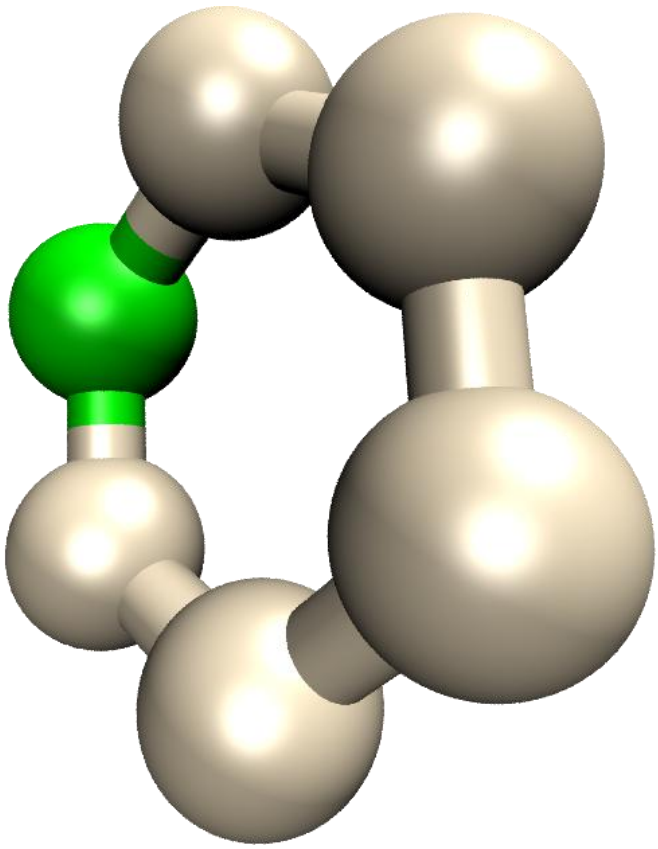
OpenMM Zephyr



User's Manual

Release 1.1

February 11, 2010

Website: Simtk.org/home/zephyr

Copyright and Permission Notice

Acknowledgments

SimTK software and all related activities are funded by the Simbios National Center for Biomedical Computing through the National Institutes of Health Roadmap for Medical Research, Grant U54 GM072970. Information on the National Centers can be found at http://nihroadmap.nih.gov/bioinformatics.

Table of Contents

1 OVERVIEW 1

1.1 Introduction to Molecular Dynamics 1
1.2 What is OpenMM Zephyr? 1
1.2.1 Limitations of OpenMM Zephyr *2*
1.3 What is OpenMM? 2
1.3.1 Running OpenMM on GPUs *2*
1.3.2 More information about OpenMM *3*
1.4 Conventions Used in this Document 3

2 INSTALLING OPENMM ZEPHYR 5

2.1 Prerequisites 5
2.1.1 Installing VMD (optional but recommended) *5*
2.2 Installing GPU software 6
2.2.1 Installing CUDA for NVIDIA GPUs *6*
2.3 Getting OpenMM Zephyr 14
2.3.1 Download OpenMM Zephyr *14*
2.3.2 Install OpenMM Zephyr application *15*
2.3.3 Contents of the OpenMM Zephyr directory *17*

3 RUNNING A SIMULATION 21

3.1 Launching OpenMM Zephyr 21
3.2 Choosing a Molecule 22
3.2.1 PDB files *23*
3.2.2 Preparing a PDB file for loading into OpenMM Zephyr *23*
3.3 Select Location for Output Files 24
3.3.1 What files are saved for a simulation? *24*
3.4 Parameters 25
3.4.1 Minimizing energy before dynamic simulation *26*
3.4.2 Choosing the length of time of your simulation *26*
3.4.3 Setting the temperature for your simulation *27*

3.4.4 Choosing a force field for your simulation *27*
3.4.5 Choosing the simulation hardware *28*
3.4.6 Solvent collision interval *29*
3.4.7 Live viewing in VMD *29*
3.4.8 Saving trajectory frames *30*
3.5 Starting a Simulation 30
3.6 Restrictions on Simulations in Zephyr 32
3.6.1 Amber96 Force Field *32*
3.6.2 Simulations are in implicit solvent *33*
3.7 VMD Console 33
3.8 Final Simulation Results 34
3.8.1 Saving a trajectory from VMD *34*
3.8.2 Contents of simulation output directory *34*
3.9 Stopping a Simulation 35
3.10 Modifying Parameters Not Shown in the OpenMM Zephyr GUI (advanced) 35
3.11 Continuing a previous simulation 36

4 TROUBLESHOOTING 39

4.1 Why is my simulation updating in VMD so slowly? 39
4.2 I am getting the message "Waiting for IMD socket connection to VMD on port 3000" 39
4.3 The "Simulation complete" message appears but the "Elapsed lab time" does not stop 40
4.4 I cannot see the full Zephyr screen and am unable to resize the window on the Mac. 40

5 GOING FURTHER: RUNNING GROMACS-OPENMM FROM THE COMMAND LINE 41

5.1 Generating Initial Parameter and Simulation Input Files 41
5.2 Preparing for a Command Line mdrun_openmm 42
5.3 Ready to Run 46
5.4 Restarting a Simulation that Finished Running in Zephyr 47
5.5 Visualizing Your Trajectory in VMD 47

6 GIVING BACK: SUBMITTING BUG REPORTS AND FEATURE REQUESTS 49

7 APPENDIX A: GROMACS-OPENMM LIMITATIONS 51

8 APPENDIX B: FFAMBER (AMBER96) PDB NAMING CONVENTIONS 53

9 REFERENCES 63

List of Figures

Figure 2.1: Download VMD from the University of Illinois 5
Figure 2.2: Window for browsing the NVIDIA code samples 7
Figure 2.4: OpenMM Zephyr can be downloaded from Simtk.org 15
Figure 2.5: The Zephyr installer installs some required Microsoft software libraries. Click "Yes" to install these libraries. 16
Figure 2.6: Sometimes the installation of the Microsoft libraries takes a while. Please be patient. 16
Figure 2.7: The installer for OpenMM Zephyr on Mac OS 17
Figure 3.1: Launch Zephyr by double clicking the Zephyr icon for Windows or Mac (left: on Windows; right: on Mac OS) 21
Figure 3.2: OpenMM Zephyr application 22
Figure 3.3: Parameters panel 25
Figure 3.4: View when a simulation is running 31
Figure 3.5: VMD can be used to watch the simulation as it runs 32
Figure 3.6: Click "Unblock" to grant permission for Zephyr and VMD to communicate through TCP sockets 32
Figure 3.7: VMD console 33
Figure 5.1: Suggested parameter settings to generate initial files to run a GROMACS simulation 42
Figure 5.2: Content of testRun directory after initial Zephyr run (files starting with # were omitted). Note the molecule name in this example was "villin." 43
Figure 5.3: Windows users: Paste commands into your Command Prompt window by right-mouse clicking the top blue bar and selecting Edit→ Paste. 43
Figure 6.1: Investigate, submit, or track feature requests and bug reports at Simtk.org 49
Figure 8.1: Standard PDB residue name format for RNA and protein in columns 18-20. Column numbers are shown vertically in black. 53
Figure 8.2: PDB atom name fields are in columns 13-16 54

Figure 8.3: Example of a water molecule PDB line. Delete this line and others like it. 55
Figure 8.4: Example of modifying protein residue names: (top) before editing, (bottom) after editing 56
Figure 8.5: Example of modifying initial N-terminal amino acid residue names: (top) Initial residue HIS 19 before editing. Notice that the initial residue is not always number "1". (bottom) Initial protein residue after editing. Four character residue name spills into column 21..... 56
Figure 8.6: Modification of final protein residue name: (top) before editing, (bottom) after editing.....57
Figure 8.7: Modification of the initial RNA residue: (top) before editing, (bottom) after editing – note the removal of the atoms associated with the phosphate group..... 59
Figure 8.8: Modification of final 3' RNA residue: (top) before editing, (bottom) after editing 60
Figure 8.9: PDB file excerpt from villin.pdb. Green text shows where atom and residue names have been changed. 61

1 Overview

1.1 Introduction to Molecular Dynamics

Researchers are increasingly using molecular dynamics (MD) simulations to model molecular motion and help expand our knowledge and understanding of biology. Examples of how MD simulations have been used include:

- Providing hypotheses for biological phenomena that cannot currently be observed experimentally and guiding new experiments [(Marszalek & al., 1999), (Herce & Garcia, 2007), (Kasson & Pande, 2008)]
- Identifying which engineered molecules would be stable or would bind with another molecule [(Kasson & Pande, 2008)]
- Augmenting information provided by static structures, i.e., from x-ray crystallography [(Glazer, Radmer, & Altman, 2008)]

The potential uses for MD simulations are exciting, but getting started can be overwhelming and time-consuming. OpenMM Zephyr was designed to get researchers running MD simulations quickly.

1.2 What is OpenMM Zephyr?

OpenMM Zephyr is a freely downloadable application with a graphical user interface that allows users to easily run the OpenMM version of GROMACS, a widely used MD package. The OpenMM version speeds up the GROMACS simulations, so that more complex molecules and/or longer processes can be simulated. Read more about OpenMM and how to take full advantage of it in Section 1.3.

OpenMM Zephyr leads the user through the basic steps required to set up and run a simulation, and also to visualize the simulation results using VMD, a popular molecular visualization program. In addition, OpenMM Zephyr displays the specific GROMACS

commands being used, enabling motivated users to teach themselves how to run GROMACS from the command line so that they can eventually use the other options within GROMACS.

1.2.1 Limitations of OpenMM Zephyr

- Supported platforms: Windows XP or Vista, Mac OS 10.5 and 10.6, Linux (with XWindows) Linux version are built and tested on CentOS5.
- 32 bit operating system (a 64-bit version is also available for Linux)
- Implicit solvent (no explicit water molecules)
- Amber96 force field only
- Support for only the 20 standard amino acids and 5 canonical nucleotide residues
- For GPU accelerated dynamics:
  - You need to have a supported GPU and drivers (Sections 1.3.1 and 2.2)

1.3 What is OpenMM?

OpenMM is a freely downloadable, high performance, extensible library that allows molecular dynamics simulations to run on high performance computer architectures, such as computer clusters and graphics processing units (GPUs). This makes it more reasonable to simulate complex molecules and/or longer processes. Performance speed ups of over 100X have been achieved in some cases using OpenMM on GPUs. [(Friedrichs, et al., 2009)]. OpenMM Zephyr 1.0.0 is built using OpenMM Release 1.0.

1.3.1 Running OpenMM on GPUs

Beginning with Preview Release 2, OpenMM enabled fast MD simulations on GPUs. GPUs are computer chips designed for generating computer graphics. Their parallel design makes them also very useful for speeding up a variety of other applications, including MD simulations.

To take advantage of these speed-ups, though, your computer needs to be equipped with one of the supported GPU cards:

Supported NVIDIA GPUs:

http://www.nvidia.com/object/cuda_learn_products.html

Your computer does not need one of these GPU cards to run OpenMM Zephyr, but your simulations will not run as fast without one.

1.3.2 More information about OpenMM

You can learn more about the OpenMM project at http://simtk.org/home/openmm.

1.4 Conventions Used in this Document

This warning icon denotes situations that require special attention or caution.

The clicky pointer icon shows where to click on screen shots

2 Installing OpenMM Zephyr

2.1 Prerequisites

OpenMM Zephyr runs on Windows XP, Windows Vista, Mac OS, and Linux (32-bit and 64-bit).

If you wish to use the visualization capabilities of OpenMM Zephyr, you will need to install VMD (see Section 2.1.1).

To take advantage of GPU accelerated molecular dynamics, you must have a supported GPU (see section 1.3.1). You will also need to have the special programming language(s) used for your particular GPU (see Section 2.2).

2.1.1 Installing VMD (optional but recommended)

OpenMM Zephyr also uses the molecular visualization program VMD. If you do not intend on visualizing the MD simulation results within OpenMM Zephyr, you do not need to install VMD. However, to take full advantage of OpenMM Zephyr, it is recommended that you install VMD.

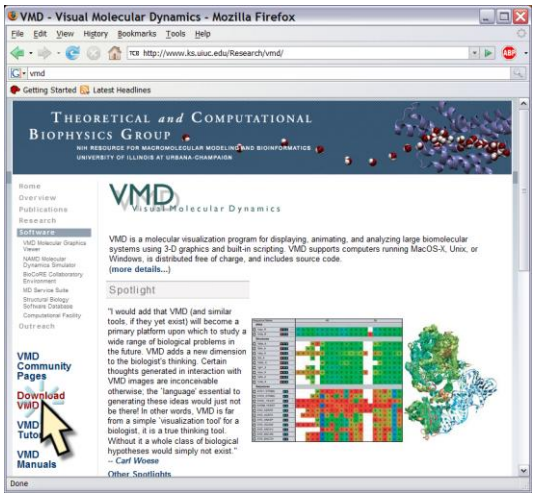


Figure 2.1: Download VMD from the University of Illinois

VMD can be downloaded for free from http://www.ks.uiuc.edu/Research/vmd/. Click on the "Download VMD" link in the left-hand column (Figure 2.1). You will be taken to another webpage, listing all the different versions of VMD. Select the appropriate distribution of the most recent version for your particular platform. Follow the on-line directions to complete the downloading and installation of the program.

2.2 Installing GPU software

If you intend to use GPU accelerated molecular dynamics, you need to install CUDA (for NVIDIA GPUs) or OpenCL (Mac only), and test it before running OpenMM Zephyr.

VMWare, another software for running Windows on a Macintosh, will **not** enable your machine to access the GPU.

2.2.1 Installing CUDA for NVIDIA GPUs

For NVIDIA GPUs, you need to have CUDA version 2.3 or later installed to get the GPU acceleration. It is recommended that you test your installation before trying to run OpenMM and the provided examples.

2.2.1.1 Windows

1. Go to http://www.nvidia.com/object/cuda_get.html

2. Download and install the CUDA Driver, the CUDA Toolkit, and the CUDA SDK code samples. We recommend using version 2.3, but later versions might work for your machine. The driver and toolkit are needed to get the GPU acceleration. The code samples are required for testing purposes.

3. To verify that you've installed things correctly, run a sample program available with the SDK code samples.

    a. Go to Start -> All Programs -> NVIDIA Corporation -> NVIDIA CUDA SDK -> NVIDIA CUDA SDK Browser

b. A window appears showing all the different sample programs you can try running (Figure 2.2).

c. Locate the program "Device Query" and click on the associated "Run" link on the right-hand side. If things are running correctly, a window will appear stating how many devices are running CUDA (there should be at least 1) and that it/they passed the test.

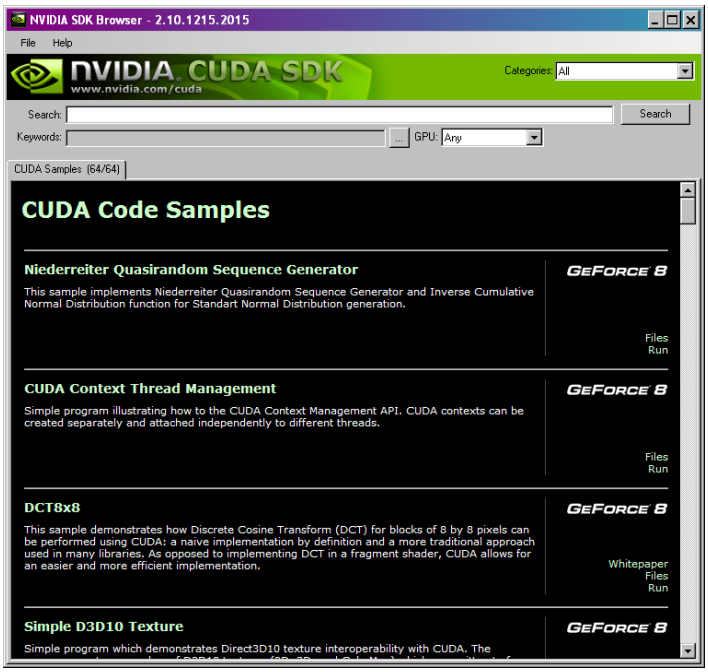


Figure 2.2: Window for browsing the NVIDIA code samples

2.2.1.2 *Mac OS X*

1. Go to http://www.nvidia.com/object/cuda_get.html

2. Download and install the CUDA Toolkit, CUDA Driver, and the CUDA SDK code samples, version 2.3.

   a. Install the toolkit using a *custom install* and make sure all boxes, including the kernel extension are checked. With version 2.2, the driver is included; with later versions, which may work depending on your setup, the driver is a

separate download, and you need to make sure you download and install that as well. The toolkit and driver are needed to get the GPU acceleration. The code samples are required for testing purposes.

b. Open a terminal window. Go to Macintosh HD -> Applications -> Utilities. Click on Terminal.

c. Set your environment variables so that your computer can locate the CUDA programs by typing the following two lines:

```
export PATH=/usr/local/cuda/bin:$PATH

export DYLD_LIBRARY_PATH=/usr/local/cuda/lib:
$DYLD_LIBRARY_PATH
```

This sets the environment variables only for the terminal you are in. To set them permanently, you will need to add them to your bash_profile.

3. To verify that you've installed things correctly, run a sample program available with the SDK code samples.

a. Within the terminal window, navigate to the location of the code samples. If you installed everything in the default directories, then you would type one of the following (depending on the version of SDK that you downloaded):

```
cd /Developer/CUDA
```

OR

```
cd /Developer/GPU Computing/C
```

b. Compile the test programs by typing:

```
make
```

c. Navigate to the location of the compiled programs by typing one of the following (depending on the version of SDK that you downloaded):

```
cd /Developer/CUDA/bin/darwin/release
```

OR

```
cd /Developer/GPU Computing/C/bin/darwin/release
```

d. Run the deviceQuery program:

```
./deviceQuery
```

If things are running correctly, you will see how many devices are running CUDA (there should be at least 1) and a printout saying that it/they passed the test.

Troubleshooting:

If no devices are found, verify that you have a supported GPU card. If you do, re-run the installer and make sure to select a custom installation verifying that all boxes, including the kernel extension, are checked.

If you have multiple GPUs and only one is activated, this may be because of the energy-saving options (this is the case for new MacBook Pros, which ship with a deactivated 9600M GPU). To change the energy-saving options, click System Preferences -> Energy Saver and set the graphics option to "Higher Performance." You will need to log out and then log back in for the new options to take effect.

Additional instructions and troubleshooting tips are provided in the "Getting Started" manual on the CUDA download site.

2.2.1.3 Linux

1. To compile the GPU code on a Linux machine, you will need gcc, version 3.4 or 4.x through 4.2. You can verify the version gcc installed on your system by typing:

   ```
   gcc --version
   ```

2. Go to http://www.nvidia.com/object/cuda_get.html

3. Download and install the CUDA Driver, the CUDA Toolkit and the CUDA SDK code samples, version 2.3. The toolkit and driver are needed to get the GPU acceleration. The code samples are required for testing purposes. We have tested this for the Redhat Enterprise Linux 5.x version (64-bit). Please refer to the CUDA website and "Getting Started" manual for a list of all supported Linux distributions and additional instructions.

   a. Open a terminal window.

   b. If you are running X Windows, you will need to turn it off to install the driver. You can do this by typing as a **superuser**:

      ```
      /sbin/init 3
      ```

   c. Run the CUDA driver installation script as a **superuser**. If you turned off X Windows, you can turn it on again after the installation is complete (try the commands `startx` or `init 5`).

   d. Run the CUDA toolkit installation script as a **superuser**.

   e. Set your environment variables so that your computer can locate the CUDA programs. Or, if you wish to be able to run OpenMM Zephyr by double-clicking the icon through a graphical user interface, you will need to set the

library path for the entire system. Please consult your system administrator before changing system-wide configurations.

For the BASH shell (for your individual account)

1. Set your PATH by typing:

```
export PATH=/usr/local/cuda/bin:$PATH
```

2. Set your library path. Depending on whether you use 32-bit or 64-bit Linux, type one of the following:

   For 32-bit, type (all on one line):

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib:
$LD_LIBRARY_PATH
```

   For 64-bit, type (all on one line):

```
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:
$LD_LIBRARY_PATH
```

These commands set the environment variables **only for the terminal you are in and **only** for your account. To set them permanently, you will need to add it to ~/.bash_profile or ~/.bashrc

For csh or tcsh shells (for your individual account)

1. Set your PATH by typing:

```
setenv PATH ".:/usr/local/cuda/bin:$PATH"
```

2. Set your library path. Depending on whether you use 32-bit or 64-bit Linux, type one of the following:

   For 32-bit, type (all on one line):

```
setenv LD_LIBRARY_PATH "/usr/local/cuda/lib:
$LD_LIBRARY_PATH"
```

   For 64-bit, type (all on one line):

```
setenv LD_LIBRARY_PATH "/usr/local/cuda/lib64:
$LD_LIBRARY_PATH"
```

These commands set the environment variables **only for the terminal you are in and **only** for your account. To set them permanently, you will need to add it to ~/.cshrc (or similar) file.

To set library path system-wide

CONSULT YOUR SYSTEM ADMINISTRATOR BEFORE CONTINUING

1. You will still need to set your PATH as above.

2. Depending on whether you use 32-bit or 64-bit Linux, have your system administrator include one of the following paths in /etc/ld.so.conf (or equivalent type file) in the list of directories:

   For 32-bit: `/usr/local/cuda/lib`

   For 64-bit: `/usr/local/cuda/lib64`

3. Then, type as superuser/root:

   ```
   ldconfig
   ```

f. Run the CUDA SDK installation script as a **regular user**.

4. To verify that you've installed things correctly, run a sample program available with the SDK code samples.

   a. Within the terminal window, navigate to the location to compile the code samples. If you installed everything in the default directories, then you would type (default directory is only valid for version 2.3):

   ```
   cd $HOME/NVIDIA_GPU_Computing_SDK/C
   ```

b. Compile the test programs by typing:

```
make
```

c. Navigate to the location of the compiled programs by typing (directory is only valid for version 2.3):

```
cd $HOME/NVIDIA_GPU_Computing_SDK/C/bin/linux/release
```

d. Run the deviceQuery program:

```
./deviceQuery
```

If things are running correctly, you will see how many devices are running CUDA (there should be at least 1) and a printout saying that it/they passed the test.

Additional instructions and troubleshooting tips are provided in the "Getting Started" manual on the CUDA download site.

2.3 Getting OpenMM Zephyr

2.3.1 Download OpenMM Zephyr

OpenMM Zephyr can be freely downloaded from https://simtk.org/home/zephyr. From the main project page, click on the "Downloads" link in the left-hand column (Figure 2.3).

You will be taken to the Downloads webpage. Click on the link for the Zephyr installation for your platform. If you are not already logged in to Simtk.org, you will be asked to log in (or register). You will then be asked to describe how you plan to use this software. Provide a description and then click "Download Now."

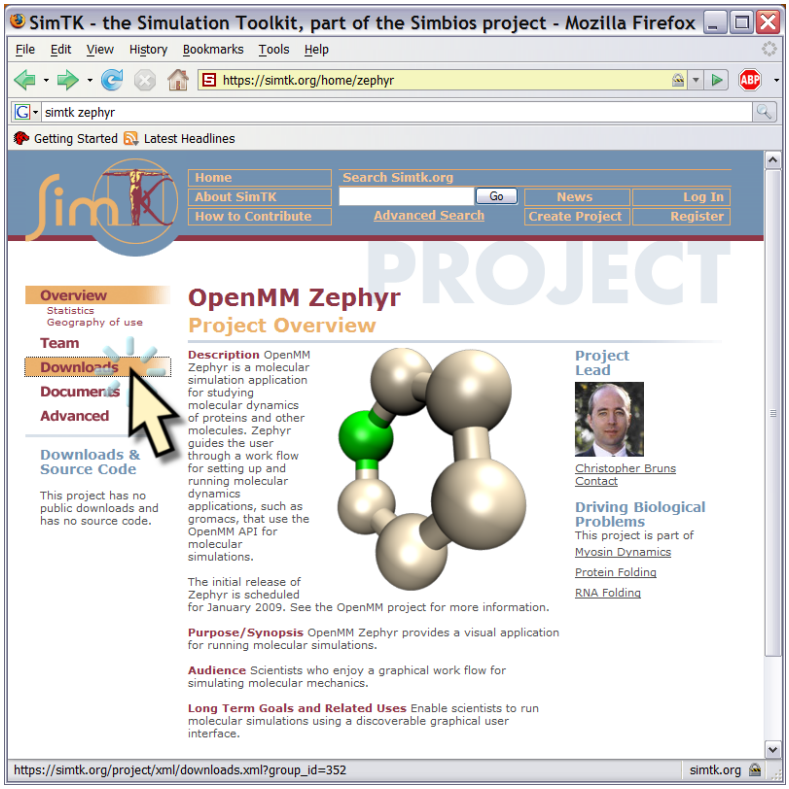


Figure 2.3: OpenMM Zephyr can be downloaded from Simtk.org

2.3.2 Install OpenMM Zephyr application

2.3.2.1 Windows

Save the installation program to your computer and then double click to run the installer.

When you install OpenMM Zephyr, you will be asked where to save all the files. You can save the files anywhere you like on your computer. However, **the path name for the installed files must not contain any spaces.** This is a limitation of the version of GROMACS used as the basis for OpenMM Zephyr. For a Windows computer, a good place to save the files is to C:\Zephyr.

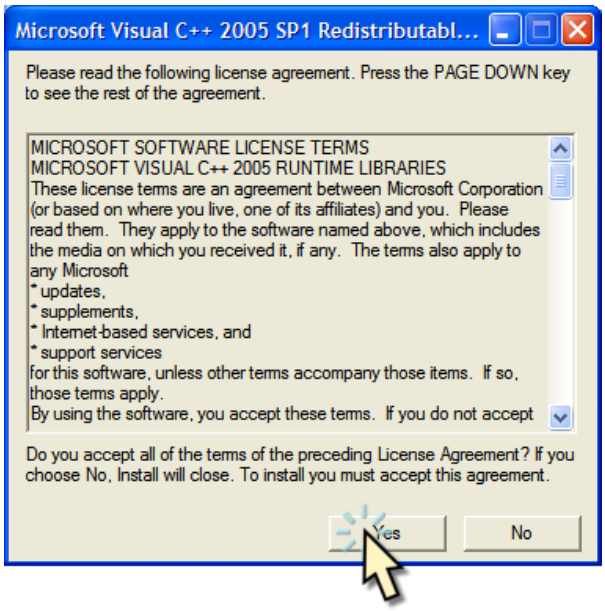


Figure 2.4: The Zephyr installer installs some required Microsoft software libraries. Click "Yes" to install these libraries.

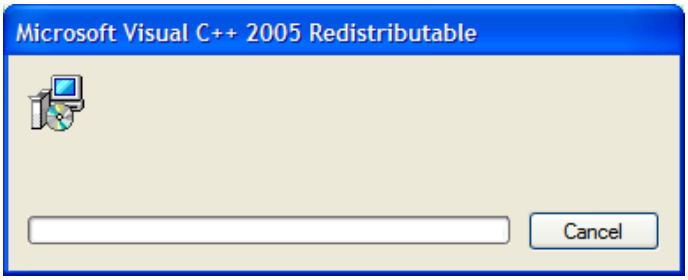


Figure 2.5: Sometimes the installation of the Microsoft libraries takes a while. Please be patient.

Installing Zephyr to "My Documents" will not work. You will be able to extract the files to that location but will not be able to run OpenMM Zephyr from there.

Remember: the path for the OpenMM Zephyr files must not contain any spaces.

2.3.2.2 *Mac OS*

After downloading the dmg installation file, a window should appear instructing you to drag and drop the Zephyr program on the left into your Applications folder (Figure 2.6). If not, locate the dmg file (typically on your Desktop or in your Downloads folder) and double-click on it. Follow the instructions for moving Zephyr into your Applications folder.

This installation also comes with the user manual, ZephyrUserManual.pdf. You should also move this file to a permanent location, e.g., in your Documents folder.

After moving Zephyr and the user manual, you can close the window and delete the .dmg file/eject the disk.

Figure 2.6: The installer for OpenMM Zephyr on Mac OS

2.3.2.3 *Linux*

After downloading the .sh installation script, open a terminal window and run the script. Follow the on-line directions to complete the installation.

2.3.3 Contents of the OpenMM Zephyr directory

If you have successfully installed OpenMM Zephyr, you should see the following files and directories.

2.3.3.1 *Windows*

Go to the directory where you installed OpenMM Zephyr (default location is C:\Zephyr). Within that directory, you should see:

- **Zephyr:** This is the OpenMM Zephyr application. Double-click on this to start the program.
- **Uninstall:** To uninstall OpenMM Zephyr, click on this. The program will be removed from your computer, and all files in the directory except for the testData directory will be deleted.
- **doc:** This directory contains the documentation files
- **testData:** This directory contains example data files to use for starting the example simulations

You will also see the **bin** directory, which provides files used behind the scenes by the OpenMM Zephyr program, and the COPYING text document that provides the licensing details for OpenMM Zephyr.

2.3.3.2 Mac OS

Go to Macintosh HD -> Applications. Press the ctrl key and click on Zephyr to view the contents. Navigate to Contents -> Resources. You should see a **testData** and a **bin** directory, as well as a COPYING text document (details about these are in Windows section, 2.3.3.1). You will also see Zephyr.icns.

2.3.3.3 Linux

Go to the directory where you installed OpenMM Zephyr (default location is /usr/local/zephyr). Within that directory, you should see:

- **zephyr:** This is the OpenMM Zephyr application. You will run this to start the program.
- **doc:** This directory contains the documentation files
- **testData:** This directory contains example data files to use for starting the example simulations

You will also see the **bin** directory, which provides files used behind the scenes by the OpenMM Zephyr program, and the COPYING text document that provides the licensing details for OpenMM Zephyr.

3 Running a Simulation

3.1 Launching OpenMM Zephyr

To start OpenMM Zephyr on Windows or Mac OSX, double-click the Zephyr icon. On Linux, go to the directory where you installed Zephyr (the default location is either $HOME/zephyr/ or /usr/local/zephyr/):

```
cd $HOME/zephyr
```

Then, run zephyr in the background by typing:

```
./zephyr &
```

(You can also run Zephyr by double-clicking on the icon in a Linux environment, but in order to get the GPU acceleration, you will need to ask your system administrator to provide system-wide access to the GPU libraries – see Section 2.2.1.3 for CUDA installation instructions).

The window shown in Figure 3.2 appears.

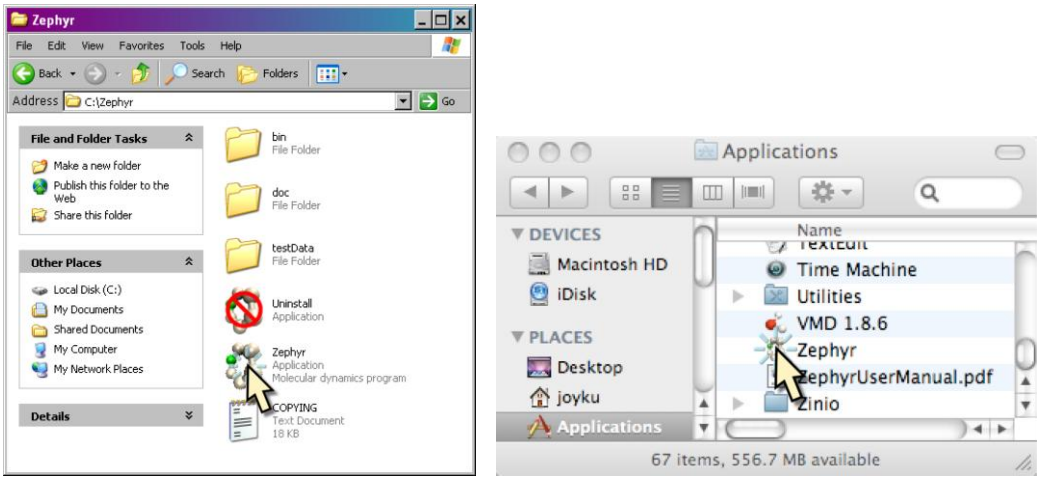


Figure 3.1: Launch Zephyr by double clicking the Zephyr icon for Windows or Mac (left: on Windows; right: on Mac OS)

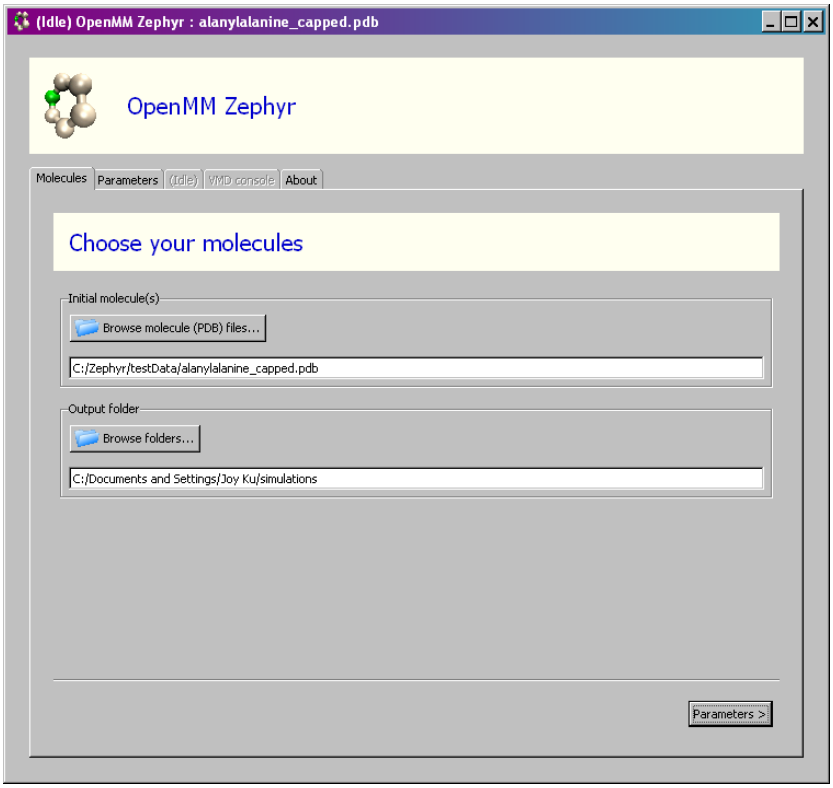


Figure 3.2: OpenMM Zephyr application

Note the series of tabs near the top of the window: Molecules, Parameters, (Idle) (grayed out), VMD console (grayed out), About. The order of the tabs guides you in setting up and running the simulation:

1. Choose a molecule to simulate (the Molecules tab) – See Section 3.2
2. Set the simulation parameters (the Parameters tab) – See Section 3.4
3. Run the simulation (simulation progress shown in the third tab) – See Section 3.5
4. If desired, visualize the simulation results in VMD (the VMD tab) – See Section 3.7.

3.2 Choosing a Molecule

Molecular simulations need an initial molecular structure to start the simulation. OpenMM Zephyr is able to read in structures that are in the Protein Data Bank (PDB) format. See section 3.2.1 for more information about PDB files.

To choose the molecule you wish to simulate, click on the **Molecules** tab. Click the **Browse molecule (PDB) files...** button in the **Initial molecule(s)** panel to select the file describing the molecular structure. You can also type in the file name.

Note: On the Mac OS, Applications/Zephyr.app is recognized as an application, not a folder, so you will not be able to directly access its contents when navigating to it from the Applications folder. This is a problem if you want to access the example PDB files provided by OpenMM Zephyr.

To get to the Zephyr.app sub-folders, go to the Applications folder. Click command–shift–G. A pop-up appears asking for a folder name. Type Zephyr.app. That will take you into the Zephyr.app folder. Navigate to Contents -> Resources -> testData to find example PDB files to use.

3.2.1 PDB files

The Protein Data Bank (PDB) format provides a standard representation for macromolecular structure data derived from X-ray diffraction and NMR studies. This format was created in the 1970s and includes information about the molecule, such as its name, the primary and secondary structure information, and details about how the data was collected.

3.2.1.1 Finding PDB molecular structure files

PDB files can be downloaded from http://www.rcsb.org/pdb/home/home.do, but they might need to be modified to be used by OpenMM Zephyr.

3.2.2 Preparing a PDB file for loading into OpenMM Zephyr

Beginning with Zephyr version 0.9.2, PDB residue and atom names are automatically transformed to be compatible with the ffamber (Amber96) force field used in Zephyr. If you are running an older version of Zephyr, consult Appendix B for details on manually editing your PDB file for use in Zephyr.

Zephyr version 0.9.2 and later performs the following operations on your input PDB file:

- Renames residues and atoms to be compatible with the ffamber (Amber96) force field. This renaming includes special rules for the first and last residues in a chain of protein or RNA.

- Removes water molecules. OpenMM Zephyr uses an implicit solvent model for simulation, and explicit water molecules might cause problems.
- Removes certain atoms that are not supported in the ffamber (Amber96) force field, including phosphate atoms at the 5' end of an RNA molecule.

However, additional modifications may be needed in order to use your PDB file with Zephyr. See Appendix B for full details. Some of the more common changes include:

- Modifying or deleting non-standard amino acids and nucleotide residues
- Filling in missing atoms

Note that hydrogen atoms are automatically added to the molecular structure by Zephyr, so hydrogens do not need to be defined in the PDB file. In fact, if any hydrogen atoms are included in the PDB file, they are first removed and then added back in by Zephyr.

The modified PDB file generated and used by Zephyr for running the simulation is saved in the file <input_file_root_name>_processed.pdb in the output directory (see Section 3.3).

3.3 Select Location for Output Files

To select the location for the simulation output files, click on the **Molecules** tab. Click the **Browse folders...** button in the panel for **Output folder** to select the folder for the output files. You can also type in the name of the folder. A default folder for the simulation output files has been created under *C:/Documents and Settings/<User Name>* on Windows, */Users/<User Name>* on Mac OS, <home directory>/simulations on Linux.

3.3.1 What files are saved for a simulation?

Molecular simulations produce what are called "molecular trajectories" or just "trajectories." By definition, a trajectory is the path of a moving body through space. However, in a molecular simulation, many bodies (atoms) are moving through space, so an intuitive way to capture all the information is to save snapshots of the molecule throughout the simulation. At the end of a simulation, you have information about the molecule (e.g., coordinates of atoms, velocities, etc.) at different time points along the molecular trajectory.

For each simulation, OpenMM Zephyr saves the trajectory information in a .trr file. This is a binary format, so you cannot look at the contents by just opening the file with a text

editor. These .trr trajectory files can be viewed in VMD, but only after loading a structure file (.pdb or .gro) containing the molecule.

A number of other files are also saved automatically with each simulation (See Section 3.8.1). These are used by GROMACS to run the simulation and could be used as a starting point to run simulations with GROMACS parameters not provided via the OpenMM Zephyr interface (See Chapter 5). The updated PDB file that Zephyr generates in order to be compatible with the ffamber (Amber96) force field naming convention (See Section 3.2.2) is also saved.

If you choose to watch your simulation as it runs, the simulation results will be sent to VMD. VMD provides a variety of options for viewing the simulation, and it allows you to save the trajectory information in a variety of file formats (See Section 3.8.1).

3.4 Parameters

A small subset of all the parameters within GROMACS is available through OpenMM Zephyr. These can be set by clicking on the **Parameters** tab. Within the Parameters view, you can also choose the hardware on which you run your simulation (CPU or GPU) and the VMD options for visualizing the simulation as it runs.

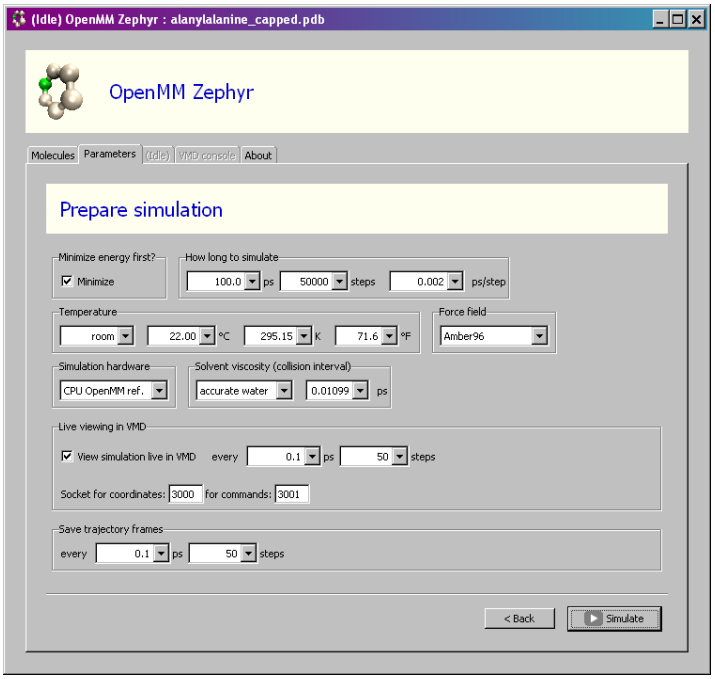


Figure 3.3: Parameters panel

3.4.1 Minimizing energy before dynamic simulation

Sometimes the starting configuration of a molecule in a PDB can have a few close atom contacts or other situation that results in a large calculated potential energy. This situation can cause the simulation to "blow up," in which case the molecular configuration becomes nonsense. It is a good idea to adjust the initial structure by energy minimization to prevent blowing up. This option can be enabled by checking the box for **Minimize** within the **Minimize energy first?** panel within the **Parameters** view. More details about this minimization process can be found by examining the file “testData/em.mdp” and consulting the GROMACS documentation.

3.4.2 Choosing the length of time of your simulation

To set the length of time to run your simulation, click on the **Parameters** tab and use the pull-down menus within the **How long to simulate** panel (located at the top of **Figure 3.3**) You can choose to set the length of time of your simulation (ps = picoseconds = 10^{-12} seconds) or the number of timesteps (steps). The number of timesteps, combined with the timestep interval (ps/step), determines the total simulation time.

The three pull-down menus (ps, steps, and ps/steps) are linked, so changing one will affect the other values. For example, if you have chosen 0.002 ps/step, then running a simulation for 10.0 ps is equivalent to running it for 5000 steps.

3.4.2.1 Guidelines for choosing the total simulation time

When setting up a simulation, you need to decide how long you are willing to wait for a simulation versus what interesting or useful biology can be observed within a given simulation time.

Proteins fold in microseconds to hours. A folding simulation has to calculate new velocities and positions for each atom every couple of femtoseconds. So to compute 1 microsecond (μs) of actual folding requires 500,000,000 timesteps (assuming a timestep of 2 femtoseconds). The computational time to simulate this many timesteps is staggering. Even with GPUs, simulations are able to compute less than one microsecond per day.

Although the largest values in the simulation time pull-down in Zephyr are 10000 ps (10 ns) and 100000 steps, you can create a longer simulation by typing a larger number into the box. Be aware that such simulations may take a long time, even on a GPU.

Some factors that affect the time needed to simulate one timestep (and thus the total simulation time) include:

- Molecule size: A larger molecule takes longer to simulate than a smaller one
- Hardware: GPU simulations should be faster than CPU simulations
- Frequency of updates to VMD and trajectory file: Sending coordinates less often to VMD and the output trajectory file can result in a faster simulation.

3.4.2.2 Choosing the timestep size

The timestep size needs to be smaller than the fastest phenomenon being simulated in order to accurately simulate it. For molecules, this means the timestep is on the order of a femtosecond (10^{-15} seconds or 10^{-3} picoseconds) to account for the fast bond stretch vibrations in the system. However, the smaller the timestep is, the longer the simulation will take to run. If the time step size is too large, the simulation may become unstable and the molecular structure can diverge to an unreasonable configuration. The technical term for this phenomenon is "blowing up."

3.4.3 Setting the temperature for your simulation

The **Temperature** panel within the **Parameters** view allows you to set the temperature of your simulation. You can set the temperature in either Celsius (^{o}C), Kelvin (K), or Fahrenheit (^{o}F) by typing in the value or using the pull-down menus. The left-most pull-down menu within the **Temperature** panel allows you to select from some commonly used temperatures (e.g., freezing temperature, boiling temperature, human body temperature).

3.4.4 Choosing a force field for your simulation

A force field determines how the particles within your simulation will interact. More specifically, it refers to the equations and parameters that describe the potential energy of a

system of particles as a function of the particles' positions. There are many different force fields that have been developed for molecular dynamics simulations. They are empirical models, attempts to reproduce what has been observed experimentally, and include approximations to simplify the calculations.

The current version of OpenMM Zephyr includes the Amber96 force field. This is a classical force field, meaning it treats each atom as a single particle and models their behavior using Newton's equation of motion (F=ma, where F=force, m=mass, a=acceleration). The forces are specified via potential energy functions that define the interactions between atoms (i.e., bonds, Van der Waals, electrostatic, etc.). These potential energy functions are empirically derived and are what distinguish one force field from another. You can read more about the Amber force fields in (Ponder & Case, 2003) and (Case, et al, 2005).

The force field is set using the pull-down menu in the **Force field** panel within the **Parameters** view.

3.4.5 Choosing the simulation hardware

Use the pull-down menu for **Simulation hardware** within the **Parameters** view to choose what hardware to use to run your simulation:

CPU OpenMM ref.	Simulation runs on your CPU using the OpenMM reference code. Don't expect this mode to be fast. The OpenMM CPU reference code is written for expository pedagogical purposes at the expense of efficiency.
CPU gromacs	USE FOR TESTING ONLY - This uses standard GROMACS simulation without OpenMM. Beware that this version of GROMACS ignores implicit solvent information, so extreme caution is advised.

GPU nVidia	Simulation runs on your NVIDIA GPU card. If you do not actually have an NVIDIA GPU card, the simulation defaults to the CPU OpenMM ref. option. If you have an NVIDIA GPU card but have not correctly installed the GPU software (see Section 2.2), the simulation will start and then immediately end.

Use the CPU gromacs option for testing purposes only. OpenMM Zephyr currently only supports implicit solvent simulations, but the version of GROMACS used by Zephyr ignores implicit solvent information, so the simulation results may not make any sense.

3.4.6 Solvent collision interval

The solvent collision interval specifies the average time it takes a solvent molecule to collide with another solvent molecule (in ps). For instance, the experimental value for water is 0.01099 ps. This number provides a measure of the solvent viscosity. A higher value means that there will be longer time gaps between solvent collisions, which means you have a less viscous solvent. A molecule in a less viscous solvent can move more easily and thus will fold faster (Zagrovic & Pande, 2003).

To set this value, use the pull-down menus in the **Solvent collision interval** panel within the **Parameters** view. Currently, you can choose between an "accurate water" collision interval of 0.01099 ps and the "faster folding" time interval of 1.0 ps. This setting corresponds to the tau_t parameter setting in GROMACS' parameters file (see Chapter 5).

3.4.7 Live viewing in VMD

You can watch the simulation as it is running by checking the box for **View simulation live in VMD** within the **Parameters** view. Set the frequency at which VMD

is updated with the simulation output using the pull-down menu for **ps** or **steps** within the **Live viewing in VMD** panel.

If the simulation appears jerky and/or updates very slowly, try increasing the frequency at which information is sent to VMD. However, increasing the frequency (decreasing the number of timesteps or the time period) will also cause your simulation to run slower.

Viewing your simulation in VMD will slow it down. You can minimize the impact of the live viewing by increasing the time period (or the number of steps) for sending information over to VMD.

Within this panel, you can also specify the sockets to use to connect OpenMM Zephyr to VMD. One socket is used for sending coordinates from the simulations to VMD and another is used for sending commands to VMD. The default values are 3000 for the coordinates socket and 3001 for the command socket. You will need to change these values if another application (e.g., another instance of OpenMM Zephyr) is already using these to communicate with VMD.

3.4.8 Saving trajectory frames

Choose the frequency at which the trajectory frames from your simulation are saved by using the pull-down menus for **ps** or **steps** in the **Save trajectory frames** panel within the **Parameters** view. Be aware that saving frames frequently can slow down the simulation, especially with GPU accelerated simulations.

3.5 Starting a Simulation

Once you have specified the molecule you wish to simulate and chosen the simulation parameters, you can start the simulation by clicking the **Simulate** button in the lower-right corner of the **Parameters** view. OpenMM Zephyr will automatically switch to the **Status** view (third tab). This view provides information about how much of the simulation has been completed. It also shows you the output from GROMACS, the underlying molecular dynamics simulation, in black in the **gromacs output** panel. Items in blue in this panel are either commands sent from Zephyr to GROMACS or status messages.

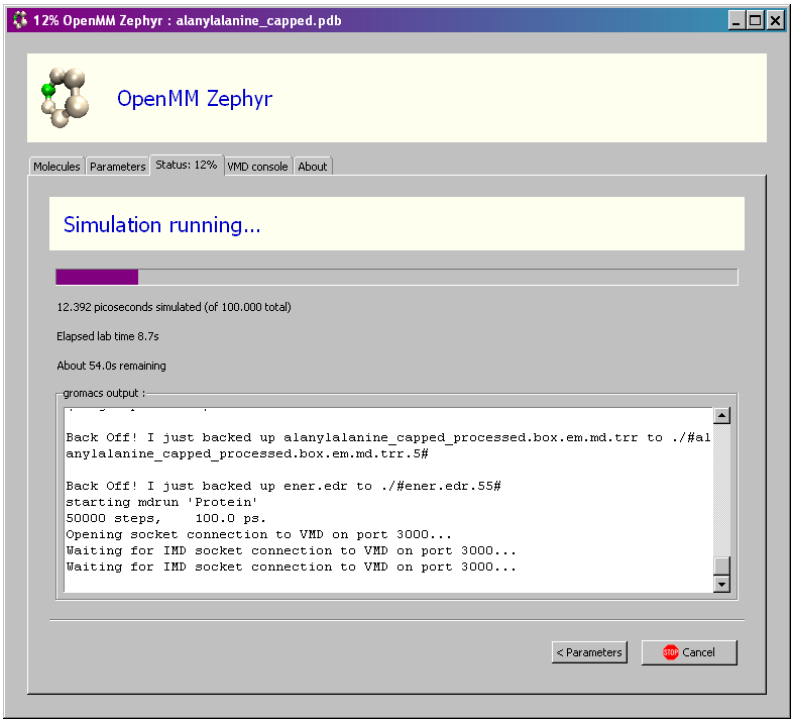


Figure 3.4: View when a simulation is running

If you chose to view the simulation as it was running (See Section 3.4.6), two VMD windows should appear, one of which shows the molecule (Figure 3.5). You may get security alert message(s), as shown in Figure 3.6. If so, just click **Unblock** to allow OpenMM Zephyr to communicate with VMD.

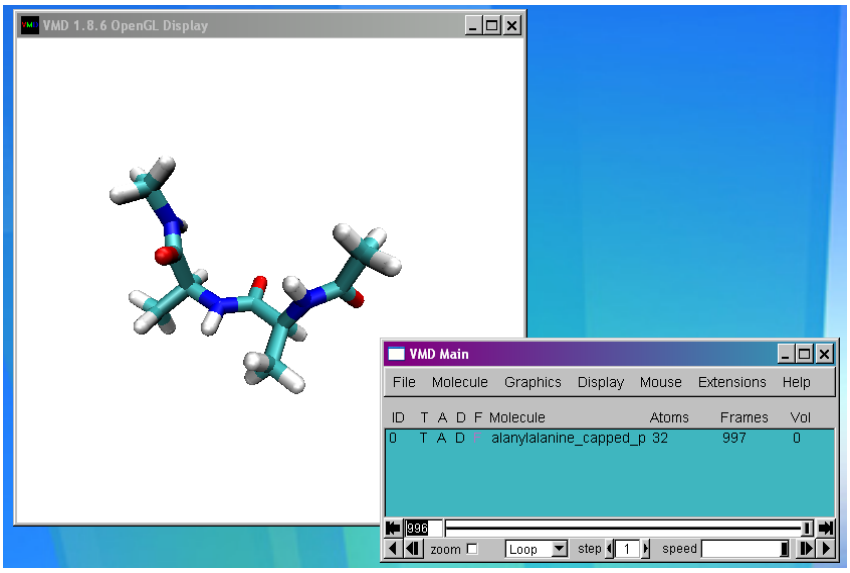


Figure 3.5: VMD can be used to watch the simulation as it runs

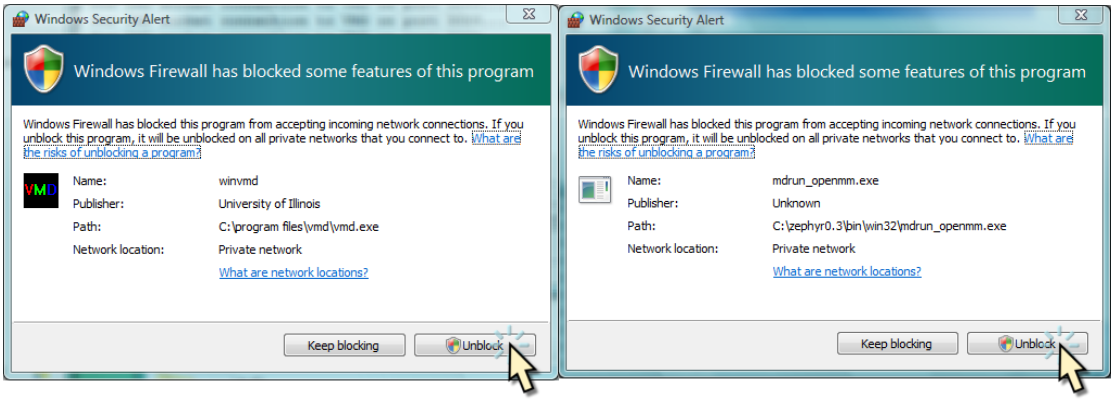


Figure 3.6: Click "Unblock" to grant permission for Zephyr and VMD to communicate through TCP sockets

3.6 Restrictions on Simulations in Zephyr

3.6.1 Amber96 Force Field

Only one force field type is available from the Zephyr interface, Amber96. Use of other force fields may be possible from the command line (see Chapter 5), and may require different conventions for the naming of atom and residue types.

3.6.2 Simulations are in implicit solvent

OpenMM Zephyr runs simulations using an implicit solvent model. This means that water molecules are not explicitly modeled (that would be the explicit solvent model). Instead, they are included in the simulation as a continuous medium.

Explicit solvent simulations are possible using the command line tools available with the original version of GROMACS; in fact, the currently released version of GROMACS (4.0) *cannot* use implicit solvent. On the other hand, the OpenMM version of GROMACS released with OpenMM Preview release 2 cannot run explicit solvent simulations.

3.7 VMD Console

The VMD console tab provides an interface between OpenMM Zephyr and VMD. VMD commands sent by OpenMM Zephyr (shown in blue), as well as VMD output (shown in black) are displayed. There is also a field at the top (**Type VMD commands here**) to type in commands and control VMD via the command line. See the VMD User's Guide for more information: http://www.ks.uiuc.edu/Research/vmd/current/ug/.

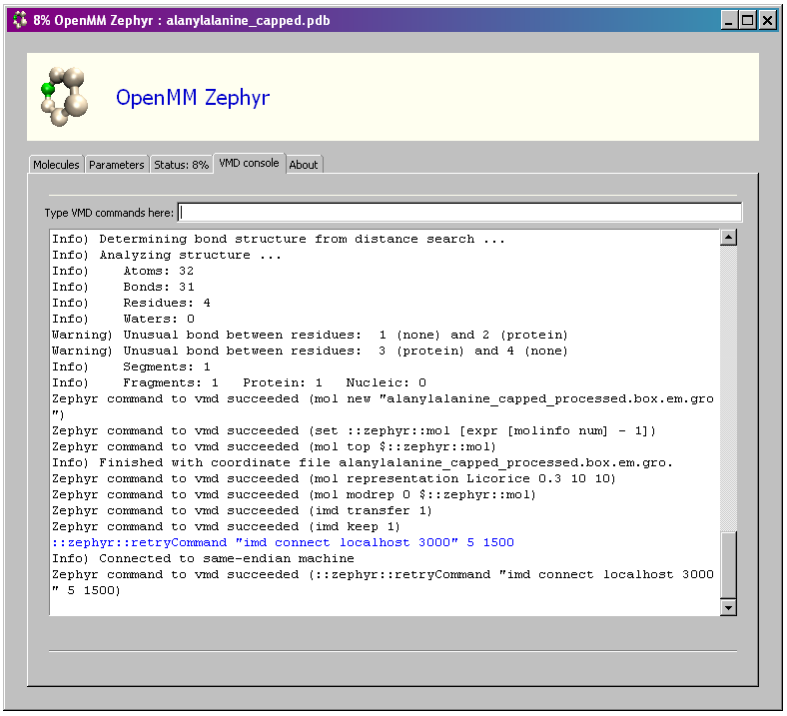


Figure 3.7: VMD console

3.8 Final Simulation Results

The trajectory produced by the simulation is saved by OpenMM Zephyr and can be loaded into VMD for viewing and additional processing (See Section 3.8.1). OpenMM Zephyr also saves other intermediate files used by the simulation (See Section 3.8.2).

3.8.1 Saving a trajectory from VMD

Once your trajectory is loaded into VMD, you can save trajectory and structure files in many file formats from within VMD. These and many other rich features of the VMD program are documented in the current version of the VMD User's Guide, available at http://www.ks.uiuc.edu/Research/vmd/current/ug/.

3.8.2 Contents of simulation output directory

GROMACS-OpenMM, upon which OpenMM Zephyr is built, produces a number of files during the simulation. These are like the standard files produced by GROMACS but they cannot be used with GROMACS, only with the OpenMM version of GROMACS. A partial list of these files is given below.

For a more complete description of the files, refer to the documentation for GROMACS 4.0 (http://www.gromacs.org/component/option,com_wrapper/Itemid,192/). The links to the Flow Chart and to Getting Started -> GROMACS files may be particularly useful.

md.out.(mdp) This is an MDP file (the mdp extension may or may not appear, depending on your browser settings). It is the input parameters file for the simulation, containing information such as how long to run the simulation, the timestep, and the temperature for the simulation.

traj.trr This file contains the trajectory of the simulation. It will include the coordinates and velocities. This is in a binary format, so it cannot be read using a regular text editor but it can be viewed using VMD.

md(.log) This text file is the GROMACS-OpenMM log file for the simulation. The information is what GROMACS outputs, which is a subset of the text printed in the **gromacs output** panel of the **Status** view.

.tpr file This file contains the starting structure of your simulation, the molecular topology, and all the simulation parameters. It is generated by several pre-processing steps and is in a binary format, so it cannot be read using a regular text editor.

OpenMM Zephyr also outputs **zephyr.log**, a text file that captures the output shown in the **gromacs output** panel of the **Status** view, including the Zephyr commands used to interface with GROMACS.

3.8.2.1 Cleaning up when the output directory gets too full

After you have run a very large number of simulations in the same directory, your simulations may stop working. Try removing backup files in the directory, i.e., those with names beginning with a '#' character. To avoid this problem (and keep better track of your simulations), you may want to use a separate directory for each new simulation.

3.9 Stopping a Simulation

Press the **Cancel** button in the **Status** panel to halt a running simulation.

3.10 Modifying Parameters Not Shown in the OpenMM Zephyr GUI (advanced)

OpenMM Zephyr uses a modified version of GROMACS (GROMACS-OpenMM) to run molecular simulations. Power users who are familiar with GROMACS may be able to run a wider variety of simulations using the command line. See Chapter 5 for a brief tutorial on how to run GROMACS-OpenMM from the command line. The GROMACS user documentation at http://www.gromacs.org/ may also be helpful.

Many of the parameters that can be modified from the Zephyr GUI are available from the “Advance Parameters…” button within the Zephyr Parameters tab.

3.11 Continuing a previous simulation

It is possible to pick up where a previous trajectory left off in OpenMM Zephyr. This feature is new in Zephyr version 1.0. Within the Molecules tab is a section called "Continue a previous run". Select a trajectory file (.trr) from your previous run and press the "Continue simulation" button to continue the run.

3.11.1.1 Extending your current simulation

To continue a successfully completed run in the same Zephyr session:

1. Open the Molecules tab
2. Verify that the latest trajectory file is listed in the "Continue a previous run" area. (The correct file should be there automatically)
3. Verify that the "Auto load mdp params" box is checked.
4. Enter the number of picoseconds you wish to add to the simulation in the "Extend time by" box.
5. Press "Continue simulation". This will create a new trajectory file, which picks up where the previous trajectory left off.

3.11.1.2 Extending a simulation with changes

If you want to change parameters in the continued simulation, the procedure is a bit more complicated. You can, for example, enable or disable VMD visualization, or change the viscosity of the simulation before continuing.

1. Open the Molecules tab
2. Verify that the latest trajectory file is listed in the "Continue a previous run" area.
3. Press "Load mdp now". This will load the simulation parameters associated with the previous trajectory. The "Auto load mdp params" box should now be automatically UNchecked.
4. Open the Parameters tab and adjust the parameters to your taste. Return to the Molecules tab.
5. Enter the number of picoseconds you wish to add to the simulation in the "Extend time by" box.

6. Press “Continue simulation”. This will create a new trajectory file.

3.11.1.3 Extending a simulation from a previous Zephyr session

1. Open the Molecules tab
2. Press the “…” button under the “Continue a previous run” section. This launches a file browser. Load the trajectory file (.trr) that you want to continue from.
3. Verify that the “Auto load mdp params” box is checked.
4. Enter the number of picoseconds you wish to add to the simulation in the “Extend time by” box.
5. Press “Continue simulation”. This will create a new trajectory file, which picks up where the previous trajectory left off.

3.11.1.4 Continuing a cancelled or crashed trajectory

(This does not work well yet)

This method can also be used to continue a run that has been terminated.

1. Open the Molecules tab
2. Press the “…” button under the “Continue a previous run” section. This launches a file browser. Load the trajectory file (.trr) that you want to continue from.
3. Press “Load mdp now”. This will load the simulation parameters associated with the previous trajectory. The “Auto load mdp params” box should now be automatically UNchecked.
4. Set “Extend time by” to a small value, such as 1.0 picoseconds.
5. On the “Parameters” tab, set How long to simulate->”steps” to a small number, such as 1. Return to the Molecules tab.
6. Press “Continue simulation”. This will create a new trajectory file, which picks up where the previous trajectory left off.

4 Troubleshooting

4.1 Why is my simulation updating in VMD so slowly?

If the VMD view of the simulation is changing very slowly, try decreasing the interval for VMD updates, and restarting the simulation. This will result in a smoother display in the VMD window. This may however also result in a longer running time.

4.2 I am getting the message "Waiting for IMD socket connection to VMD on port 3000"

Click to the VMD console view to see what is happening with VMD. Depending on the message you see, you will need to take different actions:

VMD console message/issue	**Action**
couldn't open socket: address already in use	Cancel the simulation by clicking the **Cancel** button in the **Status** view. Close the VMD windows that appeared. Click the **Parameters** tab. Change the values of **Socket for coordinates** and **for commands** (See Section 3.4.6)
VMD appears to have hung	Scroll up through the list of VMD commands and outputs, and locate the command "imd connect…" Copy and paste the command in quotes in the field for VMD commands.

4.3 The "Simulation complete" message appears but the "Elapsed lab time" does not stop

The simulation is completed when the message appears. However, there are other tasks, such as writing out the files, that are being performed and are included in the calculation of the **Elapsed lab time**. So the **Elapsed lab time** will stop when all tasks are done.

4.4 I cannot see the full Zephyr screen and am unable to resize the window on the Mac.

Due to screen resolutions, the Zephyr window may extend beyond the bottom of your screen upon launching. On Macintosh computers, this presents a problem since the location of the resize function is in the lower-right corner of the window and is not accessible. To make the Zephyr window the size you want it to be, first click the button to maximize the window. Then you will be able to resize it to the desired size.

5 Going Further: Running GROMACS-OpenMM from the Command Line

5.1 Generating Initial Parameter and Simulation Input Files

You can use OpenMM Zephyr to generate initial parameter files for your simulation. You can then modify these files to run simulations using parameters that are not accessible through the OpenMM Zephyr GUI or to continue a simulation you started with Zephyr.

1) Create a new folder named "testRun" to save the results of your command line Gromacs-OpenMM run. Be sure to remember the full path to testRun. You will need it later.
2) Locate your Zephyr install (on Windows, this is typically C:\Zephyr; on Mac OS, this is typically in the Applications folder (/Applications))
3) Launch Zephyr
    a) Select the **Initial molecule** of your choice
    b) Set the **Output folder** to the testRun directory created earlier
    c) Click on **Parameters**. You will run Zephyr to get initial parameter and simulation input files and to minimize the energy of your structure before the simulation. Suggested initial parameters are as follows (see Figure 5.1 below) :
        a. Set the length of the simulation time to 0.1 ps
        b. Make sure the **Minimize energy first?** box is checked
        c. Uncheck **View simulation live in VMD**
    d) Select the **Simulate** button

The run should finish very quickly.

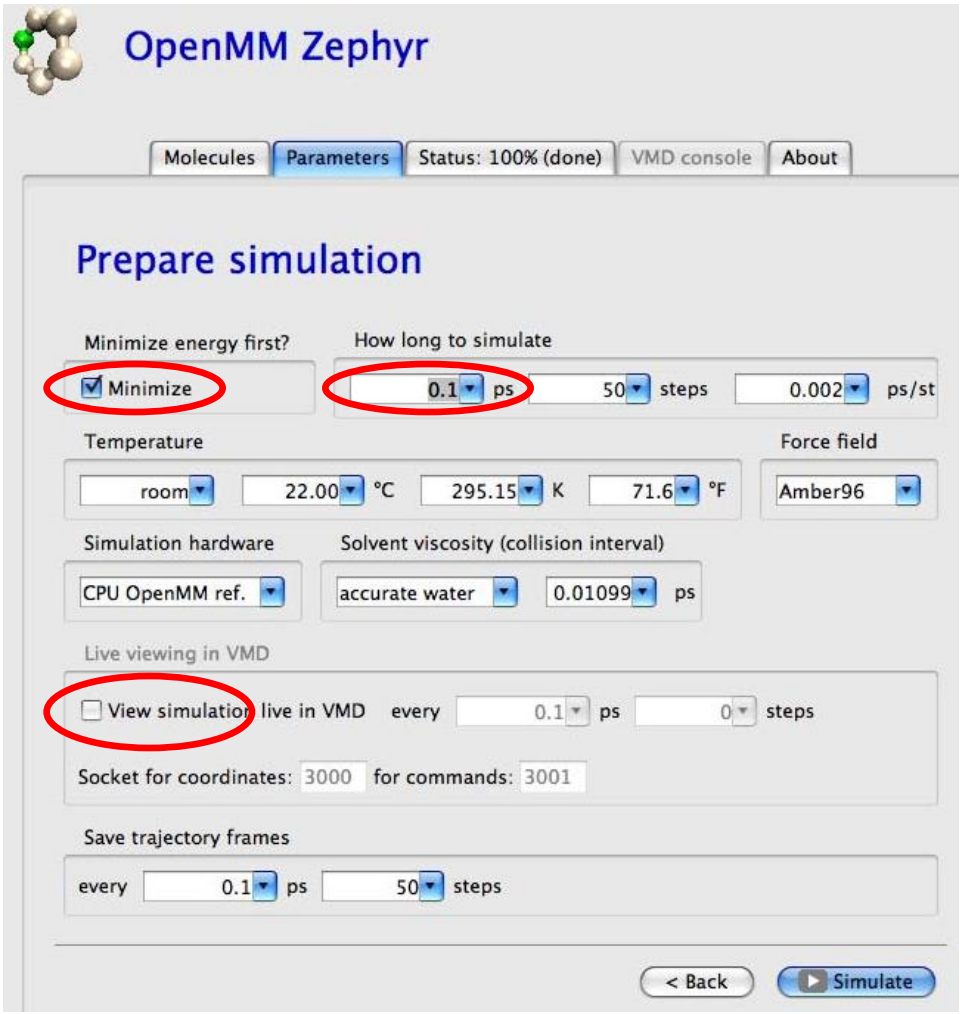


Figure 5.1: Suggested parameter settings to generate initial files to run a GROMACS simulation

5.2 Preparing for a Command Line mdrun_openmm

mdrun_openmm is the command to actually start the simulation. There are a few more steps that you need to take before you are ready to start the simulation.

1) Verify the content of your testRun folder. Zephyr should have added the files shown below in Figure 5-2 to your folder (files starting with '#' are not shown). Note the molecule name in the example below was "villin."

2) Locate the following files:
    a. md.out.mdp → this is the parameter file (it may be listed as md.out if your settings are set to not show extensions)

b. villin_processed.box.em.gro [with "villin" replaced by your molecule name]
c. villin_processed.top [with "villin" replaced by your molecule name]

```
em.out.mdp
ener.edr
md.log
md.out.mdp
posre.itp
state.cpt
traj.trr
villin_processed.box.em.gro
villin_processed.box.em.md.gro
villin_processed.box.em.md.mdp
villin_processed.box.em.md.tpr
villin_processed.box.em.md.trr
villin_processed.box.em.tpr
villin_processed.box.gro
villin_processed.gro
villin_processed.top
zephyr.log
```

Figure 5.2: Content of testRun directory after initial Zephyr run (files starting with # were omitted). Note the molecule name in this example was "villin."

3) Edit the `md.out.mdp` file to change parameters. Save the edited file to a new name, e.g., `villinMod.mdp`.

   You definitely want to set *nsteps* to at least 10000 to get a trajectory (.trr file) with a few frames. Consult the GROMACS documentation (http://www.gromacs.org) for details on other variables. Note that only some of the options have been implemented.

   To run a GROMACS simulation, you will first have to run `grompp` on your modified .mdp file and then `mdrun_openmm` (details in Section 0 below).

4) Get the terminal window, environment variables and paths ready

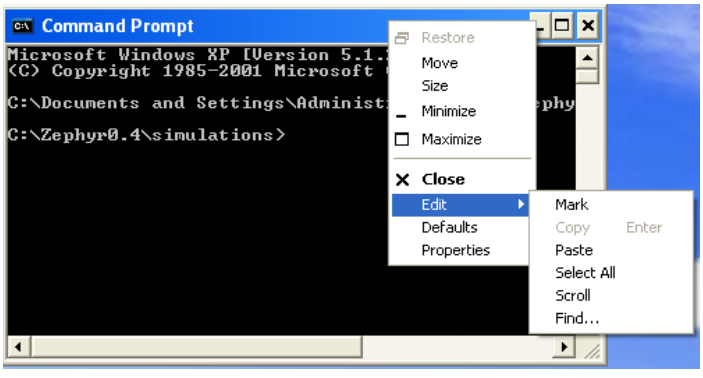


Figure 5.3: Windows users: Paste commands into your Command Prompt window by right-mouse clicking the top blue bar and selecting Edit→ Paste.

Instructions for Windows

Open a command prompt by selecting

Start → All Programs → Accessories → Command Prompt

`cd` to the `testRun` directory created earlier that has the Zephyr output files.

Set environment and PATH variables so that the program knows where to find the program files it needs. See Figure 5.3 for copy/paste instructions into the command prompt window.

```
set GMXLIB=c:\Zephyr\bin\gmxlib

PATH=c:\Zephyr\bin\win32;%PATH%
```

Type `set` (to verify that GMXLIB is defined correctly)

Type `path` (to verify that your path contains `c:\Zephyr\bin\win32`)

To set these permanently do the following:

Go to Start → Control Panel → System → Advanced → Environment variables
Under "User Variables for xxx" select "New"

Set Variable Name: GMXLIB
Set Variable Value: C:\Zephyr\bin\gmxlib

For GPU acceleration, you need to set and verify the `OPENMM_PLUGIN_DIR` following the same directions above as for setting and verifying `GMXLIB`.

To execute on an **NVIDIA GPU**, add:

```
set OPENMM_PLUGIN_DIR=c:\Zephyr\bin\win32\lib\openmm\nvidia
```

Instructions for Mac OS

Open a terminal window

`cd` to the `testRun` directory created earlier that has the Zephyr output files

Set environment and PATH variables so that the program knows where to find the program files it needs.

```
export GMXLIB=/Applications/Zephyr.app/Contents/Resources/bin/gmxlib

export
PATH=$PATH:/Applications/Zephyr.app/Contents/Resources/bin/mac

export DYLD_LIBRARY_PATH=/Applications/Zephyr.app/Contents/Resources
/bin/mac:$DYLD_LIBRARY_PATH
```

Type `printenv` (to verify variables and path settings)

To set these variables permanently, do the following:

```
echo 'export GMXLIB=/Applications/Zephyr.app/Contents/Resources/bin/
gmxlib' >> ~/.bash_profile

echo 'export PATH=$PATH:/Applications/Zephyr.app/Contents/Resources/
bin/mac' >> ~/.bash_profile

echo 'export DYLD_LIBRARY_PATH=/Applications/Zephyr.app/Contents/Res
ources/bin/mac' >> ~/.bash_profile
```

For GPU acceleration, you need to set and verify the `OPENMM_PLUGIN_DIR` following the same directions above as for setting and verifying GMXLIB.

To execute on an **NVIDIA GPU**, add:

```
export OPENMM_PLUGIN_DIR=/Applications/Zephyr.app/Contents/Resources
/bin/mac/lib/openmm/nvidia
```

5.3 Ready to Run

Once your paths are set, you are ready to run GROMACS from the command line. The first step is to run `grompp`, the GROMACS preprocessor that will produce a .tpr file, which has all the information for your simulation.

1) Run `grompp`. For more information, consult http://www.gromacs.org/ and search for `grompp`.

   - Paste the command below (after replacing "villin" with your molecule name) in your command prompt window (see also SUGGESTION below):

```
grompp -f villinMod.mdp -c villin_processed.box.em.gro -p
villin_processed.top -o villinMod.tpr
```

   Replace `villinMod.mdp` with whatever file name you used for your edited `md.out.mdp` file.

SUGGESTION: Copy the commands into a text file. Replace "villin" with your molecule name, and then copy and paste the commands from there. Note for Window users: see Figure 5.3 for copy/paste commands; you can also copy the above command into a ".bat" file (e.g., gromppRun.bat) and simply type gromppRun.bat into the command prompt window.

2) You are now ready to run `mdrun_openmm` and start your simulation.

   - Paste the command below into the command prompt window (after replacing "villin" with your molecule name) or follow the "shortcut" instructions below:

```
mdrun_openmm -s villinMod.tpr -o villinMod.trr -c villinMod.gro
```

   This creates a new .trr (trajectory) file.

   - Shortcut:
     Note that "`villinMod`" is repeated many times in the above command. To avoid re-typing this "prefix," GROMACS allows you to execute the shorter equivalent command:

```
mdrun_openmm -deffnm villinMod
```

5.4 Restarting a Simulation that Finished Running in Zephyr

To restart a simulation that terminated in Zephyr, follow similar steps as before. Note: you can skip steps 1 & 2 if you are continuing with the same parameters and VMD was off during that run.

1) Edit the `md.out.mdp` file and save the edits to a new file `villinRestart.mdp`

   This is only needed if you need to make changes to the parameter file, for example turning off VMD. You will also typically want to increase the total number of steps in the simulation, but you can also do this in step 3. If *nsteps* was 10000 and you would like to perform another 5000 steps, you would set *nsteps* to 15000 either in your .mdp file or in step 3.

2) Run `grompp` as before:

```
grompp -f villinRestart.mdp -c villin_processed.box.em.gro -p
villin_processed.top -o villintemp.tpr
```

3) Run `tpbconv`, which will modify the .tpr file (input is `villintemp.tpr`; output is `villinRestart.tpr`) to continue the simulation from the last frame of the .trr file:

```
tpbconv -s villintemp.tpr -f villin_processed.box.em.md.trr
 -nsteps 15000 -o villinRestart.tpr
```

4) Run `mdrun_openmm` (we recommend appending "part2" to the name of the new output trajectory):

```
mdrun_openmm -s villinRestart.tpr -o villin.part2.trr
```

5.5 Visualizing Your Trajectory in VMD

After the completion of mdrun_openmm, you can visualize the resulting trajectory in VMD using the .trr and .gro files as follows:

1) Start VMD:

   Windows: **Start -> All Programs -> University of Illinois -> VMD -> VMD X.X.X** (where X.X.X. refers to the version number)

   Mac: go to your Applications folder and double click **VMD**

2) In the VMD Main window, select **File → New Molecule**

   a. Enter **Filename**.
      Browse to the testRun directory and select villin_processed.box.em.md.trr, where "villin" is replaced by your molecule name.

   b. Hit **Load. Do NOT close the window.**

   c. Enter **Filename**.
      Browse to the testRun directory and select villin_processed.box.em.gro, where "villin" is replaced by your molecule name.
   d. Hit **Load**

You can now "play back" your trajectory.

See http://www.ks.uiuc.edu/Research/vmd/current/docs.html#tutorials for a complete set of options in VMD.

6 Giving Back: Submitting Bug Reports and Feature Requests

Feedback about OpenMM Zephyr is very much appreciated. Please submit bug reports and feature requests on-line at http://simtk.org/home/zephyr. Click on Advanced -> Features & Bugs -> Submit New.

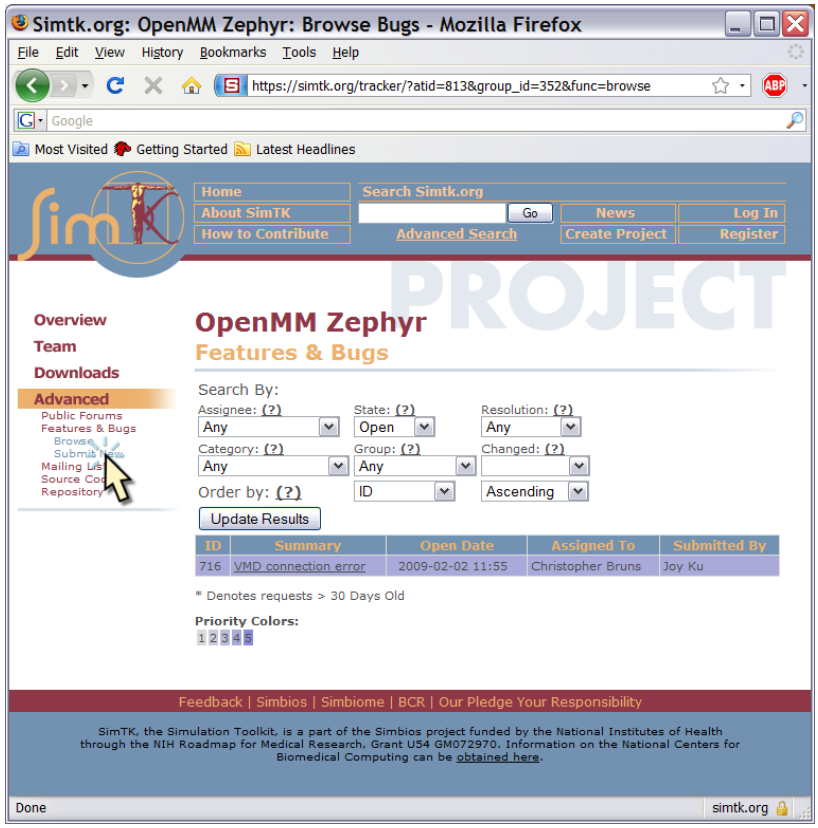


Figure 6.1: Investigate, submit, or track feature requests and bug reports at Simtk.org

7 Appendix A: GROMACS-OpenMM Limitations

Only a fraction of the features of Gromacs 4 are supported in this release of Gromacs-OpenMM. Here are some of the most significant limitations to the allowed options in the MDP file.

The only supported integrators are "md", "sd", and "bd".

The only supported settings for the "constraints" options are "none", "hbonds", or "all-bonds". Constraints are always enforced with SHAKE; the "constraint_algorithm" option is ignored.

The only supported "coulombtype" is "Reaction-Field". If you specify any other value, then no cut-offs will be applied.

The only supported value for the "tcoupl" and "pcoupl" settings is "no". Temperature coupling can be achieved by using the "sd" or "bd" integrator. Pressure coupling is not supported.

Gromacs-OpenMM supports implicit solvent, even though this is not available in the standard Gromacs 4. Implicit solvent can be added to a simulation by including the following three lines in the MDP file.

implicit_solvent = GBSA
gb_algorithm = OBC
gb_epsilon_solvent = 78.3

By default, implicit solvent only works with the AMBER99 force field. To use it with other force fields, you must add the appropriate atom types to the params.agb file.

Checkpoint files produced during a simulation are not supported.

8 Appendix B: ffamber (Amber96) PDB Naming Conventions

Versions of Zephyr prior to version 0.9.2 require you to modify your input PDB file. In order to be compatible with the force field used in the OpenMM Zephyr simulations (GROMACS AMBER-96), you will need to edit the PDB file. See http://chemistry.csulb.edu/ffamber/ for complete details.

Note that hydrogen atoms are automatically added to the molecular structure by Zephyr, so hydrogens do not need to be defined in the PDB file. In fact, if any hydrogen atoms are included in the PDB file, they are first removed and then added back in by Zephyr.

Use a text editor (*not* Microsoft Word) or write a script to make the changes to residue and atom names, described below.

8.1.1.1 PDB residue names

Residue names are three characters long (including spaces) and are found in columns 18 to 20 of PDB "ATOM" and "HETATM" lines. More details of the PDB file format specification can be found at http://www.wwpdb.org/documentation/format23/sect9.html.

```
ATOM    759  C4    U N  40       1.034   5.731  -3.177  1.00  0.00           C

ATOM     23  CA  SER    2      16.690  28.620  35.070  1.00  0.00           C

00000000011111111122222222233333333334444444444555555555566666666667777777777
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Figure 8.1: Standard PDB residue name format for RNA and protein in columns 18-20. Column numbers are shown vertically in black.

Sometimes you will need to insert a four character residue name (see below). In these cases, you may use the otherwise unused 21st column for the fourth character of the residue name.

8.1.1.2 PDB atom names

Atom names are four characters long (including spaces), and are found in columns 13 to 16 of PDB "ATOM" and "HETATM" lines. The two left-most characters of the atom name are usually the one- or two-character chemical element symbol for the atom, right-justified. For example the atom name " CA " (space-C-A-space) refers to a carbon atom, with element symbol "C". (A calcium atom, with element symbol "Ca", might also be named "CA", but the "C" and "A" characters would appear in columns 13 and 14 so C-A-space-space).

```
ATOM    759  C4    U N  40       1.034   5.731  -3.177  1.00  0.00           C

ATOM     23  CA  SER    2      16.690  28.620  35.070  1.00  0.00           C

00000000011111111122222222222333333333344444444445555555555666666666677777777771
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

Figure 8.2: PDB atom name fields are in columns 13-16

Keep columns in a PDB file in their correct alignment

When editing PDB files, it is important that you avoid changing the alignment of fields. For example, when editing a residue name or atom name field, do not change the number of characters in the line. Ensure that numbers in other columns, especially those to the right, still line up.

8.1.1.3 Remove water, ligands, and other molecules that are not protein, RNA, nor DNA

Because Zephyr simulates using implicit solvent, water molecules (residue name "HOH") should be removed from PDB files for simulation in Zephyr. Further, other

molecule types and atoms that are neither protein nor RNA (specifically, any residue types not defined in the file bin/gmxlib/ffamber96.rtp) should also be removed.

```
HETATM  605  O   HOH A  79       6.861   9.897  73.678  1.00 20.00           O
```

Figure 8.3: Example of a water molecule PDB line. Delete this line and others like it.

8.1.1.4 Handling missing atoms

Many molecule structures from the protein data bank (PDB) contain incomplete residues, missing some non-hydrogen atoms. Zephyr does not allow you to simulate molecules with missing atoms, since this would lead to inaccurate simulations. You can try to correct the file by manually adding in the missing atoms or using one of the available programs for constructing molecular models, such as Schrodinger's Maestro or the Sali lab's Modeller, both of which are free for academic users.

8.1.1.5 Proteins: Modifying residue and atom names

Protein residue names CYS, HIS, and LYS must be changed to be consistent with the ffamber force field in GROMACS:

- CYS residue name: Convert to CYN, unless it's in a disulfide bridge, in which case use residue name "CYS2". "CYS2" is an example of a four character residue name that requires using the 21st column of the ATOM line for the fourth character.
- HIS residue name: Convert to HID, HIE, or HIP, depending on whether the residue is be protonated at the delta nitrogen (HID), epsilon nitrogen (HIE), or both (positively charged - HIP). This choice depends on the arrangement of nearby hydrogen bond donors and acceptors.
- LYS residue name: Convert to LYP

```
ATOM    188  N   CYS A  42      40.714  -5.292  12.123  1.00 11.29           N
ATOM    559  CG  HIS A  90      33.017  -0.052  -3.236  1.00 16.37           C
ATOM    518  CA  LYS A  85      31.348  -0.655 -11.701  1.00 47.73           C
```

```
ATOM    188  N   CYN A  42      40.714  -5.292  12.123  1.00 11.29           N
ATOM    559  CG  HID A  90      33.017  -0.052  -3.236  1.00 16.37           C
ATOM    518  CA  LYP A  85      31.348  -0.655 -11.701  1.00 47.73           C
```

Figure 8.4: Example of modifying protein residue names: (top) before editing, (bottom) after editing

8.1.1.5.1 Initial N-terminal amino acid residues

Both the first and the last residue of each continuous protein chain require additional changes to residue and atom names. Initial protein residues (N-terminal residues) must be prefixed with the letter "N". This results in four-character residue names, and thus must make use of the 21st column in the PDB ATOM lines.

```
ATOM      1  N   HIS A  19      28.165  29.227  23.618  1.00 91.78           N
ATOM      2  CA  HIS A  19      27.004  29.173  22.731  1.00 91.74           C
ATOM      3  C   HIS A  19      26.321  27.818  22.666  1.00 81.05           C
ATOM      4  O   HIS A  19      25.105  27.739  22.805  1.00 89.56           O
ATOM      5  CB  HIS A  19      27.248  29.629  21.270  1.00 98.21           C
ATOM      6  CG  HIS A  19      27.954  30.936  21.082  1.00100.00           C
ATOM      7  ND1 HIS A  19      28.852  31.112  20.015  1.00100.00           N
ATOM      8  CD2 HIS A  19      27.882  32.111  21.798  1.00100.00           C
ATOM      9  CE1 HIS A  19      29.310  32.368  20.116  1.00100.00           C
ATOM     10  NE2 HIS A  19      28.753  32.997  21.176  1.00100.00           N
ATOM     11  N   SER A  20      27.057  26.778  22.303  1.00 58.48           N
```

```
ATOM      1  N   NHIDA 19      28.165  29.227  23.618  1.00 91.78           N
ATOM      2  CA  NHIDA 19      27.004  29.173  22.731  1.00 91.74           C
ATOM      3  C   NHIDA 19      26.321  27.818  22.666  1.00 81.05           C
ATOM      4  O   NHIDA 19      25.105  27.739  22.805  1.00 89.56           O
ATOM      5  CB  NHIDA 19      27.248  29.629  21.270  1.00 98.21           C
ATOM      6  CG  NHIDA 19      27.954  30.936  21.082  1.00100.00           C
ATOM      7  ND1 NHIDA 19      28.852  31.112  20.015  1.00100.00           N
ATOM      8  CD2 NHIDA 19      27.882  32.111  21.798  1.00100.00           C
ATOM      9  CE1 NHIDA 19      29.310  32.368  20.116  1.00100.00           C
ATOM     10  NE2 NHIDA 19      28.753  32.997  21.176  1.00100.00           N
ATOM     11  N   SER A  20      27.057  26.778  22.303  1.00 58.48           N
```

Figure 8.5: Example of modifying initial N-terminal amino acid residue names: (top) Initial residue HIS 19 before editing. Notice that the initial residue is not always number "1". (bottom) Initial protein residue after editing. Four character residue name spills into column 21.

8.1.1.5.2 Final C-terminal amino acid residues

C-terminal amino acid residue names must be prefixed with a "C". Again, this will create a four-character residue name that must use column 21 of the ATOM line. Further, for the C-terminal residues only, the atom names "O" and "OXT" must be changed to "OC1" and "OC2", respectively.

```
ATOM   2339  CG2 VAL A 313      15.802  10.026  10.329  1.00  6.43           C
ATOM   2340  N   TYR A 314      18.123   7.078   9.077  1.00 11.96           N
ATOM   2341  CA  TYR A 314      18.703   6.708   7.797  1.00 13.08           C
ATOM   2342  C   TYR A 314      17.968   5.444   7.240  1.00 12.06           C
ATOM   2343  O   TYR A 314      18.333   5.097   6.066  1.00 21.58           O
ATOM   2344  CB  TYR A 314      20.271   6.496   7.921  1.00  9.89           C
ATOM   2345  CG  TYR A 314      20.564   5.634   9.181  1.00 15.02           C
ATOM   2346  CD1 TYR A 314      20.402   4.250   9.128  1.00 12.83           C
ATOM   2347  CD2 TYR A 314      20.931   6.188  10.414  1.00 14.05           C
ATOM   2348  CE1 TYR A 314      20.671   3.408  10.205  1.00  9.53           C
ATOM   2349  CE2 TYR A 314      21.238   5.367  11.505  1.00  6.19           C
ATOM   2350  CZ  TYR A 314      21.068   3.985  11.411  1.00 10.42           C
ATOM   2351  OH  TYR A 314      21.357   3.139  12.450  1.00 11.71           O
ATOM   2352  OXT TYR A 314      17.078   4.932   7.946  1.00 17.12           O
TER    2353      TYR A 314
```

```
ATOM   2339  CG2 VAL A 313      15.802  10.026  10.329  1.00  6.43           C
ATOM   2340  N   CTYRA 314      18.123   7.078   9.077  1.00 11.96           N
ATOM   2341  CA  CTYRA 314      18.703   6.708   7.797  1.00 13.08           C
ATOM   2342  C   CTYRA 314      17.968   5.444   7.240  1.00 12.06           C
ATOM   2343  OC1 CTYRA 314      18.333   5.097   6.066  1.00 21.58           O
ATOM   2344  CB  CTYRA 314      20.271   6.496   7.921  1.00  9.89           C
ATOM   2345  CG  CTYRA 314      20.564   5.634   9.181  1.00 15.02           C
ATOM   2346  CD1 CTYRA 314      20.402   4.250   9.128  1.00 12.83           C
ATOM   2347  CD2 CTYRA 314      20.931   6.188  10.414  1.00 14.05           C
ATOM   2348  CE1 CTYRA 314      20.671   3.408  10.205  1.00  9.53           C
ATOM   2349  CE2 CTYRA 314      21.238   5.367  11.505  1.00  6.19           C
ATOM   2350  CZ  CTYRA 314      21.068   3.985  11.411  1.00 10.42           C
ATOM   2351  OH  CTYRA 314      21.357   3.139  12.450  1.00 11.71           O
ATOM   2352  OC2 CTYRA 314      17.078   4.932   7.946  1.00 17.12           O
TER    2353      CTYRA 314
```

Figure 8.6: Modification of final protein residue name: (top) before editing, (bottom) after editing

8.1.1.6 RNA: Modifying residue and atom names

RNA and DNA molecules from the Protein Data Bank use residue names “A”, “C”, “G”, “T”, and “U”. These one letter names are found in the right-most of the three residue name character positions (columns 18-20).

- Prefix RNA residue names with an “R”, and DNA residue names with a “D”
- Rename atom name “O2” to “O”
- Rename atom names with a “*” character to use a “’” (single quote) instead

8.1.1.6.1 Initial 5’ nucleotide residues

RNA residue names at the beginning of an RNA chain must be suffixed with a “5” character. Because this results in a three-character residue name, we will still be able to fit the residue name into the legal columns 18-20.

Further, 5’ phosphate groups on initial RNA residues are not supported in the ffamber force field, so phosphate atoms (P, OP1, OP2, OP3), if present, must be removed from initial 5’ RNA residues.

```
ATOM      1  OP3   G A   1       8.893  36.412   2.241  1.00 32.40           O
ATOM      2  P     G A   1       7.818  35.669   1.255  1.00 32.55           P
ATOM      3  OP1   G A   1       7.212  34.572   2.181  1.00 32.01           O
ATOM      4  OP2   G A   1       8.815  35.217   0.149  1.00 31.90           O
ATOM      5  O5'   G A   1       6.644  36.702   1.016  1.00 29.61           O
ATOM      6  C5'   G A   1       7.092  38.058   0.866  1.00 25.81           C
ATOM      7  C4'   G A   1       5.804  38.801   0.627  1.00 22.69           C
ATOM      8  O4'   G A   1       4.892  38.123  -0.209  1.00 21.38           O
ATOM      9  C3'   G A   1       5.043  38.994   1.942  1.00 21.94           C
ATOM     10  O3'   G A   1       5.763  39.898   2.778  1.00 22.29           O
ATOM     11  C2'   G A   1       3.635  39.317   1.434  1.00 20.38           C
ATOM     12  O2'   G A   1       3.696  40.673   1.016  1.00 21.09           O
ATOM     13  C1'   G A   1       3.585  38.381   0.239  1.00 17.45           C
ATOM     14  N9    G A   1       3.047  37.057   0.627  1.00 15.64           N
ATOM     15  C8    G A   1       3.596  35.863   0.388  1.00 13.98           C
ATOM     16  N7    G A   1       2.791  34.895   0.777  1.00 14.02           N
ATOM     17  C5    G A   1       1.637  35.508   1.225  1.00 13.08           C
ATOM     18  C6    G A   1       0.424  34.992   1.703  1.00 13.33           C
ATOM     19  O6    G A   1       0.070  33.829   1.882  1.00 13.54           O
ATOM     20  N1    G A   1      -0.490  35.960   1.972  1.00 12.82           N
ATOM     21  C2    G A   1      -0.254  37.283   1.852  1.00 12.50           C
ATOM     22  N2    G A   1      -1.289  38.026   2.241  1.00 13.22           N
ATOM     23  N3    G A   1       0.862  37.832   1.404  1.00 13.63           N
ATOM     24  C4    G A   1       1.779  36.864   1.075  1.00 13.72           C
ATOM     25  P     G A   2       5.596  39.608   4.451  1.00 21.60           P
```

```
ATOM      5  O5'  RG5 A   1       6.644  36.702   1.016  1.00 29.61           O
ATOM      6  C5'  RG5 A   1       7.092  38.058   0.866  1.00 25.81           C
ATOM      7  C4'  RG5 A   1       5.804  38.801   0.627  1.00 22.69           C
ATOM      8  O4'  RG5 A   1       4.892  38.123  -0.209  1.00 21.38           O
ATOM      9  C3'  RG5 A   1       5.043  38.994   1.942  1.00 21.94           C
ATOM     10  O3'  RG5 A   1       5.763  39.898   2.778  1.00 22.29           O
ATOM     11  C2'  RG5 A   1       3.635  39.317   1.434  1.00 20.38           C
ATOM     12  O2'  RG5 A   1       3.696  40.673   1.016  1.00 21.09           O
ATOM     13  C1'  RG5 A   1       3.585  38.381   0.239  1.00 17.45           C
ATOM     14  N9   RG5 A   1       3.047  37.057   0.627  1.00 15.64           N
ATOM     15  C8   RG5 A   1       3.596  35.863   0.388  1.00 13.98           C
ATOM     16  N7   RG5 A   1       2.791  34.895   0.777  1.00 14.02           N
ATOM     17  C5   RG5 A   1       1.637  35.508   1.225  1.00 13.08           C
ATOM     18  C6   RG5 A   1       0.424  34.992   1.703  1.00 13.33           C
ATOM     19  O6   RG5 A   1       0.070  33.829   1.882  1.00 13.54           O
ATOM     20  N1   RG5 A   1      -0.490  35.960   1.972  1.00 12.82           N
ATOM     21  C2   RG5 A   1      -0.254  37.283   1.852  1.00 12.50           C
ATOM     22  N2   RG5 A   1      -1.289  38.026   2.241  1.00 13.22           N
ATOM     23  N3   RG5 A   1       0.862  37.832   1.404  1.00 13.63           N
ATOM     24  C4   RG5 A   1       1.779  36.864   1.075  1.00 13.72           C
ATOM     25  P     RG A   2       5.596  39.608   4.451  1.00 21.60           P
```

Figure 8.7: Modification of the initial RNA residue: (top) before editing, (bottom) after editing – note the removal of the atoms associated with the phosphate group

8.1.1.6.2 Final 3' nucleotide residues

Final 3' RNA and DNA residues are suffixed with a "3".

```
ATOM    887  C6    C N  44       2.133  -9.174   2.039  1.00  0.00           C
ATOM    899  P     C N  45      -1.517  -3.805   3.064  1.00  0.00           P
ATOM    900  OP1   C N  45      -2.742  -3.375   3.776  1.00  0.00           O
ATOM    901  OP2   C N  45      -1.601  -4.193   1.640  1.00  0.00           O
ATOM    902  O5'   C N  45      -0.418  -2.617   3.203  1.00  0.00           O
ATOM    903  C5'   C N  45       0.977  -2.925   3.291  1.00  0.00           C
ATOM    904  C4'   C N  45       1.719  -2.050   4.296  1.00  0.00           C
ATOM    905  O4'   C N  45       2.129  -2.824   5.421  1.00  0.00           O
ATOM    906  C3'   C N  45       2.995  -1.475   3.699  1.00  0.00           C
ATOM    907  O3'   C N  45       2.755  -0.241   3.003  1.00  0.00           O
ATOM    908  C2'   C N  45       3.873  -1.280   4.933  1.00  0.00           C
ATOM    909  O2'   C N  45       3.658  -0.001   5.553  1.00  0.00           O
ATOM    910  C1'   C N  45       3.430  -2.418   5.855  1.00  0.00           C
ATOM    911  N1    C N  45       4.384  -3.554   5.829  1.00  0.00           N
ATOM    912  C2    C N  45       5.595  -3.415   6.503  1.00  0.00           C
ATOM    913  O2    C N  45       5.873  -2.380   7.076  1.00  0.00           O
ATOM    914  N3    C N  45       6.451  -4.470   6.510  1.00  0.00           N
ATOM    915  C4    C N  45       6.143  -5.612   5.890  1.00  0.00           C
ATOM    916  N4    C N  45       7.007  -6.623   5.923  1.00  0.00           N
ATOM    917  C5    C N  45       4.906  -5.761   5.197  1.00  0.00           C
ATOM    918  C6    C N  45       4.063  -4.711   5.190  1.00  0.00           C
TER     931        C N  45
```

```
ATOM    887  C6   RC N  44       2.133  -9.174   2.039  1.00  0.00           C
ATOM    899  P   RC3 N  45      -1.517  -3.805   3.064  1.00  0.00           P
ATOM    900  OP1 RC3 N  45      -2.742  -3.375   3.776  1.00  0.00           O
ATOM    901  OP2 RC3 N  45      -1.601  -4.193   1.640  1.00  0.00           O
ATOM    902  O5' RC3 N  45      -0.418  -2.617   3.203  1.00  0.00           O
ATOM    903  C5' RC3 N  45       0.977  -2.925   3.291  1.00  0.00           C
ATOM    904  C4' RC3 N  45       1.719  -2.050   4.296  1.00  0.00           C
ATOM    905  O4' RC3 N  45       2.129  -2.824   5.421  1.00  0.00           O
ATOM    906  C3' RC3 N  45       2.995  -1.475   3.699  1.00  0.00           C
ATOM    907  O3' RC3 N  45       2.755  -0.241   3.003  1.00  0.00           O
ATOM    908  C2' RC3 N  45       3.873  -1.280   4.933  1.00  0.00           C
ATOM    909  O2' RC3 N  45       3.658  -0.001   5.553  1.00  0.00           O
ATOM    910  C1' RC3 N  45       3.430  -2.418   5.855  1.00  0.00           C
ATOM    911  N1  RC3 N  45       4.384  -3.554   5.829  1.00  0.00           N
ATOM    912  C2  RC3 N  45       5.595  -3.415   6.503  1.00  0.00           C
ATOM    913  O   RC3 N  45       5.873  -2.380   7.076  1.00  0.00           O
ATOM    914  N3  RC3 N  45       6.451  -4.470   6.510  1.00  0.00           N
ATOM    915  C4  RC3 N  45       6.143  -5.612   5.890  1.00  0.00           C
ATOM    916  N4  RC3 N  45       7.007  -6.623   5.923  1.00  0.00           N
ATOM    917  C5  RC3 N  45       4.906  -5.761   5.197  1.00  0.00           C
ATOM    918  C6  RC3 N  45       4.063  -4.711   5.190  1.00  0.00           C
TER     931      RC3 N  45
```

Figure 8.8: Modification of final 3' RNA residue: (top) before editing, (bottom) after editing

```
ATOM    576  N   CPHE   76       5.995  -7.109   3.616  1.00  2.54           N
ATOM    577  CA  CPHE   76       6.033  -8.575   3.923  1.00  3.42           C
ATOM    578  C   CPHE   76       4.678  -9.203   3.574  1.00  4.09           C
ATOM    579  OC1 CPHE   76       3.672  -8.540   3.760  1.00  4.51           O
```

Figure 8.9: PDB file excerpt from villin.pdb. Green text shows where atom and residue names have been changed.

9 References

D.A. Case, T.E. Cheatham, III, T. Darden, H. Gohlke, R. Luo, K.M. Merz, Jr., A. Onufriev, C. Simmerling, B. Wang and R. Woods. The Amber biomolecular simulation programs. *J. Computat. Chem.* **26,** 1668-1688 (2005).

Friedrichs, M., Eastman, P., Vaidyanathan, V., Houston, M., Legrand, S., Beberg, A., et al. (2009). Accelerating Molecular Dynamic Simulation on Graphics Processing Units. *Journal of Computational Chemistry* , in press.

Glazer, D., Radmer, R.J., & Altman, R. (2008). Combining molecular dynamics and machine learning to improve protein function recognition. *Pac Symp Biocomput* , 332-343.

Herce, H., & Garcia, A. (2007). Molecular dynamics simulations suggest a mechanism for translocation of the HIV-1 TAT peptide across lipid membranes. *PNAS , 104*, 20805-20810.

Hess, B., Kutzner, C., van der Spoel, D., & Lindahl, E. (2008). GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput. , 4*, 435-447.

Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD - Visual Molecular Dynamics. *J. Molec. Graphics* , 33-38.

Kasson, P., & Pande, V. (2008). Structural basis for influence of viral glycans on ligand binding by influenza hemagglutinin. *Biophysical Journal , 95*, L48-L50.

Marszalek, P., & al., e. (1999). Mechanical unfolding intermediates in titin molecules. *Nature , 402* (6757), 100-103.

Ponder, J.W. and Case, D.A. (2003). Force fields for protein simulations. *Adv. Prot. Chem., 66*, 27-85.

Zagrovic, B. and Pande, V. (2003). Solvent viscosity dependence of the folding rate of a small protein: distributed computing study. *J. Comput. Chem.*, 24(12), 1432-1436.