NOMAD Summer

A hands-on course on tools for novel-materials discovery

September 25-29, 2017, Berlin

Advanced Graphics:

Interactive analysis and visualization of complex datasets

Michele Compostella

Markus Rampp

MAX PLANCK COMPUTING & DATA FACILITY
RECHENZENTRUM GARCHING DER MAX-PLANCK-GESELLSCHAFT

- # **A short introduction**

VisIt is a powerful, open source software for the analysis and the visualization of scientific datasets developed by the Lawrence Livermore National Laboratory. Together with other data analysis and visualization applications (like, for example, Paraview), VisIt is one of the visualization software supported at the Max Planck Computing and Data Facility (MPCDF) and can be easily accessed on the HPC systems of the Max Planck Society using environment modules. The software can also be easily installed on local machines using the executable version and the compiling script available for any operating system at this link[1] (for more information on how to install VisIt, please check the Install notes for the downloaded version).

VisIt offers several visualization capabilities that can be operated via a user-friendly graphical interface or a command line interface. VisIt's graphical user interface (or GUI) allows users to quickly get started visualizing their data and automatically creates time-based animations from datasets that contain multiple time steps. The GUI's controls allow users to pan, zoom, and rotate objects interactively using the mouse. It also gives users the ability to interactively size and position geometric objects such as planes and spheres.

On the other hand, VisIt's command line interface (or CLI) is a powerful, Python-based, alternative interface with respect to VisIt's GUI. It includes a large set of instructions based on Python 2.7 that allows the user to set the type of plot, its options and to produce images using single instructions or complex scripts.

The main objective of this guide is to show how the software VisIt can be used for the visualization and the analysis of data available in the NOMAD project, providing some basic steps into the use of VisIt's GUI for the rendering of complex interactions between materials. For this purpose, we
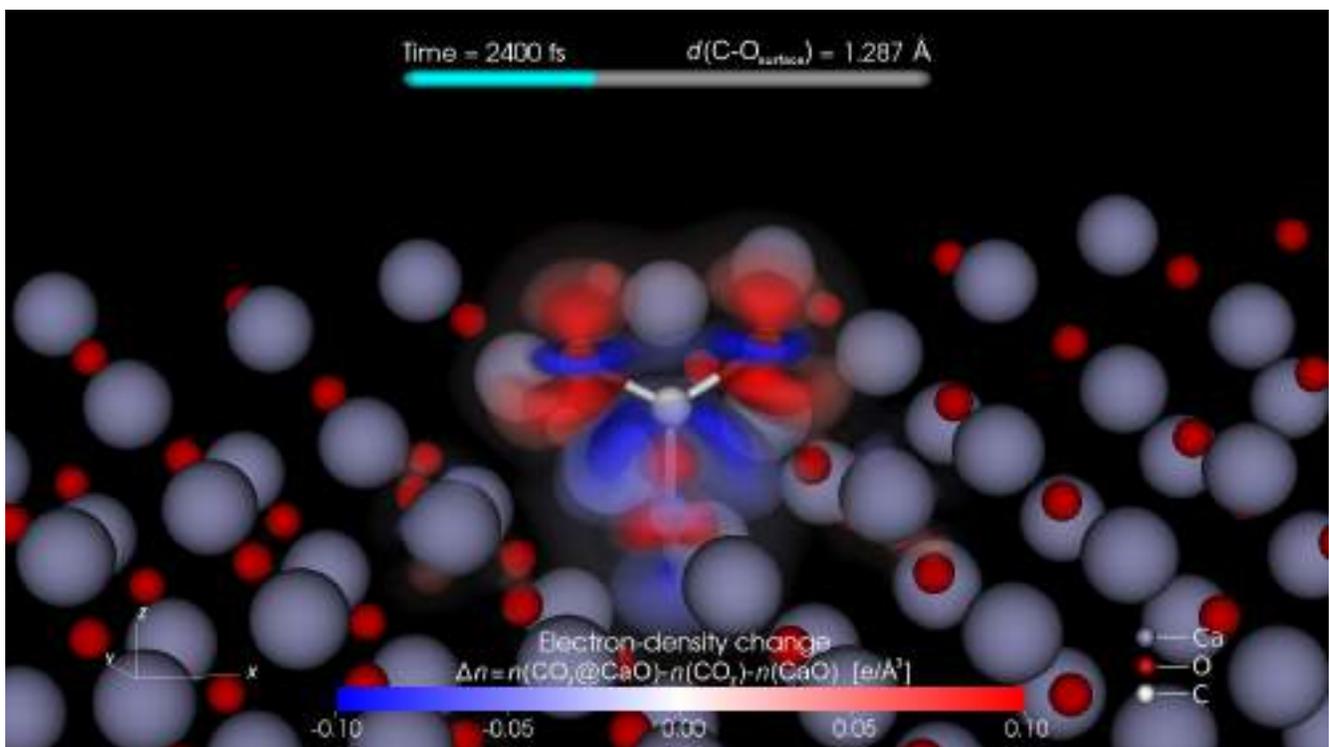

Fig. 1: Adsorption of a carbon dioxide molecule off a calcium oxide surface

1. https://wci.llnl.gov/simulation/computer-codes/visit/executables

will use the dataset generated at the Fritz-Haber-Institut in Berlin by A. Mazheika, S. V. Levchenko & M. Scheffler using the FHI-aims (Fritz Haber Institute ab initio molecular simulations). This dataset contains a time series that follows the adsorption of a CO_2 molecule onto a CaO surface. A typical image produced from this dataset is presented in Fig. 1. Atoms are represented using spheres of different colors, while changes in the electron-density due to the interaction between the CO_2 molecule and the CaO surface are depicted using red and blue regions.

- ## **Basic operations with VisIt's GUI**

  - ### *Open a database*
    Once VisIt has been launched on your machine, you can open a file by selecting the "Open" button or the "Open File" entry in the "File" menu (see Fig. 2). In the "File open" window you can browse to the */data/* directory and select the *#.total_density.cube* files. VisIt automatically tries to group files with similar name in order to open a single database containing several steps.
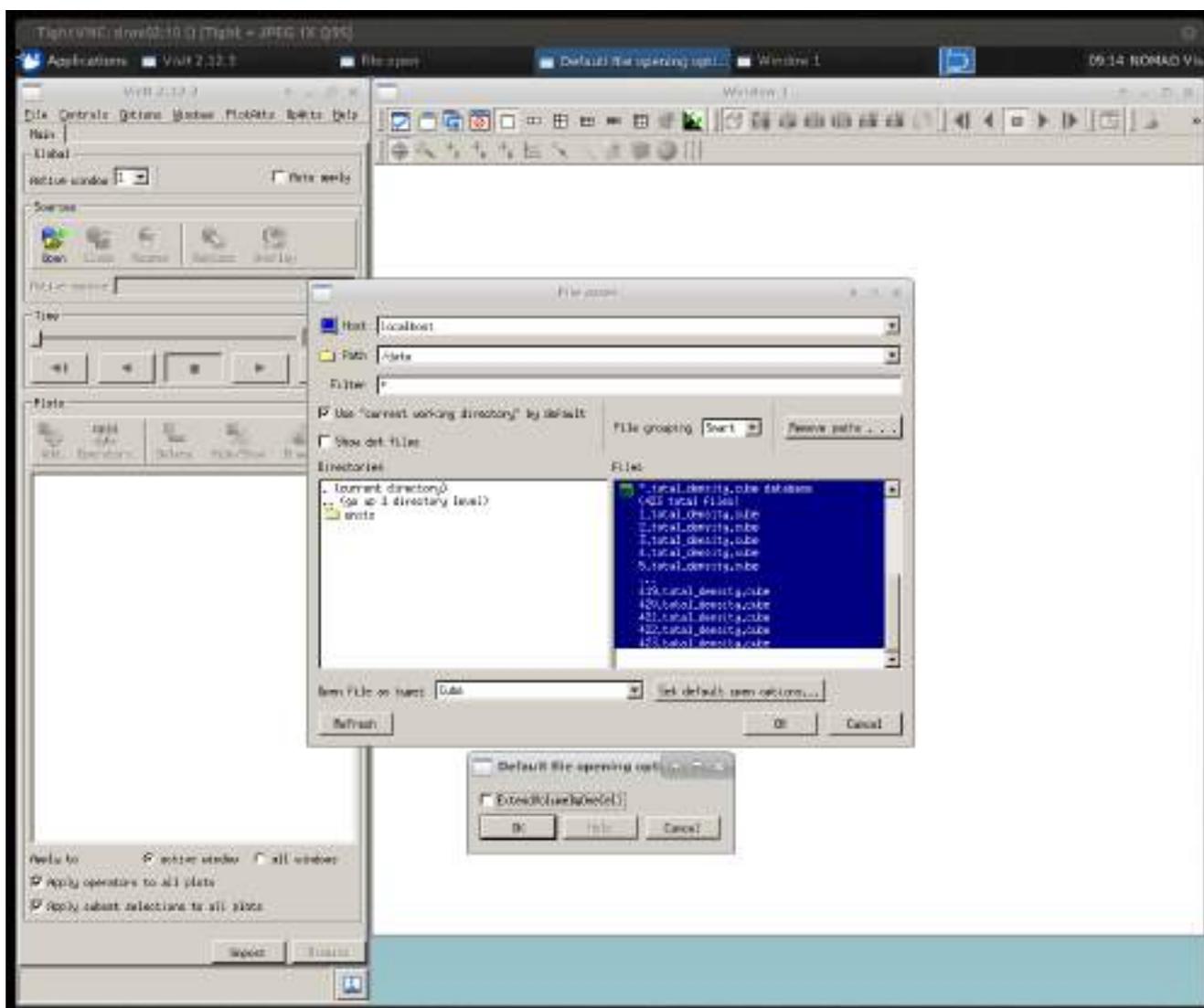

Fig. 2: Open a database

Additional options available for the specific database reader can be found selecting a database type (in our case the *Cube* plugin) and pressing "Set default open options...". Before confirming with the "OK" button, we unmark the "ExtendVolumeByOneCell" option. This will prevent a shifting in the data that

will affect our visualization.

- ○ ***Add a plot***

Once the database has been opened, it is possible to add plots and start visualizing the content of the dataset. Select the "Add" button and descend to the "Molecule" plot. This will open a list of variables contained in the open database that can be used for the specific plot. We are going to plot the variable *element,* containing the positions of the atoms involved in the adsorption reaction (see Fig. 3). The *electron_density* variable is used later in this guide.
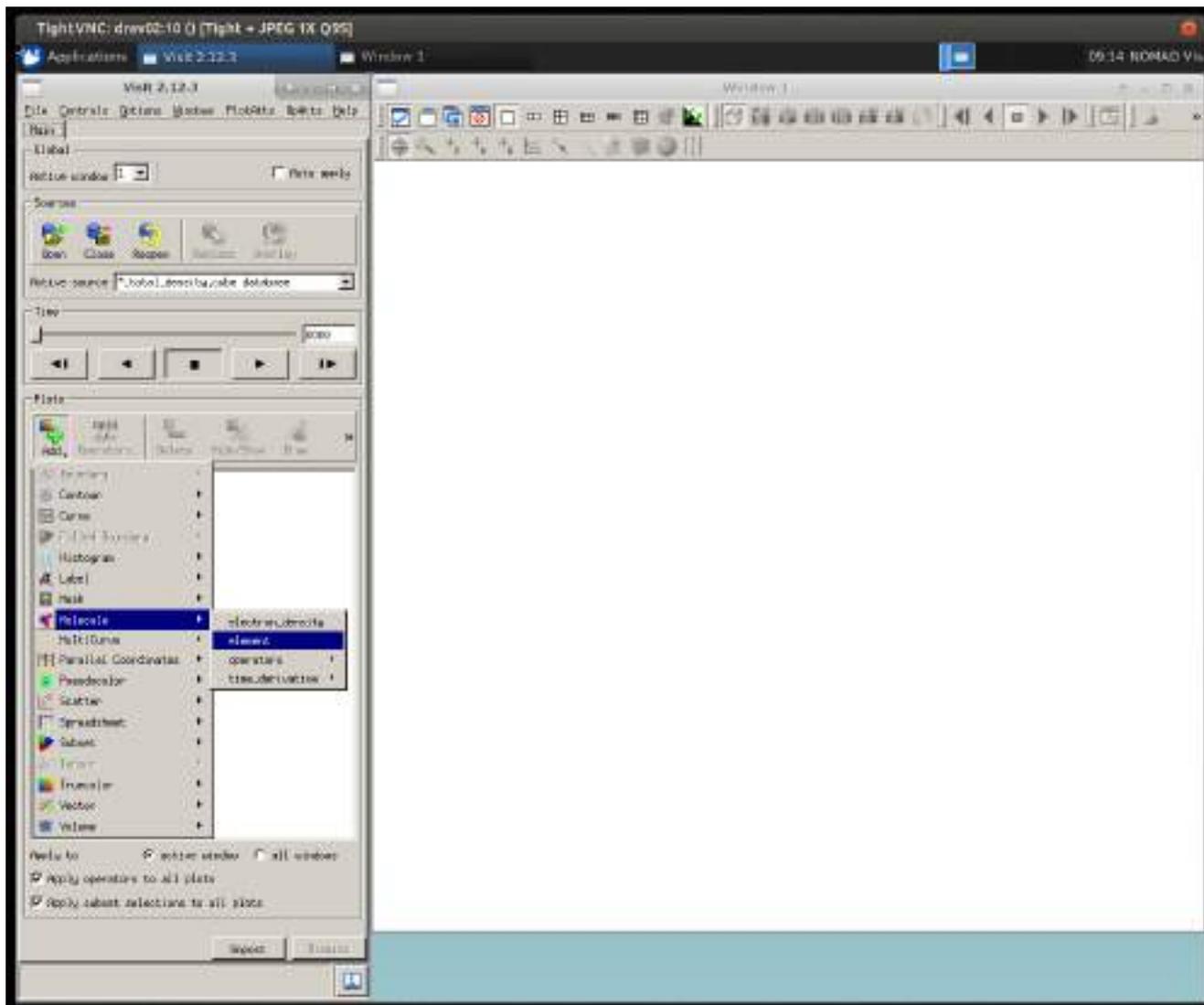

Fig. 3: Add a Molecule plot of the variable "element"

As soon as you select the variable, the entry *Molecule – element* appears in the plot list of the control window. In order to actually visualize the result of the plot, select the "Draw" button. The result of this operation is shown in Fig. 4.

By default, the first time step of the database is shown, but you can visualize subsequent time steps changing the position of the bar in the "Time" section of the control window. It is also possible to create a simple animation of the results by selecting the "Play" button.

This basic procedure can be repeated several times in order to add more and more entries to the plot list, thus creating a complex arrangement of the different plots.
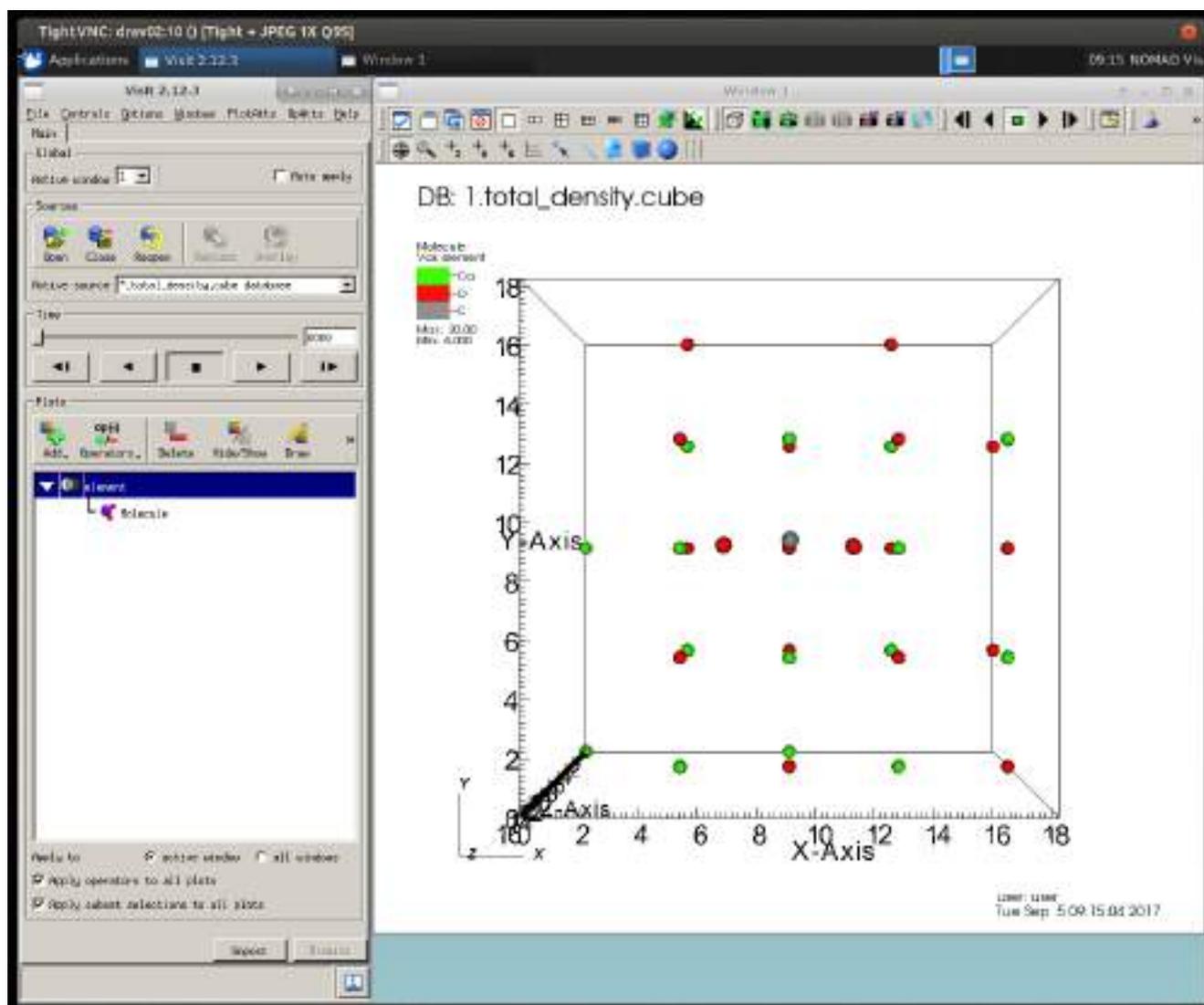
Fig. 4: Draw the Molecule plot

○ ***Change the properties of the Molecule plot***

Each plot in VisIt is characterized by some default options. In order to modify the properties of a plot, double click on the plot name and edit the attribute window that appears (see Fig. 5).
In the case of the Molecule plot, we can change the following properties in the "Atoms" tab:

```
Atom sphere quality        Super
Radius based on            Covalent radius
Atom radius scale factor   0.6
```

Switch to the "Bonds" tab and set the "Bond cylinder quality" to "Super". Finally, select the color table "cpk_rasmol" for the "Element types" in the "Colors" tab. When we "Apply" the changes, the plotting window is updated to the new properties. At this point, we can dismiss the "Molecule plot attributes" window.

Note that it may be convenient to set the new options as default using the "Make default" button, so that, for the remainder of this VisIt's session, all Molecule plots have the same properties and we don't have to manually change each Molecule plot we add.
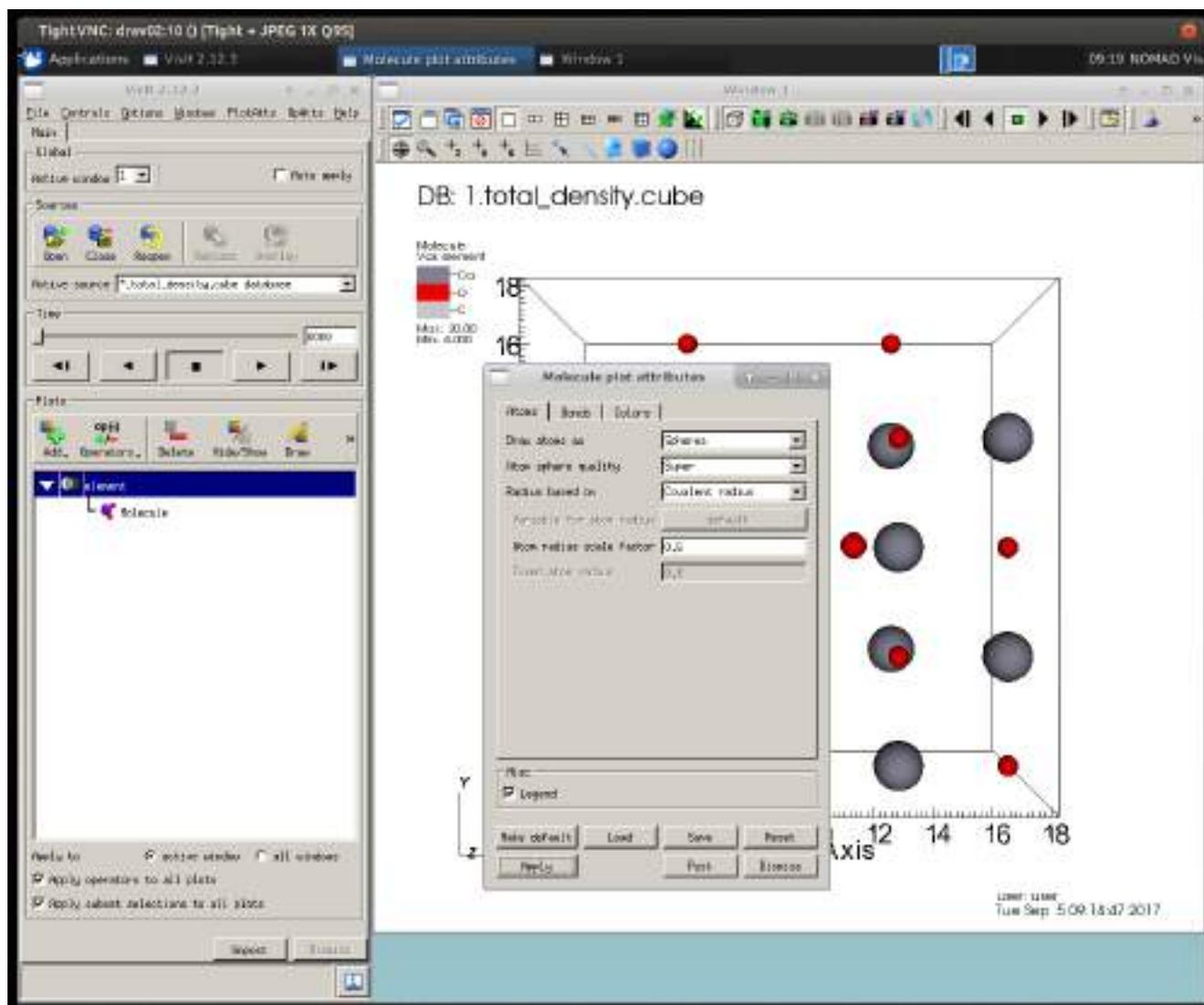
Fig. 5: Modify the properties of the Molecule plot

○ ***Add a new plot from a different database***

Similarly to what we did with the *#.total_density.cube* database, we can select the "Open" button and add the *#.lattice.cube* dataset to the list of available databases. At this point, we can add a new Molecule plot ("Add", "Molecule") using the *element* variable available in the just opened database. Note that VisIt automatically produces plots using the most recently opened database, but you can select a different input file in the "Active source" field of the control window. VisIt also recognizes that the two open databases contain the same number of steps and the same grid structure and tries to create a correlation between the two (see Fig. 6). This correlation binds the two datasets and allows us to update all the plots at the same time when changing time step. When prompted, select "Yes" in the "Correlate databases" window.

The newly added plot is not directly visible, since the spheres representing the atoms of the CaO surface hide the atoms from the lattice database. In order to temporarily hide a plot without completely removing it from the plot list, we can select the **.total_density.cube database:element* entry in the plot list and hit the "Hide/Show" button. We can now double click on the Molecule plot of the lattice database (**.lattice.cube database:element*) and adapt the properties to the one we used for the previous plot. Note that this step is not necessary if we set the previous attribute of the Molecule plot as default.
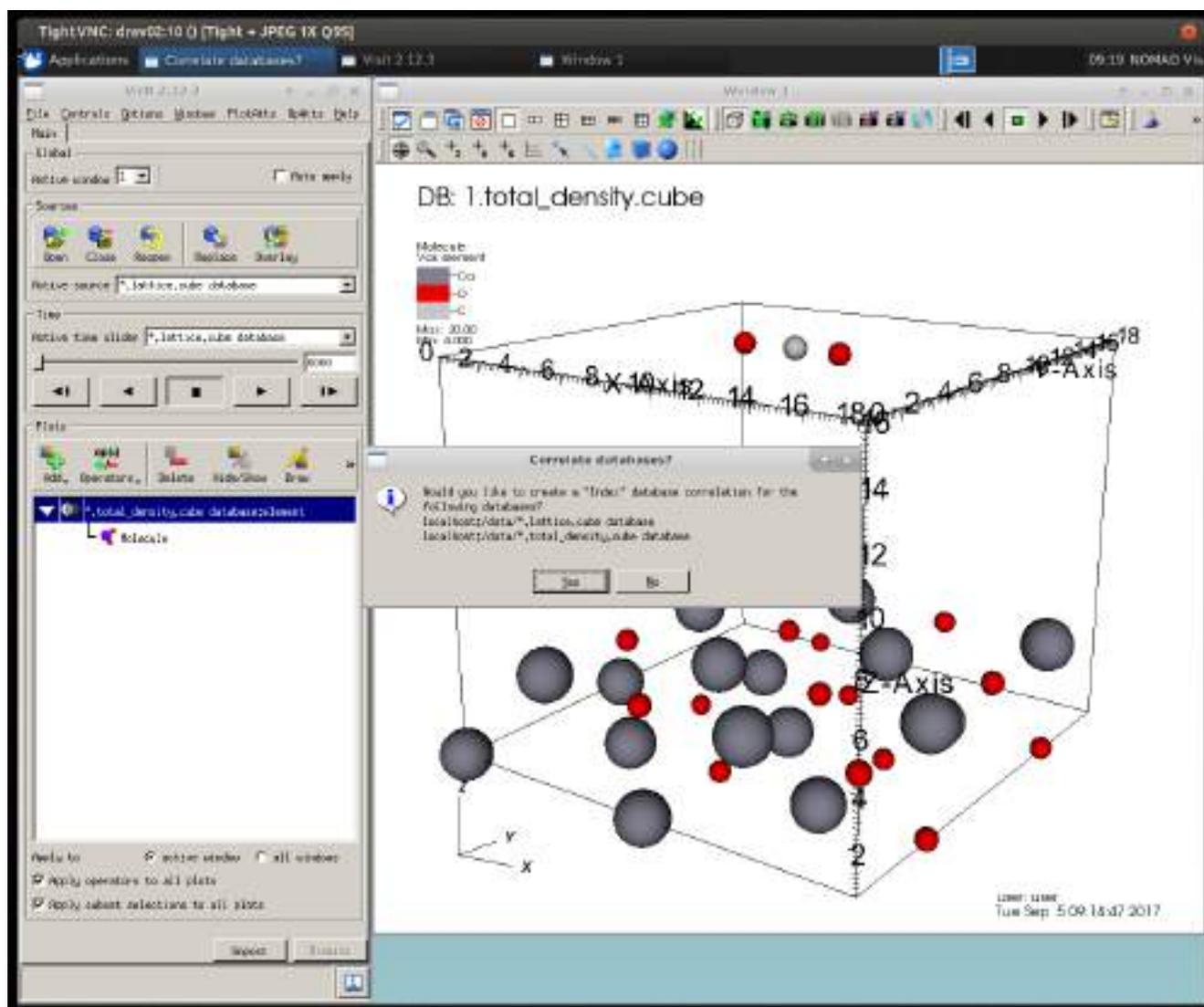
Fig. 6: Create a database correlation

○ ***Add an operator***

      While the "Add" button allows to create new plots and add them to the plot list, the "Operators" button allows to modify existing plots. Before applying any operator, it is convenient to de-select the "Apply operators to all plots" option at the bottom of the control window. This ensures that any new operator we add will affect only the selected plot.

      We can now shift the position of the atoms of the CaO lattice and replicate them in order to create a larger surface for the adsorption to take place. Therefore, we select the **.lattice.cube database:element* Molecule plot, and apply a "Transform" operator from the "Transforms" tab with the following properties (see also Fig. 7):

```
Translate                       ✓
X                               0.1
Y                               0.1
Z                               0
```

As soon as an operator is added to an existing plot, the plot's color in the plot list becomes green and it

disappears from the visualization window. In order for the changes to take place and be displayed, we have to select the "Draw" button again.
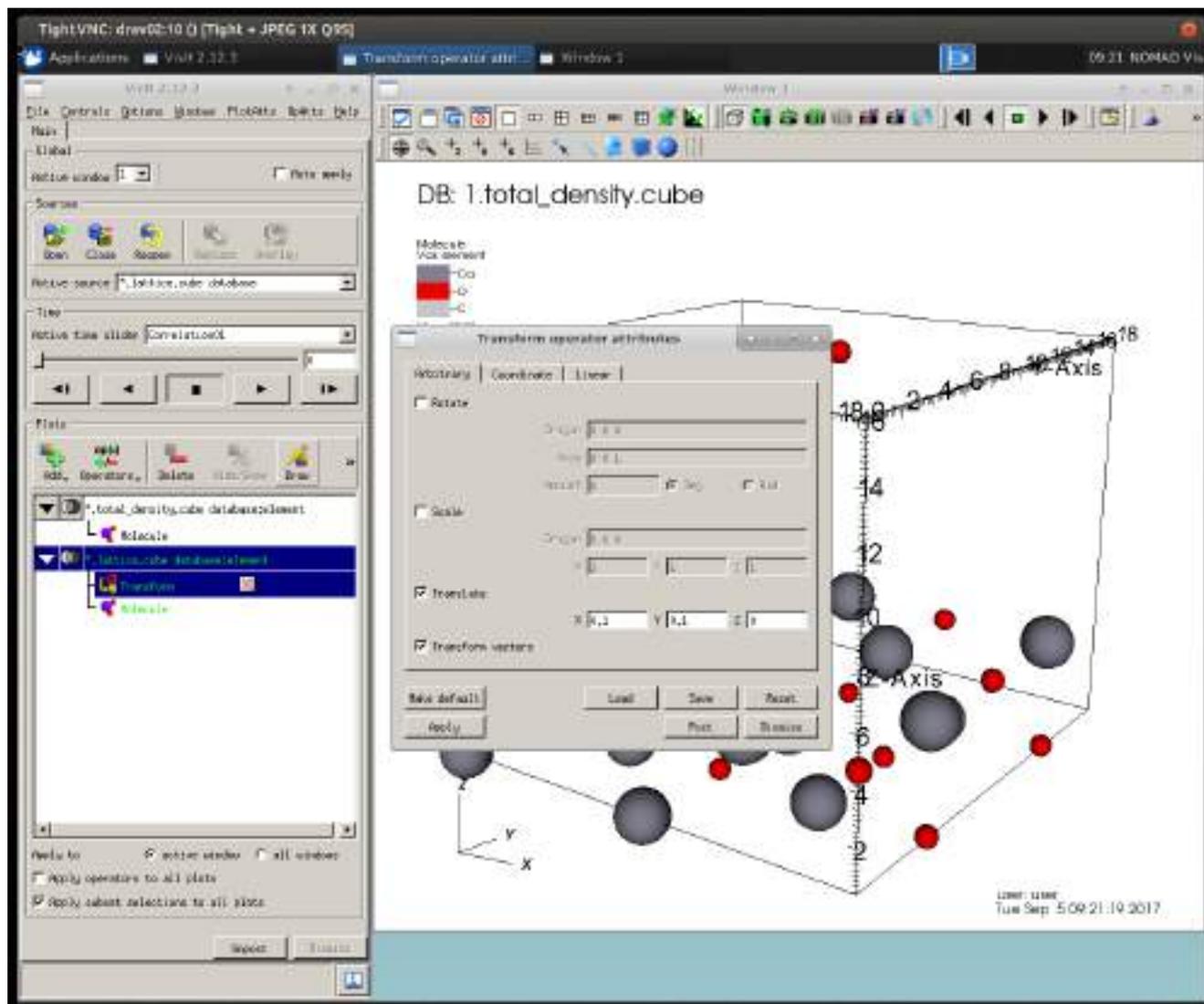

Fig. 7: Transform the atoms of the CaO lattice

If you look closely, you should see that now the atoms appear deformed, since we are shifting only the CaO atoms of the lattice file, but not the ones in the *total_density* input (this separation can be clearly seen if we use a larger value for the translation). We will take care of this later when we shift the position of the *total_density* Molecule plot.

In order to create the effect of a surface, we can replicate the atoms of the lattice using another operator available in the "Transforms" group: the "Replicate" operator. After adding this operator, we can change its properties as follows (see Fig. 8) and select "Draw":

```
Use provided unit cell vectors      ✓
Replications in X                   3
Replications in Y                   3
Replications in Z                   1
For molecular data, periodically    ✓
replicate atoms at unit cell
boundaries.
```
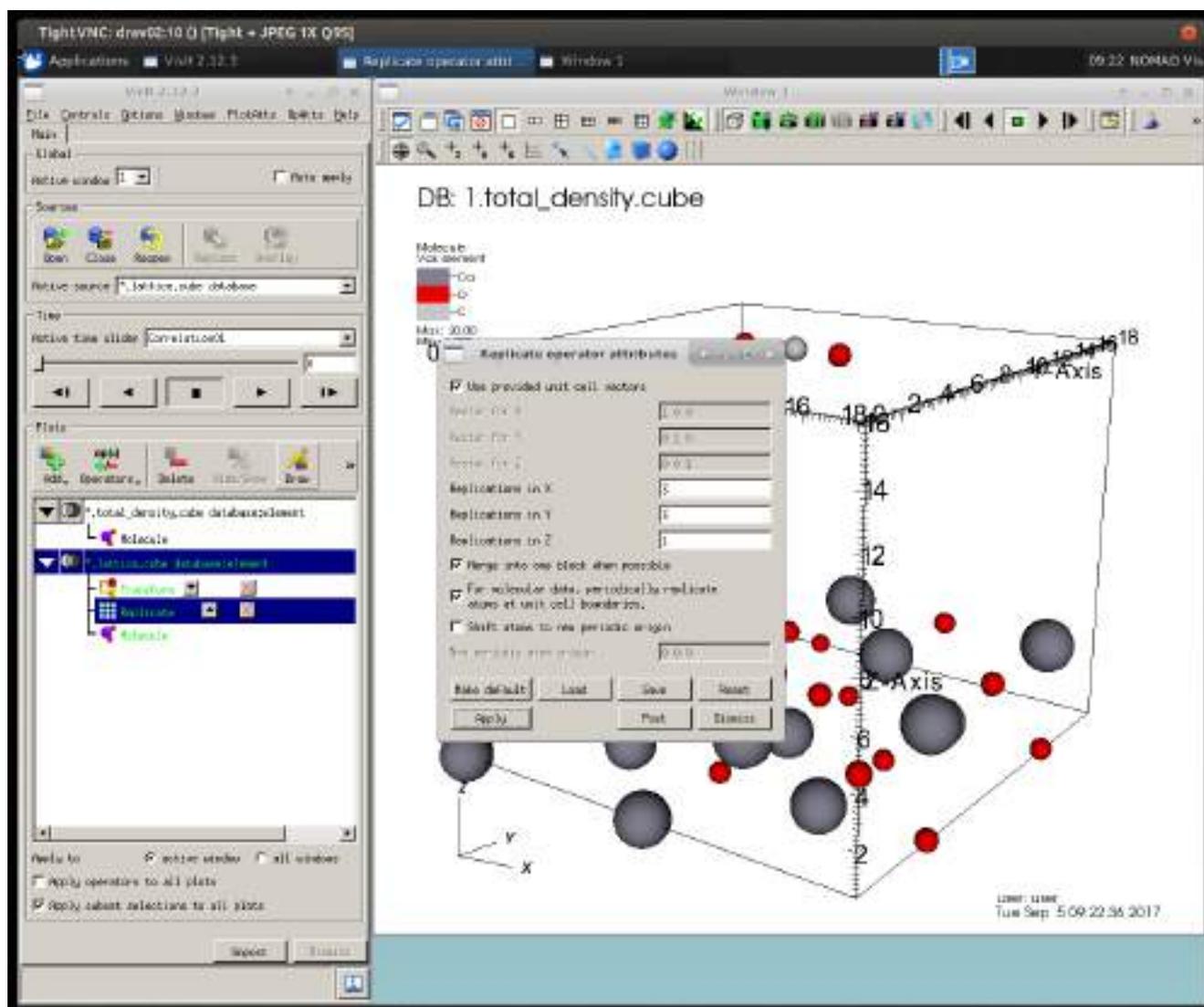
Fig. 8: Replicate the atoms of the lattice

We now have what resembles a surface of atoms with a single CO_2 molecule over it. It is time to test what the time series will look like by pressing the "play" button in the control window: VisIt will then show the list of plots we created for each time step available in the database. We can return to a static view by pressing the "stop" button.

The presence of periodic boundary conditions in the volume generates a quick motion of the atoms close to the edge of the computing volume, that move from one side to the opposite side of the grid in a single time step. We can prevent this effect by removing the atoms close to the volume edges with the "Selection" operator "Box". After adding the operator, we can modify its properties (see Fig. 9) as:

```
X-Minimum                        0.5
X-Maximum                        53.5
Y-Minimum                        0.5
Y-Maximum                        53.5
Z-Minimum                        -5
Z-Maximum                        20
```

This will restrict the visualization to the atoms inside the box limits and will remove the quick motion for most of the atoms at the edges of the volume. Atoms associated with the *total_density* database will still show this effect.
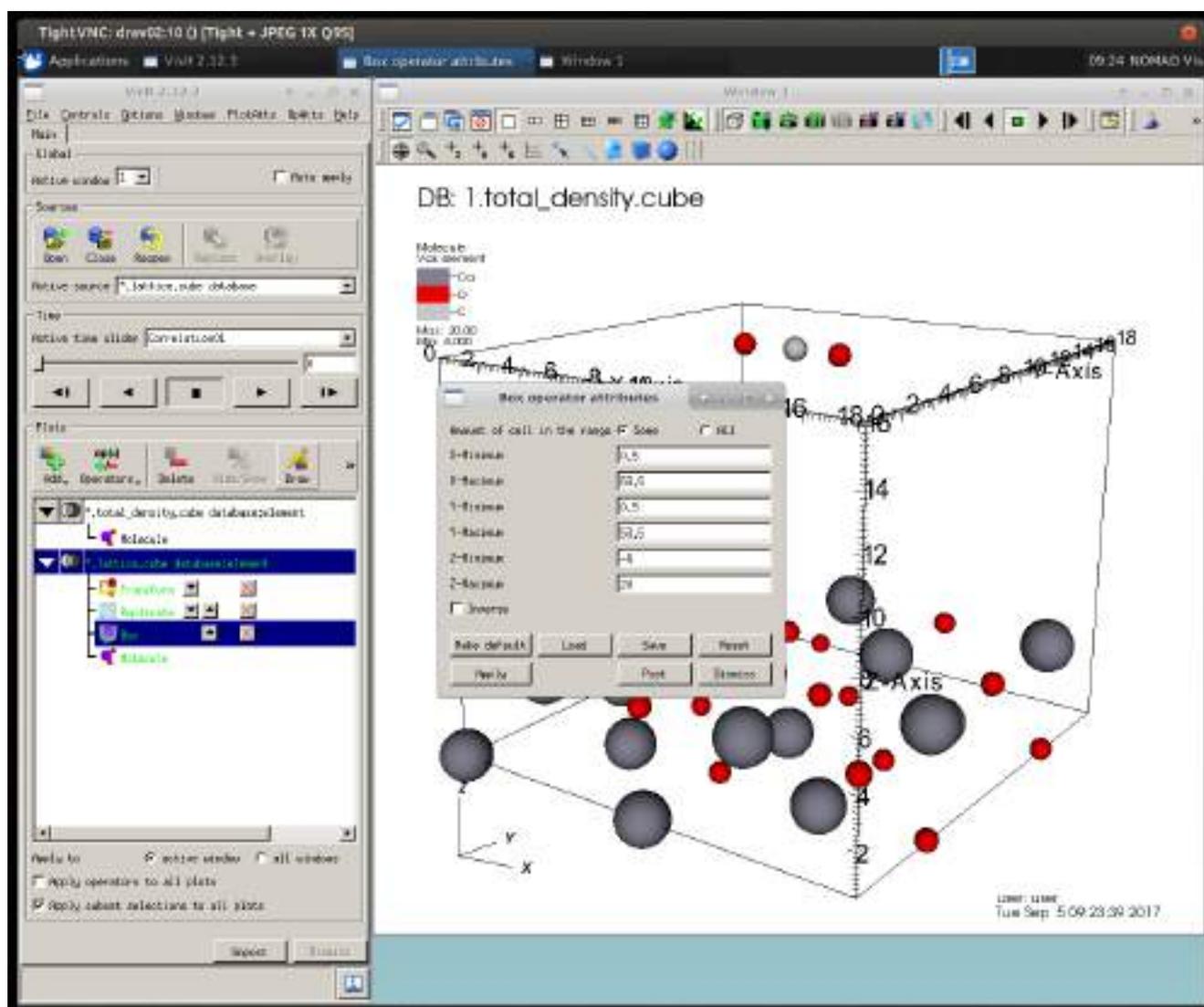
Fig. 9: Box operator

Note that the order in which we add the different operators influences the final output: if we first select some data and then replicate them, only the data included in the selection will be replicated. On the other hand if we first replicate and then select the data, the entire dataset is replicated and the selection is applied only to the final result.

It is now time to arrange the CO_2 molecule at the center of the CaO surface and we can do this by applying a "Transform" operator to the *total_density* Molecule plot with the following properties:

```
Translate              ✓
X                      18
Y                      18
Z                      0
```

Finally, we add the molecular bonds to the same plot using a "Molecular", "CreateBonds" operator. First, remove the default entries and then create the following new one between Carbon and Oxygen:

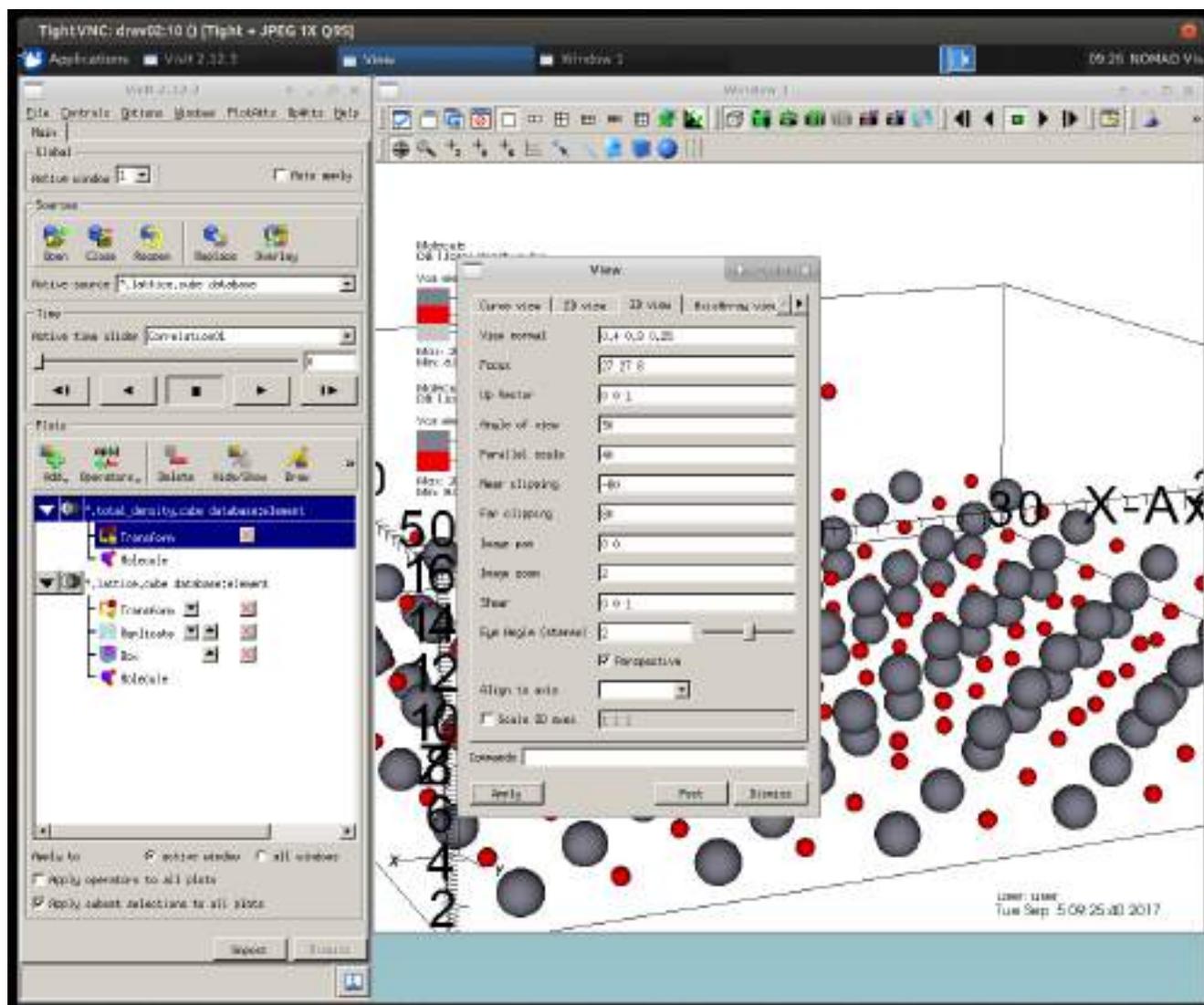| 1st	2nd	Min	Max
C	O	0	3

Fig. 10: Modify the view

○ ***Change the view***

The easiest way to change the camera view in VisIt's GUI is to use the mouse. By dragging the content of the visualization window, it is possible to rotate the view and to zoom in and zoom out using the scroll wheel. Although this method is quite useful when exploring the data of a new dataset, it is not the best approach if we want to produce a set of images with the same view. We can therefore use the "Controls" menu and select the "View" entry. This will open the window shown in Fig. 10, where we can set the following values in the "3D view" tab:

```
View normal              0.4    0.9    0.25
Focus                    27     27     8
Up Vector                0      0      1
Angle of view            30
Parallel scale           40
Near clipping            -80
Far clipping             80
Zoom                     2
```

○ ***Add a subset plot***

We now add a new plot using the *lattice.cube* database that shows the boundaries of the different replicas we produced, but first we can invert the background and foreground colors of the plot, so that the final image presents a larger contrast. This is done by selecting the button in the visualization window with black and white triangles.

Once the *lattice.cube* database has been selected, we add a "Subset" plot (see Fig. 11) of the *electron_grid* variable. We then modify its properties as follows:

- `Single`   *Select white color*
  `Opacity`   `10 %`
  `Wireframe`   ✓

This will highlight the boundaries of the computational box used for the calculation without interfering with the subject of our visualization.

We now translate the new Subset plot the same way we did for the CO_2 molecule and we replicate the result in order to stress the presence of 9 different regions in the plot.
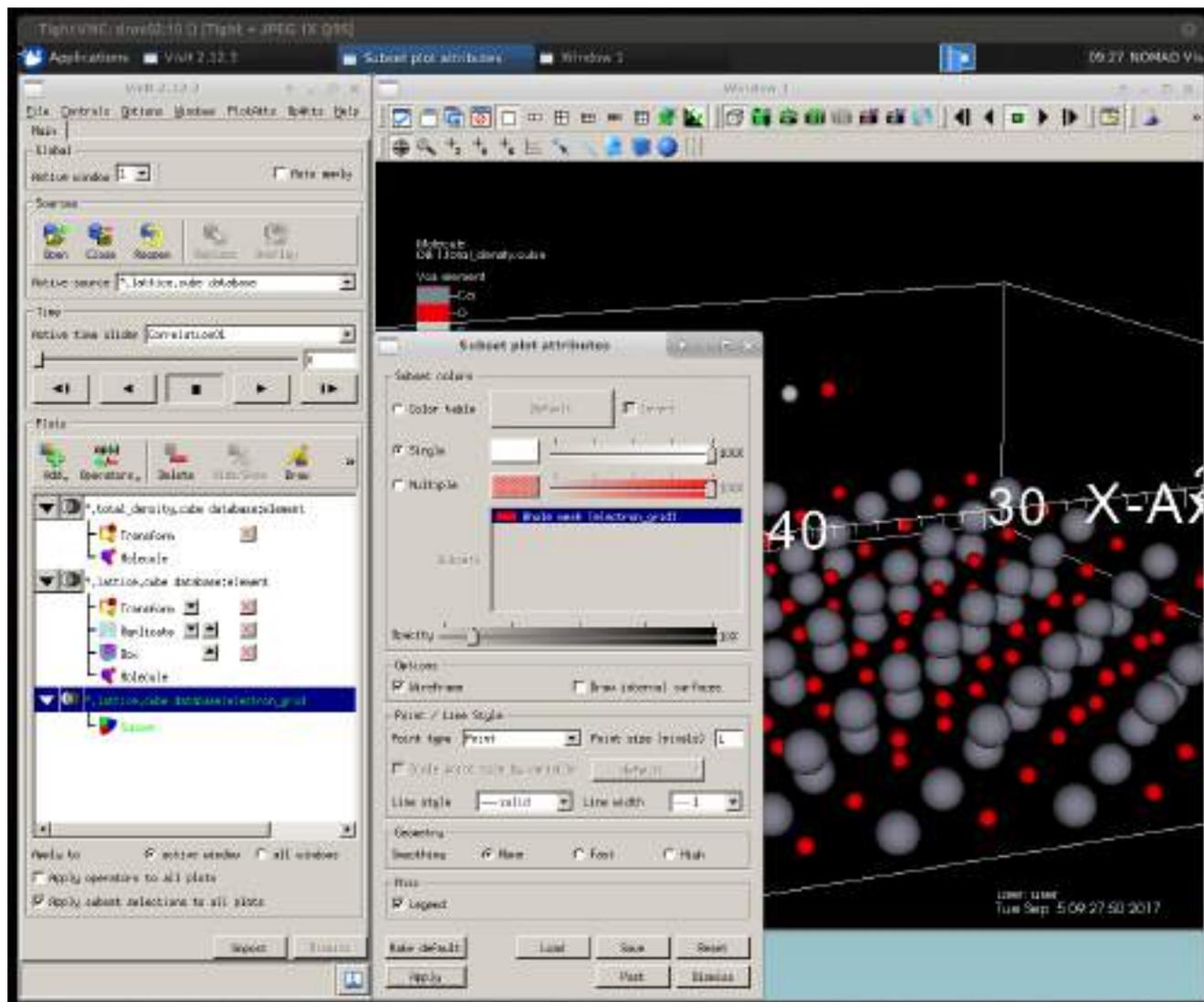

Fig. 11: Add a Subset plot

Therefore we use the following properties for the "Transform" operator

```
Translate                          ✓
X                                  18
Y                                  18
Z                                  0
```

and we replicate the Subset plot with a "Replicate" operator with the same properties used for the atoms of the lattice (3 replicas along X and Y and a single replica along Z).
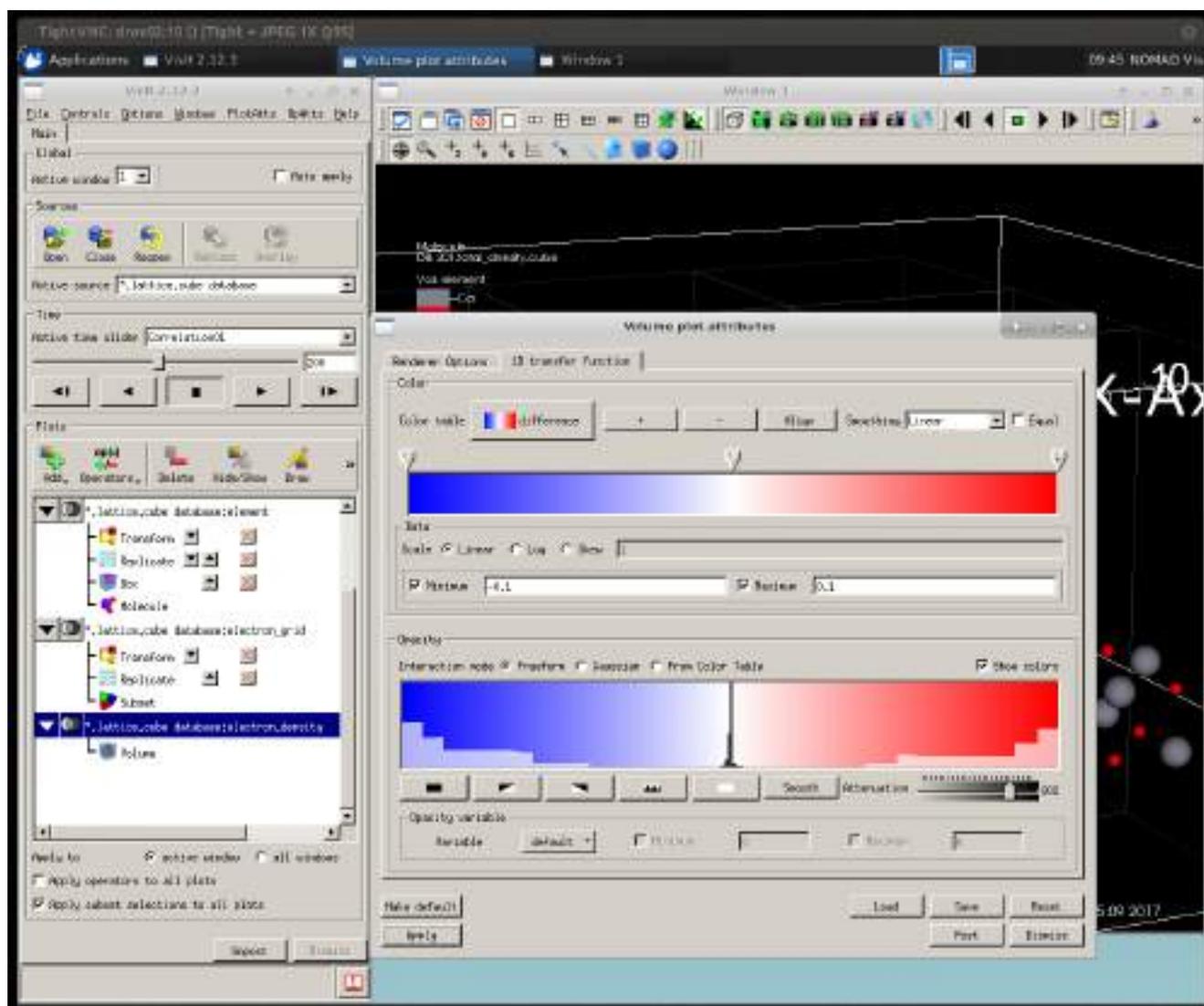

Fig. 12: Transfer function of a Volume plot

- ○ *Add Volume plot*

The main scientific result contained in the input databases is the change in electron density due to the adsorption process. We can visualize this using a "Volume" plot: select the *total_density* database, then "Add", "Volume" and the *electron_density* variable.

Opening the plot properties, we select the "Ray casting: compositing" as "Rendering Method" and we increase the number of samples per ray to 1000. Switching to the "1D transfer function" tab, we should now define the color scale and its opacity. During this step we can associate a different transparency to

each color shade. In order to reproduce the image in Fig. 1, we select the "Color table" called "difference" and we create some steps in the opacity that will highlight the regions where the absolute value of the electron change is large. First, we activate the "Minimum" and "Maximum" fields and we set their values to -0.1 and 0.1, respectively. Then, we select the black rectangular button (this will make the opacity completely transparent for each color shade) and, using the mouse, we create some opacity steps similar to the ones shown in Fig. 12. As a last step, we set the value of the "Attenuation" parameter to 80%.

Note that the black shade in the opacity window conveniently shows a histogram of the distribution of the data in the Volume plot.

Finally, we change the spatial position of the Volume plot using a "Transform" operator with the following properties:

```
Translate                          ✓
X                                  36.2
Y                                  36.2
Z                                  0
```

○ *Modify the annotation*

At this point, the drawn plots are very similar to the image of the adsorption process we are trying to reproduce, but there are several information displayed that we may want to modify. This can be done accessing the "Annotation" window in the "Controls" menu.

In the "General" tab we remove the information on the user and on the dataset by setting the following properties:

```
Database                           remove ✓
User information                   remove ✓
```

In the "3D" tab, we can remove the axes around the simulated volume

```
Show axes                          remove ✓
Show bounding box                  remove ✓
```

but we keep the triad, so that we can still have some information on the spatial orientation of the data. In this tab it is also possible, if needed, to modify the properties of the axes, to give each axis a user-defined name and to select the range and properties of the ticks on the axes.

We now want to add some labels and refine the objects shown in the visualization window. Therefore, we select the "Objects" tab and we add a "Time slider" (we can leave the default name for all the objects we are going to add) with the following properties:

```
Lower left                         0.3   0.87
Text label                         single space
```

Then we add three text labels on top of the plots we produced: select "Text" (default name) and change its properties to

```
Lower left                         0.3   0.93
Height                             2 %
Text                               Time = 0.5 fs
```
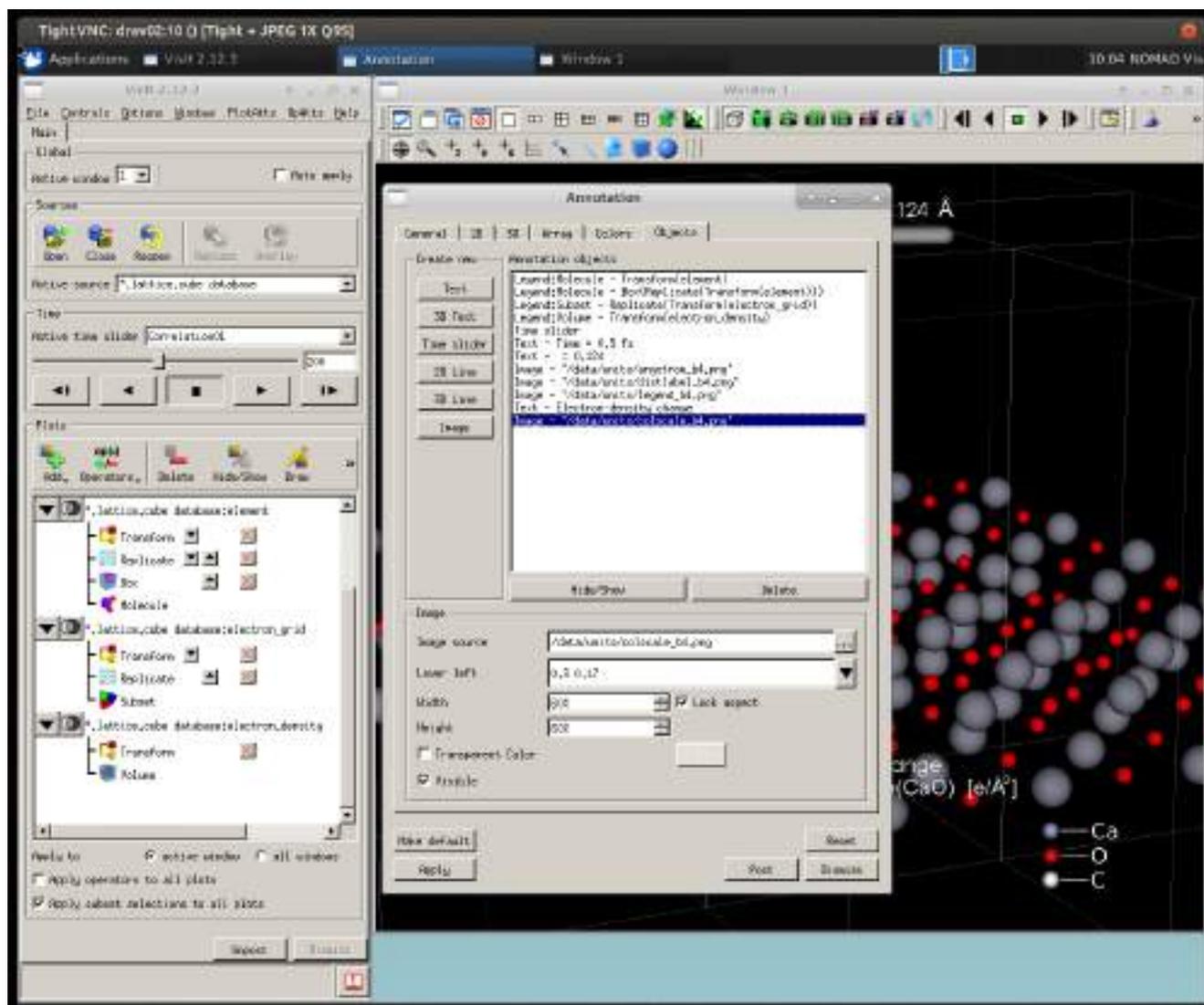
Fig. 13: List of objects in the annotation window

We add a second "Text" object with the properties:

```
Lower left                          0.58    0.93
Height                              2 %
Text                               = 0.124
```

and a final one characterized by

```
Lower left                          0.37    0.2
Height                              2 %
Text                               Electron-density change
```

At this point, we add some images prepared in advance that will improve the look of our visualization and overcome some of VisIt's limitations. As a matter of fact, VisIt can only produce simple labels of text that do not contain any special symbol or character, like, for example, the Ångström symbol. We work around this problem using some images in the */data/units/* folder. Select "Image" (default name) and modify the properties as follows:

```
Image source                      /data/units/angstrom_b4.png
Lower left                        0.68   0.93
Width/Height (✓ Lock aspect)      50 %
```

We now add and modify the properties of the image containing the label of the distance between the CO_2 molecule and the CaO surface:

```
Image source                      /data/units/distlabel_b4.png
Lower left                        0.46   0.925
Width/Height (✓ Lock aspect)      50 %
```

We also add an improved version of the legend for the Molecule plots:

```
Image source                      /data/units/legend_b4.png
Lower left                        0.8   0.05
Width/Height (✓ Lock aspect)      60 %
```

and finally we add the image containing the equation used to calculate the electron-density change in the Volume plot:

```
Image source                      /data/units/colscale_b4.png
Lower left                        0.27   0.17
Width/Height (✓ Lock aspect)      60 %
```

Fig. 13 shows how the list of objects should look like once we add all the images and text labels.

Note that the values reported above for the position and height of the different objects we added are based on the standard size of the visualization window used by VisIt. If you changed the size of the window or you are planning to save the image with a resolution different from the current one, the size and position of the objects may require some changes.

○ *Create a new legend for the plot*
The default legend for a Volume plot shows the selected transfer function, but there are cases where we want to show only the color scale used, without overwhelming the viewer with too much information. We can create a legend without transfer function using an additional plot. Select the *total_density* database, "Add" and "Pseudocolor" plot, specifying the *electron_density* variable. Then, change the plot properties as shown in Fig. 14, reproducing the same settings used for the Volume plot, but setting the opacity to 0:

```
✓  Minimum                        -0.1
✓  Maximum                        0.1
Color table                       difference
Opacity                           Constant
Opacity value                     0 %
```

This will return a completely transparent Pseudocolor plot, where only the legend is visible.

At this point we open the properties of each of our previous plots (not the Pseudocolor one) and one by one we hide the legend by removing the tick from the corresponding field. The only legend visible at the end of this process will be the one associated with our transparent Pseudocolor plot.
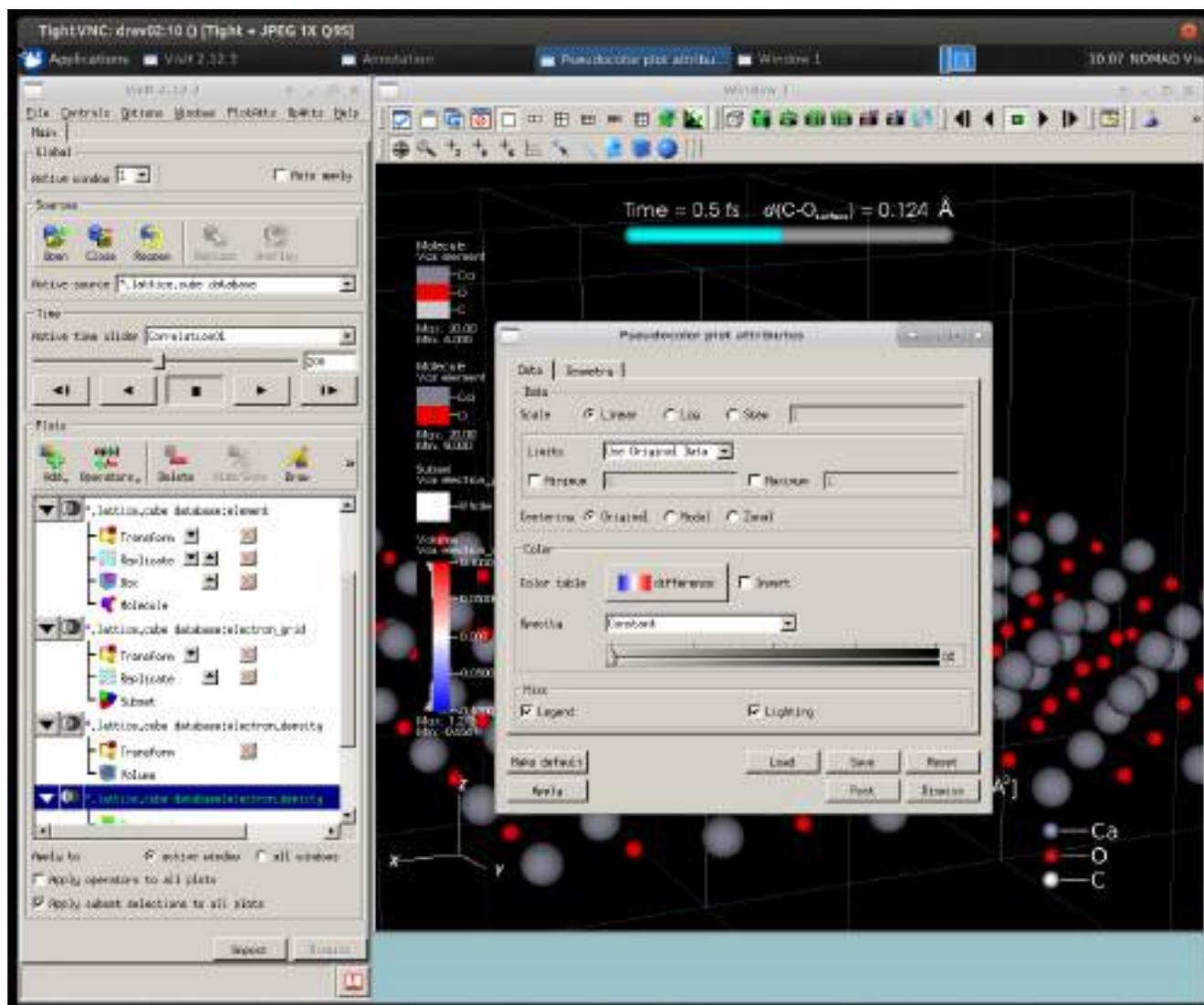
Fig. 14: Add the legend from a transparent Pseudocolor plot

Finally, we return to the annotation window ("Controls", "Annotation") and we open the "Objects" tab. Here, we select the *Legend: Pseudocolor – electron_density* entry and we modify its properties as follows:

```
        Position tab:
Let VisIt manage legend position     remove ✓
Legend position                      0.25    0.15
X-scale                              200 %
Y-scale                              30 %
Orientation                          Horizontal, Text on Bottom


        Appearance tab:
Draw title                           remove ✓
Draw min/max                         remove ✓
Font height                          0.08
Number format                        %# -5.2f
```

If some of the modifications have altered the view in the visualization window, we can re-open the "View" panel and set its values as we did before.

This concludes the creation of the image. The final result is shown in Fig. 15.
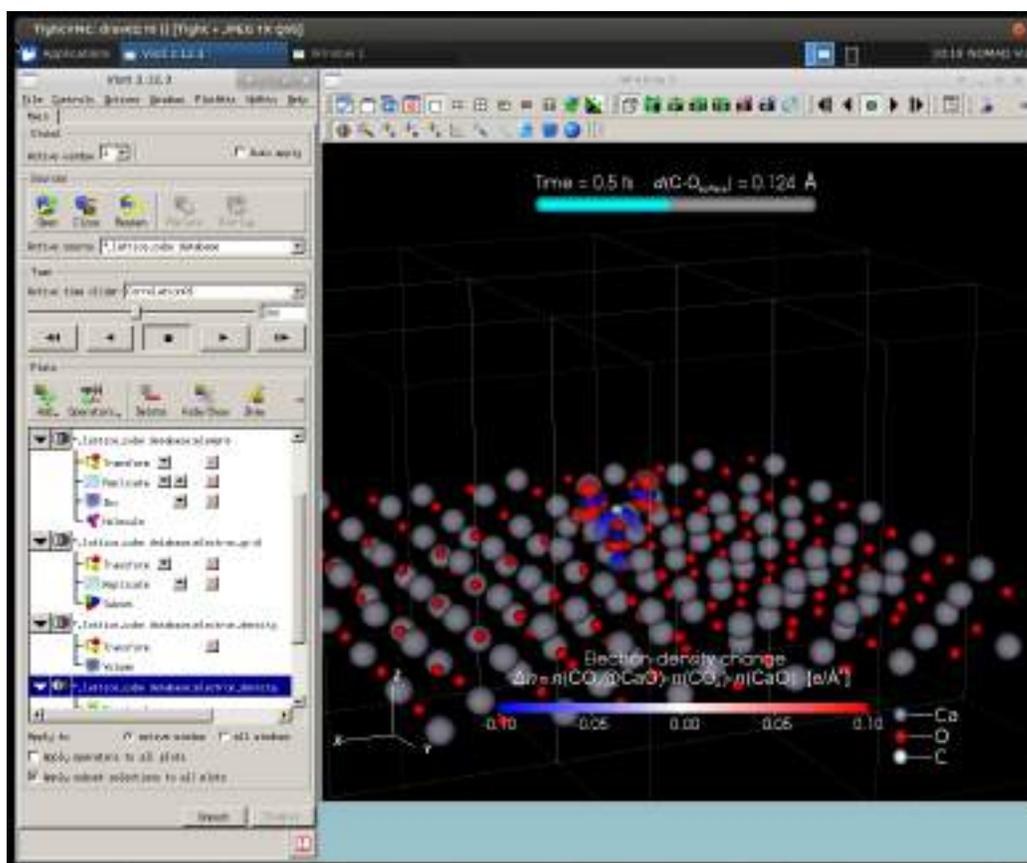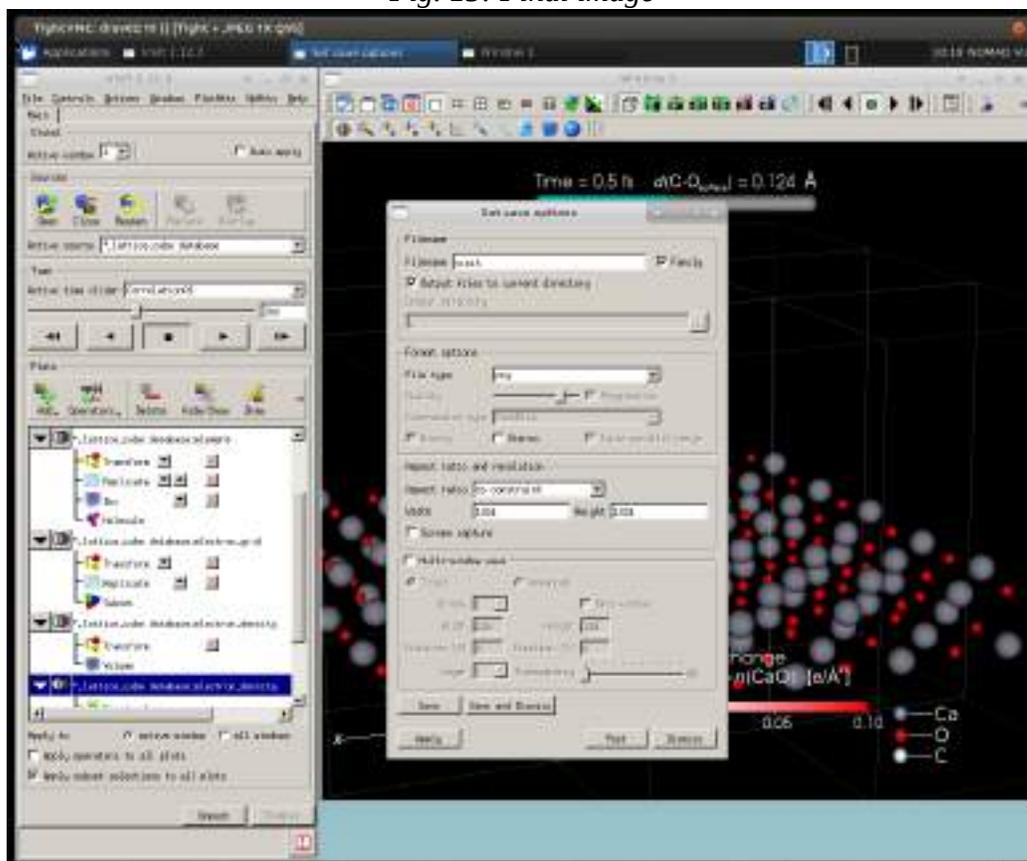
Fig. 15: Final image


Fig. 16: Save attributes

- ○ *Set Save options*

In order to save the image we just created, we can simply select "Save window" in the "File" menu. This will save the image with the default options, but we may want to specify different properties for the output. We can access these properties using the "Set Save options" entry of the "File" menu (see Fig. 16). In this panel we can select the output directory, the file name, type and resolution (remember that changing the resolution with respect to the default one may move the annotation objects in your image) and, in the case of multiple plotting windows, we can specify how the different windows should be arranged in the final output.

• **Additional operations**

- ○ *The pick tool*

VisIt's GUI offers useful tools for the interactive exploration of data. An example is the pick tool, represented by the following icons in the visualization window:
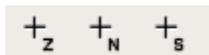
By using the pick tool it is possible to quickly determine the position of atoms in the active plot or the location (in terms of coordinates, zone and node) of specific elements in our plot. In order to obtain information on extended regions instead of single elements, the point, line, plane, box and sphere tools may also be used:

- ○ *Define new expressions*

Sometimes, when analyzing the results of numerical calculations, it is useful to create combinations of the variables available in the input database or create some functions that will restrict the visualization to a subset of the available data. In VisIt this operation is possible using the so-called "Expressions" that, in essence, create new variables available for plots and calculations.

In order to create a new variable, select the "Expression" entry in the "Controls" menu. Use the "New" button and assign a name and a type to the new variable. As an example, we can create a new variable called *AbsValue* calculated as the absolute value of the variable *electron_density*. We select "Insert Function" and, in the "Math" tab, we choose "abs". Once the cursor is between the parenthesis, we select "Insert Variable" and "electron_density". At this point we can create a plot of the newly defined variable *AbsValue* as if it was contained in the original input database.

- ○ *Queries*

Although VisIt can not be considered a complete replacement for the comprehensive quantitative analysis offered by other software (Python, GNU-R, IDL, MATLAB, etc.), it provides some basic tools for the analysis of a dataset that are called "Queries".

As an example, we can study the evolution over time of the maximum electron-density change. In order to do so, we select the Psedocolor plot of the *electron_density* variable in the *total_density* database (so that the query will be executed on this dataset) and choose "Query over time options" in the "Controls" menu. Here, we select "Timestep" as value for the "X-Axis". Then, we open the "Query" panel in the same menu and we look for the query called "Max", taking care of activating the "Do Time Query" option before pressing the "Query" button. After some calculations, a new window will show the evolution with the time step of the maximum value of the electron-density change.

More complex queries can be assembled in the "Python Query Editor" tab using the standard Python 2.7 programming language.

○ *Add new color tables*

VisIt offers the possibility to modify existing color tables or create new ones with the color scale we prefer. Select the "Color table" entry in the "Controls" menu and choose from the "Manager" list the color table that you want to use as a starting point. Assign a new name to the color table and select the "New" button. At this point we can select the color points at the bottom of the Color table window and alter their position or their color in order to create a completely new color table that we can use for our plots.

○ *Command*

The "Command" window in the "Controls" menu offers the possibility to translate the actions we perform in the GUI to instructions in the Python language (and vice versa). This option is particularly useful when preparing a Python script for VisIt's CLI, since it simplifies the process of setting the parameters of each plot (think, for example, about the opacity of the transfer function in a Volume plot). On the other hand, if used excessively, the output of the "Command" window can reduce the readability of your Python script with redundant information and it can hide small errors in many lines of code. Therefore, it should be used with caution.

As an example, open the "Commands" window and select the "Record" button. Return to the control window and open the properties of the Molecule plot of the variable *element* in the *total_density* database. At this point push the "Apply" button (even without changing anything) and press the "Stop" button in the "Commands" window. The bottom part of the window will now display all the properties of the Molecule plot in Python language, as reported below:

```
MoleculeAtts                              = MoleculeAttributes()
MoleculeAtts.drawAtomsAs                  = MoleculeAtts.SphereAtoms
MoleculeAtts.scaleRadiusBy                = MoleculeAtts.Covalent
MoleculeAtts.drawBondsAs                  = MoleculeAtts.CylinderBonds
MoleculeAtts.colorBonds                   = MoleculeAtts.ColorByAtom
MoleculeAtts.bondSingleColor              = (128, 128, 128, 255)
MoleculeAtts.radiusVariable               = "default"
MoleculeAtts.radiusScaleFactor            = 0.6
MoleculeAtts.radiusFixed                  = 0.3
MoleculeAtts.atomSphereQuality            = MoleculeAtts.Super
MoleculeAtts.bondCylinderQuality          = MoleculeAtts.Super
MoleculeAtts.bondRadius                   = 0.12
MoleculeAtts.bondLineWidth                = 0
MoleculeAtts.bondLineStyle                = MoleculeAtts.SOLID
MoleculeAtts.elementColorTable            = "cpk_rasmol"
MoleculeAtts.residueTypeColorTable        = "amino_shapely"
MoleculeAtts.residueSequenceColorTable    = "Default"
MoleculeAtts.continuousColorTable         = "Default"
MoleculeAtts.legendFlag                   = 0
MoleculeAtts.minFlag                      = 0
MoleculeAtts.scalarMin                    = 0
MoleculeAtts.maxFlag                      = 0
MoleculeAtts.scalarMax                    = 1
SetPlotOptions(MoleculeAtts)
```

Even for a relatively simple plot, the amount of instructions is rather large and it becomes even larger for more complicated plots (e.g. Pseudocolor and Volume plots).