A Graphical User Interface to Specify Parameters for Molecular Dynamics Simulations

Desirée E. Velázquez Ríos
842-06-9756
Department of Mathematics
University of Puerto Rico at Humacao
Faculty advisor: José O. Sotero Esteva

I. Introduction

I.1 Molecular Dynamics

A Molecular Dynamics (MD) simulation requires the specification of many parameters, including the name and localization of several data files; physical parameters such as temperature, and simulation parameters such as the amount of time-steps, periodic boundary conditions and many others. These configuration files are written most of the time using a text editor. This is a difficult task and the source of many errors that can result in the waste of execution time in sophisticated computer systems.

The reason is that the configuration files are composed by files of the extension type *.pdb* and *.psf* that contain the following information: name of the compound, species and tissue from which is was obtained, authorship, revision history, journal citation, references, amino acid sequence, stoichiometry, secondary structure locations, crystal lattice and symmetry group, and finally the ATOM and HETATM records containing the coordinates of the protein and any waters, ions, or other heterogeneous atoms in the crystal. These files are organized in a specific way so that each file can be reused without having to change much the information contained. As mentioned before, these files are then used to create the configuration files, with the extension .conf for NAnoscale Molecular Dynamics (NAMD).

I.2 MOSDAS-GUI

A Graphical User Interface (GUI) to specify parameters for MD simulations was developed in order to simplify the earlier mentioned process and reduce the number of errors the can waste execution time. The interface presents to the user a set of templates to the specification of the configuration parameters. The templates group the parameters by categories according to their nature: physical, data, etc. Special attention has been put into the usability of the GUI. Suggestions of the values of the parameters are made based on previous choices such as commonly used data sources and physical parameters, as well as the choices of parameters that the user is making during the session.

At the start, the GUI is opened through the Visual Molecular Dynamics (VMD) program, which shows the menu bar from where the user will work. It prevents the user from accessing anything while nothing is on screen, or has been created. Once the user

creates a molecule or system of molecules, such as fibers that can include a strand of DNA with a carbon nanotube or different types of polymers such as Copolymers or Homopolymers that include Polyaniline, the GUI opens up new options that the user is now able to use. When creating a system one is able to determine the length or components of said system

But before the user is able to access options like moving or centralizing the systems, he must first select which system or systems of molecules he would like to work with. Moving the system allows the user to change the xyz coordinates of the selected system, while centralizing it puts the geometrical center of the system in the 000 coordinate. If the user selects two or more systems, the merging option is available. This option merges the two or more selected systems, creating one whole, new system but the previously selected systems will be unable to be accessed as separate systems again. Other options, such as boxing the systems for simulations, are available as well.

The GUI also guides the user through the parameter specifications process by visually suggesting the order in which they should be entered. This guarantees that the user has finished entering a set of data before moving onto the next. And thus the GUI has proven to be a valuable tool for the specification of parameters for a MD simulation.

II. Window for MD configuration parameters

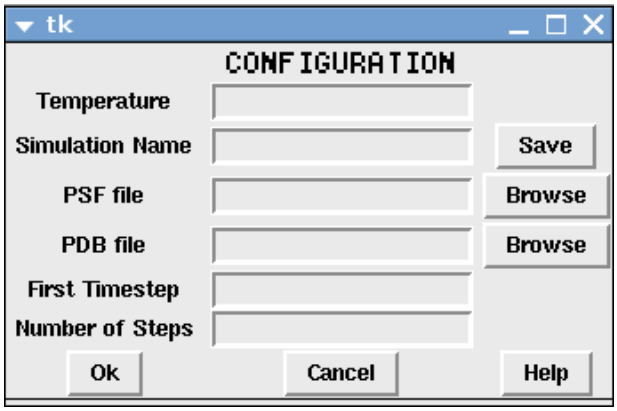
At first, I constructed the configuration window, that would help in the simplification of configuring parameters in MD simulations, for the GUI (spoken about in the last report), in order for it to be incorporated into the MOSDAS-GUI. This window was created using the Python programming language, the Tkinter library and the tkFileDialog module. In order to program well, the DrPython Interpreter was used. It has a feeling of the DevBloodshed C++ so it was easy to get accustomed to it.

To begin, we create a window using the Tkinter library. The window is divided by eight rows and three columns. For each phase that describes what is required of the user, a Label is used. A variable, containing the label is appended to the desired position of this 8x3 table window. In the first row, only the label of the title of the window is contained in the second column.

In the second, sixth and seventh rows, labels that require only text written by the user are contained in the first column. The second column for these rows contains an empty space box each where the user will write the required information. For the third, fourth and fifth rows the same process just mentioned is applied. The only difference is that the user need not write the names for all three rows. The third row requires the user to write a line of text, but he is able to browse where he’ll save the configuration file, along with the option of just selecting a previously saved file to replace. The fourth and fifth rows share a similarity to the third row, but it only requires the user to select the name of a *pdb* and *psf* file to add to the configuration file. This process is accessed through the use of buttons and thanks to the tkFileDialog module and its

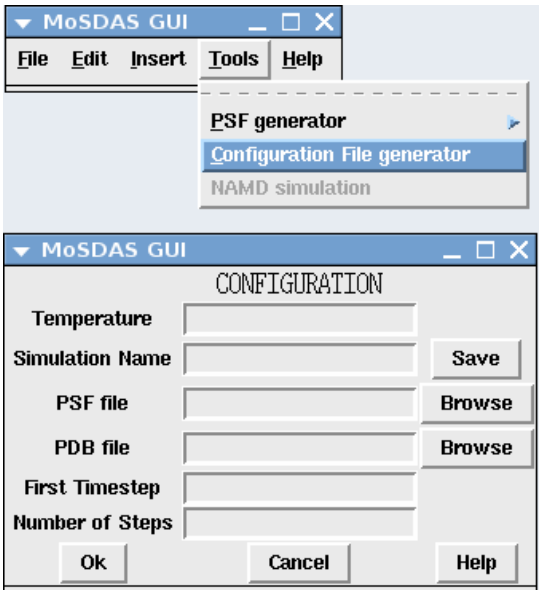
asksaveasfilename() and askopenfilename() functions which can be selected through the use of said buttons.

The last row contains three important functions. The first column contains the *ok* button, where once the user has pressed it, the configuration window saves the information written and sends it the default folder or the folder selected by the user through the button on the third row. Once it has saved and sent the configuration file, the window closes. The *cancel* button does as its name implies, it cancels the activity and closes the window without saving anything. Lastly, the *help* button will give suggestions in case the user does not know what he should write. A picture of the finished window running from the computer directly can be seen below.



III. Results and Discussion

Once incorporated and the user has done the earlier mentioned steps, he will be able to save the system(s) through the VMD menu or through the configuration window, shown below again, this time running through the VMD program.



With this, we are able to tell that our window works just fine. But we can improve the configuration file generator by adding suggestions in the empty box spaces.

IV. Future Work

To include suggestions for each option that the user utilizes. By adding suggestions in the space boxes that are empty right now, the user will be able to have a clearer idea of what it is that he needs to put in. Another way of improving the configuration window is by adding a bar that the user can slide for options such as temperature or number of steps, allowing him to know from where to where he should select the information.

V. References

- *Un Interfaz Gráfico de Usuario para Especificar Parámetros en Simulaciones de Dinámica Molecular* (poster), SIDIM 2008, UPR Carolina, March 1, 2008

- *A Graphical User Interface to Specify Parameters for Molecular Dynamics Simulations* (poster), Symposium on Nanotechnology, Palmas del Mar, Humacao, May 2, 2008

- *Python Essential Reference 3rd Edition*, David M. Beazly, Sams Publishing (2006)

- *Tkinter reference: a GUI for Python*, John W. Shipman, New Mexico Tech (2004), http://www.nmt.edu/tcc/help/pubs/tkinter

- *An Introduction to Tkinter*, Fredrik Lundh (1999), http://www.pythonware.com/library/tkinter/introduction/

- http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-unix-html/namd-tutorial-unix.html (NAMD Tutorial by the Computational Biophysics Workshop of the Beckman Institute from the NIH Resource for Macromolecular Modeling and Bioinformatics in the University of Illinois at Urbana-Champaign)