integer) number of shells. Bellow are some examples of this command.

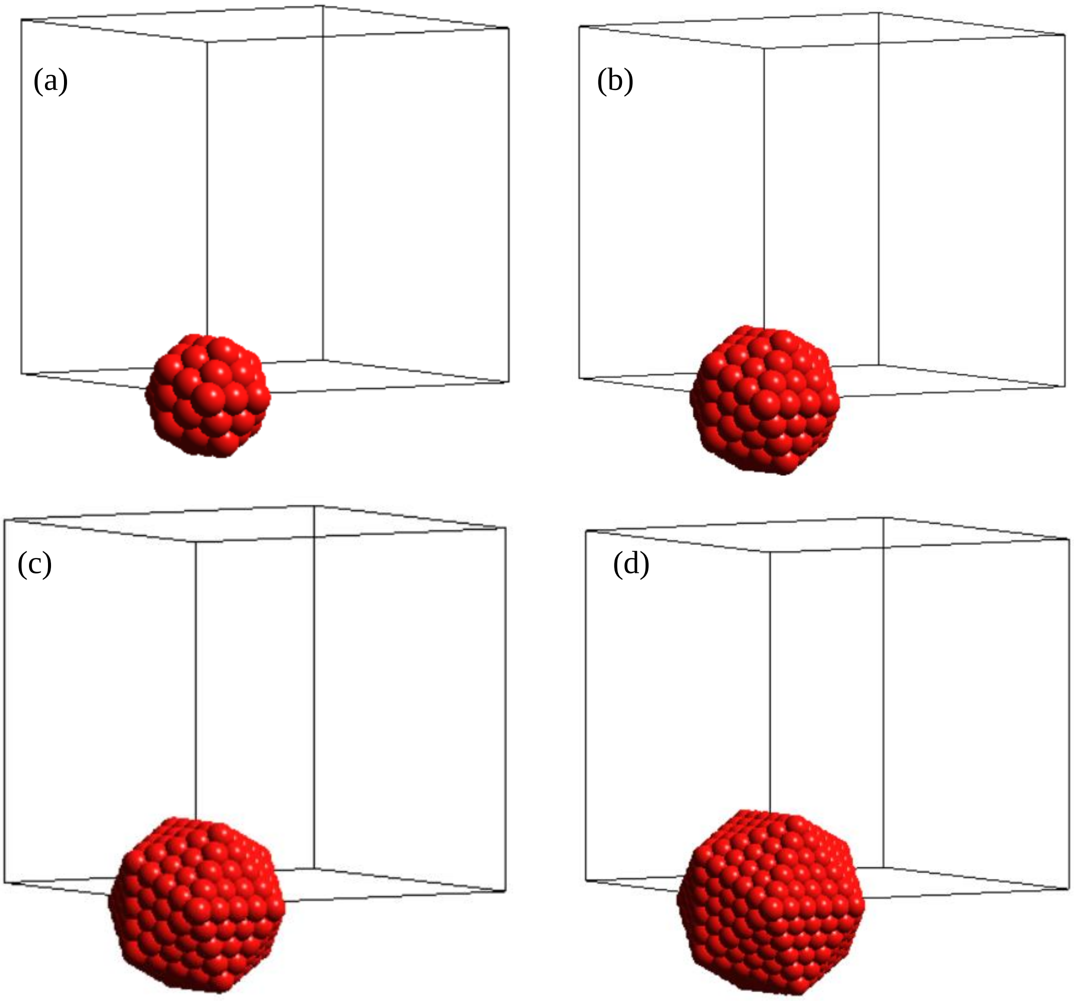


Figure 21: Icosahedron NPs using the commands (a) icosahedron(2,3) (b) icosahedron(2,4) (c) icosahedron(2,5) (d) icosahedron(2,6).

The *decahedron(p, q, r, a)* creates a decahedron NP. Where *a* (in Angstrom) is the lattice constant, *p* is the number of atoms perpendicular to the 5-fold axis, *q* is the number of atoms parallel to the 5-fold axis, and *r* depth of the marks re-entrence at the pentagon corners. r = 0 corresponds to no re-entrence. Bellow are examples that explain the *p*, *q*, and *r* parameters.

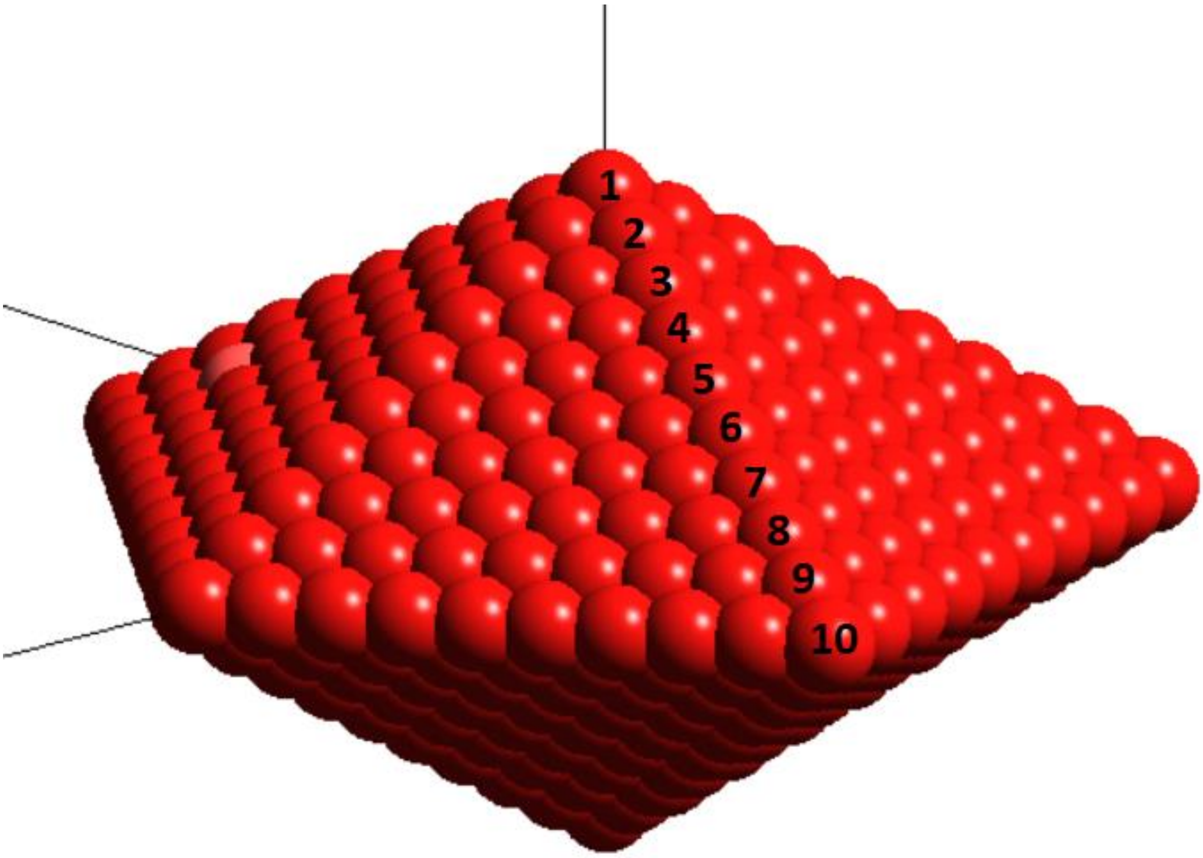


Figure 22: Decahedron NP using the *decahedron(10,1,0,2)* command. In this example the parameter *p* is equal to 10. The picture numbers and shows the logic of the *p* parameter.

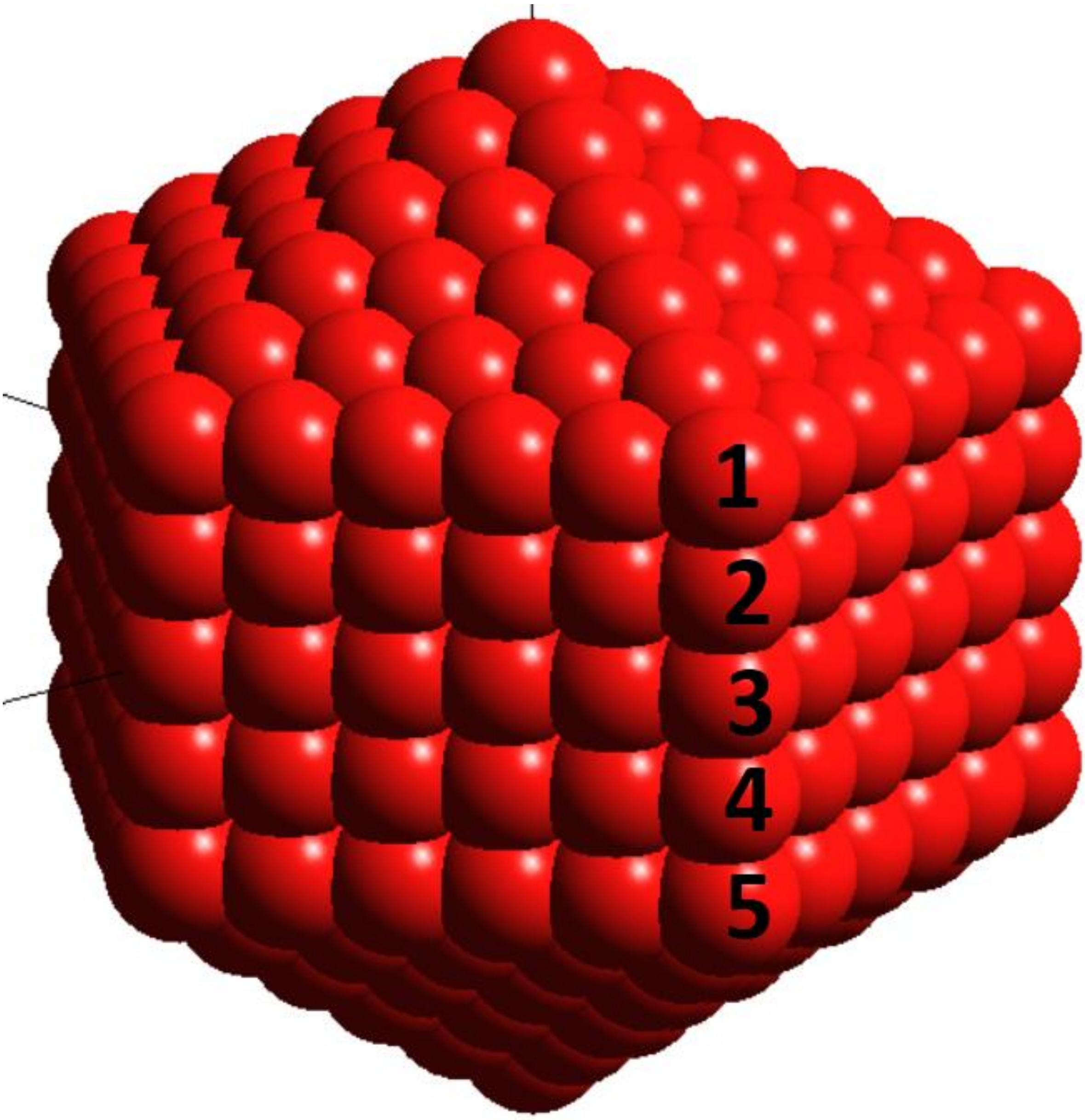


Figure 23: Decahedron NP using the decahedron *(6,5,0,2)* command. In this example the parameter *q* is equal to 5. The picture numbers and shows the logic of the *q* parameter.

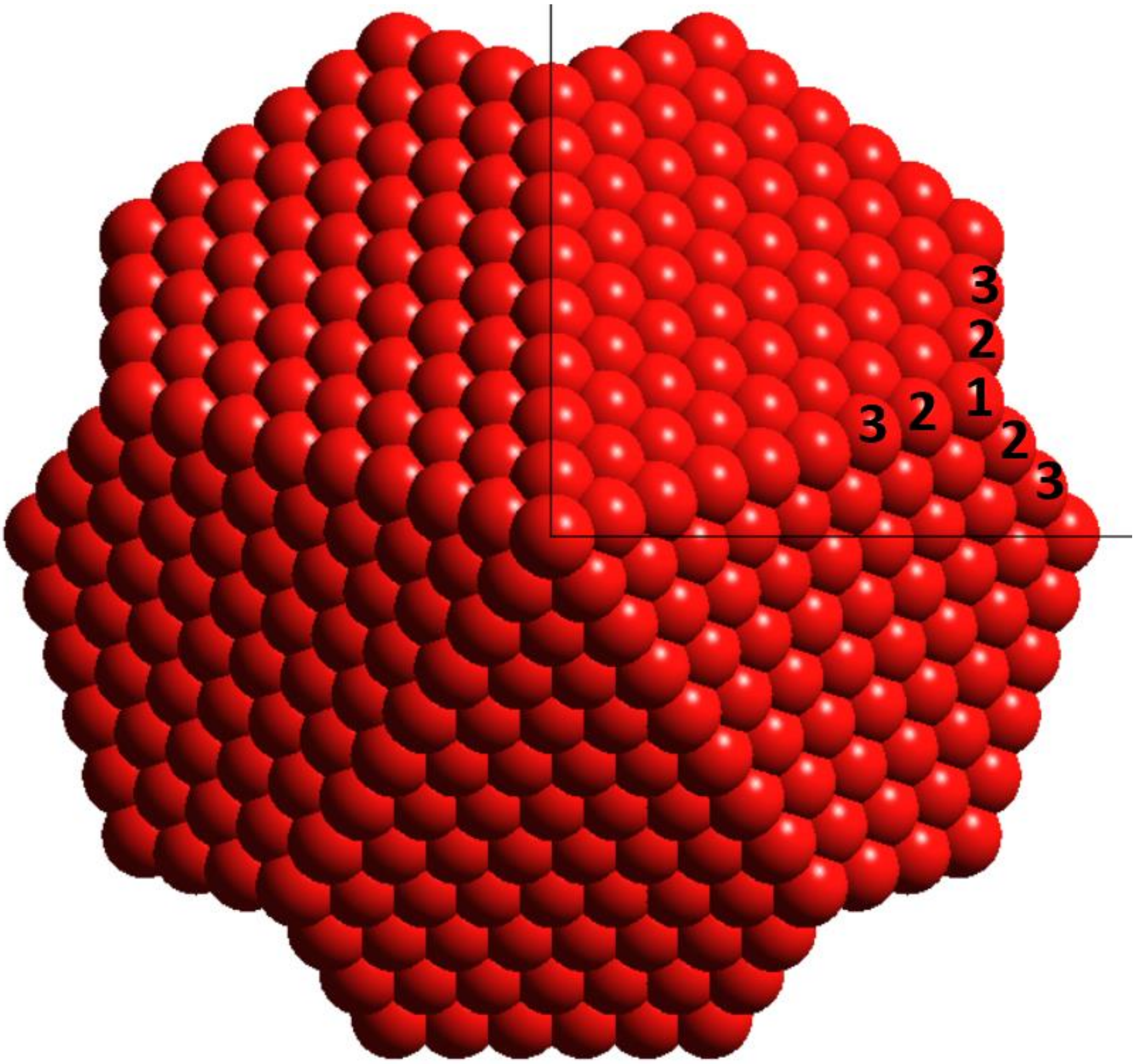


Figure 24: Decahedron NP using the decahedron *(6,1,3,2)* command. In this example the parameter *r* is equal to 3. The picture numbers and shows the logic of the *r* parameter.

References

[1] S. J. Kim, P. Lei, K. Zhang, C. Zhou, G. W. Graham, and X. Pan "Tunable, Endotaxial Inclusion of Crystalline Pt-Based Nanoparticles Inside a High-Quality Bronze TiO2 Matrix" Chemistry of Materials 29 (5), 2016-2023, (2017) DOI: 10.1021/acs.chemmater.6b03620

[2] V. Gorshkov, A. Zavalov, and V. Privman "Shape Selection in Diffusive Growth of Colloids and Nanoparticles" Langmuir, 25, 14, 7940–7953 (2009) DOI: 10.1021/la900613p 7953

[3] P. Grammatikopoulos, C. Cassidy, V. Singh and M. Sowwan "Coalescence-induced crystallization wave in Pd nanoparticles" Scientific Reports, 4, Article number: 5779 (2014) DOI: 10.1038/srep05779

[4] J. Zhao, E. Baibuz, J. Vernieres, P. Grammatikopoulos, V. Jansson, M. Nagel, S. Steinhauer, M. Sowwan, A. Kuronen, K. Nordlund and F. Djurabekova, "Formation mechanism of iron nanocubes in magnetron sputtering inert gas condensation," *ACS Nano*, vol. 10, no. 4, p. 4684, 2016. DOI:10.1021/acsnano.6b01024

[5] G. D. Barmparis, Z. Lodziana, N. Lopez and I.N. Remediakis "Nanoparticle shapes by using Wulff constructions and first-principles calculations" Beilstein J. Nanotechnol. 2015, 6, 361–368. DOI:10.3762/bjnano.6.35

[6] Evelyn Auyeung, Ting I. N. G. Li, Andrew J. Senesi, Abrin L. Schmucker, Bridget C. Pals, Monica Olvera de la Cruz & Chad A. Mirkin, Nature 505,73–77 (2014) DOI:10.1038/nature12739

[7] H. Hofmeister, G.L. Tan, M. Dubiel, “Shape and internal structure of silver nanoparticles embedded in glass” Journal of Materials Research, 20(6), 1551-1562(2005). DOI:10.1557/JMR.2005.0197

[8] Christine Vauthier, Gilles Ponchel (Eds.) “Polymer Nanoparticles for Nanomedicines” ISBN 978-3-319-41421-8

[9] Riccardo Ferrando (Eds.) “Properties of Nanoalloys-Elsevier” ISBN: 978-0-08-100212-4