

Construction of Kinemages through the use of PREKIN

Copyright 1999, Gary Manfready

Overview of the PREKIN and MAGE program

The MAGE program is a three dimensional vector modeling service. Although any image may be rendered, the usefulness for our purposes is its ability to make clear and understandable images of complex molecular structure. Once an image (called a kinemage) is generated and displayed many manipulations are possible through the use of control buttons and menu bar options.

The PREKIN program, along with the PREKIN LIBRARY makes the visualization of a kinemage possible. The procedure follows a general principle of locating a file of coordinates for the desired molecule (done through the use of the Internet). The file (called a PDB for protein data base) is downloaded to your computer and used in the PREKIN program to build an image based on YOUR SELECTED PARAMETERS. Once it has been created, the file is essentially formatted to comply with the MAGE program's directives for producing a kinemage image. You will notice the file ending is .kin or .kip which refers to the formatted PDB file being ready for MAGE viewing. The two programs work hand in hand. You will not be able to view a kinemage of a downloaded PDB file until PREKIN has formatted the information for use by MAGE.

PREKIN TUTORIAL

To begin this tutorial you will need to obtain the PDB file for the insulin dimer. You will also need to have the PREKIN program available on your computer. To obtain this information you will need an Internet access and appropriate browser.

Obtain the PREKIN Program (if you already have the programs skip this section)

1. Once online go to the URL site: <http://kinemage.biochem.duke.edu/website/download.htm>
2. Depending on the type of computer you are using select either PC PLATFORM or McINTOSH PLATFORM .
3. For PC or Mac download one of each of three files if you do not have them.
 - a. For PC: PKIN 5 4.exe This is the PREKIN program
PREKINAA.LIB This is a reference library for PREKIN
MAGE 5 4.exe This is the MAGE program for viewing kinemages
 - b. For Mac PKIN 5 4.PPC.bin
PREKINAA.LIB
MAGE 5 4.PPC.bin
4. Close the window and exit the Internet if you are not continuing with the tutorial at this time. Otherwise remain online to download the information in the next section.

Obtain the PDB file

1. Once online go to the URL site: www.rcsb.org/pdb/
2. Identify the file you wish to obtain
 - a. In the **SEARCH WINDOW** type the PDB file number for this molecule, 1ZEH. Then click on the **EXPLORE** button.
 - b. A **SUMMARY INFORMATION WINDOW** appears giving you pertinent information regarding the selected PDB file. Click on the **DOWNLOAD/DISPLAY FILE** selection. (Second selection on the left side of the screen)
 - c. A **DOWNLOAD/DISPLAY FILE** window appears. Scroll the window down using the scroll bar on the right side of the window. Locate the second screen in the lower half of the screen labeled **DOWNLOAD THE STRUCTURE FILE** . Select the second choice **NONE** and **PDB** . This allows you to download a non -compressed PDB version of the file.
 - d. When prompted, save the file to an appropriate location on either your hard drive or floppy disk. Write the name and extension assigned to this file in a safe place for future reference.
3. Exit PDB by closing the window
4. Sign off the Internet.

*You now have a PDB file named insulin and saved under an appropriate name. Remember this point... as you move through construction of a kinemage **NEVER** save a file under the same name as the downloaded PDB file. Consider the original downloaded file as an archive to which you may return over and over to use without altering the original PDB format.*

Preparing a Simple Kinemage

In this lesson you will learn how to build a simple image from the downloaded PDB file. We will attempt to construct the insulin dimer in such a manner as to illustrate some simple characteristics through a general kinemage.

1. **OPEN THE PREKIN PROGRAM**
 - a. Locate the icon for the program and double click on it.
 - b. When you see the window with three buttons, click on the **PROCEED** button.
2. **OPEN THE PDB FILE**
 - a. Locate the PDB file for insulin that you have downloaded (remember you wrote the name down for this purpose). Click on it to enter it into the command box.
 - b. When prompted for a name to save it by, choose a convenient name that you will

remember (write it down as well). PREKIN is now going to prepare a file according to your input commands and save it under a name you select. Remember, DO NOT SAVE IT WITH THE SAME FILE NAME AS THE ORIGINAL PDB FILE NAME. Something like **INSULIN1** will do nicely. Click to save the file.

3. SELECTING THE STARTING RANGES

- a. The next screen you will see contains a list of choices for the method by which the molecule will be built. The screen should be titled **STARTING RANGES** if using a PC or **NEW PASS** if using Mac. You have 5 choices on this screen.
 1. **BACKBONE BROWSING SCRIPT**: the program will build a simple chain of all alpha carbons, disulfide bonds, and het groups other than water.
 2. **SELECTION OF BUILT - IN SCRIPTS**: the program will ask you to choose from one of nine possible arrangements of components.
 3. **NEW RANGES**: the program will build according to your selection of components and the range of those components.
 4. **FOCUS ONLY**: the program will select only components within a specified radius or point.
 5. **RANGE LIST FROM EXTERNAL FILE**: the script will be obtained from another file

FOR THIS LESSON WE WILL CHOOSE OPTION ONE

*click on the first button **BACKBONE BROWSING SCRIPT** then click **OK**.*

- b. PREKIN will now build the molecule according to this scripting demand. Wait until the program completes its task.
- c. Depending upon how PREKIN and MAGE are linked you may be sent directly to the MAGE program and the kinemage for this molecule will appear.
- d. If this does not happen then open the MAGE program, click on file from the menu bar and then select open from the menu. Locate the new file created by PREKIN (remember what you have named it) and select it. MAGE will then display the kinemage.

AT THIS POINT YOU MAY VIEW THE KINEMAGE AND EXPLORE THE FUNCTION COMMANDS.... REFER TO THE MAGE TUTORIAL ONE PORTION IN THIS DOCUMENT...WHEN DONE RETURN TO THE FOLLOWING SECTION.

SELECTING PARTICULAR COMPONENTS FOR A KINEMAGE

In this lesson you will learn how to run PREKIN in such a manner as to create a custom image built to your specifications. We will build the insulin molecule again but this time we will focus on showing the main chains and hydrogen bonds of the monomer.

1. OPEN THE PREKIN PROGRAM

- a. Locate the icon for the program and double click on it.

- b. When you see the window with three buttons, click on the **PROCEED** button.
2. **OPEN THE PDB FILE**
 - a. Locate the PDB file for insulin that you have downloaded (remember that this is the **ORIGINAL** name of the PDB file, the file you were told to keep as an archive).

Click on it to enter it into the command box.

When prompted for a name to save it by, again choose a convenient name that you will remember (write it down as well). Since this is the second insulin molecule we are building we will name it **INSULIN2**. Click to save the file.

3. **SELECTING THE STARTING RANGES**
 - a. This time we will not choose the first option....we would like to see the main chain of the monomer along with the hydrogen bonds. Select the second button **SELECTION OF BUILT IN SCRIPTS** . This will allow us to manipulate what we desire to see in the kinemage. Click **OK** .
 - b. Remember this screen? You have nine options to choose from. Choosing the first would give you the same results as the previous lesson (it gives the same backbone browsing results). We do not wish to see that again. The second option will provide us with the desired results. Click on the second button **mcHb MAINCHAIN AND ITS H-BONDS** , then click **OK** .
 - c. The next screen gives us the option of choosing the number of subunits of the molecule for which we wish to see the main chain and hydrogen bonds applied. Rather than seeing all four components of the dimer (that is the A, B chains and A', B' chains). We will only select the first monomer (A, B). Leave the **FIRST BOX** as it is with the number one, but change the **SECOND BOX** to the number **TWO**. Then click the button for **DO SUBUNITS FOR RANGE GIVEN ABOVE** . Click **OK**
 - d. Again **PREKIN** will build the molecule as you have described it.

WHEN IT IS COMPLETE YOU MAY AGAIN ACCESS THE KINEMAGE. ON THE PREKIN FILE MENU SELECT LAUNCH MAGE WITH THIS KIN AND EXPLORE THE KINEMAGE USING WHAT YOU HAVE LEARNED FROM PART I OF THE MAGE TUTORIAL. RETURN HERE TO CONTINUE WITH THE TUTORIAL.

USING PREKIN TO ALTER RESIDUES

There are few small variations to specific amino acids on the peptide chains that account for the insulin differences between different organisms. As an example, the following kinemages will demonstrate the small variations that account for the difference between sheep insulin and that which represents both pigs and whales. When threonine and isoleucine at sites 8 and 10 respectively are replaced by alanine and valine, the insulin monomer representing pig or whale then becomes the insulin of sheep. This being the only difference in the entire molecule.

1. **OPEN THE PREKIN PROGRAM:** We will first generate a kinemage to represent the insulin of pigs and whales.
 - a. As in previous lessons you need to access the archive file of insulin and save an output

- file under another name. This time call it **INSULIN3**.
- b. In the first PREKIN screen choose the second option called **BUILT IN SCRIPTS** then click **OK** .
 - c. In the screen which appears next, choose the 4th option **aasc+R** which will build the molecule with the alpha carbons as a chain, the amino acids on that chain. Click **OK** .
- In the subunit screen leave the box fields set to the default **1** and click the second button. This will only create the first chain of the monomer. Click **OK**
- e. When PREKIN is finished launch the image in MAGE.

SEE THE MAGE TUTORIAL PART TWO FOR INSTRUCTIONS ON EDITING THIS IMAGE BEFORE PROCEEDING TO THE NEXT LESSON IN PREKIN.

IN THIS PORTION OF THE LESSON YOU WILL LEARN HOW TO CREATE ALTERNATE RESIDUES ON A CHAIN, AND THEN ISOLATE AND REMOVE THE INITIAL RESIDUES.

2. OPEN PREKIN AND CREATE ANOTHER INPUT FILE...SAVE AS "INSULIN4":

We will now change residue number **9** to glycine to represent the horse insulin.

- a. In the first screen choose **SELECTION OF BUILT IN SCRIPTS** and click **OK** .
- b. Choose the **caSS** option (first one). Click **OK**
- c. Choose **DO ONLY FIRST SUBUNIT** . Click **OK** .
- d. From the file menu choose **NEW PASS** .
- e. Choose the **NEW RANGES** button. Click **OK** .
- f. In this screen you have the option to identify that range of the chain that you wish to edit. We are going to change sites **8** and **10** by replacing the residues. For this pass, type in **8**, **8**, and **ALA** in the first three boxes. Then select the last button, **RO: ROTATION +/- MUTATION OF ONE RESIDUE** . This allows PREKIN to mutate the residue at site **8** from the original format of **THR** to the **ALA** you have just requested. To the right on this screen locate the set of buttons that begin with the word **ok** . Click on **OK ACCEPTS AND END RANGES** .
Now click **OK** .

For MacIntosh Users: At this point you will be prompted to access the PREKIN.LIB. The MacIntosh version of PREKIN does not automatically search for the replacement vectors. You must locate the directory where the library file is kept. It should be in the same place that you downloaded the PREKIN program. Select the file and proceed.

- g. Select **NO FOCUS AT ALL** , click **OK** .
- h. Select **DO ONLY FIRST SUBUNIT OF INPUT FILE** , click **OK**

We must now mutate site 10 on the same molecule from ILE to VAL. On the file menu select NEW PASS and repeat steps d through h .

YOU HAVE NOW CREATED A KINEMAGE OF INSULIN THAT SHOULD APPEAR VERY MUCH LIKE THE ONE BUILT AS INSULIN3, ONLY THE RESIDUES AT SITES 8 and 10 HAVE BEEN MUTATED. LAUNCH MAGE AND THE INSULIN4" FILE. PROCEED TO PART TWO OF THE MAGE TUTORIAL EDITING INSULIN 3". WHEN COMPLETED RETURN TO THIS PREKIN TUTORIAL FOR THE REMAINING LESSON.

USING PREKIN TO ILLUSTRATE A RIBBON SCHEMATIC

In this lesson you will learn how to use another PREKIN `BUILT IN SCRIPTS` command. There are four possible configurations which center on the theme of a ribbon schematic. The default for the program is a simple ribbon with no secondary - structure information. To display beta ranges a ribbon arrow can be formatted. For helices a spiral configured ribbon is available. To demonstrate a coil there is yet another format. Remember the ribbon commands are extremely memory intensive and will slow down the action under MAGE considerably. Use ribbons wisely.

1. **OPEN PREKIN AND CREATE ANOTHER INPUT FILE...SAVE AS INSULIN5"**
 - a. In the first screen choose `BUILT IN SCRIPTS` click `OK` .
 - b. Choose the eighth selection down on the next screen `RIBBON VARIABLE WIDTH DEPENDENT ON LOCAL CURVATURE` . Click `OK`
 - c. For this tutorial we will accept all the defaults on the next screen. You may, however, change these to your suiting as you become more familiar with the program. Click `OK` .
 - d. In the next screen choose the third button `DO ALL SUBUNITS OF INPUT FILE` so that the entire dimer will be represented as ribbons. Do not change the values in the boxes since we are doing all subunits.

You can now view this ribbon version of insulin through the MAGE program.

THERE ARE MANY MORE AVENUES FOR YOU TO EXPLORE USING THE PREKIN PROGRAM. THROUGH THIS TUTORIAL YOU HAVE LEARNED HOW TO FIND AND USE EACH MAIN SCREEN. IT IS UP TO YOU NOW TO INVESTIGATE OTHER OPTIONS. REMEMBER TO ALWAYS SAVE THE INPUT FILE AS SOME USEFUL NAME SO THAT YOU CAN ALWAYS REFER BACK TO THAT MAGE.KIN FILE.

MAGE TUTORIAL

PART ONE

THE BASICS

This tutorial is a brief look at the functions of the menu bar and button bar of a simple kinemage as displayed through the MAGE program. It has been designed to follow along with the PREKIN tutorial. You may use it independently of the PREKIN tutorial.

KINEMAGE INSULIN1"

In the first lesson of PREKIN you developed a kinemage and saved it as INSULIN1. This lesson refers to that file. You will learn some of the more common functions of the MAGE screen. The less used functions will be left out of this simple tutorial. If you already have MAGE open and the file is available then skip step 1.

1. Open the MAGE program by double clicking the icon for the program.
 - a. On the menu bar click on **FILE** and then **OPEN** . Select the file you have saved called **INSULIN1"** and open it.
2. The screen is composed of several areas of information.
 - a. To the left is a white window with text and it is titled **TEXT** . You can click anywhere on the screen to bring it to the forefront of the screen. Click on it to try this. Now you can reduce the text window size to better view the remainder of the screen. To do this move the cursor over the right edge of the text window until you see a double arrow, click and drag the screen closed enough to view the kinemage better.
 - b. To the bottom of the screen is another white screen and it is titled **CAPTION** . You can also change the size of this window with the same procedure. For both windows you can also change the horizontal size by the same method. Arrange both windows so that you can see both all three windows equally well.
 - c. To the right of the screen is a list of **BUTTONS** and some **SCROLL BARS** . Remember, when you prepared the kinemage through PREKIN you requested certain attributes for the image. These were prepared and buttons for each were added for you to control the image. The scroll bars are automatically added to any kinemage.
 1. Notice the way the buttons line up vertically. They are not in a straight line. This is because as you created the kinemage in PREKIN certain attributes took dominance over others. The left justified buttons are **DOMINANT** . Any button under it before the next dominant button is controlled by the dominant button.
3. Determining the function of **BUTTONS** and **SCROLL BARS** .
 - a. Click on and off each of the dominant buttons and notice what changes in the kinemage. Remember that Insulin Dimer is composed of two sets of chains. Each dominant button controls the view of those chains. The last dominant button controls the **het** groups (the sugars and metals in this case). Try clicking the buttons under the first dominant chain. Can you determine the hierarchal

arrangement of the buttons? There is also a button called **MARKERS** which allows you to mark any point on the kinemage you select. Click it on and click a few spots on the kinemage.

- b. The **SCROLL BARS** to the right control functions regarding the entire image.
 - 1. **ZOOM** allows you to move in or out for selected areas of viewing
 - 2. **Z SLAB** and **Z TRAN** allows you to control the depth of field that you see.

4. Moving the kinemage

- a. **MAGE** makes it possible to rotate the image in any of the three dimensions. This can be accomplished one dimension at a time or a composition of three at once.
 - 1. Place the arrow (your cursor pointer when on the kinemage) at the extreme top center of the kinemage. Click and hold the mouse. Notice the part of the arrow change to a white box with a black dot. While holding the mouse button down move left or right. This function allows you to rotate the kinemage in one dimension. Notice you can only go so far before the motion is reversed.
 - 2. Place the arrow on the extreme left or right center. This time when you click and hold you should see a plus sign on the arrow. Move the mouse straight up or down and notice the dimensional move.
 - 3. If you move the cursor to any other position on the kinemage and click the arrow again is present. You can move in multidimensional directions by controlling the mouse movement.

5. The Menu Bar

- a. If you wish to return the kinemage to the original position there is a function on the menu bar to do so. Locate the **VIEWS** title on the menu bar above the kinemage. Click and select view one. Notice that the view returns to the original position.
- b. Under the **DISPLAY** menu there are three options for line width on the kinemage. Click each one and notice that the line thickness changes.
- c. Under the **EDIT** menu click on **CHANGE COLOR** . Notice on the button bar that
 - a new button has been created and it has been clicked on.
 - 1. Now click anywhere on the white strand (chain A). Notice that a color bar selector appears to the left on the screen. Depending on whether you are using a PC or MAC platform the selection process will vary slightly.
 - 2. For PC select the color purple. Notice that the entire chain has changed.
 - 3. Look at the top of the color selection window. You will see two buttons. The **LIST** button is probably checked... this makes the entire chain change color. Click the **POINT** button and notice only one small portion of the chain changes color. You can control how much of a chain or residue is color changed with these functions.
 - 4. Click **CANCEL** on the color selection window to exit that function.

NOTE!!! if you do not click off the color selection button on the right side of your screen you will be continuously prompted to change colors every time to click in your kinemage.

- d. Under the **EDIT** menu click on **PRUNE** . Again, notice that new buttons have been created to the right of the screen. The **PUNCH** button should be clicked on automatically.
 1. Now carefully select one of the ring structure by click on one end of a line. Notice the line disappears. This is a dangerous tool but very useful. Be very careful when using it to remove certain vectors from the kinemage.
 2. The **PRUNE** function is much the same as **PUNCH** . It can remove larger sections at a time. You should use the **PUNCH** when in doubt.
- e. Under the **EDIT** menu click **DRAW NEW** . Notice that a new button has appeared

at the bottom of the right side button list. It is called **DRAWLINE** . It should be already clicked on. For now click it off.

 1. Rotate the kinemage so that you have a good view of one of the het groups and the nearest chain.
 2. Click on the end of the het group. Notice that in the bottom left corner of the screen the identification of the group is shown along with a number.
 3. Double click on that point, the number should revert to 0.000.
 4. Now click on the nearest chain point. Notice the number has changed. This is the distance from the het group to the chain.
 5. Find a het group - chain distance of about 4 - 6 angstroms.
 6. Now click on the **DRAWLINE** . Click the two points from het group to chain. A line should have been drawn. This is useful tool for drawing bonds.

Under the **EDIT** menu click **DRAW NEW SETUP** . The screen you see allows you to vary the **LENGTH** of the line you draw. In the box type .85 and do not click on any of the buttons. Notice the descriptions at the bottom of the window. These relate to the buttons that will show up after you click ok. So now click **OK** .

1. Make sure **DRAWLINE** button is clicked off for now.
2. Rotate the kinemage so that you can see the het groups of atoms signified as balls. These are the zinc and chloride ions responsible for creating hexamer molecules with other monomer insulins.
3. Now click on the **DRAWLINE** button and click on each ball nearest each other. You should see a line formed, but the line is shorter than the distance between the balls. You have drawn it 85% of the distance.

*If you accidentally draw a line that you did not intend there is a way to erase it, providing it is the last operation you have made. A button called **ERASE LAST** at the bottom of the button panel will perform this operation.*

- g. Choose **VIEW** from the menu and select view 1 again. **MAGE** allows the viewer to select many views of the kinemage and store them for easy retrieval.
 1. Rotate and zoom the image until you have a suitable view of the four chains from a side perspective.
 2. Click on the **EDIT** menu and select **KEEP CURRENT VIEW**
 3. The next window allows you to select the number of the image and a name. It usually defaults to the next number. If you desire to overwrite

any previous view with a new view simply type the number you wish to overwrite.

1. VIEW ID allows you to name it with a short (9 characters maximum) descriptive identity.
 2. Leave the buttons as they are. Click OK
4. Rotate the kinemage then select VIEWS and view 2. Your saved view should reappear.
- h. Notice that the dominant buttons are labeled by the original PDB file name. There is a function which allows you to change the names of the buttons, as well as hide them, and switch their dominance.
1. Under EDIT on the menu bar select SHOW OBJECT PROPERTIES .
 2. Click on the first chain of the dimer. This should be the one colored white and to the extreme right when viewing kinemage view 1.
 3. The window that appears allows you to manipulate the functions as listed above. In the first box remove the existing name and type A CHAIN . Do not click any other buttons. Click ACCEPT CHANGES .
 1. Notice what has happened to the first button name in the button list.
 4. Now do the same operation for each of the remaining chains using names B CHAIN, A CHAIN, B CHAIN. See that the buttons in turn change their names.

Although we will not use the APPEND FILE function found under the FILE menu, it is worth mentioning for future use. This function allows MAGE to literally write one KINEMAGE over the top of the other. This is useful for comparing partially similar sequences of molecular structures. It is important to remember that in order for this to work successfully the two files must have the same COORDINATE VECTORS. You can not append two files which have come from separate PDB files. (Unless of course they have exactly the same coordinates, which is highly unlikely).

This concludes the MAGE tutorial for selected functions. There are others as well and, as you learn the program better, you can explore their functions. Quit the MAGE program but when prompted, select PROCEED NO WRITE . You do not want to save the changes you have made during this tutorial.

If you were working on the PREKIN tutorial prior to this MAGE tutorial, now is the time to return to the next lesson in PREKIN SELECTING PARTICULAR COMPONENTS FOR A KINEMAGE .

**MAGE TUTORIAL
PART TWO
EDITING INSULIN3**

In this part of the MAGE tutorial you will apply some of what you learned in the last lesson. You will prepare the INSULIN3 file for another upcoming activity. In this lesson you will use the PRUNE and CHANGE COLORS functions.

1. OPEN THE MAGE PROGRAM AND OPEN THE INSULIN3 KINEMAGE FILE.
 - a. In the EDIT menu select the PRUNE option. (Remember the difference between prune and punch.)

BEFORE you remove anything always TURN OFF THE PRUNE, PUNCH BUTTON check the identity of what you are about to remove.

1. Identity check: remember to click on the location and the identification presents itself in the lower left corner.

When you are satisfied it is a suitable removable item turn the button back on and remove it.

2. Remove all residues EXCEPT at sites 8, 9, and 10.
3. Using the SHOW OBJECT PROPERTIES function under the EDIT menu hide the SIDECHAIN buttons. To do this click on one

of

the remaining residues. When the window appears select NO BUTTON under the group name box. Click ACCEPT CHANGES

QUIT THE MAGE PROGRAM. WHEN PROMPTED TO WRITE OUT THE CHANGES, CHOOSE OK AND NAME THE FILE INSULIN3B . RETURN TO THE PREKIN LESSON USING PREKIN TO ALTER RESIDUES STEPS 2 AND 3.

**KINEMAGE TUTORIAL
EDITING A KINEMAGE FILE THROUGH A WORD PROCESSOR**

Once a kinemage has been created there are two ways that editing the file may be accomplished. You are now familiar with the process of editing through the MAGE program functions. There is also the ability to edit line by line of the actual program using specially selected commands. (Created for particular use in the MAGE programming language) This editing is accomplished through the use of a word processor. For this tutorial the Microsoft WORD processor is used.

1. OPEN THE WORD PROGRAM AND "INSULIN3B"
 - a. Components of a typical Kinemage file
Every file begins with the @KINEMAGE command even though there is text prior to this line command. If you type anything after the @TEXT and before

the @kinemage it will appear in the text box of the kinemage. You can also edit the caption box by typing the extension after the @CAPTION command

Special Note: In the MacIntosh platform of MAGE the text can be edited directly without the use of the word processor. There is a command on the menu bar which allows you to write the text out while still in MAGE.

- b. There are also a few more commands before the first set of vectors appear. The ones that concern us for now are @GROUP, @SUBGROUP, and @VECTORLIST.
 - a. @GROUP: this refers to the dominant button in the kinemage. It controls the parameters listed below it until the next @group is encountered.
 - b. @SUBGROUP: one lower level of dominance
 - c. @VECTORLIST: the actual list of xyz coordinate points for a plotted object
- c. What appears in between { } are identities of many related components.
 - a. After @group, etc. It is the name that will appear on the button panel. You can edit that name here as well as in the MAGE program (as you have already done).
 - b. Before the vector line. It is the name of the carbon and residue at that vector address
- d. Notice that you can also edit colors. On the end of the first @vectorlist line is a command for color to be white.
- e. The P and L commands refer to line segment start and stop points. The P is required as a start for a line segment if inside the vectorlist. The L is a command to continue to draw from previous point. Since the program defaults for continuing to draw the L command does not need to be manually placed.

As you scroll down the screen you will see repetitive patterns of @commands, labels, and vector lists. This is the language of the kinemage program. As you become more familiar with the use of MAGE, make use of the command list at the end of this tutorial.

2. TEXT, CAPTIONS AND LABELS

In the area just below the @text command erase the given information and type in some pertinent information about the insulin molecule as described within this tutorial.

- b. In the area just below the @caption command do likewise but be more specific with regard to this particular kinemage.

Locate the @GROUP command near the beginning of the file. It will be followed by {1zeh 1}. Remove this name and type in A CHAIN. Now the group button will display this name when the kinemage is viewed.

- d. You can also create labels for any part of the viewable kinemage. These labels will appear as text on the kinemage wherever you command the program to place them. Scroll to the end of the file and click the cursor on the end of the last line. Press the return key to create a new line.
 - 1. Enter the following information:

@labellist {A lbl } color=white master={rec labels}
{A CHAIN} -7.777, 17.654, 13.489

We have just added a command that will write the phrase A CHAIN at the end of the viewable chain in the kinemage. Lets explore each part of the command.

@labellist: The command to encrypt a label
{A lbl}: A directive for the actual name to be inserted as a label type.
Color=: Allows formatting the color of the label
master=: Allows a button to be made to control the label.
{A CHAIN}: The actual name of the label as directed by the second command.
The numbers: The coordinates for label placement.....they were taken from the first vector line since this is where we wanted the label to be placed.

SAVE THE FILE. SELECT SAVE AS AND NAME THE FILE INSULIN3C". CLOSE THE FILE. YOU WILL BE WARNED THAT THE DOCUMENT CONTAINS FORMATTING THAT CAN NOT BE WRITTEN AS TEXT. CHOOSE THE YES OPTION TO SAVE. IN THE NEXT SCREEN SAVE AS TEXT ONLY THIS IS VERY IMPORTANT!!!!

2. ANIMATION

In this portion of the lesson you will learn how to animate or manipulate certain portions of the kinemage upon command of a button. This is called ANIMATION .

- a. Open the file "INSULIN1" in the word processor.
- b. Locate the command line @GROUP {A CHAIN} near the beginning of the file.
 1. At the end of this line type ANIMATE.
- c. Locate the next command line @GROUP {B CHAIN}.
 2. Again type the word ANIMATE after the line
- d. Repeat this procedure for each @GROUP line.

Save the file under the same name.

- f. You can view this kinemage in MAGE and notice that there is an ANIMATE button at the bottom of the button list.
 1. Click the button several time and notice each portion of the chain in the dimer including the het groups appear in succession.
 2. You can also view the entire kinemage at once by clicking on each dominant button.

In this next portion of the lesson we will perform some complex copy, cut, paste, delete, and original insertion procedures.

Open file "INSULIN4". In this file we wish to animate the kinemage so that either the insulin for pig and whale, or that of sheep is visible.

- g. Locate the line @group {1zeh 1} .
 1. Inside the brackets rename this pig/whale.

2. At the end of this line type ANIMATE
- h. In the next line @ subgroup {sidechain} make the following changes
 1. Inside the brackets rename this to THR
 2. At the end of this line type DOMINANT MASTER = {SUBUNIT 1}
- i. In the next line @vectorlist {sc} color = cyan make the following change.
 1. Remove the brackets and SC contents.
- j. After the vector line there is a line @group {1 zeh}.
 1. Remove the ENTIRE LINE.
- k. In the next line @subgroup {rotate 10} dominant make the following changes
 1. Inside the brackets rename this to ILE .
 2. At the end of this line type MASTER = {SUBUNIT 1}
- l. In the next line @vectorlist {rol} color = sea 1bondrot -61.7 make the following changes.
 1. Remove the brackets and rol contents.
 2. Remove 1bondrot -61.7".
- m. After three lines of vectors there is a line @vectorlist {rotation} nobutton 0bondrot 0.0.
 1. Remove this ENTIRE LINE
 2. In its place type the following @ group {sheep} animate

Open file INSULIN3c . We will be copying some blocks of vector commands from this file and inserting them into the INSULIN4" file we are currently editing.

- a. Locate the line @subgroup {ser} dominant master = {subunit 1}.
 1. Copy from this line through {og ser a 9} -12.881 7.239 17.064" (a total of five lines)
 2. Open the window for the file INSULIN4"
 - a. Locate the line @subgroup {thr} dominant master = {subunit 1}
 - b. Count from this line down through the line {cb ala a 8} L -8.315, 8.674, 18.863"
 - c. After this line paste in the clipboard copy from INSULIN 3c
- b. Return to the INSULIN3c file window.
 1. Locate the line @subgroup {ile} dominant master = {subunit 1}
 - a. Copy from this line through the line {cg2 thr a 8} -7.092 9.553 18.982"
 2. Open the window for the file INSULIN4" again.
 - a. Locate the line @group {sheep} animate
 - b. After this line paste in the clipboard contents from INSULIN3c

SAVE THIS FILE UNDER THE SAME NAME INSULIN4".

3. ADDITION OF OTHER KINEMAGES TO EXISTING FILE

This portion of the lesson will instruct you as to how to make several kinemages part of one presentation (as opposed to opening new files for each kinemage).

- a. Open the file INSULIN1" and then open INSULIN2". You now should have two windows available in the word processor, one with each file.
- b. You now want to add the file INSULIN2" to the end of the INSULIN1" file.
 1. Use the SELECT ALL function under the EDIT menu in the

- INSULIN2" window
2. Under the EDIT menu select COPY .
- c. Under the WINDOW menu select the INSULIN1" file.
1. Place the cursor at the END of the entire INSULIN1" file.
 2. Under the EDIT menu select PASTE . The entire file has now been written after the first file.
- d. Close the INSULIN2" window. You do not need it anymore.

We will now add a few more kinemages to the INSULIN1" file.

- e. Using the same procedure again, copy and paste INSULIN3C , INSULIN4 , AND INSULIN5" TO THE END OF INSULIN1" AND INSULIN2".

SAVE THE FILE AS MYNAME (PLACE YOUR NAME IN THE FILENAME). REMEMBER THAT WHEN YOU EXIT THE WORD PROCESSOR YOU WILL BE PROMPTED TO SAVE TEXT. SAVE AS TEXT ONLY .

4. POLISH UP THE PRESENTATION

You now have a nice presentation of the insulin molecule. At this point you can finish editing the caption portions of each kinemage in your file. EDITING THE SUBSEQUENT @TEXT WINDOWS WILL NOT CHANGE THE KINEMAGE PRESENTATION. When several kinemages are added together as one file, only the text window of the first kinemage will be displayed regardless of what kinemage in that file is presented. The @CAPTION windows, however change with each kinemage presentation. Also walk through your program in MAGE and determine if any buttons need to be renamed, or removed. You might add LABELS to the chains of INSULIN2" by accessing the word processor.

WHEN YOU ARE FINISHED THE FILE SHOULD CONTAIN FIVE SEPARATE KINEMAGES. VIEWABLE THROUGH THE KINEMAGE FUNCTION ON THE MENU BAR OF THE MAGE PROGRAM!!!

KEYWORDS AND LINE COMMANDS

Key to annotation flags in this file:

-> should not be used by authors in distributed kinemages.
\$ annotations for programmers (authors should not use features \$ marked).
\$-> programmers need NOT use these: will NOT be supported in the future.
(I stands for an integer, 'f' for a floating-point number)

Note: Keywords can have additional characters beyond those the MAGE parser actually recognizes. As a matter of style and long term robustness, however, authors should use fully spelled out keywords. Official definition of keywords are in lower case; however, the MAGE parser converts potential keywords, so that upper case should work. All MAGE keywords are defined to start with \n@: i.e. @ is the first character of a new line. This will avoid problems with email addresses where @ is used but never as the first char of a line. However, early MAGE versions were permissive about this, so there may be trouble with early kinemages.

TEXT Parser (MAGEFILE.c)

@text Whatever follows is put in TEXT WINDOW, until @kinemage or EOF. Valid only when MAGE first opens a kin file. No other @keywords recognized until an @kinemage is encountered, except @MAGE f where f is the MAGE version. This keyword will be recognized even if it comes before @text when MAGE first opens a kin file. Carriage returns are omitted except for paragraphs or lists, to allow text wrap when windows are resized.

Within text region:

{ } Delineates "go to" setups: when reader clicks inside, MAGE goes to kinemage & view specified and turns on or off master buttons as specified. e.g. *{Kinemage 6, View 2, with master= {side ch} on}* short form: *{Kin 6, v 2, m= {side ch} on}*

{Q: } Delineates "Questions": when reader clicks inside, question dialog box comes up. Note: MAGE adds question number and a line where answer will appear. e.g. *{Q: What is your name?}*

@MAGE f MAGEVERSION for version control comments

@kinemage I Starts a new kinemage; numbers should be positive, unique, and increase monotonically.

KINEMAGE Parser (MAGEINPT.c)

MAGE filters the input character stream: passes only tab,LF,CR control characters, all ASCII characters above 32 and less than 127. This should allow a kinemage to be interpretable no matter how saved by a word processor program, but it is a lot better to save kinemages as plain text.

@kinemage I Ends current and starts a new kinemage; numbers should be positive, unique, and increase monotonically.

@MAGE f MAGEVERSION for version control comments, can go anywhere.

@caption What follows goes in CAPTION WINDOW, until next valid keyword.

Input controls determine what is in the GRAPHICS WINDOW and adjust the appropriate items in the menus and button panel:

@onewidth Makes all lines 2 pixels wide (if omitted, width depends on z).

@thinline Makes all lines 1 pixel wide (good for superpositions). Default is multi width depth cueing from 1 in back to 5 closest to viewer.

@perspective Replaces normal orthographic projection. Kinemage style guide: do NOT use for molecules, use with geometric objects where the eye expects perspective.

@whitebkg White background, not black (uses alternate color palette)

@whiteback: synonym

@compare Makes side-by-side comparison of animate groups.

@flat Makes mouse-drag control 2D scrolling instead of 3D rotation

@xytranslation:synonym. Use one of these for authored 2D kinemages; readers can toggle by scrolling mode with keyboard "f".

\$ @multibin I Archaic: once used to improve fineness of hidden-line removal, now optimized and built in. Do nothing, but recognize as valid, since lots of old files use @multibin.

The section below distinguishes things that are the author's business to specify for how a kinemage should be viewed, at least initially, from things that are reader's choice for viewing. E.g.: line quality (thinline, etc.), and perspective are author's choices, while stereo or enlarged font size are a reader's choice. The idea that readers would develop preferences for viewing kinemages in general seemed reasonable when associated with limitations of early computers, but has now been rejected because the reader should try a kinemage, at least to start with, the way the author specified.

-> @keepstereo Obsolete: Invokes stereo, for rest of session (can be turned off manually). Authors should NOT set this in a distributed kinemage; readers can still toggle keepstereo using keyboard "s".

-> @stereoangle f Changes stereo angle (default is 6 degrees, wall-eye). Authors should NOT set this in a distributed kinemage. Readers can toggle cross-eye/wall-eye using keyboard "c" or change parameters using menu.

-> @keepthinline Obsolete: Invoked thinline for rest of session (speeds rotation, esp.on early PC's). Authors should NOT set this in a distributed kinemage; instead use single-kinemage form.

@thinline. Readers can select thinline using keyboard "t".

-> @keeperspective Obsolete: Invoked perspective, for rest of session.
("@keeperspec" actually recognized by parser) Authors should NOT set this in a distributed kinemage; instead use single-kinemage form: @perspective. Readers can toggle perspective using keyboard "p".

@noscale Initial scaling and centering will not be done. (Watch out!)

@scale f Reads in scaling factor to be used, and initial scaling and centering will not be done. (Watch out!)

@zclipoff Kinemage comes up with zclip OFF. (Lets zslab control depth cueing without clipping image). Zclip toggle button always available anyway.

\$ @ignore MAGE ignores everything until another valid keyword is found. (Historically used to let MAGE and CHAOS share input files.)

\$-> @pickcenter kinemage will come up with pickcenter ON (Beware!). Pickcenter toggle button always available anyway.

\$-> @zoomonly Obsolete, now dummied. Zoom slider always available anyway.

\$-> @sideclip Obsolete, now dummied. Sideclipping always done anyway.

@plotonly (before all points) Gives 2-D plot with no rotation, and no limit on number of points (after all points) Gives no-erase "kaleidoscope" image; rotation smears. Readers can toggle "kaleidoscope" using keyboard 'k'

Maximum number of views = 60 in MAGE 5.4, specified with the following information:

@viewid {string} String identifying startup view (max characters = 15) (appears on VIEWS menu)

@1viewid {}: synonym

@2viewid {string} String identifying View2 (appears on VIEWS menu)

....

@60viewid {string}

Zoom and span are exclusive alternative ways of setting magnification. On input, MAGE calculates scale so that zoom of 1.0 nearly fills window with entire object.

Span is good for precalculated closeup views, with a scale independent of overall object size.

@zoom f Scaling bias for startup view (default 1.0 nearly fills window)

@1zoom f: synonym

@2zoom f Scaling bias for View2

....

@60zoom f

@span f adjusts Scaling so that startup view is f units across window.

@2span f adjusts Scaling for View2 adjustment is to lessor of window height or width.

@60span f

@zslab I Zslab (depth of window in and out of screen) in startup view

@1zslab 1: synonym

@2zslab I Zslab for View2

....

@60zslab I

\$-> @zclip Archaic synonym for zslab

@center f f f X, y, and z of center of rotation and center of screen (in orig. coords.)

@1center f f f: synonym

@2center f f f Center for View2

....

@60center f f f

@matrix f f f f f f f f 3x3 orientation matrix for startup view on same or following line(s)

@1matrix f f f f f f f f: synonym

@2matrix f f f f f f f f 3x3 orientation matrix for View2:

@3matrix a11,a12,a13,a21,a22,a23,a31,a32,a33

@4matrix operation:

@5matrix x,y,z | a11, a12, a13 | x',y',z'

@6matrix X | a21, a22, a23 | =

@7matrix | a31, a32, a33 |

@60matrix f f f f f f f f

@master { } Master declaration: lets author specify order of master buttons no matter what order they are actually first used. Otherwise, order is that of first use. Cosmetic - not necessary, but convenient for standard layouts like presentation of contact dot calculations.

@lens f Sets radius of sphere of enhanced detail at center of image. Corresponding lists with parameter "lens" then pass their items for display only when within that sphere. This state is always in effect with a default radius (7.0 in MAGE 5.4) for lists with the "lens" parameter. Readers can toggle this behavior with keyboard e

@command {...} String is read in, to be the command line presented in remote-update dialog as the possible command line to be sent.

@localrotation f f f f f f f f Rotation matrix applied to part of file. ("@localrotat" actually recognized by parser) ("@localmatrix": synonym). Coordinates are changed during input and stored changed.

@endlocalrotation Ends part to be rotated; use if more items follow ("@endlocalrot" actually recognized by parser). ("@endlocalmatrix":synonym)

Note that local transformations (and bond rotations) are the ONLY things in the kinemage format that need to be ended explicitly.

@localcenter f f f Centering translation: puts center of rotation at point ("@localprecenter":synonym). If a localrotation is specified, translation done first. Coordinates are changed during input and stored changed.

@localpostcenter f f f Centering translation: puts center of rotation at point. If a localrotation is specified, translation done after.

@endlocalcenter Ends scope of localcenter,localprecenter,localpostcenter. ("@endlocalcen" actually recognized by parser) ("@endlocalprecenter":synonym) ("@endlocalprecen" actually recognized by parser) ("@endlocalpostcenter":synonym) ("@endlocalpostcen" actually recognized by parser)

Note that this sequence:

@localprecenter x,y,z
@localrotation...
@localpostcenter -x,-y,-z
intervening groups,subgroups,lists, etc.
@endlocalcenter
@endlocalrotation

Will do rotation of intervening stuff around the point x,y,z. If modified kinemage is written out, it has the changed coordinates.

These font size controls should not be used by authors in distributed kinemages, since fonts vary

from machine to machine. They are useful for setting up a presentation on a particular computer, but usually reader (or lecturer) changes them from menu.

@fontsizeinfo I Font size for the information line (pointID & distance)

@fontsize label I Font size for labellist item

@fontsize word I Font size for wordlist items

\$-> @specialps Do non-clipped floating point recalculation for PostScript output, done now by PostScript output control dialog box. Obsolete: this is now how Postscript is always done.

@listcolordominant Lets author determine that list color will come up toggled so that it over-rides point color. Reader can always toggle this state with keyboard "l" (el).

("@listcolordom " actually recognized by parser)

->@projection Obsolete: sets flag so construct4 extends line or dot by cos of angle, thus plotting onto a plane (film) rather than a sphere. Now controlled in the construct dialog box.

\$->@constructdot Obsolete: sets flag so construct4 makes a dot rather than drawing a line to the constructed point. Now controlled in the construct dialog box.

Graphics: Hierarchical Display Objects

A redraw of the graphics does a pass through the hierarchy, displaying only those objects flagged as "on" (== checked button box).

@group { } [param...] High-level display object. Group parameters must be on same line as keyword. Button name in { }, up to 14 char., (max = indicator + 14), indicator space before name is blank or has animate indicator "*" or 2animate indicator "%". Its "button" has on/off indicating check mark box.

@subgroup { } [param...] Mid-level display object.

\$-> ("@set": obsolete synonym) Subgroup parameters must be on same line as keyword. Button name in { }, up to approx. 10 char. (max = 15 but limited by width of button panel). Its "button" (_indented) has on/off indicating check mark box.

@xxxlist { } [param...] List of low-level display objects. xxx == vector, label, dot, ball, sphere, triangle, ribbon, word. List parameters must be on same line as keyword. Button name in { }, up to approx. 9 char. (max = 15 but limited by width of button panel). Its "button" (_ _indented) has on/off indicating check mark box. Items in the list are on following lines.

Internally, points can be a type that overrides their list type, which is used by MAGE (in the "New Group" of reader-drawn objects) but is not at present specifiable in the kinemage file itself. In actual usage, the points in vector, triangle, & ribbon lists group into P,L,L sets like poly-lines, triangles, etc. Note that pruning actually works on poly-lines, so can wipe out a whole Calpha list!

Attributes of individual items in a list: {pointID}, P, < >, etc. precede the x,y,z triple to which they belong: [attributes] x, y, z. They can be either lower or upper case. Attributes not relevant to type of list are ignored. Each field within a list item must be ended: A number (x, y, or z) can be ended by a space, comma, or newline A ')' or '>' ends its field, for a PointID or comment. One-character attributes (eg 'U') need white space to end them except 'P' is a special case,

{pointID} Shown on info line when point picked. Up to 255 char, stored efficiently in offset, indexed array.

P (Point) begins new line segment (required if inside vectorlist, presumed at beginning of list)

L (Line) means vector drawn from previous point (L is the default, so not required)

M moveto == P synonym

D drawto == L synonym

U Unpickable, used on shortened or extended lines where distance measurements would be confusing (can be overridden with 'superpick' on Edit menu)

B Forces line shortening by radius given as this vectorlist param; alternate way of shortening vectors for an accompanying balllist.

< > Anything between < > is stored as comment attribute of the point. <> comment should precede its point triple, for correct writeout. Up to 255 char, stored efficiently in offset, indexed array.

\$ ' ' Anything between single quotes is ignored. Historical relic of shared MAGE and CHAOS files, unfortunately embedded in some distributed kinemage labellists.

All colors are their own attribute names, for point color which is by default dominant over list color - though that can be controlled by input keyword @listcolordominant or by toggle of keyboard "l" (el).

Line or point attribute of width:

width1 == thin

width2 == 2 pixels wide

width3

width4
width5
width6
width7 (limit is 3 bits, stored along with point's color info)

Color and width for a line is determined by the drawto point attributes. Radius of a ball or sphere at this point: $r = f$

@vectorlist { } [param...] list of lines, button name in { }
{ } P x, y, z Individual vectorlist item,
{ } L x, y, z pointID in { }, shown on info line when point picked
{ } L x, y, z
{ } P x, y, z { } L x, y, z (Can have more than one on a line)

@dotlist { } [param...] list of dots, button name in { } { } x, y, z Individual dotlist item,
pointID in { }, shown on info line when point picked

@labellist { } [param...] List of labels, button name in { }
{ } x, y, z Individual labellist item. The pointID in { } is written on screen as label at that point. PointID also shown on info line when point picked. Note that for MAGE 4.4 and later a < > comment, if present, is what will be shown on the info line]

@wordlist { } [param...] List of words; button name in { }.

{pointID} <wordy stuff> x, y, z Individual wordlist item . PointID in { }, shown on info line when point picked. String in < > placed on screen starting at x,y,z <limit of 255 characters; can have newlines, as below> {pointID} <string line 1 string line 2> x, y, z (Multiple lines accepted). Within string, < or > needs "\<" or "\>"; "{" will subscript contents and "}" will superscript

@ballist { } [param...] List of balls, button name in { }. { } x, y, z Individual ballist item (pointID in { }, shown on info line when point picked) Makes colored disk with small highlight disk to simulate a ball. Illusion requires impinging vectors to be shortened by ball radius. This is done automatically if ballist in same group, and before any vector lists with vectors to these points.

@trianglelist { } [param...] Overlapping list of point triples for triangles; first 3 make first triangle and each successive point defines a new triangle with the 2 previous points. Making a completely unconnected triangle currently requires starting a new trianglelist. If update rate in MAGE is too slow, trianglelist will temporarily be shown as vectors.

@ribbonlist { } [param...] Overlapping list of point triples as above, except that the triangles are dealt with in pairs which are given a common normal and are colored by a highlighted lighting model rather than depth-cued. If update rate in MAGE is too slow, the ribbonlist will temporarily be shown as "cross-tie" vectors. A backbone ribbon is shown in MAGE with both a vectorlist for the splines of the two edges and a ribbonlist of successive lines across between the edges. Rendered as rectangular segments.

\$ @vector == @vectorlist Short form is what parser recognizes,

\$ @label == @labellist but authors should use full form.

\$ @dot == @dotlist A number of old kinemages use short form,

\$ @word == @wordlist so need to preserve this

\$ @ball == @balllist

*PARAMETERS (optional) for Graphics Objects @group, @subgroup, @__list:
(in any order, between keyword and line end*

master= { } Master button, will control all objects with same { } in their master parameters. {12 char max} MAGE creates a button for each unique master= { }, which when toggled resets the show-flag for all objects with that particular master= { }. Master buttons have on/off-indicating check boxes, and appear on the button panel below heirarchical-object buttons and above markers, etc.

off Object will start out turned off (button box NOT checked)

\$-> "-" Obsolete symbol for "off"

dominant No buttons for objects below in heirarchy (for groups, subgroups)

nobutton No button for this object (esp. useful for groups)

animate Group in an animation series (only for groups; group button shows *)
2animate Group in 2nd animated series (only for groups; group button shows %)
("animat", "2animat" actually recognized by parser, so "animation" works)
ANIMATE (2ANIMATE) button created whenever there are 2 or more groups flagged with animate (2animate).

\$-> "*" Obsolete symbol for "animate" for a group. Obsolete symbol for "dominant" for a subgroup. Obsolete symbol for "off" for a list

moview= I Use view I (only for animate groups, and takes effect only when that group is turned on by an "animate" control)

instance= { } Repeats the previous list, subgroup, or group that had the unique exact name given in { }

< > Totally ignored comment if with parameters. Not stored, so lost when kinemage written out from MAGE. (comment stored only if in TEXT region, in a wordlist, or preceding an individual point triple). Can have "<" or ">" in text if done as "\<" or "\>"

lens Special display flag: points in this list will only show within center lens (default radius = 7.0, reset with @lens f). Subgroups and lists inherit this parameter.

static Makes this display object immune to rotation, scaling, view, etc., but can be turned on & off by buttons or animation, and adjusts with window size

Parameters only used with lists:

color= string (for any type of list) String is color name (default white). Currently defined color names (see Kin.1 of Demo5_4b.kin) are: red, green, blue, cyan, yellow, magenta, white, pink, orange, purple, sky, brown, gray, gold, yellowtint, sea, pinktint, bluetint, greentint, hotpink. Each color name == a number in a lookup table (in the above order). \$ rust==orange, skyblue==sky, seagreen==sea, paleyellow==yellowtint

The following wordlist parameters should not be used by authors until we get better descriptions and defaults for the fonts on various platforms.

\$ font= I (wordlists only) Coded integer for font to be used.

\$ face= I (wordlists only) Coded integer for font face to be used.

-> size= I (wordlists only) Coded integer for font size to be used.

Font size as parameter to a wordlist only affects that wordlist, so though computer specific, does not mess up general machine defaults.]

nozclip Avoids zclip discrimination for items in this list; used to let something stand out in a surround of clipped objects.

detail Puts this list under control of level of detail which is controlled by update rate, settable by a pull down menu.

radius= f (balllist) Radius of the balls in this balllist. (default = .2). Vector lists after balllists within a group will have appropriate vectors shortened to make the illusion that disks are balls.

\$ radius= f (vectorlist) "radius" to shorten vectors at a point flagged with a B. (see vectorlist param. B)

radius= f (labellist) Lets label "float" forward on surface of ball or sphere, but revert to center point if ball/sphere turned off

[width= I (vectorlist) linewidth overrides general linewidth control, & is in turn overridden by individual drawTo point's line width. NOT yet implemented in 5.4: prototyped for 5.5+

1bondrot f n = 1-9: Next two points define an axis. Then allows hierarchical rotation (level 1>2>3...) about that axis, of all following points until another rotation of the same level, or lower n, is specified. f is starting value of torsion angle (default = 0.0), used to position slider and as label above it. MAGE creates a slider to right of button panel for

each rotation; maximum is 10 rotation axes.

2bondrot f

9bondrot f

0bondrot f Ends scope of rotations.

(EOF or "@vectorlist {dummy} 0bondrot 0.0" ends scope of all rotations). Can be off, with no button, quite hidden.

Example of rotation setup for an Ile sidechain:

```
@subgroup {sc rotat}
@vectorlist {chi1} color= sea 1bondrot -162.7
{ca ile 13} P 10.458 10.692 -.195 {cb ile 13} 9.952 10.935 1.283
{cb ile 13} P 9.952 10.935 1.283 {cg2 ile 13} 8.509 11.472 1.366
@vectorlist {chi2} color= sea 2bondrot 87.1
{cb ile 13} P 9.952 10.935 1.283 {cg1 ile 13} 10.024 9.657 2.032
{cg1 ile 13} P 10.024 9.657 2.032 {cd1 ile 13} 11.44 9.503 2.645
@vectorlist {dummy} nobutton 0bondrot 0.0
```

In its capacity as a graphics workbench, MAGE recognizes a number of keywords and parameters used in experimental kinemages but not defined as part of the official kinemage format.

For example, hplot and vplot are used for successively plotting value of last two rotations (on command) onto a screen plane overlay, e.g. phi-psi plots. The parameters Lbrotoptions 1-10 are used for both precession geometry and docking, while @control, @parameter f, precession, selection, samescope2, etc. are needed for the non-Eulerian angles of a precession camera.

Therefore, the following parameters are reserved for such special uses:

screen, hplot, vplot, Lbrotoptions 1-10, precession, selection,
x-zrotation, x-ztranslation, phirotation, psirotation, samescope2-4

Similarly, the following keywords are reserved for such special uses:

@control, @parameter f, @beginselect, @endselect, @copyright, @title]