

Visualization in the integrated SimPhoNy multiscale simulation framework

Joan Adler, Sampo Kulju, Jari Hyväluoma, Keijo Mattila, Kit Choi, Ioannis Tziakos, Didrik Pinte, G. A. Garca, Y. Makushok, V. Makushok, J. Lama, G. Romn-Prez, Adham - for Aviz - you, me, Amihai, Nathan and Mehdi?

Technion - IIT, Haifa, Israel, 32000, Sgenia, Madrid, Spain

E-mail: phr76ja@tx.technion.ac.il

Abstract. to be written

1. Introduction - summary of SimPhoNy concepts and structure

Progress in modelling, e.g. in the field of materials research, requires evermore detailed considerations of a system or phenomena under investigation. This may be accomplished by resorting to multi-scale or multi-physics modelling. The related models are complex to a degree which rarely allows analytical treatment and, thus, numerical simulations are routinely used for obtaining approximate solutions. However, an in-house implementation of multi-scale or multi-physics simulators is laborious and may compromise the cost-effectiveness of computational experiments in comparison to laboratory experiments. The situation is particularly demanding when short development cycles are critical: a new application may require implementation of new features which, in the case of complex simulators, can take more time to complete than a rapid reaction permits.

A solution is to rely on well-established, specialized simulators which alone can resolve only part of the system or phenomena at hand. Two or more of these simulators are then set up to cooperate in an effort to treat the whole system at once. Here the main challenges are

configuration and operation of the various simulators, orchestration of necessary data exchange between the simulators.

The SimPhoNy integrated framework is developed specifically to address these difficulties and, above all, to facilitate simulation of systems requiring multi-scale or multi-physics modelling. The principal design aspects of SimPhoNy include the Common Universal Data Structure (CUDS) which not only specifies the data format for dynamic variables, but also for the simulation and configuration parameters. This design feature is complemented by the specification of simple SimPhoNy interfaces for the data movement and management of the simulators. In other words, SimPhoNy relies on the wrapper technology, i.e. each of the native simulators are encased with a software layer, a wrapper, which implements the above common interface specification and thus allows a uniform operation of the simulators as well as the orchestration of data exchange. The wrappers are written per simulator, and their main responsibilities include

translation of the simulation data in SimPhoNy format into the format supported by the native simulator, and vice versa, extraction of the necessary configuration data in SimPhoNy format and then utilization of the data in the set up and control of the simulator.

Of course the simultaneous utilization of several native simulators inflicts a burden on the related pre- and post-processing steps: there is no single tool capable of pre- and post-processing data for very disparate simulators. Or even if there is, the single tool may not be practical or may involve unacceptable compromises, e.g., in the results provided. This is particularly true for visualization tools where high-quality illustrations typically require specialized tools. This is acknowledged in SimPhoNy where a specific interface is defined for visualization tool wrappers. These wrappers are analogous to the simulation wrappers and, in particular, allow visualization of simulation data in SimPhoNy format independently of the particular simulator used for producing the data. Here we demonstrate visualization of simulation data using the SimPhoNy framework.

2. Visualization - different styles for different scales

There is no “one-size fits all“ for multiscale visualization, and this paper should help researchers find a suitable approach for their project. The codes in the different subsections are all to be integrated into the SimPhoNy framework, so we will move thru the lengthscales considering our options. Of course there are many useful packages outside SimPhoNy in fact far too many to even list so we will concentrate on “in-house“ possibilities.

At the smallest electronic scale we have the 3D electronic density visualization from AViz. This is basically binned colored “smoke“, using the “off-label“ application of the AViz dot representation. Of course any code that draws contour plots can be used for 2D slices, and so the other SimPhoNy codes ,..... could be used here.

At the atomic scale we have AViz (IIT/IWM) and n-CAD (Sgenia). Both are adapted to atomic systems with possibly more atoms, in less rigid structures than the many common and excellent tools used by chemists, which will not be discussed further here.

AViz is LINUX public-domain based with OpenGL/Mesa and QT4 underpinnings, whereas n-CAD is Microsoft windows and propriety.

When also include the micro/meso/macro scale we often need to visualize fluids. This is a well-researched field, especially in the context of CFD (Computational Fluid Mechanics), with both many codes and many approaches for both scalar and vector fluid flow. Several SimPhoNy applications fall into this category, and the n-CAD fluid code of Sgenia is discussed in the next section.

At the micro/meso scale there are also visualization needs for solid objects which are well satisfied by many codes, but not particularly relevant for the SimPhoNy project.

In addition to the above packages, several vtk based tools are integrated into SimPhoNY, including Paraview and mayavi from Enthought. can someone fit this into above discussion please where appropriate.

3. Some specific codes with examples

4. AViz - atomistic

AViz, short for atomistic vizualization, grew out of earlier OpenGL codes in Joan Adler’s Computational Physics group at the Technion. The first version was created in the days when Vizualization was usually done in fancy centers by specialists, by Adham Hashibon and Geri Wagner [13], for our own use, with modest aims. There were (and remain) some 15 demands for the code including vizualization on every desktop at minimal cost for hardware or software, The full list can be reviewed in [14], and many are standard in many packages.

The objects of AViz can be atoms (dots or spheres with x , y and z coordinates), cylinders for quadrupoles or liquid crystals (with an angular orientation as well), or vectors (with a

direction to the orientation). The vector case was designed for modeling Ising, Heisenberg and XY models, but is also applicable for vector fields. The atoms can have an inbuilt connection between specified atoms to model polymers. In full acknowledgement that atoms actually have a wave-like nature, they are drawn as solid objects which are the primary concern of the code.

AViz has many features that aid in three dimensional intuition. Most of these are standard for visualization tools, relating to viewing angles, zoom, slicing etc., but some like analoglyphic stereo less so. Another of these is bonds which are simply connections between the primary objects, helpful to indicate perspective, but are representations of the wavefunction hybridization, rather than objects in their own right. An important distinguishing feature of AViz is the option enabling the user to draw bonds of specified distance (and/or thickness) only between specified types of atoms, and also to indicate different configurations in different colors, see Fig. ?? of a split interstitial. Two cases showing the interactive interface options are given in Fig.??, the first is an example of thickness varying with distance, and the second, Fig.??, shows the effect of changing the fovy (fixed angle of viewing in the y-direction), both from [15]. The latter case is from a paper about the formation of diamond membranes, showing a graphitized region ready for removal in yellow. These possibilities are of paramount importance when studying defects or crystal interfaces, where interest centers on the disturbed atoms rather than the translationally invariant ones. This feature was born from frustration with those chemical visualization packages which drew bonds of specific lengths at specific angles between atoms of specific types. A recent addition to AViz is an analoglyphic (red/cyan) stereo feature, [16], as shown in Fig. ?? further enhances three dimensional vision.

Figure ?? A split interstitial defect in a diamond lattice??, showing atoms colored according to their and their neighbours' coordination numbers and bonds colored according to the coordination numbers of their neighbours.

yoelfig.png

Figure ?? A diamond sample including the split interstitial defect with bond thicknesses according to their length [15] and showing the AViz panels that were used to draw this sample.

newAviz.png

Figure ?? An example of fovy (angle of viewing in the y direction) variation, with perspective at left and a straight-on view at right, from [15]. Amihai1.png and Amihai2.png

Figure ?? An example of a nanotube in a red-cyan stereo view, from [16]

peledfig.png

AViz can be used during or post production, with images snapped and later turned into animations as needed. Because it is LINUX based AViz runs equally well on large parallel machines (convenient if there is a lot of data) and desktops. It is gnu-licensed, installable from SimPhoNy Github with a homepage containing advice, examples and FAQs. Examples of AViz' use are given in SimPhoNy use cases U1 (MD modeling of a nanotube resting on pillars) and U7 (MD modeling of damaged diamond). Since AViz works from datafiles, it can be combined with any simulation code, and several examples of combining AViz with codes such as LAMMPS are given on the website [17].

4.1. AViz - electronic

One of the object options in the original AViz was the dot one that enabled a quick sample build to check object locations. This possibility has been developed into an offlabel application to visualize field density and can be used as smoke density to show electronic density with options for color binning [18] and stereo [19, 20]. In the same style as the other AViz realizations, the above papers and the websites linked therein provide clear guidelines for materials scientist to use this approach. In Fig. ?? we show a binned colored hydrogen atom wavefunction, and the analoglyphic stereo electronic density of a nanotube. An example of this electronic density visualization in conjunction with QUANTUM ESPRESSO codes is given in use cases U8 and

U13 related to DFT modelling of nanotubes. On a typical LINUX desktop box, several hundred thousand dots can be drawn with ease. Full instructions for AViz use in combination with QUANTUM ESPRESSO are given in [19].

f2.png bastien2-90.png

Figure ?? The electronic density of wavefunction of a $n=3$, $l=1$, $m=1$ hydrogen atom state using binned color [18] (left), and a stereo image of the electronic density of part of a nanotube [20](right).

4.2. *mayavi*

Mayavi2 is a general purpose, multi-platform visualisation tool for scalars, vectors and tensors data in 3D space [21]. It consists of an interactive application that can be run on Windows, Mac OS and Linux; a Python library that provides scripting capability; a plugin framework that allows Mayavi to be embeded in other applications. Mayavi is also highly customisable that project specific elements can be added to the application easily. These features make Mayavi a suitable component for the SimPhoNy integrated framework. Since it is based on the state-of-the-art Visualisation Toolkit (VTK; [23, 24]) which has an abstract representation of data organised as structures and data attributes¹, Mayavi2 is capable of visualising particles, bonds, lattices and mesh datasets compliant with the SimPhoNy API. In other words, Mayavi2 can be used for the full range of physical systems currently in the SimPhoNy project. However, the degree of sophistication and limitations vary among the types of datasets, as are described in the following subsections.

Atomistic domain In terms of atomistic visualisation, atoms would be represented as Point Data and bonds are represented as Lines or PolyLines 4.2. Data attributes such as temperature, velocity, etc can be attached to these data structures and be selected for visualisation individually. The elements (sphere/cube/arrow) used for representing the atoms and the thickness of the bonds can be modified in the Mayavi2 pipeline. In 4.2, the varying thickness of bonds is achieved by manually separating the bonds into independent datasets. If instead all the bonds are in the same datasets, the thickness of the lines can only be varied uniformly; colours can be associated with the bonds but can only be defined at the points connected by the bonds and not on the bonds itself.

Mayavi-particles.png

Figure: . Hypothetical atomic structure to demonstrate Mayavis capability of visualising atoms and bonds. Colors depict a scalar data (e.g. mass) for each atom.

Electronic domain In the electronic domain, electron density functions can be a represented as point scalar data and visualised using the volume rendering method (Fig. 4.2)

Mayavi-electronic-example.png

Figure: Electron localisation function of an H2O molecule. From Enthought Mayavi Documentation [22]

Continuum domain Similar to particles, lattice nodes are also represented using point data. The topology and geometry of the nodes, however, may be represented using different VTK datasets depending on the type of lattices. For cubic, tetragonal and orthorhombic lattices that have very high degree of symmetry and rectilinear structures, the simplest and most compact VTK dataset representation, ImageData, can be applied to minimise memory requirement and maximise data access efficiency. Other types of lattices are represented by the a more general VTK dataset called PolyData, where both topology and geometry can be specified more generally. Similarly, crystals with defects, i.e. no symmetry, will have to be represented using PolyData. Examples of visualising lattices of high and low degrees of symmetry are shown in 4.2.

¹ VTK assumes datasets that consist of structures and data attributes. The topology and geometry of the datasets are the structures, whilst scalar/vector/tensor data such as a temperature value are data attributes.

Mayavi-lattice-example.png

Figure Hexagonal lattice (left) and orthorhombic lattice (right) visualised using Mayavi2. The orthorhombic lattice is represented using a simpler and more compact VTK dataset, made possible by the lattices high degree of symmetry.

Fluid dynamics systems generally require representation of mesh datasets containing points which are connected to form edges, faces and cells. These types of data are represented using the most general VTK dataset called UnstructuredGrid. Currently, all mesh datasets are represented using UnstructuredGrid. Future plans involve using PolyData for mesh datasets that only contain points for enhanced efficiency. An example using the Kratos CFD engine to simulate pressured flow in a tube is shown in Fig. 4.2.

Mayavi-cfd-example.png

Figure: Visualisation of fluid in a tube. Colors on the tube show the pressure values and vectors (and its color) show the velocity (and speed).

4.3. Paraview

Paraview is a general tool for visualisation in 3D space. Like Mayavi2, it is also based on VTK and can be used for visualising particles, lattices and mesh in the SimPhoNy project. Examples of using Paraview to visualise lattices, mesh and particles are shown in Figs. 4.3, 4.3 and 4.3.

Paraview-lattice-example.png

Figure: Cubic lattice example visualised using Paraview

Paraview-mesh-example.png

Figure: Mesh example visualised using Paraview. Colors show the temperature defined at each point.

Paraview-particles-example.png

Figure: Particle example visualised using Paraview. Bonds are shown as white lines.

4.4. nCAD

nCAD (nanotechnology CAD) is an innovative software that connects 3D Computer-Aided Design (CAD) programs and atomic visualization fully developed by Sgenia. nCAD provides an user-friendly and common environment for pre-processing, modelling, visualization and post-processing complex systems in the nano/atomic scale for any dimensionality with a powerful graphical user interface (GUI). nCAD is able to work with assemblies of pieces designed in traditional CAD software, both importing the geometry from any CAD or plugging-in to Solidworks [25]. Furthermore, nCAD provides the possibility to design simple pieces in 2D and 3D that can be combined with pieces designed in CADs. Each one of the pieces, which are internally named components, is independently treated to design the ad hoc atomic structure. Afterwards the assembly is built and all the components can be commonly displayed and manipulated. Figure 1 shows an example of different components in the assembly view. nCAD is a standard alone application that has been integrated in SimPhoNy platform. This integration has been done through the development of a specific Application Programming Interface (API) implemented in python and makes available many of the nCAD features from SimPhoNy, ensuring complete compatibility with the software utilities included in SimPhoNy and allowing a multiscale approach.

Designing the atomic structure nCAD provides a complete set of user-friendly windows (Atomic Editor, see Figure 2) that guide the user in the modelling process, providing a wide range of tools for a detailed design of the atomic structure. All these windows include simultaneous and embedded visualization to provide an easy and quick check of the structure. The unit cell of each component can be chosen from a library, imported from external files (CIF [26] and CD [27] formats) or fully designed from scratch within the Atomic Editor. Symmetry operations are automatically implemented by nCAD and the user can create supercells with just one click.

Changes implemented by the user in dedicated visualizer to the unit cell (moved atoms, created bonds, included new atoms, etc) can be saved and stored in the library. The Atomic Editor also permits implementing punctual and pair defects not only in the unit cell but distributed in the material and the real shape of the component as well. Hence, occupation, interstitials, substitutionals, Frenkel, divacancies, Schottky and other defect types can be implemented on the structure with different distributions in a simple manner, just giving the doping percentage and the species involved and choosing the distribution function. nCAD imports complex shapes designed in external CAD or similar software in STL format, orienting the imported pieces in the real space directions chosen by the user. It is also possible to create the real shape of the components or subsystems using the implemented shape generator for 2D and 3D components. Besides, nCAD permits setting the primary (Miller indexes) and secondary crystallographic orientation and it tunes these orientations with the real space directions as well as with the real shape of the components. The Atomic Editor is slightly different for 3D, 2D and 1D objects, see Figure 3. This last one also implements a nanotubes constructor that can be used for any material just choosing the unit cell of a bidimensional material. The 0D Editor is reserved for unstructured matter, molecules or macromolecules, fullerenes, etc. In this case, the atoms are imported from external sources, chosen from a library (fullerenes) or introduced by the user.

Visualization Once the structure has been modelled, nCAD offers multiple possibilities to visualize and manipulate the structure. nCAD has three different visualizers: the unit cell, the material and the component visualizer. The unit cell visualizer includes all the visualization tools provided by traditional atomic visualizer [28, 29, 30] like visualization and manipulation of atoms, bonds, boundaries, replication of unit cell, possibility of change species, positions, rotations, zoom, etc. Besides, it includes the possibility of visualize vacancies, occupation and impurities, depending on the percentage defined with three different schemas of visualization. The material visualizer has been designed to visualize the bulk material and thus impurities and defects are included in their final relative positions, according to the distribution function chosen by the user. The size of the material to be represented by this visualizer (measured in units of cell unit) is also tunable and eligible by the user. The third visualizer is the component one. This visualizer shows the real shape imported or designed of a component with the final position of the atoms, taking into account the unit cell, the defects implemented, the real space orientation and the crystallographic orientation. Another visualizer similar to this last one is available for the assembly, that is, for all the components simultaneously. The operations for these representations would be, depending on the size, computationally expensive and consequently they are the only operations on demand. All the rest of features have been implemented to be instantaneously updated. Figure 4 shows an example of the different visualizers previously explained and the capability to work with several components simultaneously. Figure 4 Upper panel shows unit cell, material and component visualizers for silicon. Down panel shows three different components with three visualizers for each one. All the visualizers described above include a wide variety of visualization features and graphical tools. nCAD has three different modes of visualization, orthogonal, perspective and first person (see Figure 5), where the user can move inside the structure. The atom and bond sizes are controlled by widgets and two different colour schemas for species are available. The user can also create objects in the scene like planes, angles, centroids, axes or distances that are referred to atoms and bonds positions or to other objects, and are automatically recalculated if the position of these ones changes (see Figure 6). Graphical gizmos have been implemented to manipulate atoms and bonds, using visual rulers to provide a better control of the manipulation. Translation can be done along any direction and along Cartesian planes and rotations are also eligible to be done around any direction. Different selection modes and show/hide capabilities have been implemented. All the visualizers are connected with a dock widget that gives useful and editable information to the user depending on the visualizer and the chosen selection, like position, bonds, cell parameters,

number of replications, total atoms selected, etc. Besides, another dock widget gives detailed information about the different components of the assembly. There are three visualization environments (i.e., components, assembly and library) than can be switched at any moment. Different labelling is also available in all the visualizers.

Post-processing nCAD also includes post-processing tools that provide the capability of representing results of the simulations simultaneously with the modelled atomic structure. nCAD is able to create isosurfaces and it permits equidistance representation from eligible ranges (see Figure 7a). These isosurfaces can be projected to planes and other geometries like spheres, cylinders, cubes, etc (see Figure 7b). Moreover, nCAD builds colour maps and four different schemas have been implemented (see Figure 7c), including ranges representation or same slope approach to a value. The colour maps can also be projected on the unit cell boundaries and combined with isosurface projections (see Figure 7d). nCAD allows the user to choose the colours, show, hide and delete the planes and isosurfaces, as well as to export to PNG format.

4.5. *nFLUID*

Structure of channels nFluid nFluid is a pre-processor software with GUI to design nano and micro channels fully developed by Sgenia. nFluid provides the capability to build complex and multi-connected channels with any geometrical section (cylindrical and prismatic) and ad hoc shape in a simple and intuitive manner. The software includes an automatic solver for the geometry that calculates the resulted assembly and identifies possible errors introduced by the user. All the features are available through an API for Python and through a fully functional GUI, which also provides visualization features. nFluid is completely integrated in SimPhoNy and thus fully compatible with other SimPhoNys software.

nFluid has a powerful and user-friendly Channel Editor where the user can design the geometry of the channels. For this purpose, pre-designed individual pieces have been developed lending the user the possibility to create any set of connected and complex channels, including flow adapters, expansion chambers, tees and any type of elbow joints. The user just has to care about the general characteristics of the channel elements under construction, like the radius, the bending angle of a joint, the flux direction of the channels, etc. Due to the automatic solver, the new pieces inherit the parameters of the channel where they are located, unless the user defines them. With this minimal information, nFluid creates, connects and adapts the pieces to build the channels and creates a mesh to define geometrically the designed channels. Besides, ad-hoc pieces with complex geometry can be also designed just giving several positions that the channel must fit and follow (see Figure 8). These pieces can be designed at any moment and simply connected as another individual piece to the channels, with the same properties than the pre-designed ones. The Channels Editor allows to modify the channels at any moment, adding, inserting or deleting pieces, and it also permits visualizing the results on demand.

Meshing The geometry of the channels is defined by a triangular mesh that fits to the shape of the whole channels structure. This mesh is continuous and perfectly adjusted to the geometry. The precision of this mesh can be chosen by the user and it permits not only improving the definition of the geometry but also defining a finer mesh useful for subsequent simulation goals. nFluid can also construct the negative mesh of the channels, that is, it is able to create a bulk structure where the channels are perforated inside the bulk. Moreover, it also allows to choose if the structure, channels and negatives, are closed or open structures. nFluid exports the mesh to different formats like STL or SimPhoNy Mesh class, and creates projects that can be straightforward imported by OpenFoam [31]. nFluid also includes an utility to get a text summary of the channels, where the geometry and a sketch, as well as the main characteristics of the channels and the parameters of the pieces, are included. GUI All the features described above are integrated in a GUI that implements a clear and easy workflow. Figure 9 shows the nFluid GUI and the four main parts that the GUI is divided in are marked in red. The Channel

Editor, as it is explained above, provides to the user a simple manner to design the channel. Two windows offer the information about the channels: the first one shows the modifiable list of pieces, while the second one offers a detailed hierarchical sketch of the channel. The main window is dedicated to a visualizer that displays the channel on construction during the design process.

5. Mayavi & Paraview – visualization of structured grid data

5.1. *Mayavi & Paraview*

Mayavi and Paraview are open-source scientific data visualization softwares [21, ?] based on Visualization Toolkit (VTK) [?], an open-source visualization library. VTK supports large number of data formats, like for example structured grid, unstructured grid and polygonal data. In the following examples the data was stored as a structured grid and visualized with Mayavi and Paraview.

Structured grid data was got from the lattice Boltzmann (LB) simulation of a water droplet hitting a solid surface. LB calculation was run with code based on a single-component multiphase model [?]. LB-code generates density (scalar) and momentum (vector) data in given lattice and stores this to a binary file. Solid objects, like walls, are given density -1.0 and momentum (0, 0, 0) when printed to a data file. By Python script, this file was read and transformed to SimPhoNy CUDS, which was then feed to Mayavi via SimPhoNy plug-in.

Simulation presents a case where a water droplet (surrounded by a gas) with diameter of approximately 58 μm hits the surface at 1.6 m/s (Weber number 2).

For the simulation the $250 \times 250 \times 250$ grid was used. One iteration of

visualization consisted always following phases; 20 time steps of LB simulation, reading the data file and converting to CUDS and visualizing with Mayavi.

5.2. *Structured grid data of scalars*

Density data of the system, presented in a structured grid of scalars, was visualized by presenting two isosurfaces; one for the gas-liquid interface (value 1.5, color light blue) and one for the solid surface (value -0.9, color dark grey). Mayavi also has a possibility to render surfaces with different opacity values

from 0.0 (totally transparent) to 1.0 (solid). Value 0.8 was used for the water droplet and 1.0 for the solid surface. The drop and the surface are presented in

Fig. 1 where are snapshots from iterations 0, 100, 200, 300, 400 and 500.

All the calculations and visualizations were run on a normal desktop computer. Rendered frames were also combined to a video with a free video and multimedia software package FFmpeg [?]. The video is available online.

5.3. *Structured grid data of vectors*

Vector data can be visualized in many ways, like vector fields or stream tracing. Both Mayavi and Paraview can do these. In Fig. 2 is presented a stream tracing image from the water droplet (surrounded by a gas), which has hit the surface and is bouncing back. Visualization was done with Paraview. Geometry was visualized again by the isosurface representation and with the same values which were used in previous section. For stream tracing the magnitude of the vectors were visualized with Runge-Kutta method, integrating to both directions and with line seed diagonally along the surface.

5.4. *Conclusions*

Both softwares, Mayavi and Paraview, offer a large variety of visualizing possibilities for scalar or vector type datasets in VTK format. Both also have a possibility to run the software either

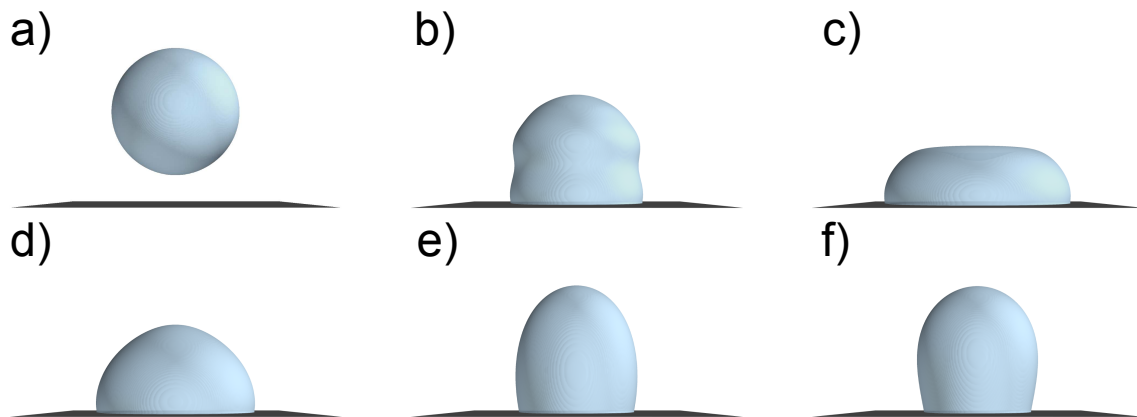


Figure 1. Snapshots from the simulation of the water droplet (light blue isosurface) hitting the solid surface (grey isosurface) from iterations a) 0, b) 100, c) 200, d) 300, e) 400 and f) 500. Visualizations were made with Mayavi.

Table 1. Comparison of visualization codes

	AViz (atomistic)	AViz (electronic)	nCAD software	nFLUID software	mayavi	paraview
<u>Atomistic scale</u>	yes	no	yes	no	yes	yes
<u>Electronic scale</u>	no	yes	yes	no	yes	yes
<u>Continuum scale</u>	no	no	no	yes	yes	yes
<u>Microsoft</u>	no	no	yes	yes	yes	yes
<u>Mac</u>	no	no	no	no	yes	yes
<u>LINUX</u>	yes	yes	no	yes	yes	yes
<u>OpenGL/Mesa</u>	yes	yes	yes	yes	yes	yes
<u>Stereo</u>	yes	yes	no	no	yes	yes

from a graphical user interface or by Python scripting. Especially the scripting option gives flexibility and provides a way to automate the data visualization of large datasets like time series, and this way render for example frames for videos.

6. Summary

Do we want to finish with something like

What is(are) a good choice(s) for your problem. questions like ..Is it continuum or discrete?

Acknowledgements: These works were supported by the program Plan Avanza from the Ministerio de Industria, Energía y Comercio from Gobierno de España, grant number TSI-020100-2011-300, in the case of nCAD, and by the FP7 program of the European Commission (FP7-NMP-2013-SMALL-7) with grant number 604005, in the case of nFluid.

Our group carries out atomistic simulations in carbon systems, with an emphasis on geometric structure and vibrations, especially near defects and interfaces. We need to observe the system development, and are often concerned with the structure deep inside the sample, which may not be accessible to non-destructive experiment. Analog models are helpful for gaining insight into 3D structures, but it is hardly practical to construct precise analog models for each timestep frame of a simulation. The next best is a 3D stereo visualization.

Figure 2. Stream tracing visualization of the water droplet which has hit the surface and is bouncing back, made with Paraview.

Table 2. Comparison of visualization codes

	AViz (atomistic)	AViz (electronic)	nCAD software	nFLUID software	mayavi	paraview
<u>Atomistic scale</u>	yes	no	yes	no	yes	yes
<u>Electronic scale</u>	no	yes	yes	no	yes	yes
<u>Continuum scale</u>	no	no	no	yes	yes	yes
<u>Microsoft</u>	no	no	yes	yes	yes	yes
<u>Mac</u>	no	no	no	no	yes	yes
<u>LINUX</u>	yes	yes	no	yes	yes	yes
<u>OpenGL/Mesa</u>	yes	yes	yes	yes	yes	yes
<u>Stereo</u>	yes	yes	no	no	yes	yes

We will introduce anaglyphic stereo in the next section, our AViz code in the following one, and conclude with a description of the new AViz 6.1 and some sample results.

Figure 3. Crystalline silicon with ORTEP.

Figure 4. Amorphous silicon with PGLOT.

Figure 5. Analyphic random walk with PG-PLOT.

Acknowledgments

We thank

References

- [1] Carroll K. Johnson, 1965. "ORTEP: A FORTRAN Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations". ONRL Report no. 3794. Oak Ridge, Ten, Oak Ridge National Laboratory)
- [2] Michael N. Burnett and Carroll K. Johnson, ORTEP-III: Oak Ridge Thermal Ellipsoid Plot Program for Crystal Structure Illustrations, Oak Ridge National Laboratory Report ORNL-6895, 1996.
- [3] Silverman A M. Sc. Thesis, Technion, 1990.
- [4] Sprott J C "Simple programs create 3D Images" 1992 *Computers in Physics* **6** pp 132-8
- [5] <http://phycomp.technion.ac.il/~aviz>
- [6] Adler J, Hashibon A, . Schreiber N, Sorkin A, Sorkin S and Wagner G 2002 "Visualization of MD and MC Simulations for Atomistic Modeling" *Computer Physics Communications*, **147** pp 665-9
- [7] Adler J 2003 "Visualization in Atomistic and Spin Simulations" *Computers in Science and Engineering*, **5** pp 61-65
- [8] Adler J, Koenka Y and Silverman A 2011 "Adventures in carbon visualization with AViz" *Physics Procedia* **15** pp 7-16
- [9] <http://phelafel.technion.ac.il/~peledan>
- [10] <http://phycomp.technion.ac.il/~newaviz>
- [11] Adler J, Fox J, Kalish R, Mutat T, Sorkin A and Warszawski E 2007 "The essential role of visualization for modeling nanotubes and nanodiamond" *Computer Physics Communications* **177** pp 19-20
- [12] Silverman A, Adler J and Kalish R 2011 "Diamond membrane surface after ion implantation induced graphitization for graphite removal:molecular dynamics simulation" *Phys. Rev. B* **83** 155410
- [13] J. Adler, A. Hashibon, N. Schreiber, A. Sorkin, S. Sorkin and G. Wagner, "Visualization of MD and MC Simulations for Atomistic Modeling", *Computer Physics Communications*, (2002) 147, p. 665-9.
- [14] J. Adler, A Column for the Simulations section, edited by D. Stauffer in *Computers in Science and Engineering*, "Visualization in Atomistic and Spin Simulations", (2003) *Computers in Science and Engineering*, 5, p. 61-65.
- [15] J. Adler, Y. Koenka and A. Silverman, "Adventures in carbon visualization with AViz", (2011) *Physics Procedia*, 15, 7-16.
- [16] D. Peled, A. Silverman and J. Adler "3D visualization of atomistic simulations on every desktop", (2013) *IOP Conference Series*, 454, p. 012076.
- [17] <http://phony1.technion.ac.il/aviz>
- [18] Joan Adler, J. Fox, R. Kalish, T. Mutat, A. Sorkin and E. Warszawski "The essential role of visualization for modeling nanotubes and nanodiamond", (2007) *Computer Physics Communications*, 177, p. 19-20.
- [19] Bastien Grosso, Valentino R. Cooper, Polina Pine, Adham Hashibon, Yuval Yaish and Joan Adler, "Visualization of electronic density", (2015) *Computer Physics Communications*, 195, p. 1-13.
- [20] Joan Adler, , Omri Adler, Meytal Kreif, Or Cohen, Bastien Grosso Adham Hashibon, Valentino R. Cooper, "Mini-review of Electron Density Visualization" (2015) *Physics Procedia*, 68, p. 26.

- [21] Prabhu Ramachandran and Gal Varoquaux, “Mayavi: 3D visualization of scientific data”, (2011) IEEE Computing in Science and Engineering 13, 2, pp 40-51
- [22] Enthought, “Chemistry example.” (n.d.). Retrieved January 27, 2016, from <http://docs.enthought.com/mayavi/mayavi/auto/examplechemistry.html>
- [23] Schroeder, W., Martin, K., Lorensen, B. (2006). “The visualization toolkit: An object-oriented approach to 3D graphics”. Clifton Park, NY: Kitware.
- [24] Avila, L. S. (2010). The VTK user’s guide: Install, use and extend the visualization toolkit. [1] Mayavi Project: 3D Scientific Data Visualization <http://code.enthought.com/projects/mayavi/>, accessed: 2016-02-04 and Plotting, [2] ParaView: An open-source, multi-platform data analysis and visualization application, <http://www.paraview.org/>, accessed: 2016-02-04 [3] The Visualization Toolkit (VTK): An open-source, freely available software system for 3D computer graphics, image processing, and visualization, <http://www.paraview.org/>, accessed: 2016-02-04 [4] P. Yuan and L. Schaefer, Equations of state in a lattice Boltzmann model, *Physics of Fluids*, 18(4), 2006 [5] FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video, <http://www.ffmpeg.org/>, accessed: 2016-02-04
- [25] SolidWorks (<https://www.solidworks.com/>) by Dassault Systmes
- [26] S.R. Hall, F. H. Allen, I. D. Browns The Crystallographic Information File (CIF): a New Standard Archive File for Crystallography, *Acta Cryst.* (1991). A47, 665-685
- [27] nCAD proprietary format
- [28] MaterialsStudio / Biovia by Accelrys of Dassault Systmes
- [29] CrystalMaker by CrystalMaker Software Limited
- [30] K. Momma and F. Izumi. VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data *J. Appl. Crystallogr.* (2011), 44, 1272-1276.
- [31] OpenFoam by OpenCFD Ltd. H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Comp. Phys.* (1998), 12, 620.