

CrystalSim Manual

1.	Running CrystalSim	1
2.	Editing the source code.....	3
2.1.	Installing the Java Development Kit.....	3
2.2.	Installing Eclipse.....	4
2.3.	Installing EGit	5
2.4.	Installing Visual Molecular Dynamics (VMD) for viewing the simulated crystals.....	6
2.4.1.	Loading structures and movies with VMD	6
3.	Explanation of Simulation Parameters	9
3.1.	Nucleation and Growth Parameters -- Figure 13.....	11
3.1.1.	Crystal Shape.....	11
3.1.2.	Axial Growth Velocity.....	11
3.1.3.	Nucleation	12
3.1.4.	Crystal Orientation	12
3.2.	Sample Parameters -- Figure 14.....	12
3.3.	Termination Condition Parameters -- Figure 15	13
3.4.	Fitting Parameters -- Figure 16 to Figure 18	13
3.5.	Misc Parameters -- Figure 19	15
3.6.	Input/Output Parameters -- Figure 20	16
3.7.	Simulation Runtime -- Figure 21	17
4.	Output file types	17
4.1.	.xyz files (Structures).....	17
4.2.	.xyz files (Movies).....	18
5.	References	20

1. Running CrystalSim

Download “CrystalSim.jar” from <https://github.com/JamesDMartin/CrystalSim> by clicking the “Download ZIP” button on the right hand side of the web page. Extract the zip file to a directory and double click on “CrystalSim.jar” to run it. You should then be greeted with something that looks like Figure 1.

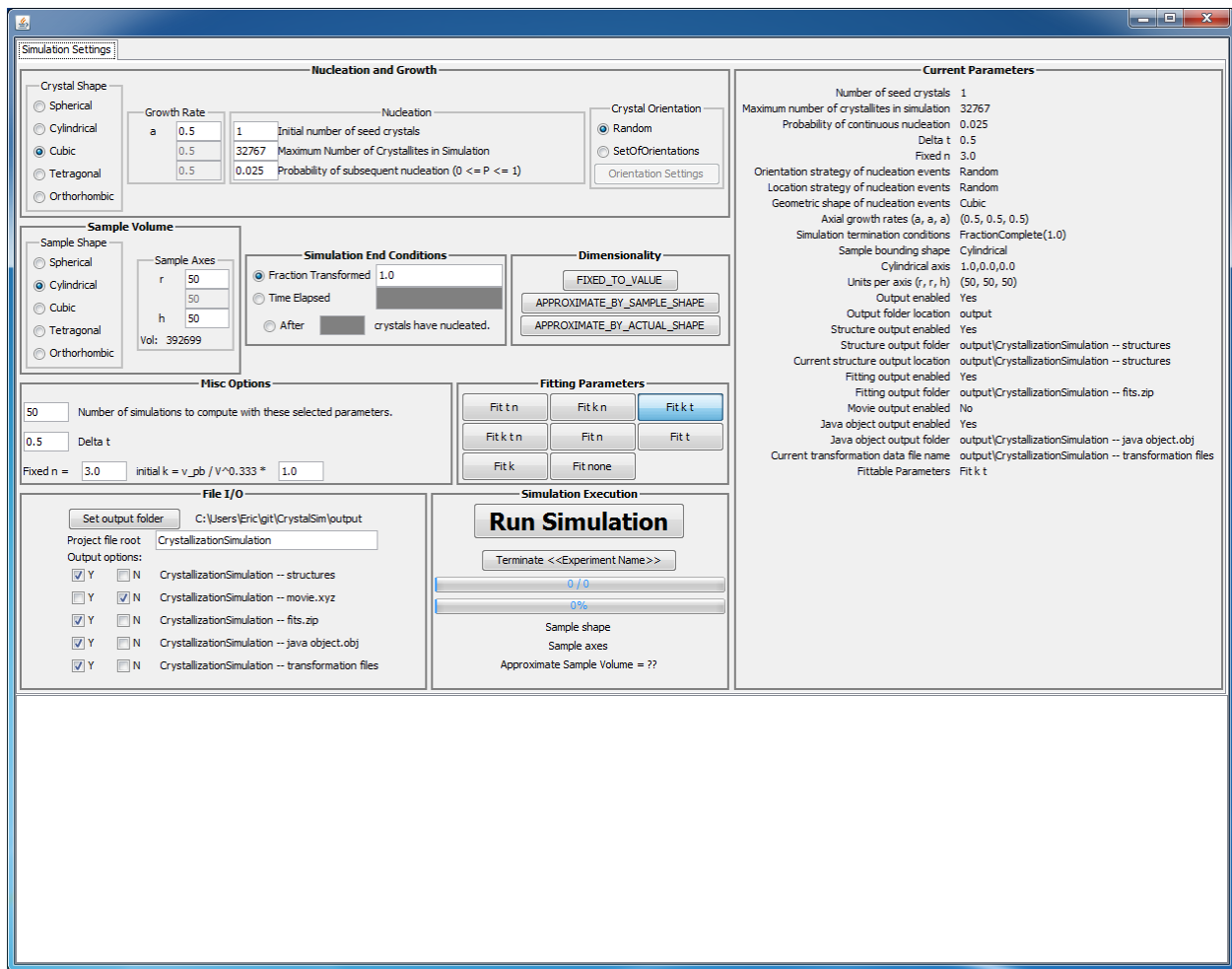
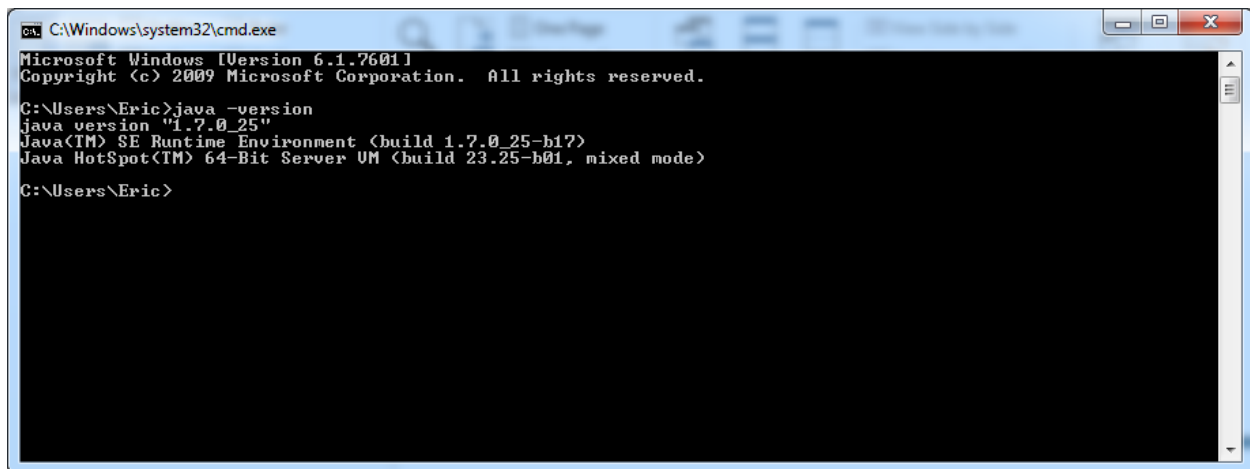


Figure 1. Screenshot of the CrystalSim GUI on Windows 7.

If nothing appears it is most likely that you do not have the Java Development Kit (JDK) or Java Runtime Environment (JRE) installed. To determine whether or not these are installed, open up a command window (windows key + r, then type cmd, then press enter) or a terminal and run the command:

java -version

If your output looks like Figure 2, then you have a version of the JRE or JDK installed and you can skip the next paragraph.

A screenshot of a Windows command prompt window. The title bar reads 'C:\Windows\system32\cmd.exe'. The window content shows the following text:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Eric>java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

C:\Users\Eric>
```

Figure 2. Verifying that the Java Development Kit (JDK) or the Java Runtime Environment (JRE) is installed.

If your output says something along the lines of “java is not recognized as an internal or external command...” then you need to install the Java Runtime Environment (JRE) or the Java Development Kit (JDK) installed. I would recommend using version 7 of either of them. If you want to edit the source code you will need the JDK and Eclipse, otherwise the JRE is fine. See Installing the Java Development Kit on page 3. After you have installed either of these, double click CrystalSim.jar and you should be greeted with Figure 1. If this still does not pop up, please rerun the `java -version` command and ensure that you have the JDK or JRE installed. If you still get an error, restart the computer and run the `java -version` command. If you’re still having issues, please get in touch with your local IT support and get java installed.

If you are positive that you have Java installed but you are not greeted with Figure 1, the next step is to run CrystalSim.jar from the command line:

`java -jar CrystalSim.jar.`

(Protip: Shift + right click on the “CrystalSim-master” folder and a new command will appear called “Open command window here.” Click that and then type in the aforementioned command.) If you get an error while doing this, please take note of exactly what the output was and send me (eddill@ncsu.edu) or Prof. Jim Martin (jdmartin@ncsu.edu) an email with the error text. Please include your Java version as well (the output from the `java -version` command).

2. Editing the source code

To view the source code, navigate to <https://github.com/JamesDMartin/CrystalSim/tree/master/src>. The source code was developed with Java 1.7 in the Eclipse IDE (Kepler). To edit the source code, the [Java Platform \(JDK\)](#) and the [Eclipse IDE](#) must be installed. For ease of use it is recommended to install the Eclipse extension “EGit” as the github repository can be cloned directly into Eclipse and it will be trivial to update the source code as the main repository is edited.

2.1. Installing the Java Development Kit

This software was developed with Java 1.7 revision 21 (1.7u21). I advise working with Java 1.7, but different versions will likely not have any issues. Download your Java Development Kit (JDK) of choice

from <http://www.oracle.com/technetwork/java/javase/downloads/index.html> and then install it like a standard program.

2.2. Installing Eclipse

This software was developed in the Eclipse Integrated Development Environment (IDE), Kepler edition. The software repository is configured to work with the Eclipse IDE and the setup is very easy if you decide to use Eclipse. Eclipse does not need to be installed, just downloaded and extracted from the zipped file. To download Eclipse, navigate to <http://www.eclipse.org/downloads/> and choose your desired flavor. I use “Eclipse Standard” edition. Select the appropriate version (32-bit or 64-bit) and commence the download on the following page. Once downloaded, navigate to the download location and you should see a file named something like “eclipse-standard-kepler-SR1-win32-x86_64.zip” which needs to be extracted into a folder (try right-clicking on it and look for the word “extract”). Once extracted Eclipse is operational and can be run by double-clicking on “eclipse.exe” within the extracted folder. In Windows, I like to copy the folder into my “Program Files” directory and then create a shortcut from “eclipse.exe” to the Desktop or Taskbar, but this is not a necessary step.

Run eclipse by double clicking on “eclipse.exe” or the shortcut that you created. You will be greeted by a sequence of screens where one will ask you to “Select a workspace.” For most users the default location is fine, so click the “Use this as default and do not ask me again” checkbox and click ok. If you decide that you would like a workspace in a different location, you can change that from within the IDE after it loads (**File->Switch Workspace**).

Eclipse will now load into the “Welcome to Eclipse” screen. Feel free to navigate around and explore or just go directly to the workbench by clicking the arrow in the top right corner labeled “Workbench.”

To configure your new Eclipse install to access the CrystalSim source code, you will need to install EGit. See the “Installing EGit” section for how to install this. Return to this point once you have installed EGit. Moving forward, I am assuming that you have EGit installed. If you haven’t yet done that, please go do it. It makes editing the source code and keeping it in sync with my changes extremely easy.

To add CrystalSim to your Eclipse IDE, click on **File**, then **Import**, then expand “Git,” select “Projects from Git and click “Next”. Then click “Clone URI” and click “Next.” Type “<https://github.com/JamesDMartin/CrystalSim.git>” into the URI: text box. Alternatively you can navigate to <https://github.com/JamesDMartin/CrystalSim> and click the ‘copy to clipboard’ icon on the right hand side of the web page and the little popup should change from “copy to clipboard” to “copied!” Once that has been copied, return to Eclipse and right click in the URI box and click paste (or Ctrl + V). After the URI box has is populated, click Next, then Next again, one more time, make sure that “Import existing projects” is selected and click next, then click Finish.

You will now need to clone another project, this time located at “<https://github.com/ericdill/GlobalPackages.git>”. Do this the exact same way that you just did added CrystalSim to the workspace.

At this point you will return to the Eclipse IDE with two new folders in the “Package Explorer” window on the left hand side of the screen named something like “CrystalSim [CrystalSim master]” and “GlobalPackages [GlobalPackages master]”. Expand the CrystalSim project by double clicking on the root folder or clicking the arrow to the left of the name. There is a default runnable file called

“CrystalSim.jar” that will run a version that I compiled. There will be a folder called “src” which contains the source code. There is also a folder named “doc” which contains this file among a few others. In the src folder there are seven package folders containing various aspects of the program source files. To run CrystalSim in the Eclipse IDE navigate to the “ui” folder and double-click “UIViewer.java” or “UI.java”. Then click the green circle with the play arrow in it (arrow in Figure 3), click “Run As” and then click “Java Application.”

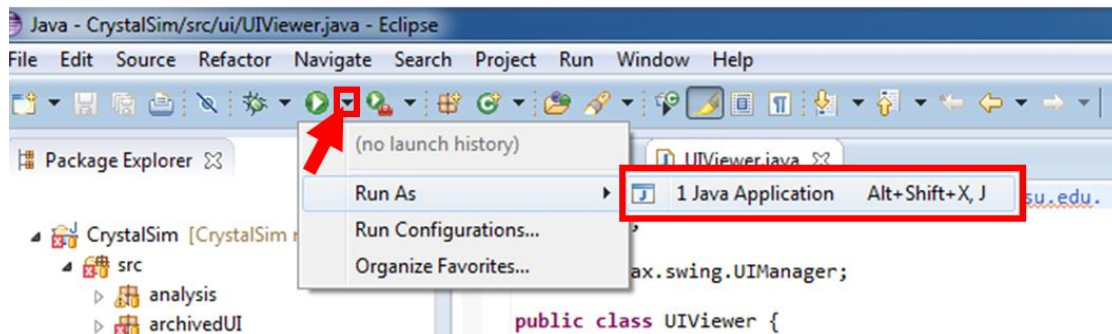


Figure 3. Running a program from the Eclipse IDE.

Because I’ve included various files that were useful at some point in the past but are now broken due to changes in other files, a popup will appear that is telling you that there are “Errors in the required project(s): Global Packages and CrystalSim. Proceed with launch?” The errors that are in these projects do not affect the runtime behavior of CrystalSim, so go ahead and “Proceed.” If you so desire, you can check the box “Always launch without asking” to never see this message again. I would advise against this, but it’s up to you. At this point, you should see a GUI that resembles Figure 1.

ECLIPSE PRO TIP: Add line numbers by right clicking just to the left of the text in the editor and click “Show Line Numbers.”

ECLIPSE PRO TIP: Change the font size by going to Window -> Preferences. A popup will appear, go to General -> Colors and Fonts. Now in the right side of that window click on “Java” and then “Java Editor Text Font” and then “Edit” on the right hand side of the window. I like 14 point font, but I would not change the font style. Looking at source code that is not monotonically spaced is just weird.

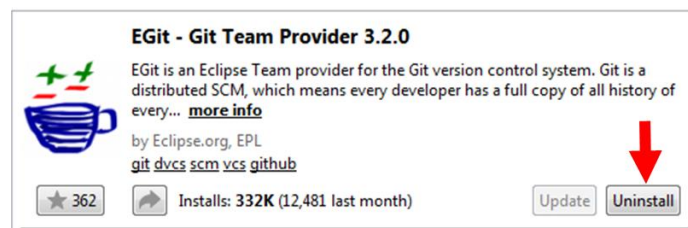


Figure 4. Click where the arrow is to install EGit from the Eclipse Marketplace. Yes, I realize that it says “Uninstall” in this image, but it will say “Install” if you have not yet installed it.

2.3. Installing EGit

Installing EGit is very easy from within the Eclipse IDE. Navigate to “Help” on the menu bar and then “Eclipse Marketplace.” Once this loads, search for “EGit” and click the “Install” button. (I realize that my image shows an “Uninstall” button; that’s because I already have it installed. If you don’t have it

installed your button will say “Install.”) Click through the installation, accepting the various agreements. Once finished, Eclipse will require a restart. After restarting you will have EGit installed.

2.4. Installing Visual Molecular Dynamics (VMD) for viewing the simulated crystals

Download VMD here:

<http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD>

Install it by double clicking on the downloaded executable (in Windows).

2.4.1. Loading structures and movies with VMD

Simulation structures are located in the folder at: [output directory]\[simulation name] – structures\, where the square brackets denote a variable defined in the UI. The structures are all .xyz files (see .xyz files (Structures) on page 17). To load these files with VMD, begin by opening the program. Once it has loaded (3 new windows should appear), click File->New Molecule on the VMD Main window, Figure 5.

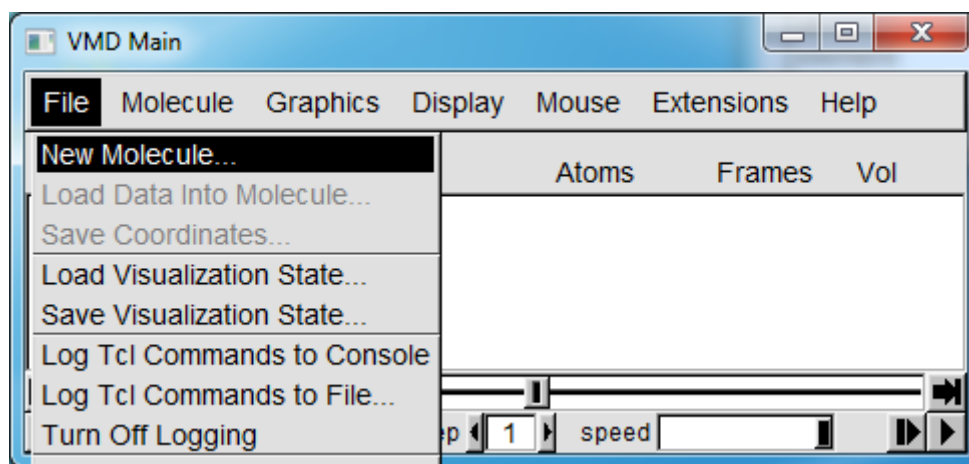


Figure 5. Main VMD Window.

This will open up a new window, the “Molecule File Browser,” Figure 6. On this window, click “Browse” and locate the .xyz file that you wish to load. Once you find the file you want and return to the “Molecule File Browser,” click “Load” and your file will be loaded into memory and you should see something appear in the “VMD Display” window, Figure 7.

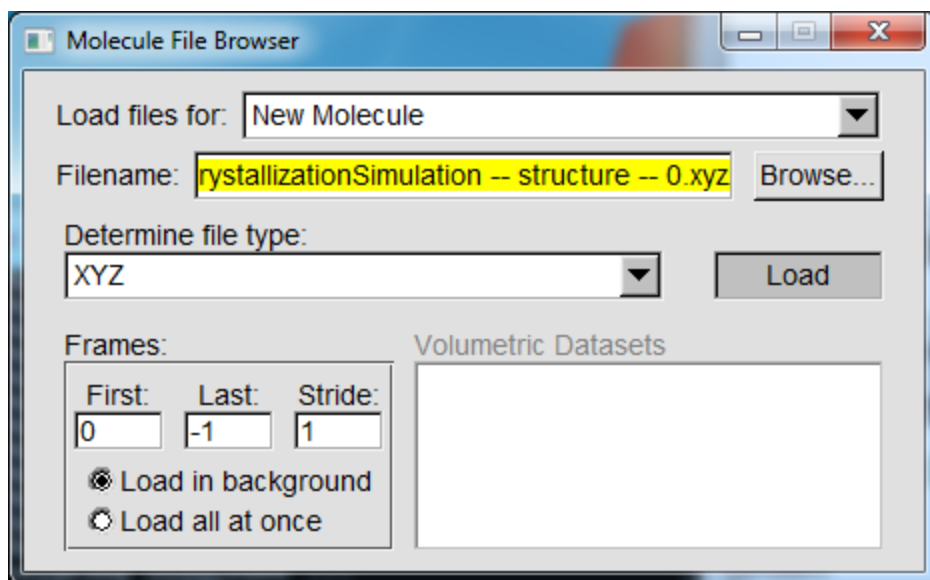


Figure 6. VMD window that allows the user to load a file.

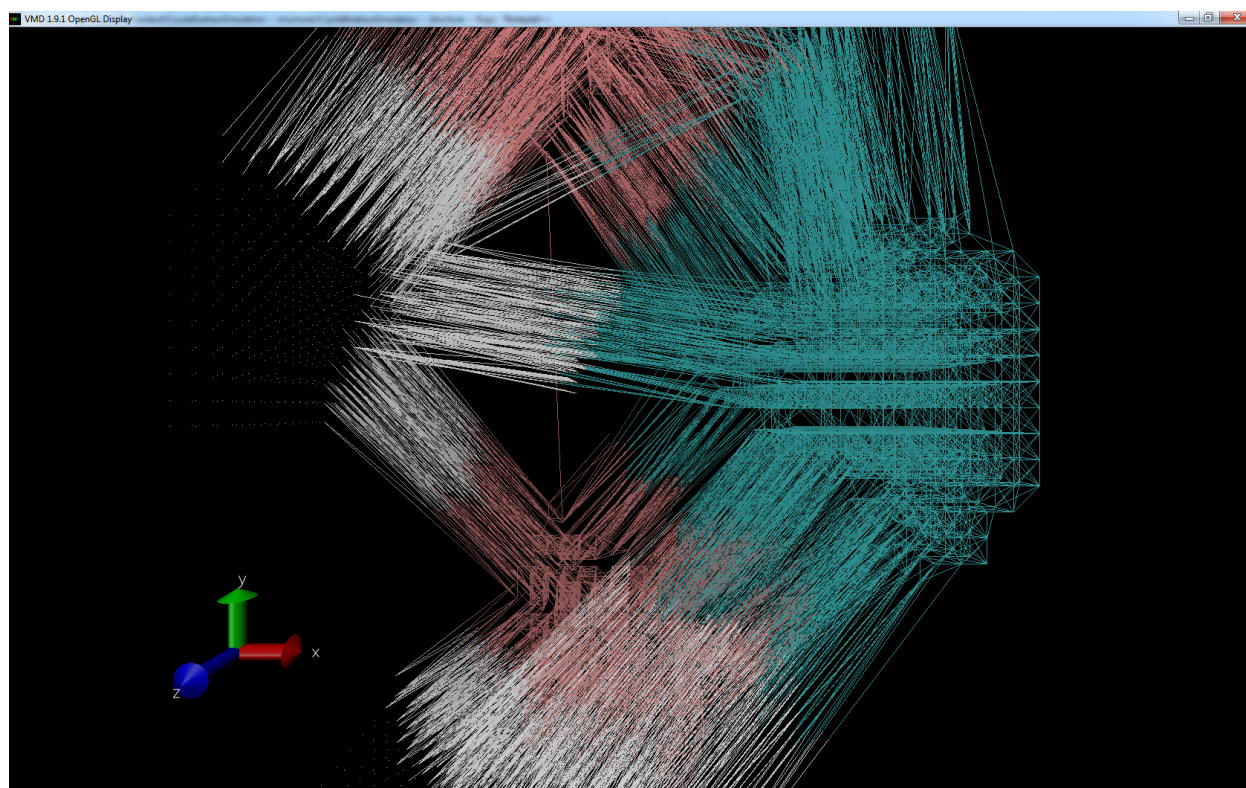


Figure 7. VMD display window showing the final exploded crystallite arrangement from a crystallization simulation.

When you first load a .xyz file from the crystallization simulation, the VMD window will display a variety of colored lines. This is not a helpful view from which to understand your crystallization simulation output, but not to worry! Your “VMD Main” window will now be populated with various things, as explained in Figure 8.

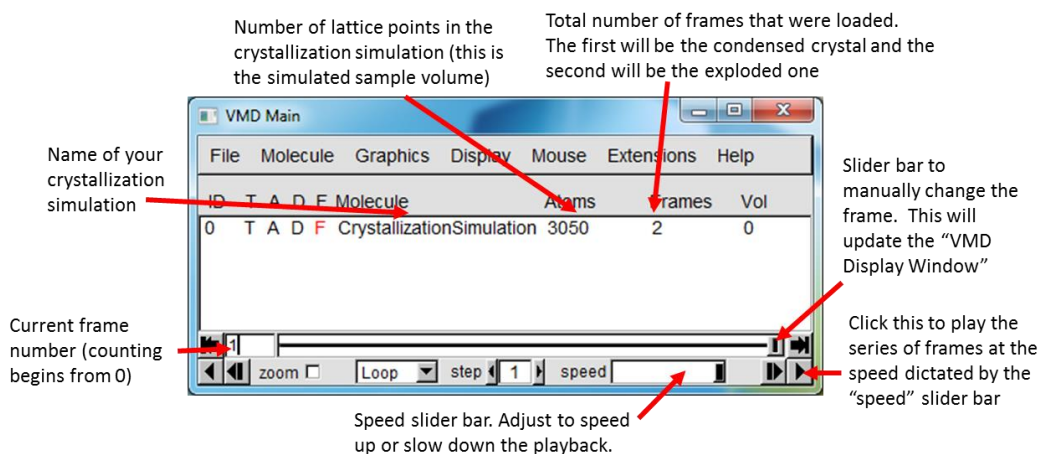


Figure 8. Explanation of "VMD Main" window after loading a crystallization simulation structure.

With the VMD display window and the "VMD Main" window visible, drag the main slider bar back and forth and you will see the VMD display window change. The default VMD method of displaying these crystals is not necessarily ideal so make the following changes as pictorially displayed in Figure xxx to Figure yyy.

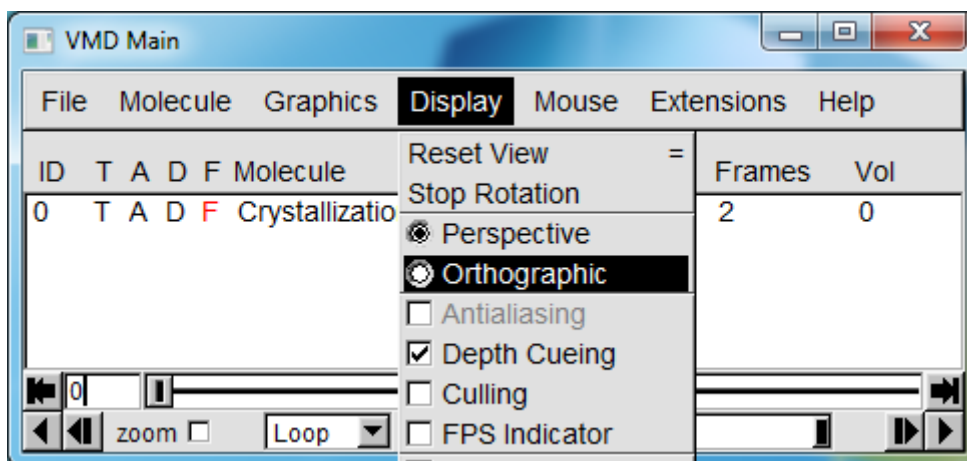


Figure 9. Change the Display mode to Orthographic

Click Graphics -> Representations and a new window will pop up,

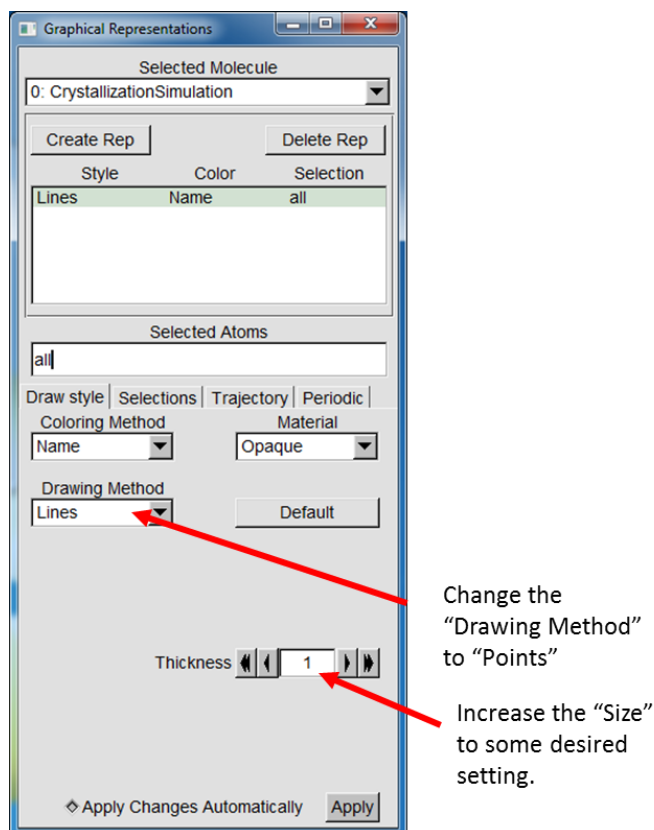


Figure 10. Changes to make to the plotting method to make the simulated sample volume more easily visible.

At this point, you should be able to easily visualize your simulated sample volume with different crystals having different colors. You can view individual crystallites by changing the “Selected Atoms” line from “all” to something like “name H”. This will show the first crystallite that nucleated. “name He” will display the second crystallite. “name H He B” will display the first, second and fifth crystallites that nucleated.

3. Explanation of Simulation Parameters

The CrystalSim GUI consists of three main panels: modify-able simulation parameters, a list of the current simulation parameters, and a message box at the bottom of the screen where relevant output is visible, as shown in Figure 11. The list of current simulation parameters will auto-update as the toggle-able parameters in the modify-able simulation parameters window are changed, and the changed parameter will be noted in the message box. The simulation parameters are further delineated by color in Figure 12 and described subsequently.

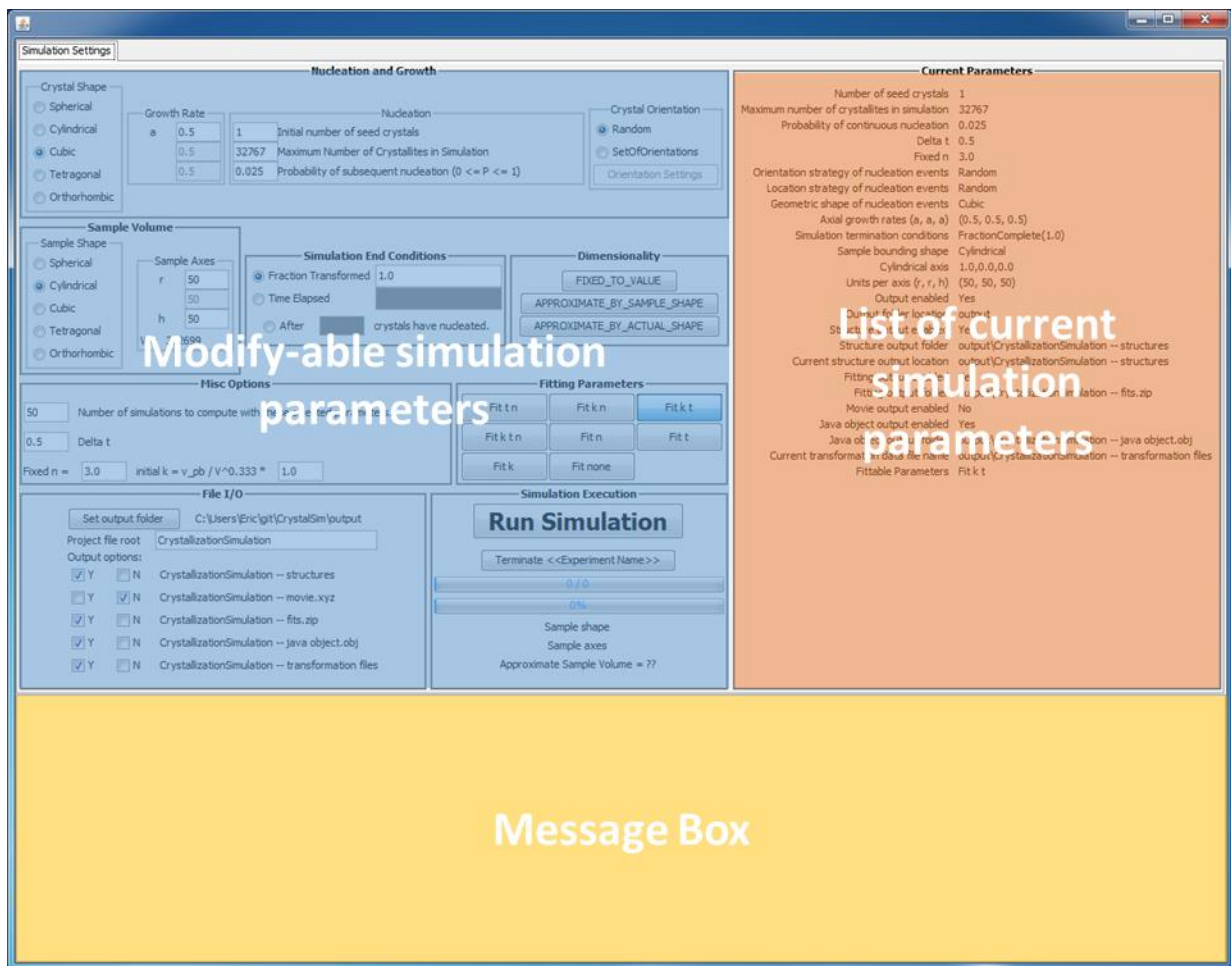


Figure 11. Three main sections of the UI highlighted in color.

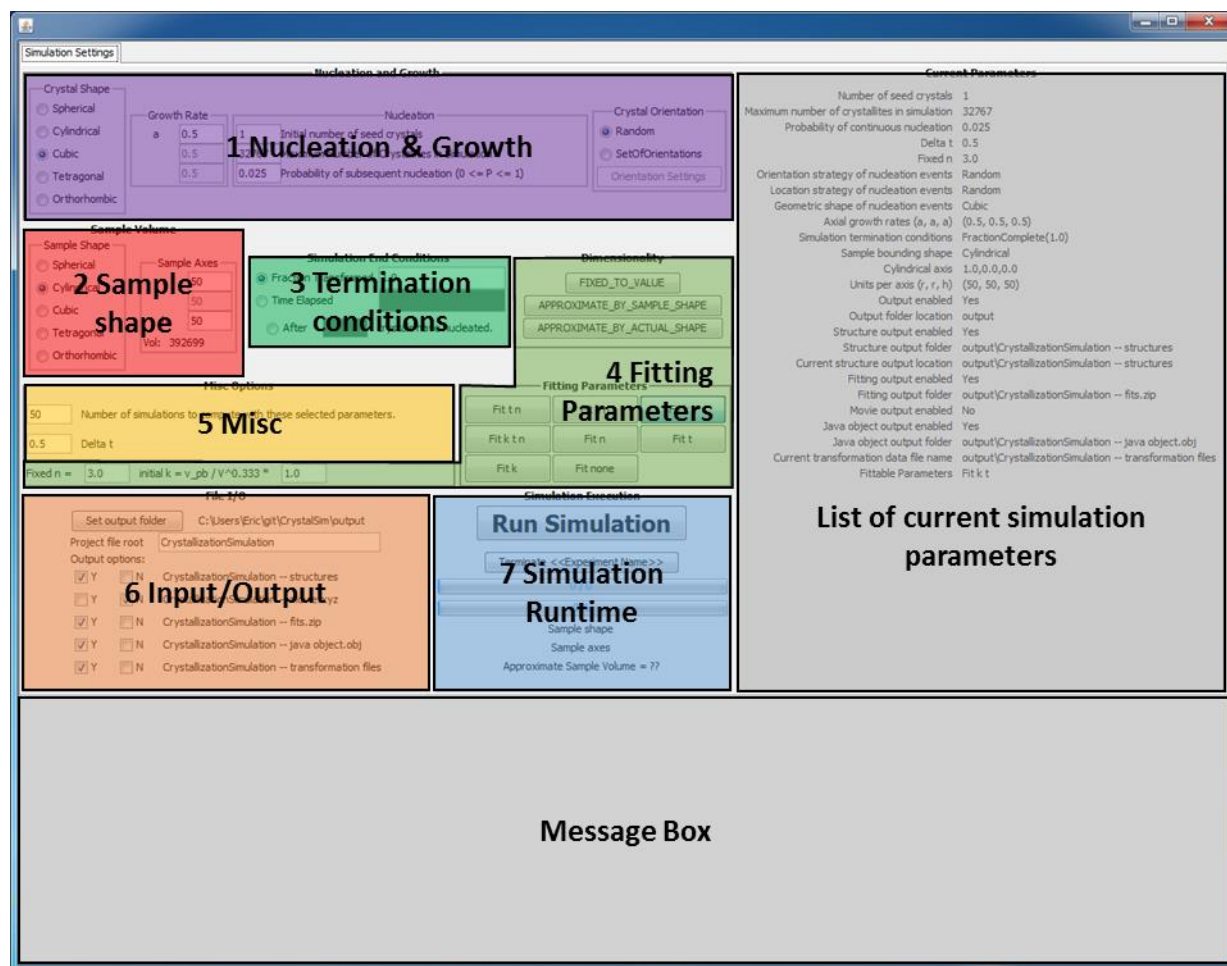


Figure 12. Simulation parameters grouped by color.

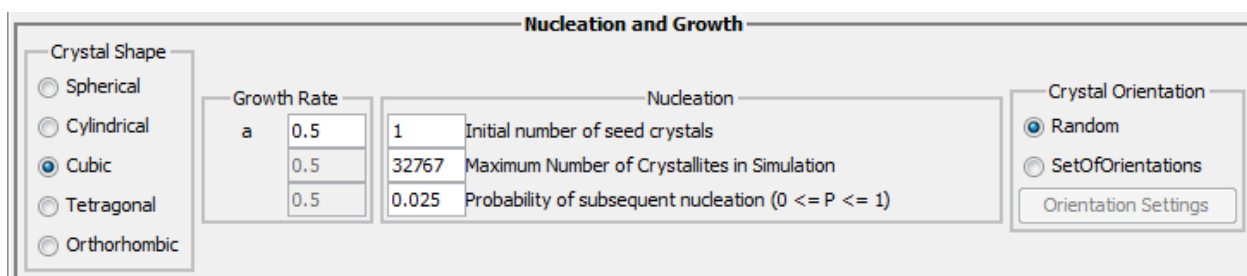


Figure 13. Nucleation and growth parameters grouped by color.

3.1. Nucleation and Growth Parameters -- Figure 13

3.1.1. Crystal Shape

There are five options for the shape of the crystallites to be grown in the simulation: Spherical, Cylindrical, Cubic, Tetragonal and Orthorhombic.

3.1.2. Axial Growth Velocity

Based on the radio button that is selected for the crystal shape, the appropriate axial growth velocities, in green, will be editable. The axial growth velocity is the number of distance units that each axis

increases during one unit time. One distance unit is the distance between two of the simple cubic lattice points.

3.1.3. Nucleation

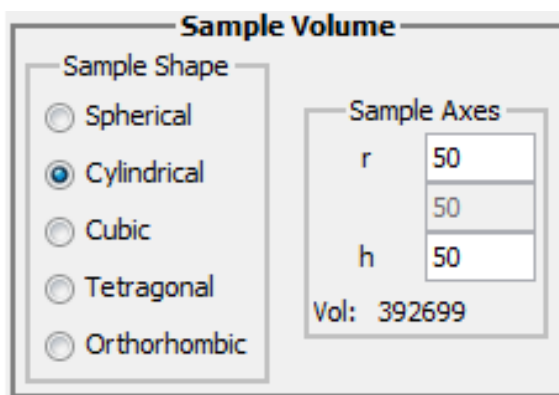
The sample will be “seeded” with the “**Initial number of seed crystals.**” These crystallites are placed in random locations and will begin growing at $t=100$. An initial time of 100 is used because the non-linear fitting algorithm does not behave when fitting parameters drop below zero and $t_{\text{init}}=100$ is usually sufficient to ensure that the t_0 fitting parameters do not drop below zero.

The parameter “**Maximum number of crystallites in simulation**” provides an upper bound for the number of crystallites that can grow during a simulation. After this limit has been reached, no more nucleation will occur.

The “**Probability of subsequent nucleation**” (P_{nuc}) parameter provides a method to allow nucleation to occur after the initial nucleation events that occur at $t=100$. The value provided in this box is per unit time and is scaled appropriately based on the value provided in the “**Time step**” text box, such that the effective nucleation probability is $P_{\text{eff}} = P_{\text{nuc}} \times \Delta t$. A random number, R , is generated every time step and compared with P_{eff} . If $R < P_{\text{eff}}$, then a nucleation event is allowed to occur *in the untransformed volume*. The nucleated crystallite has an orientation according to the **Crystal Orientation** settings.

3.1.4. Crystal Orientation

The Kolmogorov-Johnson-Mehl-Avrami condensed phase reaction framework assumes that nucleated crystallites have no preferred orientation, so the default mechanism for determining crystallite orientation is to give them random orientations. It might be useful in certain cases for the user to be able to provide a set of pre-determined crystal orientations that the system is allowed to choose from. This functionality has not yet been implemented and if the “SetOfOrientations” radio button is selected this currently results in the crystallites being oriented only along the Cartesian axes in the positive and negative directions, for six total orientations.



Sample Volume

Sample Shape

- ☐ Spherical
- ☒ Cylindrical
- ☐ Cubic
- ☐ Tetragonal
- ☐ Orthorhombic

Sample Axes

r: 50

h: 50

Vol: 392699

Figure 14. Parameters controlling the size and shape of the sample

3.2. Sample Parameters -- Figure 14

The simulation volume is a 3D volume gridded as a simple cubic lattice of size $a \times b \times c$. The shape of this lattice is determined by the radio buttons, currently allowing the sample shape to be Spherical, Cylindrical, Cubic, Tetragonal or Orthorhombic. Based on the radio button selection, the appropriate

axes will be editable. The total number of lattice points which are inside the sample shape with the specified units is given after “Vol:”

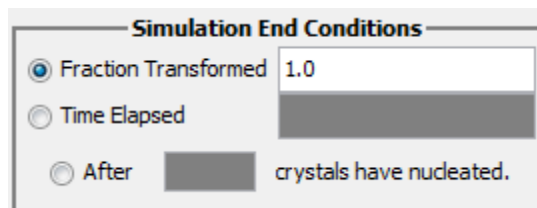


Figure 15. Parameters controlling how the simulation ends.

3.3. Termination Condition Parameters -- Figure 15

The user is given the option to specify how the simulation is terminated. By default, the simulation ends after the entire simulation volume is transformed. However, there additional selectable options for “Time elapsed” which ends the simulation after the specified amount of time has elapsed and also to end the simulation after some number of crystallites have nucleated. Note that the number of nucleated crystallites also includes the number of crystallites that the sample was seeded with.

3.4. Fitting Parameters -- Figure 16 to Figure 18

It is strongly encouraged that multiple simulations are run with each set of parameters. For the initial paper where this software was used, I did ~600 simulations for each set of parameters that I used. Because of the randomness built in to the simulation, specifically the way in which new crystals are nucleated, that they are nucleated in random locations and with random (or a small set of) orientations, no two simulations will produce the same output, unless the same random number seed is used for the random number generator. As a result of needing multiple simulations to increase the reliability of the results, dozens to hundreds to thousands of crystallites will have been simulated. It is impossible to fit all of these simulations by hand, so I wrote an automated fitting routine to automatically fit the individual crystallite transformations and output all results into a tab-delimited text file that can be directly copied and pasted into Microsoft Excel (or your favorite spreadsheet software) for further analysis and visualization. This text file is described in a subsection of Input/Output Parameters -- Figure 20 on page 16.

The model that I am using to fit the crystallite transformations is that of Kolmogorov,¹ Johnson and Mehl² and Avrami³⁻⁵ (KJMA), as given in Eq. 1.

$$\alpha(t) = 1 - \exp\{-[k * (t - t_0)]^n\} \quad 1$$

This model has three fitting parameters, k , t_0 and n . The parameters are hideously correlated (>0.9) and their physical interpretation can be found in Dill et al., *Chem. Mater.* **2013** 25(20) pp 3932-3940 & 3941-3951, but are briefly summarized as:

k : Related to the phase boundary velocity of crystallization by the sample volume and sample shape

t_0 : The nucleation time of the individual crystallite or the bulk sample, depending on what is being fit

n : The dimensionality of growth, 1=rod, 2=disc, 3=3D object.

As the parameters are hideously correlated, it may be desirable to fix some or all of them to a specific value and fit the others. This option is controlled by the set of toggle buttons as given in Figure 16.

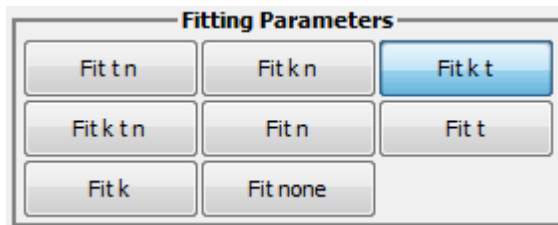


Figure 16. Toggle buttons to control which parameters are fit.

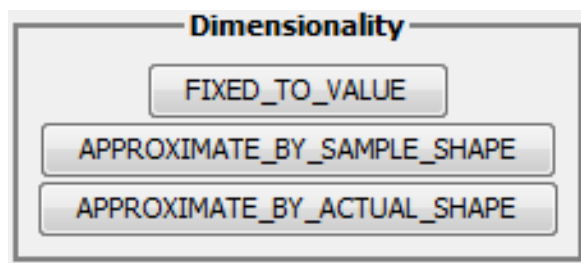


Figure 17. Toggle buttons to control how n is determined.

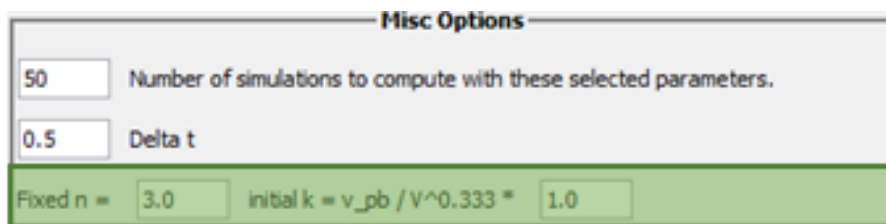


Figure 18. Specific values to use when fixing various parameters in the KJMA model as highlighted in green.

When the dimensionality (n) parameter is fixed, there are a few options as to how its value is determined. This is controlled by the toggle buttons given in Figure 17. When the “FIXED_TO_VALUE” button is selected, the fitting algorithm will use the value given in the “Fixed n =” text box, shown in Figure 18.

When the value “APPROXIMATE_BY_SAMPLE_SHAPE” is selected, the dimensionality is approximated based on the dimensions of each of the sample shape axes (a , b , c), as given in Eq. 2. In the isotropic limit when all sample axes are equivalent n_{sample} degenerates to 3.

$$n_{sample} = \frac{a + b + c}{\max(a, b, c)} \quad 2$$

When the value “APPROXIMATE_BY_ACTUAL_SHAPE” is selected, the dimensionality is estimated based on the shape of the actual crystallite when it was 50% crystallized. The 50% mark was selected because the algorithm is automatically fitting from 0% crystallized to 50% crystallized and thus the sample shape after 50% is irrelevant for the fit. I should update this software to give the option for the user to specify the specific region of alpha where the crystallite is being fit. If I still have not done this and you would like this feature implemented, please let me know at eddill@ncsu.edu or jdmartin@ncsu.edu. The way

that I approximate the shape of the individual crystallites is to compute the center of mass of the crystallite at 50% and then find the longest distance from the center of mass of the crystallite to the surface of the crystallite. I then find the smallest distance from the center of mass to the surface of the crystallite that is orthogonal to the long axis. The third axis is then defined as the cross product of the long and short axes. The estimated dimensionality is then determined with an expression similar to Eq.2, except a, b, and c are the lengths of the three crystallite axes.

Finally, the “initial k” variable allows the user to tweak the starting guess for my non-linear fitting algorithm. As non-linear fitting can be very sensitive to the starting conditions, and the fit value of the rate constant is modified by very many parameters, this is essentially an empirical value that allows the user to help me automatically fit the resultant transformations more effectively. If you find that when you use this program your crystallites are not being automatically fit, adjust this term until the crystallites are being automatically fit.

Figure 19. Specific values to use when fixing various parameters in the KJMA model as highlighted in yellow.

3.5. Misc Parameters -- Figure 19

The first option is self-explanatory: It determines how many simulations are run with the set of selected parameters.

The second option controls the time step of the simulation. This option is useful in conjunction with the Axial Growth Velocity on page 11. As the axial growth velocity increases, the time step should be reduced to ensure that enough time points are samples during the simulated transformation. Similarly, as the axial growth velocity decreases, the time step can be increased to reduce the simulation time. A reasonable guideline is that the inverse of the time step should equal the maximum of the growth velocity in any of the three dimensions.

Figure 20. Control the simulation output location and choose the specific things to output.

3.6. Input/Output Parameters -- Figure 20

Clicking the “**Set output folder**” button will open a file chooser that allows the user to select a specific location to output simulation results.

The “**Project file root**” text box allows the user to change the name of the simulation. Changes to this name will be reflected in the text below.

There are five output options:

- Structures
 - This option will output an xyz file which is simply a list of the coordinates of each of the lattice points in the simulated volume. An xyz file contains three coordinates and the Z value of each atom, all are tab delimited. The Z value of each atom, in this case, is the crystallite index. Because of the large number of data points in these simulations, I would strongly advise opening these with VMD, which is extremely effective at handling a large number of coordinates to plot. See Loading structures and movies with VMD on page 6.
- Movies
 - This option will output a file which is formatted similarly to the xyz files but contains a list of coordinates of each of the lattice points *in each of the time steps of the simulation*. For this reason, these files can be extremely large (multiple gigabytes) and so this option is disabled by default. Similar to the structure files, these files work well with VMD. See Loading structures and movies with VMD on page 6.
- Fits
 - This software will automatically fit the transformations of all simulated crystallites and the total transformation of the simulated volume when this option is selected. The first row of these output files are the fitting parameters in order k , t_0 , n . The remainder of this file are three tab-delimited columns which contain the simulation time, the simulated transformation and the fit transformation, respectively. If your fits are not very good, consider adjusting the initially estimated rate constant to a more appropriate value. If you still have trouble, email me.
- Java Objects
 - This software is written in Java and, if this option is selected, will output the native Java objects which can be used at a later date to re-analyze the simulations. This is an advanced feature which currently requires coding knowledge to utilize.
- Transformation Files
 - The transformation files contain all relevant information for the simulation. It is strongly recommended to output this file.
 - The header of this file is the majority of the simulated parameters that were output.
 - The next component of this file is a series of columns which correspond to the simulation time, the number of untransformed lattice points, the number of lattice points transformed during that time step, the total number of transformed lattice points. The remaining columns are the number of lattice points that each individual crystallite transformed during that time step.

- The next component of this file is the crystal information, which contains the nucleation location, crystallite orientation, axial growth rates, initial dimensions and the nucleation time for each crystallite that contributed to the transformation.
- The next component of this file is a fitting summary which gives the fitting parameters and the standard deviation of these fitting parameters.
- Finally, a summary of each crystallite is given in a row-tab-delimited format which can easily be copied into excel for further analysis. The header of each column should be sufficiently descriptive. If it is not, please contact me and I will clarify that. This is the same information that is provided in the “[simulation name] -- automated fitting.txt” file.

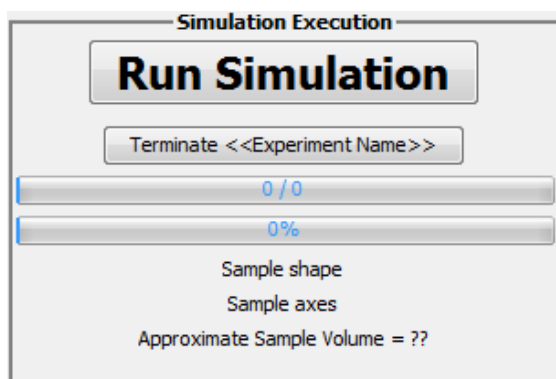


Figure 21. Section of the UI dedicated to updating the user as to the runtime status of the simulation

3.7. Simulation Runtime -- Figure 21

To run the simulation, click the “Run Simulation” button. Once this button is clicked, the two progress bars will begin to update. The first progress bar shows the number of simulations that are complete and the number of simulation remaining. The second progress bar shows the progress of the current simulation. What is currently showing as “Sample shape” will change to show the currently chosen shape of the sample container. The “Sample axes” text will change to show the unique axes and the number of units in those directions. Finally, the “??” after “Approximate Sample Volume” will change to show the approximate sample volume of the simulation.

4. Output file types

4.1. .xyz files (Structures)

[From Wikipedia:](#) The XYZ file format is a chemical file format. There is no formal standard and several variations exist, but a typical XYZ file format specifies the molecule geometry by giving the number of atoms with Cartesian coordinates that will be read on the first line, a comment on the second, and the lines of atomic coordinates in the following lines. The file format is used in computational chemistry programs for importing and exporting geometries. The units are generally in Angstroms. Some variations include using atomic numbers instead of atomic symbols, or skipping the comment line. Files using the XYZ format conventionally have the .xyz extension.

My .xyz output file has two sections. The first line is the number of atoms in the file. The remaining lines are formatted as: “Z[tab]x[tab]y[tab]z” where Z is the name of the element and (x,y,z) are the Cartesian coordinates of that atom.

4.2. .xyz files (Movies)

My .xyz movie output file is essentially a concatenated list of xyz files that correspond to each of the time steps of the simulation. Each of the time steps has three sections. The first line is the number of atoms in the file. The second line is an integer index corresponding to the time step. The first time step is 0, the second is 1, etc. The remaining lines are formatted as: "Z[tab]x[tab]y[tab]z" where Z is the name of the element and (x,y,z) are the Cartesian coordinates of that atom.

5. Class Diagram

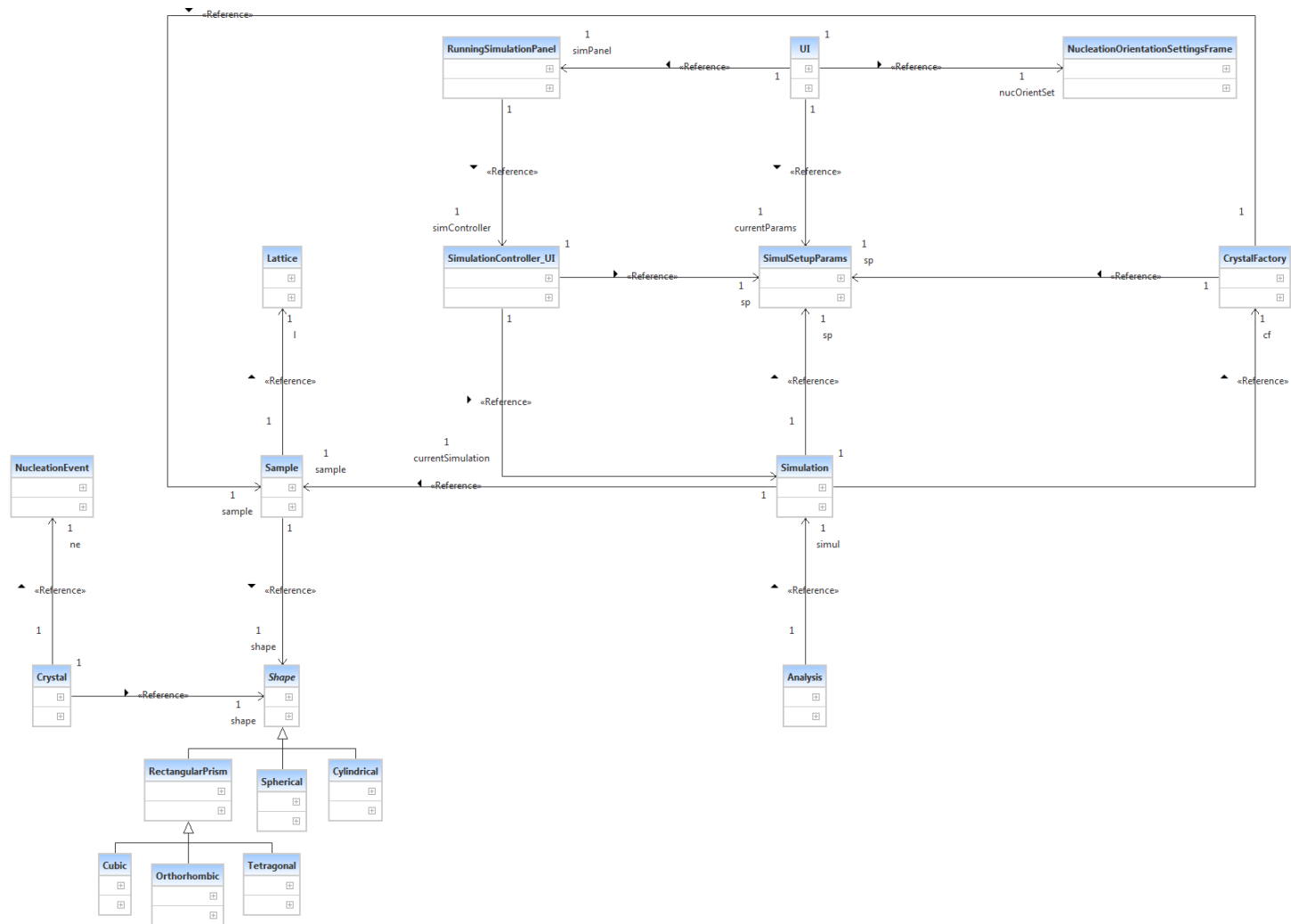


Figure 22. Class diagram for the CrystalSim program. This is available as a .uml file in the “doc” folder at the Github repository.

6. References

1. Kolmogorov, A., *B ACAD SCI URSS SCI* **1937**, 3, 355.
2. Johnson, W. A.; Mehl, R. F., Reaction kinetics in processes of nucleation and growth. *T Am I Min Met Eng* **1939**, 135, 416-442.
3. Avrami, M., Kinetics of phase change I - General theory. *J Chem Phys* **1939**, 7 (12), 1103-1112.
4. Avrami, M., Kinetics of phase change II Transformation-Time Relations for Random Distribution of Nuclei. *J Chem Phys* **1940**, 8 (12), 212-224.
5. Avrami, M., Granulation, Phase Change, and Microstructure - Kinetics of Phase Change. III. *J Chem Phys* **1941**, 9 (2), 177-184.