Introduction

This is a guide for data analysis of computational simulation data for IBAMR in the data visualization program VisIt. The guide was written using VisIt 2.5.2 for Mac, your results may vary from the exact steps here. Don't be alarmed; they should be pretty close.

Before visualizing data

You'll need to download the data from your simulation in the files that VisIt can read. These are in the `viz_IB2d/` folder. Download everything in this folder, and stick it somewhere you can find it.

Basics

Loading data

After opening VisIt, you'll want to open two databases that you'll use: the `lag_data.visit` and `dumps.visit`:

- Click the "Open" button on the VisIt window, listed under "Sources."
- Navigate to the `viz_IB2d/` folder that you downloaded previously.
- Select either `lag_data.visit` or `dumps.visit`. (You'll need to add them one at a time.)
- Repeat this process and and select the other file. You will probably not have to re-navigate to the `viz_IB2d/` folder.

Now, you should see under "Active source" a drop-down box with two options: `dumps.visit` and `lag_data.visit`.

Make a plot

After loading the data, you'll probably see a plot already made of the sub-levels from IBAMR. They will look like different colored bricks. You can delete this plot by highlighting `dumps.visit:Subset-levels` and then hitting the "Delete" button right above it. Most of the time, these just clutter your plot. If you don't want to remove them, click the "Hide-Show" button under "Plots."

The `lag_data.visit` file indexes the Lagrangian point data from the boundaries of your simulation. To plot out your **boundaries**:

- Go to Active source and select `lag_data.visit`. If you don't do this, then you will not see your boundary
- Under Plots, select "Add." A drop down menu will appear with several selections. (Many of these will be OK to use, but if they aren't and you select them, it will tell you that it isn't OK. Choose something else.)
- Select "Mesh" and then you should see a drop-down menu of the different boundary files from your simulation.
- Select each boundary separately. They won't appear at first, this is OK.
- After selecting a boundary (or all of them), hit the "Draw" button that looks like a pencil. This will produce your plots.

The `dumps.visit` file indexes all the flow-field data produced by the simulation. To plot variables having to do with **fluid flow**:

- Go to Active source and select `dumps.visit`.
- Under Plots, select Add. You will see the same drop down menu as for the boundaries, but many more of them will work with the dumps data.
- Here are some common things to plot. Under "Pseudocolor," select `U_x` for the x-component of velocity, `U_y` for the y-component of velocity, `U_magnitude` for the fluid speed. `P` is pressure and `Div_U` is the flow field's divergence. Under "Vectors" you can plot `U` which will give you velocity vectors.
- The program may ask you if you want to create a joint database between the dump data and the lag data. Click yes.
- Select the item you want. Again, they won't appear immediately and that's OK.
- After selecting the items, hit Draw button and your plots should appear.
- If you would like to view one plot at a time, say you drew both pressure and speed but can't see them both, the Hide-Show button will allow you to toggle between hiding and showing (but not deleting) one plot.

To change the attributes (color, style, labeling, etc.) on any variable of your plot:

- Select the plot you would like to change in the list under Plots. Double click that plot name.
- Change what you'd like about your plot in the pop up. I like to uncheck "Legend" because the legends look cluttery.
- Click "Apply" to apply the change to your plot.
- Click "Dismiss" if you are finished and would like the pop up to go away.
- Clicking "Post" will embed the pop up in the VisIt window.

View an Animation of the Simulation

Under "Time," there is a slider and play buttons. Hitting the play button will progress the plots through time. You can use the other buttons to navigate through time similar to other video players. The player will loop automatically, so hit the stop square if you wish the animation to stop.

Make a Movie of the Simulation

Sometimes, it's useful to make a movie that you can view in another, more standard video player for use later in a presentation or to show your boss or to remind yourself how the simulation looked qualitatively. You can make an animation movie in VisIt, or have VisIt output individual files that you can then stitch together to make a movie.

Follow the steps to make a mpeg movie within VisIt:

- Make your plot with all variables and flow data that you want to include in the animation, be sure that it is exactly the way that you wish it to appear in the animation. Most of the time, things like legends will need to be removed as to not clutter the plot.
- Make sure that your Active Window is the window of the plot that you are trying to make the movie of.

- From the top menu bar, select "File" and then "Save movie . .." This will open up a pop up window wizard that will help you out.
- Select "New simple movie" and then "Continue."
- Under "Format and Resolution" select the Format that you'd like. (VisIt makes OK mpeg movies, but if you are readying this movie for a presentation, it will be better quality if you output TIFF or PNG images and stitch them together in a video player like Quicktime Pro.) Select the movie size (if you care).
- Be sure to click the `->` button to add your movie to the queue. Hit Continue.
- Choose your length and speed parameters. Hit Continue.
- Choose an output director and base filename. (If you have a set of images as output, it will add sequential numbers after the base filename.) Hit Continue.
- Select whether or not you want an email when the movie is done. Hit Continue.
- Select one of the three options. Standard is "Now, use currently allocated processors." There have been some issues (off and on) with the other option, sometimes this doesn't work.
- Hit Done to start your movie. For Macs, this will open a terminal window and begin making the plots one at a time. If you have an image sequence, it will stop after making each image. If you are making an mpeg movie, it will then encode the movie after it is done with the images.

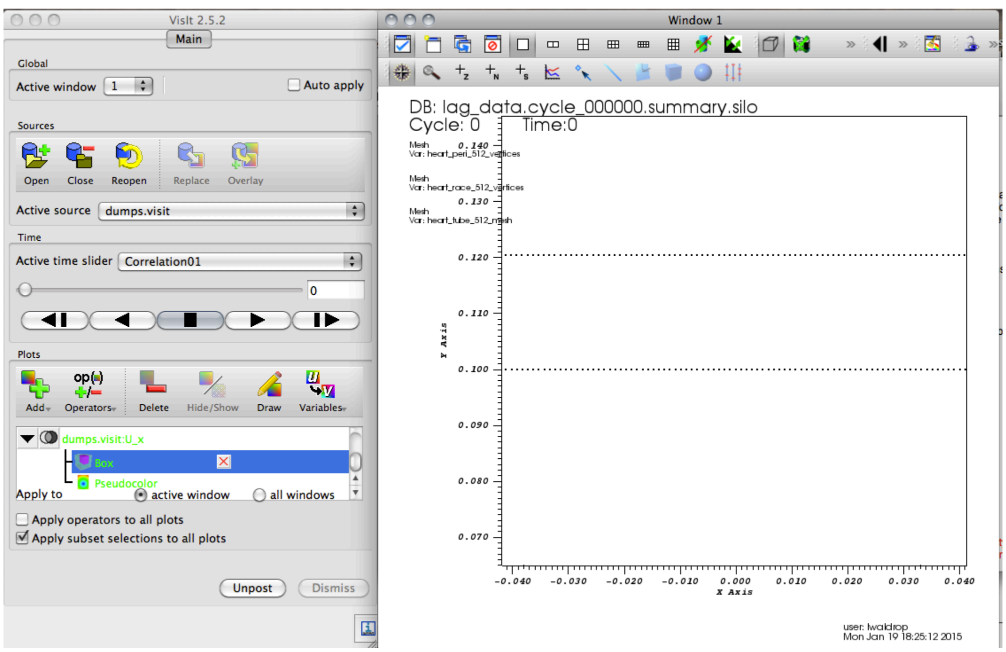

Figure 1: Screenshot of VisIt. From this zoomed-in view of the simulation geometry, the vessel goes from about 0.10 to 0.12 on the y-axis.

Data Analysis

This section covers the process for analyzing and exporting data from VisIt.

Best Practices for Data Analysis

Here is a sobering fact: ***In up to 80% of publications in which results could not be reproduced, the reason was not fraud or wrongdoing.*** Failure to properly record protocols and data storage issues were prominent amongst the reasons that data could not be reproduced in these troubled studies. ***This represents an incredible loss of time, effort, and money because people don't write things down when they do them.*** Which is more than a little ridiculous.

As researchers, we are responsible for leaving a clear, understandable record of what we did to reach the conclusions in our papers. When you leave your graduate lab, your adviser will have to be able to field questions about what you did and how you did it. This includes: documenting any code that you produce, leaving a clear protocol for reproducing analysis, and leaving notes so that all data can be easily found for anyone who wants to go looking.

Remember: transparency and reproducibility are the ultimate goals of all scientific endeavors.

A **written protocol** should include:

- The coordinates of each box or position where data were collected or analyzed. Include a labeled figure for best results (see Fig. 3 as an example).
- The method(s) by which the data were analyzed (*i.e.*, were the data in the box averaged? was the max value found? etc.).
- The step-by-step instructions for reproducing each output file. These should be detailed enough that

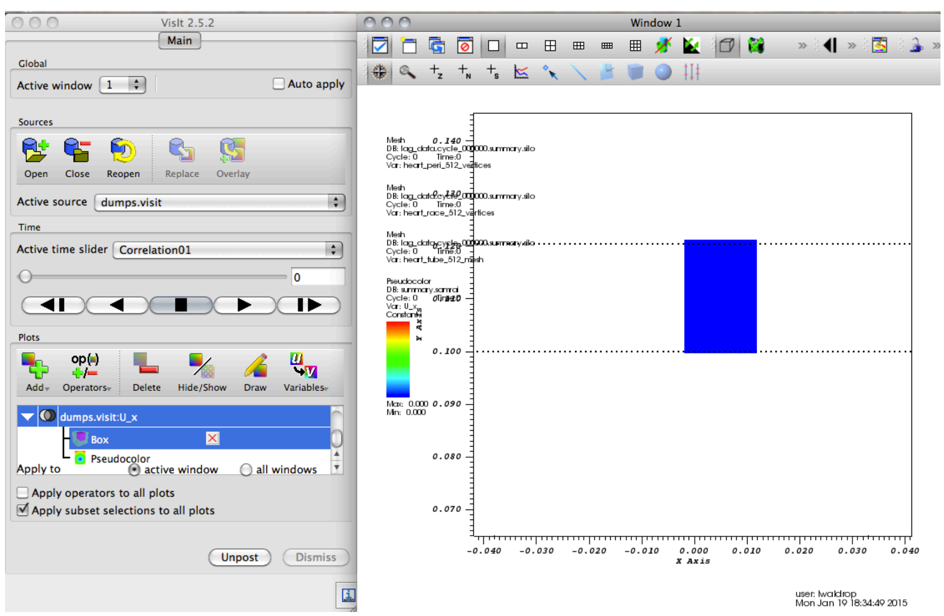

Figure 2: Screenshot of VisIt. From this zoomed-in view of the simulation geometry, only U_x is plotted within the restricted data-capture box.

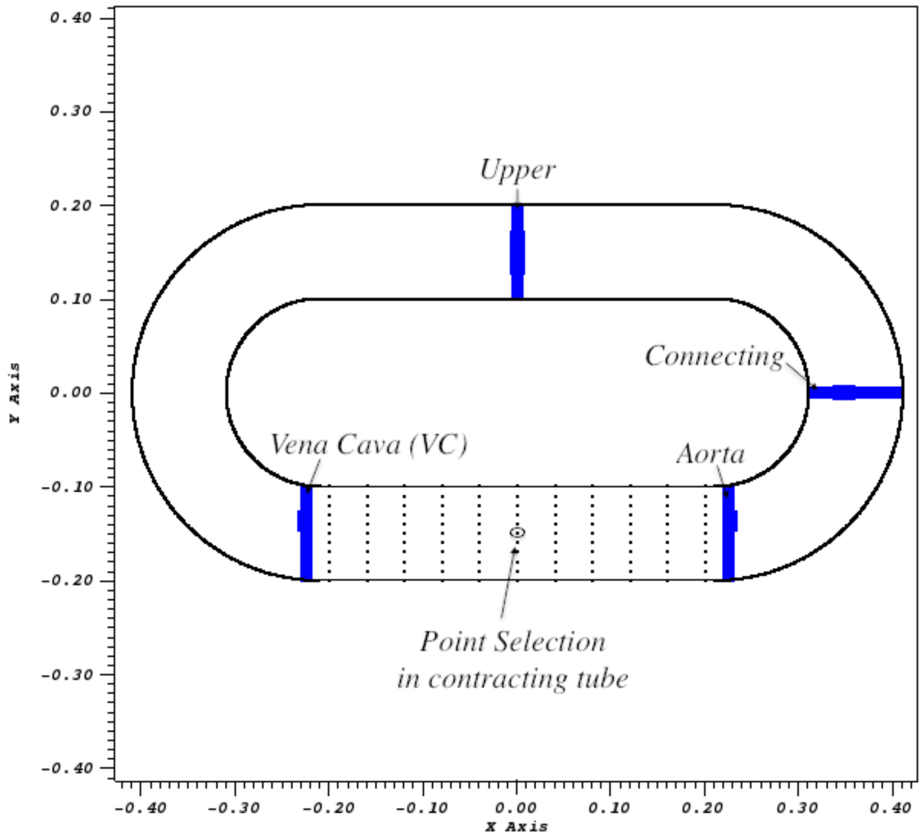

Figure 3: Screenshot of VisIt. Locations of vessels on plot of simulation geometry for the peristalsis paper.

an undergraduate with a basic familiarity with VisIt could reproduce your plots without consulting you directly.

- Any plots and the code (documented appropriately) for reproducing the plots.

Consider also leaving behind a **data sheet** in each folder for each simulation so there is a clear, handy record of the parameters that the simulation used and any other identifying information so that whoever comes behind you (or you several years into the future) won't have to remember where the parameters are located in your code or when you ran it.

Another good idea is to produce a more condensed version of your analysis protocol with the results of your analyses. This can be an excellent first draft of subsequent writing, and it will be a very effective tool in having a productive conversation with your adviser about your work. An **analysis report** can contain:

- A condensed version of the data analysis protocol on what was done at what positions. This doesn't have to be a step-by-step guide, but more like what you would write in the Materials and Methods section of a paper, although always err on the side of more details than necessary (rather than fewer). Include any statistics that you've done on the data thus far.
- Any graphs or plots that you think may be helpful or interesting. Write thorough and appropriate captions.
- Rough notes about interesting things found in the results. You can view this as a first draft of a results section, pointing out things that may be important.

- Any notes and comments that you get from your adviser regarding next steps. Saved as subsequent drafts, you can keep a clear record of your progression (which is good for you and your adviser).

Examples of a data-analysis protocol, data sheet, and analysis report should be included with these files. You are free to use and alter them to suit your needs!

Other general tips:

- **Err on the side of writing down too many details.** You can easily and quickly delete things later for a paper or thesis, but remembering things you didn't write down is much harder.
- **Record as you go along.** Don't wait to document your code or write your protocol at the end, even if this seems like the most efficient way to do things. You'll be tired and very sick of dealing with your data when you're done. (Seriously, you'll feel like throwing it out of a window.) You'll be in a rush, do a crummy job, and then kick yourself for cutting corners later. Additionally, spending a little time writing every day is the best way to be productive and publish a lot!
- **Pick a clear naming convention and stick with it.** Pick a convention that makes it easy for you and others to sift through your data quickly. For example, when running parameter sweeps, I keep copies of simulations in that sweep together in a folder (ex. "Wosweep" folder contains all the simulations in a sweep of Womersley numbers) and name each folder based on the parameter value at which the simulation was run (ex. "Wo10" and "Wo0.5" are folders within "Wosweep" and indicate the Womersley number at which the simulation was run, $Wo = 10$ and $Wo = 0.5$, respectively). Also, be sure to record all of this on a spreadsheet for quick look-ups.
- **Leave a READ ME file in the first directory to help people navigate your work quickly and efficiently.** This file should give a quick description of the study, explain your naming system, give the location of your spreadsheet explaining where each file is, give the name and location of your analysis protocol and the analysis report, and explain any simulations that were run and exist, but were not used. This will save hours of your own time later, and make it possible for someone with little knowledge of your personal habits to sift through your data later.
- **Clean up old files and simulations as you go along.** The worst thing is having too much extra data and not being able to figure out what was used and what was not used. Either delete it, or if you think that it will be used later, put it all in a clearly marked folder. If you wait to do this, it will be ten times harder to figure out what you were thinking.
- **Pretend like you won't remember anything in a week.** This is now a reality for some of us (ahem, me). It will eventually happen to you, too.

Restricting data capture with a box

It is sometimes useful to restrict your data analysis to a very small part of your total graph. In this case, we want to restrict the data analysis to a small box that covers one of the vessels of the circulatory geometry. For this, take note of the min and max x and y values that will form the corners of the box. The box should span the diameter of the vessel, and should be a small stripe along the vessel. It doesn't matter how wide the box is, so long as you use the same box for every step of the analysis (and ever simulation you analyze).

- Under the Plots section, you should see two check boxes labeled "Apply operators to all plots" and "Apply subset selections to all plots." Uncheck both of these boxes.
- For the cross vessels, add a plot of `U_x`.

- With `U_x` highlighted, either right-click or click the "Operators" above the list. There will appear a drop down list.
- Select "Selection" and then "Box."
- The label should turn green (meaning it is unplotted). Click the arrow on the left of the `dumps.visit:U_x` label and a list will appear. One will be the Box and one will be the Pseudocolor plot. Fig. 1 shows you what this list will look like after adding the box.
- Double click on Box and fill in the min and max for x and y axes. With the geometry in Fig. 1, we will use `min x = 0, max x = 0.01, min y = 0.10, max y = 0.12`. Be sure to hit Apply before exiting.
- Hit Draw to plot `U_x` only within this box. It should look like Fig. 2. If you don't see the colors plot only in the box, repeat the steps and try again.

Using Query to analyze data

Now that you have restricted data capture, we need to analyze the data. We can do this with the Query function under the top bar drop down menu "Controls." For more information about using Query, see http://www.visitusers.org/index.php?title=VisIt-tutorial-data-analysis.

- First, look at the top of the Main window and make sure your Active Window is 1 or whatever window your dump plots are in.
- Before your first Query, you need to select "Query over time options..." right below Query. In this pop up box, there will be a line of toggle buttons along the top with the options "Cycle, Time, Timestep." Select "Time" and hit Apply. You may Dismiss this box.
- Next, select Query. In this box, there is a list of options on the left that is long. Select the option you want. For this demo, we'll select "Average Value" from the list.
- To the right, in the box "Query parameters," select the checkbox that says "Do Time Query" and fill in the beginning and end times if you are not analyzing the whole simulation.
- Hit the "Query" button and wait for a graph. The graph should pop up with the requested information, like Fig. 4.

You can plot several queries on the same plot by repeating the steps above with a different variable (`U_y` or `U_magnitude`, for example) or a different query operation (Min or Max). Each line should show up as a different color.

Using Query to analyze data at a point

If you want to analyze data at a single point in your plot area, you can use Query to do this as well. In order to do this, you need to know the coordinates of the point of interest. With these coordinates, you can repeat the follow steps using `U_magnitude`, `U_x`, or another variable separately to create three lines on your graph.

- Be sure that "Time" is selected as a default in the "Query over time.." window.
- Open the "Query Window. Under "Display" menu, select "Variable-related." Then, under Queries, select "Pick."
- Set your variable using the drop-down box OR by typing in the variable. If you use the drop-down box, be sure to manually erase the previous variable (or the word "default") because selecting a new variable will not erase the last one used.

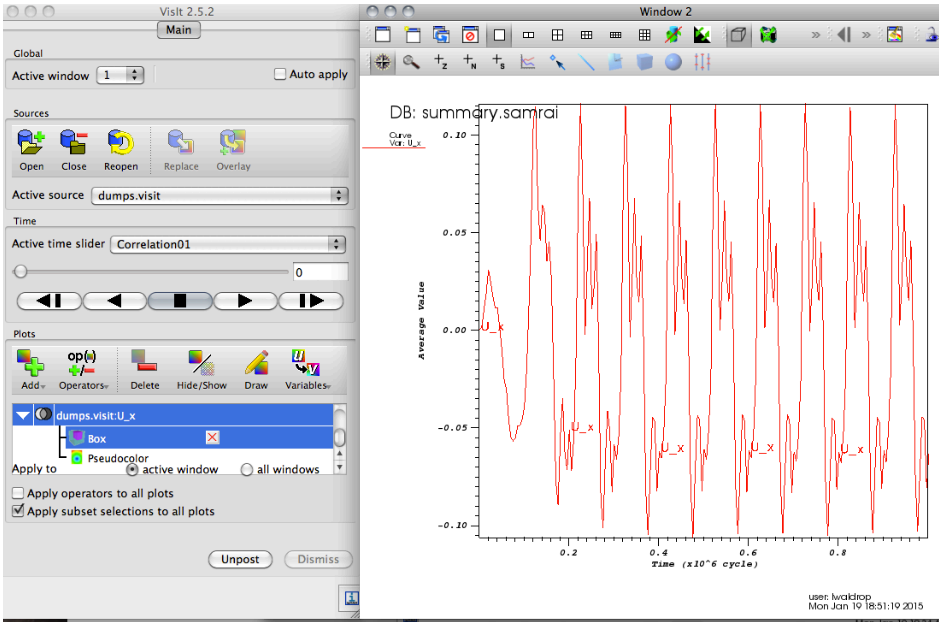

Figure 4: Screenshot of VisIt. Graph produced by Query, Average Value of U_x that is plotted within the restricted data-capture box.

- Enter in your coordinates for single point. These coordinates appear as follows:

  ```
  0 0 0
  ```

  where the first number in the ordered triplet corresponds to the x coordinate of the point, the second is the y coordinate, and third is z. If you are working with a 2D plot, leave the third number as 0.

- Select "Pick using coordinate to determine zone" from the drop-down box.
- Under "Time Curve Options:" select "Preserve Picked Coordinate."
- Under "Multiple-variable Time Curve options:" select "Create Single Y-Axis plot."
- Check the box labeled "Do Time Query." Enter any restrictions on your start and end steps (default is to run entire simulation).
- Hit Query.

Exporting the data on your graph

Now that you have a graph, you may want to graph it in another program such as Python, MATLAB, or *R* for a nicer graph or to run statistics on the data.

- At the top of the main window, select Active Window: 2 or whichever window your plot is in.
- Highlight the curve graphed in the Plots section of the main window.
- On the top bar menu, under "File," select "Set Save Options..." It will open another pop up box.
- Under "Format Options," select the file type "curve." This will output the data in a file type `filename.curve` which MATLAB can import.

- Change the file name and output directory.
- Click "Save." And then Dismiss.
- VisIt outputs data as a two-column text file with a short header, delineated by spaces (instead of commas, like a csv file). Any basic text-editing program should open this file, but since it is not csv, you will probably have to write a bit of code to read it into another program.
- In MATLAB, you will need to import the data with MATLAB's import tool. It will appear as a two-column matrix, with time as the left column and your queried value on the left-most column.
- In order to to import into *R*, you will need to use the `read.table` command. This works for me:

  ```
  data <- read.table("visit10000.curve", header=FALSE, sep="")
  ```

  This line will read the data in as two column vectors and give you default headers (`data$V1` for time and `data$V2` for the query variable).

Other Resources for VisIt

For more help on VisIt, see the following outside resources.

- VisIt Community Wiki: http://www.visitusers.org/index.php
- VisIt Users Forum: http://www.visitusers.org/forum/forum.pl
- VisIt Tutorials: http://www.visitusers.org/index.php?title=VisIt_Tutorial

A Closing Note About Data Analysis and Writing Stuff Down

I know it may seem right now that you will remember exactly what you did 3 years ago when you wrote a piece of code or analyzed a bit of simulation data, or will be able to figure it out. Take my advice: no you won't. People get old and their memories aren't as good. People get better at analyzing data or writing code, and all of the sudden, what you did 3 years ago makes no sense at all. You'll spend hours trying to reconstruct what on Earth you were thinking, if you're able to figure it out at all!

Do yourself a favor: **write it down.** Start with a draft of a protocol of steps that you can follow exactly, a protocol good enough that you could give it to a competent undergrad to complete the analysis for you. If it needs to change, then edit it. Pretend like you won't remember. Additionally, you'll sing your own praises when, months or years later, you have fantastic notes and a working first draft when it comes time to write it up for your dissertation or thesis! Your future self will thank you!