The Ymol manual. (partly updated for Ymol 0.8.117)

July 25, 2014

Contents

List of Figures

List of Tables

1 Introduction

This section currently contains copies of the files README and INSTALL.

1.1 Readme

This is Ymol version 0.8.x. Ymol is a molecular visualisation and animation program for the X Window System written from scratch. It can be used to create production quality images, and movies, using either the built in renderer or by using POVRAY (Persistence of Vision). Ymol contains a programming language of its own, which can be used to create presentations that run non-interactively.

Ymol is still in the alpha phase, but is released because it may be useful even if all features are not yet implemented. Ymol may also contain bugs. :-(

As soon ymol reaches version 0.9.x it will be considered to be in the beta stage.

1.2 Install

1.2.1 Choosing the right files to download

If ymol has been ported to your platform, it is very likely that there are precompiled binaries available. If there are no binaries for your platform, you can download the sources and try to compile Ymol yourself. It is fairly easy to port ymol to a new platform.

1.2.2 Downloading the binaries

The ymol binary archives are named as follows: package name - version number - operating system .tar.gz or .tar.Z

If all you want is the ymol binary and files required by ymol, download the package named ymol. If you also want the Xco library (for creating buttons, menus etc using only Xlib) you want the package name Xco. For the Y4 language interpreter download the package y4vm. It also contains a precompiled, freestanding, text based, interactive shell.

1.2.3 Downloading the source

The ymol source archives are named as follows: package name - version number .tar.gz To compile ymol you need all the source packages.

1.2.4 Unpacking the archives

Create a directory to hold the archive contents. Place all packages in this directory. All the packages are compressed. If your archives are named something.tar.gz you need gunzip (already installed on your system, or available on the internet) to unpack them. If they are named something.tar.Z you need uncompress (already available on all (?) systems). Using gunzip type: ”gunzip something.tar.gz” on all packages. Using uncompress type: ”uncompress something.tar.Z” This will leave you with packages named something.tar Then you need to use tar to unpack the archives. Type ”tar xf something.tar” on all packages.

1.2.5 Compiling the source

If you downloaded the binaries you don't need to do this. If ymol has been ported to your platform you just need to type "make" in the directory you created and go for a cup of coffee while ymol compiles. If it is not ported to your platform, see the file PORTING (may not be available yet).

1.2.6 Installing the binaries

Installing ymol is just a matter of typing "make install" in the directory you created. You may want to change the default installation directories in the bourne shell script make-install.sh. If you are root, ymol will be installed for all users on the system. If you are not root, ymol will be installed in a directory named "bin" in your home directory. Make sure that this is appropriate for you and that the bin directory is in your path. The installation script will also install some files needed by ymol in the directory named ".ymol" in your home directory.

For information of what you may and may not with this/these packages, see the file LICENSE.

Look at http://www.spaangberg.se/daniels/ymol.html for updates.

You can also send email to daniel@spaangberg.se

2 Basic Ymol operations

2.1 Starting Ymol

Ymol can be started by typing "ymol" at the shell command prompt. Ymol will give some (informative ;-)) messages and open a window (fig. 1).

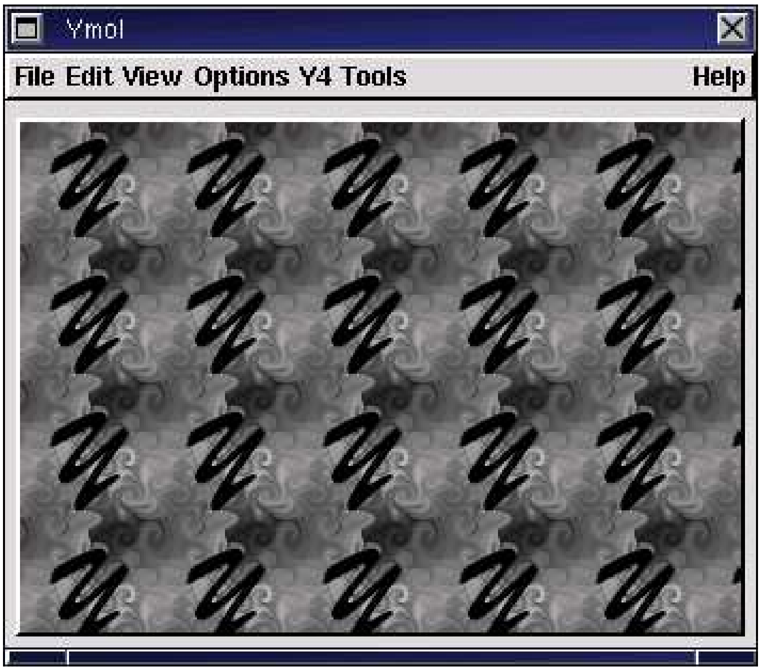


Figure 1: The Ymol window

2.2 File operations

2.2.1 Loading an Ymol file

Selecting the "Load" item in the "File" menu will bring up a file selector. In the file selector there are two "large" boxes containing text, two lines where you can input text and a pulldown list containing the filter (the file extension). The large box to the left shows the directories in the current directory. The current directory is shown in the top input field. The large box to the right shows the files in the current directory ending in the file extension shown in the pulldown list. If you want to see more files, or files of another type, use the pulldown list to select the appropriate extension. The bottom input field shows the filename. You can go to another directory by *single* clicking on the directory name. You can select a file by *single* clicking its name. Opening/Importing/Saving files is then done by clicking the appropriate button. (fig. 2).

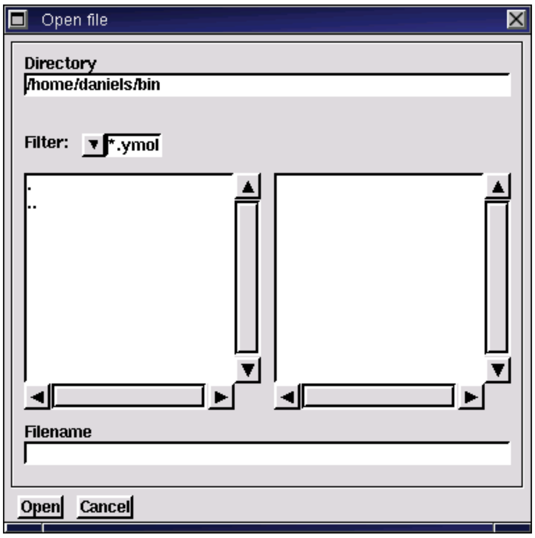


Figure 2: The file selector

2.2.2 Importing a file

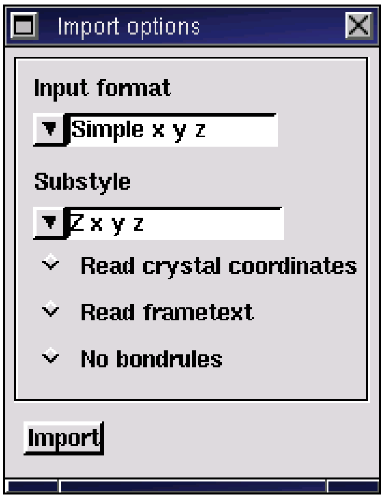
Ymol can import files consisting of any number of frames[1]. Each frame can consist of any number of atoms[2]. The frames can contain different numbers of atoms.

Selecting the "Import" item in the "File" menu will bring up a the file selector. After having chosen a file to import, a window with a selection of different import formats will appear (fig. 3).

Ymol can import files in either a format that simply contains a list of coordinates and atom types, or read output files from the Gaussian 90/92/94 program. The simple format can be varied in several ways. The basic structure is as follows:

```
number of frames
number of atoms in first frame
atom number, x, y and z of first atom in first frame
atom number, x, y and z of second atom in first frame
.
.
.
atom number, x, y and z of last atom in first frame
number of atoms in second frame
atom number, x, y and z of first atom in second frame
.
.
.
atom number, x, y and z of last atom in second frame
```

[1] Up to available memory.
[2] Again, up to available memory.



Input format
Simple x y z

Substyle
Z x y z

- Read crystal coordinates
- Read frametext
- No bondrules

Figure 3: The import format selector

```
.
.
.

.
.
.
number of atoms in last frame
atom number, x, y and z of first atom in last frame
.
.
.
atom number, x, y and z of last atom in last frame
```

In the import format selector there are some pulldown lists and some toggles with text beside. The topmost pulldown list selects the input format. The second pulldown list selects the variation of the input format. The toggles change some mode of operation. The substyles of the simple format shows a text line describing the line that contains the atom number, x, y and z in the import file. The large Z represents the atom number, the x, y and z are the atom coordinates in Ångström. The id is an atom identification number, useful to put in the file if you are planning to edit an animation in Ymol. The id is a positive integer between 0 and 2147483647. It should be set to the same number for the same atom in different frames. If you don't put the id in the file, Ymol will try to create id numbers anyway, but the algorithm to do this can fail (and does). The label is a string that is currently discarded (for compatibility with other programs). The R is the atom radius in Ångströms, the W is the atom weight in atomic units, the r, g and b are the atom color components (red, green and blue components respectively) ranging from 0 to 1. Examples of import files are found in figs 4 and 5.

Figure 4: One frame water molecule. Import format Z x y z

```
1
3
1   -0.538534    0.998759    0.000000
8    0.121740    0.326894    0.000000
1    0.937520    0.797922    0.000000
```

2.2.3 Saving an Ymol file

Saving ymol files can be done by selecting the “Save as” item in the “File” menu.

2.3 Rotating and zooming

The easiest way to rotate a molecule is just to point at it with the mouse, hold down the left mouse button and drag the mouse around. This will rotate the molecule, as if it was enclosed in a transparent sphere, and the mouse pointer is touching the surface of this sphere and moving this point of the sphere to where you drag the mouse. Another way to rotate the molecule is to bring up the control panel (fig. 6), under the “View” menu. The control panel consists of two blocks of buttons. One of the blocks consists of nine buttons and the other consists of four buttons. The block consisting of nine buttons is for rotating and zooming. The block consisting of four buttons is covered in the next section. Clicking any of the eight buttons surrounding the nineth (?) button, will rotate or zoom one time. Clicking the button in the middle toggles continous rotation. By first clicking on the middle button and then on another (or several) button, a continous rotation or zoom will start. The continous rotation or zoom is stopped by clicking the middle button again. The third way to rotate a molecule is to use the keyboard. Make sure that you have the window with the molecule active[3]. You can now rotate the molecule using the cursor keys. The space key acts as the continous rotation toggle.

2.4 Playing an animation

To play an animation, you need to open the control panel. The four buttons at the bottom controls the playing of an animation. The two buttons to the left will play the animation backwards. The two buttons to the right will play the animation forwards. The two buttons to the far left and far right will single step the animation, that is, play one frame at each click. The two buttons in the middle plays the animation continously. To find out the number of the frame you are currently looking at, you can select the “Frame number” item in the “View” menu.

[3] And currently you must, because of a “feature”, place the mouse pointer inside the drawing area

Figure 5: Two frames DMSO. Import format id Z x y z

```
2
10
14   16  0.      0.      .1219
72   8   0.      0.      2.129
35   6   0.    -1.4267 -.9772
19   6   0.     1.4267 -.9772
27   1   0.    -1.1068 -2.0143
9    1  -.8804 -2.0399 -.8065
62   1   .8804 -2.0399 -.8065
12   1   0.     1.1068 -2.0143
56   1   .8804  2.0399 -.8065
3    1  -.8804  2.0399 -.8065
10
14   16  .0493  0.      .1219
72   8  -.0328  0.      2.129
35   6  -.0329 -1.4267 -.9772
19   6  -.0329  1.4267 -.9772
27   1   .0019 -1.1068 -2.0143
9    1  -.9462 -1.9968 -.821
62   1   .8146 -2.0829 -.792
12   1   .0019  1.1068 -2.0143
56   1   .8146  2.0829 -.792
3    1  -.9462  1.9968 -.821
```

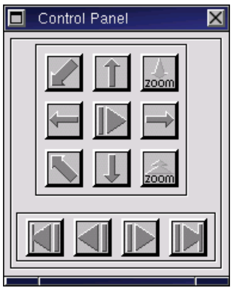


Figure 6: The control panel

3 More Ymol operations

3.1 Context sensitive menus

By right-clicking on the drawing area, a menu will appear. Everything before the line in this menu is context sensitive. This means that the menu will change depending on what you were pointing at with the pointer while right-clicking. If you click on an atom you will, among other things, get the option of deleting it.

3.2 Default atom properties

Default atom properties controls things like the atom radius and atom weight of the elements. When selecting the “Default atom properties” item in the “Edit” menu, the “Edit atom properties” window will appear (fig. 7). You can open a window for editing the properties of different atoms by pressing the buttons in the periodic table. Figure 8 contains an example of such a window. The windows for editing atom properties contain three fields. The first one is for the color. Pressing the “Browse” button will open a color selector. The second field contain the atom radius and the third the atom weight. Pressing the “Update” button will update the properties.

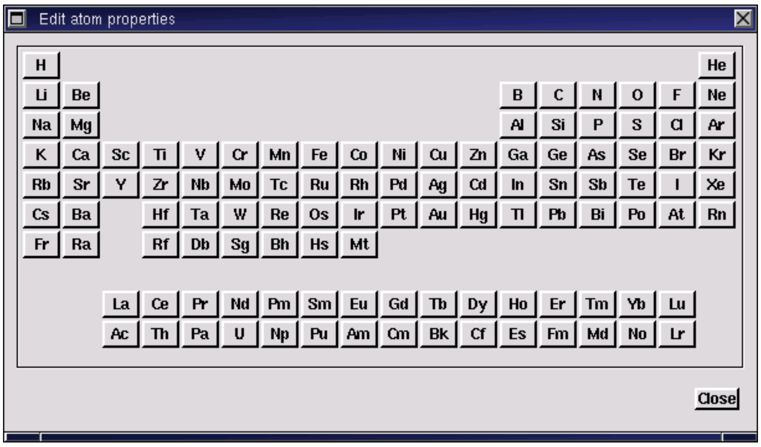
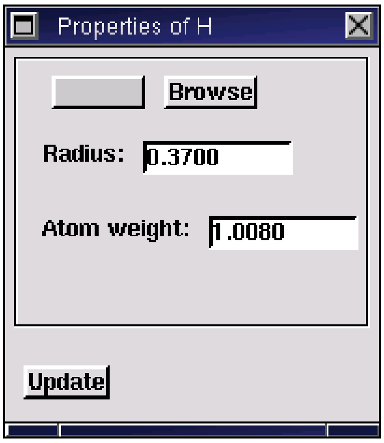

Edit atom properties
H He
Li Be B C N O F Ne
Na Mg Al Si P S Cl Ar
K Ca Sc Ti V Cr Mn Fe Co Ni Cu Zn Ga Ge As Se Br Kr
Rb Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn Sb Te I Xe
Cs Ba Hf Ta W Re Os Ir Pt Au Hg Tl Pb Bi Po At Rn
Fr Ra Rf Db Sg Bh Hs Mt
La Ce Pr Nd Pm Sm Eu Gd Tb Dy Ho Er Tm Yb Lu
Ac Th Pa U Np Pu Am Cm Bk Cf Es Fm Md No Lr
Close

Figure 7: The edit atom properties window

3.3 Bondrules

Bondrules are used to automate the creation of bonds between atoms of certain kinds. When selecting the “Bondrules” item in the “Edit” menu, the bond rules window will appear (fig. 9). You can add a bondrule by clicking on the add button. You can modify an already existing bondrule by first selecting the bondrule you wish to change, and then click on the modify button. You can also delete a bondrule by selecting it and clicking on the delete button. When you add or modify a bondrule a window containing some fields to control the bondrule opens (fig. 10). The first field contains the name of the bondrule.



Properties of H

Radius: 0.3700

Atom weight: 1.0080

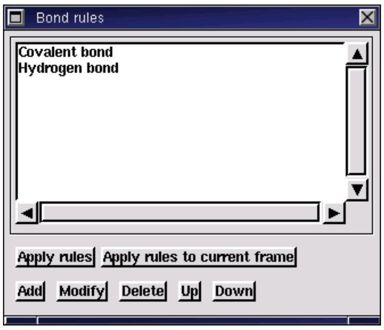
Figure 8: Window for editing the properties of Oxygen

This will be displayed in the bondrule window. The second and third fields contains the donor and acceptor atoms for this bond. If you enter “All” or nothing in these fields the bond rule will apply to all kinds of atoms. If you click on any of the browse buttons, a periodic table will appear. From this periodic table you can select up to 10 different atoms as donors and up to 10 different atoms as acceptors. If the donor field contains, say, hydrogen and the acceptor field contains, say, oxygen, the bond rule will apply only to bonds between hydrogen and oxygen. It does not matter whether you put hydrogen as donor and oxygen as acceptor or vice verse. The fourth field contains a parameter called “distance criterion”. It controls how close the atoms must be for a bond to be created. If the difference of the distance between the atoms are less than the “distance criterion” parameter a bond will be created. If the toggle “distance between ”electrons”” are selected, the sum of van der Waals radii of the atoms will be subtracted from the distance, so one gets the effect of “automatic” selection of bond distances when having many donors and/or acceptors, like in the default covalent bond. Next is a box showing the color of the bond. The fifth field contains the radius of the bond itself, that is, how thick the bond will be. The priority of bond rules are in the order they are displayed in the bond rule window. So for the default bond rules, a covalent bond will always be shown instead of a hydrogen bond if the distance between the atoms are short enough. After having added or modified a bond rule you must click on the “apply” button in the bond rule window. You can also click on the “apply to current frame” button if you want to see the result quickly without having to wait for the entire animation to be updated.

3.4 Selecting atoms and bonds

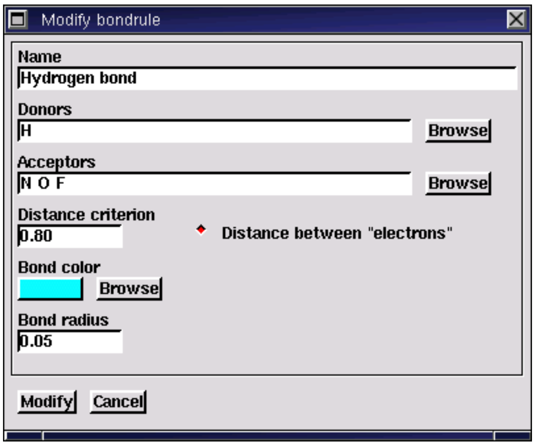
Pointing with the mouse on an atom or a bond and clicking with the middle[4] mouse button will select that atom or bond.

[4] On some systems with two-button mice you can use both buttons to emulate a middle button



Covalent bond
Hydrogen bond

Figure 9: The bondrule window



Name: Hydrogen bond

Donors: H

Acceptors: N O F

Distance criterion: 0.80

◆ Distance between "electrons"

Bond color: 🟦

Bond radius: 0.05

Figure 10: The window used to add and modify bondrules

3.5 Creating bonds using the mouse

Bonds between atoms can be created without using the bond rules. By selecting all the atoms you want bonds to be created from and then right clicking on the atom you want all the bonds to be created to, you will get the option to create bonds to selected species. There are four sub options to this option. The first (and default) is to create the bonds only in the current frame. The second is to create the bonds in all frames. The third is to create the bonds in all frames before and including the current frame. The last option is to create the bonds in all frames after and including the current frame.

3.6 Deleting atoms and bonds

By right clicking on an atom or a bond, you will get the option to delete the atom or bond. The suboptions are the same as in creating bonds using the mouse.

3.7 Changing properties of atoms and bonds

By selecting all the atoms or bonds you want to change the properties of and right clicking on the free drawing area[5] you will get the option to change the properties of selected atoms or bonds. The dialog boxes of the atom and bond properties are considered self-explanatory ;-).

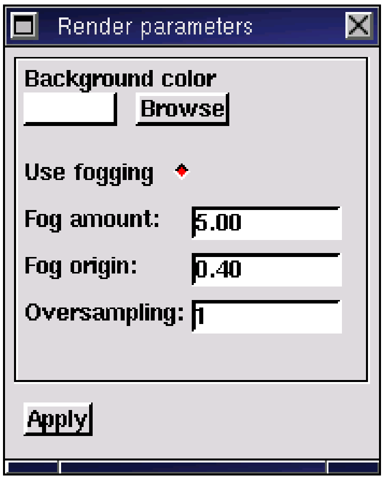
3.8 Drawstyles

You can select how atoms and bonds are drawn using the "Style" item under the "View" menu. The "Wireframe" mode will display bonds as lines and atoms as points. The "Circles and polygons" mode will display the atoms as filled circles and the bonds as filled polygons. The "Shaded spheres and cylinders" mode will display atoms and bonds using Phong and Gouraud shading for realistic lit scenes. If you prefer using povray instead of the built in renderer for "Shaded spheres and cylinders", you can select this in the "Options-Graphics-Shading-Use POVRAY" menu.

3.9 Rendering parameters

Selecting the "Rendering parameters" item under the "Edit" menu will bring up the rendering parameters window (fig. 11). The first field in the rendering parameters window shows the current background color. This can be changed by clicking on the "browse" button. The toggle "use fogging" controls whether fogging will be used or not. Fogging is useful when viewing large molecules, or sets of molecules. Fogging will cause the atoms and bonds far away from the viewer to be dimmer, by blending some of the background color with the color of the atoms and the bonds. More of the background color is blended in for atoms and bonds farther away from the viewer. The amount of fog and where to start using fog is controlled by the two fields just below the toggle. The higher the value of the "fog amount" parameter the steeper the blending curve. By increasing the fog amount value, atoms far away from the viewer will become dimmer and atoms close to the viewer will become clearer. The "fog origin" parameter controls where the center of the fog is, that is, where half of the color displayed comes from the background and half from the atom or bond. The fog origin ranges between 0 and 1. 0 is the clipping plane close to the viewer and 1 is the clipping plane far away from the viewer. The last parameter in the rendering parameters window is the oversampling. Oversampling is only used for the "Shaded spheres and cylinders" mode. If you use a oversampling of 1 only one point is computed for each pixel. An oversampling of 2 will compute four points for each pixel, two points in the x direction and two in the y direction. An oversampling of 3 will compute nine points for each pixel. Oversampling is always a positive integer. The picture resolution times the oversampling must not exceed 16384, due to the fixed point calculations used for the "Shaded spheres and cylinders" mode.

[5] Needing to click on the free drawing area is a "feature".



Background color

Use fogging ◆

Fog amount: 5.00

Fog origin: 0.40

Oversampling: 1

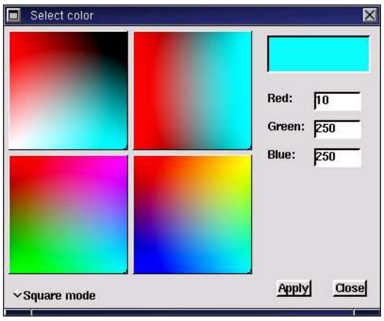
Figure 11: The rendering parameters

3.10 Using the color selector

Whenever clicking on the browse button close to a color box the color selector will appear (fig. 12). Using the color selector can be a bit tricky. Since there are more than 16 million available colors (2^{24}) it is impossible to display them all at once. In the top right corner the current color is shown. The four boxes to the left contains different colors interpolated to the current color. The current color is in the bottom right corner of all the four boxes. In the top left corner of all the four boxes is the contrast color to the current color. In the bottom left and top right corner of the top left box the colors black and white are shown. In the bottom left and top right corners of the other three boxes the base colors and their contrast colors are shown. Respectively they are: red-cyan, green-magenta and blue-yellow. By clicking inside one of the boxes the color at the mouse pointer will be selected as the current color and the other three boxes will be updated to reflect the change. So if you want the color orange starting from white, you should first click on either red or yellow, then click in any of the other three boxes where the orange you want is shown. If you want a slightly different orange you can click again in any of the four boxes. To the right of the four boxes there are three input fields containing the current rgb color values. The color values are integers with the range 0-255. You can also choose a color by typing in a number in any of these fields and press enter.

3.11 Saving the current image as a pixmap

By selecting the “Save as PPM image” item in the file menu a pixmap can be saved to disk. The saved PPM image can be read into another program like the GIMP and converted to whatever format you like (.gif,.tif,.jpg etc.).



Red: 10
Green: 250
Blue: 250

Figure 12: The color selector

3.12 Saving the current image as a POVRAY scene

By selecting the “Save as POVRAY scene” item in the file menu a povray scene can be saved to disk. If you save a scene named “test.pov” a file named “test.pov” and a file named “test.inc” will be created. The “test.pov” file contains all the coordinates of atoms and bonds. The “test.inc” file contains all the texture definitions and definitions of lamp positions etc. The scene can be rendered using the command line “povray +d -i test.pov”[6].

3.13 Creating a MPEG movie or an animated GIF

By selecting the “MPEG-Create” item under the “Tools” menu, an MPEG file will start to be created. All images shown in the window will be stored to a subdirectory and compressed using *gzip*. When you have moved or rotated the molecule or played an animation the MPEG movie is created by selecting the “MPEG-Save as” item under the “Tools” menu. Note that Berkeley *mpeg_encode* and GNU-zip (gzip) is required to be in the path for this to work. An animated GIF is created in a similar way. The program *ppmtoanimgif* must be in the path for this to work.

[6] given that povray is installed on your system, of course.

4 Interfacing Ymol with another program

Ymol can be used as a visualisation tool for another running program. This chapter contains a fairly small example of what have to be done to make your program use ymol as a visualision tool. The basic idea is that your program should write atom coordinates to a file and then give ymol a signal to read this file and display it. To make this work a small y4[7] program must be running. This y4 program is called "mdinterface". You can run this program by opening the y4vm console under the *Y4* menu. In the input field at the bottom of the console type `'mdinterface' load`. Y4vm should respond with `Loading mdinterface. Filesize is xxx bytes.`. Now you have added the command *mdinterface* to the y4vm. Start the program by typing `mdinterface`. Once you have the mdinterface program running you can run your program.

4.1 A fairly small example

This example will create a rotating circle of 8 hydrogen atoms. It is assumed that you are familiar with the C language. Of course you can use any programming language you like, most likely `FORTRAN`, but you need to send signals to ymol. This can probably be done by calling a C routine from your programming language.

The first thing we need to do is to open and read a file called `ymolpid` that ymol always writes to the current directory when it is run. This file contains the ymol PID. This is needed so your program sends the signal to the correct process. Then we starts a loop to generate configurations. After each configuration we signal ymol to read the data. In this example program we actually need to sleep for a little while, because this program is generating configurations a lot faster than ymol can handle them. This will normally not be a problem, since MD code will not generate configurations fast enough. Ok, here's the first part of the example:

```
#include <stdio.h> /* Needed for file operations */
#include <sys/types.h> /* Needed for pid_t etc */
#include <signal.h> /* Needed to send signals */
#include <unistd.h> /* Needed by sleep in this example program */
#include <math.h> /* Needed by this example program */

#define MYSIGNAL SIGUSR1 /* We will use this signal */

int main()
{
  FILE *myfile;
  unsigned long ymolpid;
  double time=0.; /* For this example program */
  myfile=fopen("ymolpid","r");
  fscanf(myfile,"%lu",&ymolpid);
  fclose(myfile);
  while(1) /* Forever */
```

[7] See: The y4 language Section 5

```
    {
      /* Here the code to generate a
         new configuration will be placed */

      kill((pid_t)ymolpid,MYSIGNAL);
      /* We need to sleep for a while,
         so Ymol doesn't choke on too much
         input data */
      sleep(1);
    }
}
```

The second part of the example shows the code needed to generate the rotating ring. It should be inserted in the program above where indicated:

```
      int i;
      /* The filename can be changed in mdinterface.y4 */
      FILE *conffile=fopen("interface.mol","w");
      fprintf(conffile,"1\n"); /* One frame */
      fprintf(conffile,"8\n"); /* Eight atoms */
      for (i=0; i<8; i++)
        {
          double x,y,z;
          x=cos(2*3.14*i/8.+time);
          y=sin(2*3.14*i/8.+time);
          z=0.;
          /* Write atom number and x, y and z coordinates to file */
          fprintf(conffile,"1 %f %f %f\n",x,y,z);
        }
      fclose(conffile);
      time+=0.1;
```

This example can be found in the “interfaceexample.c” file in the ymol source directory. It should probably be enough to type “cc -o interfaceexample interfaceexample.c -lm” to compile it on most systems.

5 The y4 language

5.1 Introduction

Ymol contains its own language. It actually contains its own virtual machine. When you type text into the y4 console, the text will be compiled into “machine instructions” for the virtual machine. Y4 is somewhat similar to the language Forth invented in 1970 by Moore. The largest difference between y4[8] and languages like C, Pascal, `FORTRAN` and `BASIC` is that it is using reverse Polish (postfix) notation. This is the same notation as Hewlett-Packard calculators use. The notation used in the named languages is called infix notation. To add two numbers using a “regular” language you would type `1+2`. In y4 you type `1 2 +`.

5.2 The stack(s)

Most of the things in y4 circle around the stack[9], or more precisely, the stacks. When you type a number, the number is stored on a stack. When typing two numbers they are stored on the stack, the second after the first. The + operator will then take the first two numbers on the stack and add them together, leaving the result on the stack. The advantage with using the stack is that most operations can be done without using variables. The operator “.” is used to show the result of a calculation. Some examples:

infix: `(1+2)*(3+4)` rpn: `1 2 + 3 4 + *`, infix: `sin(0)` rpn: `0 sin`.

I am very sorry, but I haven’t currently got the time to write this section :-(. You can always try to look in the /.ymol/*.y4 files. I will produce some example files at the same time I write this section some day.

[8] and Forth

[9] A LIFO (last in first out) buffer

6 Tips

6.1 Using a style file

Bondrules, atom colors, etc. can be imported from previously saved ymol files by selecting “File-¿Import-¿Style from ymol file”. If a file named “style.ymol” exists in the current directory when importing a file the styles from the file will be loaded automatically.

6.2 Loading/Importing files from the command line

An argument to ymol may be given. This should be the name of a file to be loaded into ymol at startup. If this file is named “something.ymol” ymol will load it as an ymol file. If it is named something else ymol will import it using the options used the last time a file was imported in the same directory.

6.3 Importing files from the command line in the fastest way

- Import a file with the correct options.
- Set all the bondrules, colors, etc.
- Save the file as “style.ymol”

Then you can load files of the same type, with the same style on the command line:

```
ymol file.mol
```