**Software and server access – UA Users**

**HPC Documentation: https://hpcdocs.hpc.arizona.edu/**

Step 1: Get an HPC account

Getting an Account

If you are affiliated with the University of Arizona but are not faculty, you will need to request sponsorship from a faculty member. This can be done through our web portal.

Step 1: Create an HPC account by navigating to https://portal.hpc.arizona.edu/.

Step 2: Request sponsorship from a UArizona faculty member. Note: Your faculty sponsor will need their own HPC account before they are able to sponsor others.

To request sponsorship, navigate to https://portal.hpc.arizona.edu/portal/sendlink.php. On the right-hand side, enter your sponsor's email address and click send.

A blue and red squares with white text

Description automatically generated

**On a Mac**

Use your spotlight finder in Mac and search for terminal and open it. You will then want to type ssh netid@hpc.arizona.edu

**On Windows**

Probably the best program to use is Putty. You can download it here:

<http://www.putty.org>

You will want to log into hpc.arizona.edu using port 22 with your netid and password.

Step 3: VPN installations and clients to login off campus

Please install the cisco client here:

https://uarizona.service-now.com/sp?id=sc\_cat\_item&sys\_id=8ab35af01bb830507947edf1604bcb90

and follow the instructions here:

https://uarizona.service-now.com/sp?id=kb\_article\_view&sys\_id=6c53dcdc1b0d4e503578773bdc4bcb99

Step 4: SFTP (moving files between local computer and the hpc)

I recommend using filezilla to move your files between your computer and the cluster.

<https://filezilla-project.org>

Step 5: Download and install Visit Visualization

<https://wci.llnl.gov/simulation/computer-codes/visit/downloads>

Step 6: Make sure you have a text editor that won’t add random hidden characters (like Word)

I like Notepad++

<https://notepad-plus-plus.org>

And Sublime

<https://www.sublimetext.com>

Step 7: Make sure you have MATLAB installed

**Running IBAMR/IBFE on super computers!**

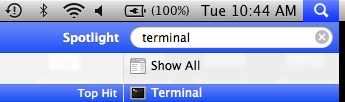
(…or how to run simulations on the hpc)

adapted from Nicholas Battista, Aug. 2014

**Introduction:**

Running simulations on the clusters is a very valuable tool in scientific computing research, especially when the code is parallelized and has the ability to run on multiple processors. This allows the code to be partitioned into parts that can all be run simultaneously and thereby finish running theoretically a lot sooner than if it was being run on a serial machine.

**Part I: Loggin’ in, submittin’ jobs, checkin’ in on simulations, and all that jazz…**

1. First I will begin this tutorial by giving a step-by-step guide to logging into the hpc and where to actually submit jobs and run simulations.
2. The first thing is that we have to *remotely* log into the hpc through the terminal. (MacOSX or Linux platforms) or using Putty (Windows).
3. Use your spotlight finder in Mac and search for terminal and open it.
4. Once the terminal is opened, we can begin searching through directories (folders) and files through your local personal computer, i.e., it is a way to search through your computer without using a mouse, but moreover you can create, copy, move, remove files, and open programs.
5. The following are a few helpful commands you can use in the terminal (or command-line).

SEE WHAT FOLDER YOU'RE IN:

pwd : lists in sequence folders you're currently in

CHANGING DIRECTORIES

cd .. : moves up a directory

cd *[folder\_name]* : change directory to a folder named *[folder\_named]*

LIST ALL CONTENTS IN A FOLDER

ls : lists all files/folders in a directory

ls -a : lists all files/folders/hidden files in a directory

MAKE A NEW DIRECTORY:

mkdir *[name]* : makes a folder of the name, *[name]*

REMOVING (deleting) A FILE OR FOLDER

rm *[name]* : removes a file of the name *[name]*

rm -r *[folder\_name]*: removes a folder of the name [*folder\_name]*

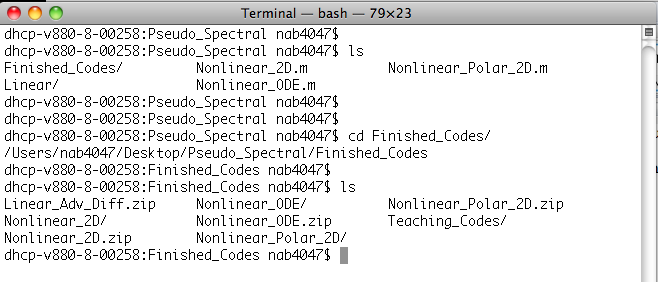
COPYING FILES/FOLDERS

cp *[file1] [file2]*: copies *[file1]* to a new file called *[file2]*

cp -r *[folder\_name1] [folder\_name2]*: copies an entire folder of name *[folder\_name1]* to *[folder\_name2]*

1. For example I’ll illustrate a few of these commands below,

Lists all files in current directory you’re in. Note: folders end with a “/” after their name



Tells you name of current directory you’re in





Changes directory into Finished\_Codes folder.

1. Now let’s start logging into the hpc In the terminal window, type

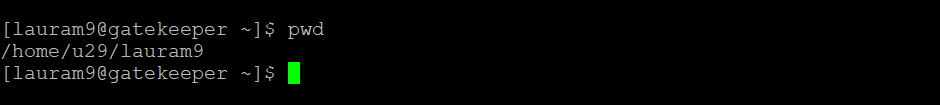
ssh [your netid]@hpc.arizona.edu

and click enter. Now you will be prompted to enter the password associated with your UA netid.

A screenshot of a computer

Description automatically generated

1. Now you should be at the screen listed above. Congratulations - you are now remotely connected to a supercomputer! If you type “pwd” you should be listed in a similar directory to the one below, but with your associated netid.



1. It does not show it, but you are currently logged into the *Bastion Host.* You must now log into puma, ocelot, or elgato.on the cluster. Let’s log into puma.

A screenshot of a computer

Description automatically generated

1. We are now in the login node for puma. This is essentially a processor that all users use when they first log in. Now this is very important, but repeat after me – **WE DO NOT RUN SIMULATIONS ON THE LOGIN NODE ☺** Users may run their work on HPC by submitting a job request to a scheduler. A scheduler, in this case SLURM, is software that will reserve resources and run work on the cluster's compute nodes when space becomes available. To do this, you will need to write a script that requests compute resources and gives the system a blueprint of how to run your work.
2. Software is not available on the login nodes. To see and use software, you must request an interactive session. This sends a request to the scheduler to take you from the login node to a compute node. An alias is set up on Puma to do this quickly and easily. Enter interactive on the command line to request a single core interactive session for one hour, e.g.:

A screenshot of a computer screen

Description automatically generated

1. The change in the command line prompt shows you are now on a compute node. From there, you may view and access individual software packages which are made available as modules. Modules allow for the customization of your environment, including allowing access to different versions of the same software. Take a look at what’s installed by running the command module avail. To use a particular module, type module load <software>.

If you ever want to see the modules you currently have loaded, use the command module list. To unload a piece of software, use the command module unload <software>.

1. To run simulations that will generate large amounts of data, we must change directory to an xdisk. To change directories type:

cd /xdisk/lauram9

A screen shot of a computer

Description automatically generated

Note that each user has a folder. You may need to submit a request to get your own folder. Also, the xdisk is deleted every 300 days, so be sure to download and save relevant simulations and codes to an external hard drive.

1. It is now time to learn how to submit jobs to run. Well let’s not get too excited, we have a little more work to do before we can do that. Namely, we must copy the simulation we wish to run to the hpc to actually be able to run it! We will run a simulation called *3-IBAMR-Example\_2Drubber\_band*. (Please see the associated .zip file). It is a 2D IBAMR example of a rubber band.
2. This example should be in your github folder. We need to move the folder from your home computer to the hpc, in the location where you want to run the simulation. I prefer to use Filezilla to do this. Download Filezilla

<https://filezilla-project.org/>

1. We need to do a little more work to get around the two-factor authentication.

* In FileZilla, select the File menu
* Select Site Manager
* Select an existing site or select New Site to create a new connection
* Enter the following information in the General tab:
* Protocol: SFTP
* Host: enter the server name “filexfer.hpc.arizona.edu”
* Logon Type: Interactive
* User: Your NetID

A screenshot of a computer

Description automatically generated

* Select the Transfer Settings tab
  + Select "Limit number of simultaneous connections"
  + Set simultaneous connections to: 1
* Click Connect
  + In the first password box, enter your NetID password
  + In the second password box, type a 1 for Duo push or enter a passcode from your Duo device

A screenshot of a computer

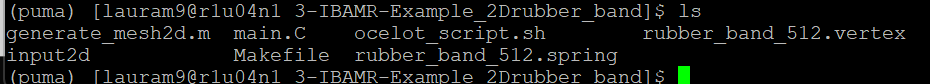
Description automatically generated

1. Now move the example folder from your home computer to your xdisk folder.

A screenshot of a computer

Description automatically generated

1. Now go back to your other terminal, the one which is connected to the hpc. You should be in your /xdisk/lauram9*/[your netid]* directory. Try typing *ls* to see what folders are now in your directory and the 3-IBAMR-Example\_2Drubber\_band directory should be present. Cd into that folder.



1. Now we are ready to begin our journey of “making” *main2d* (or *main3d*) executables and submitting our jobs to be run on the hpc! We will not worry about all the files and what their purpose is now – we will just assume they are all necessary files to run a simulation using IBAMR.
2. Now we will “make” the executable. Simply type

make clean

make main2d

and wait for the code to finish running. This is essentially compiling our code, making sure there are not syntax mistakes, and connecting all necessarily bits of software to one another.

Note: There will what appears to be a ton of information that gets printed to your terminal screen. What is important is for it all to end without saying it has found any error. Furthermore, if there is a warning present, that is fine for our purposes (hopefully).

1. Now if you type *ls* you will see a few files have been added – main2d, main.o, stamp-2d, and update\_target\_point\_positions.o. For our considerations, we don’t really need to pay much attention to these files as long as the *make main2d* went smoothly and finished compiling.
2. Now it’s finally that time- let’s submit our first job to the queue! We will submit a job using sbatch. You can check the status of your submitted SLURM jobs with the command “squeue -u ” (note squeue shows jobs from all users, provide your netid to just show your jobs). If you need to kill/end a running job, use the “scancel” command: scancel [JobID]
3. We will need to create a bash script using our favorite editor. If you don’t have a favorite editor, use nano (for now).

nano script.sh

The script contains job submission options followed by application commands. Please enter the following into your script as an initial example (Note: that each SBATCH switch below has two ‘-‘ characters, not one):

#!/bin/bash

#SBATCH --job-name=IBAMR-test

#SBATCH --ntasks=1

#SBATCH --nodes=1

#SBATCH --time=04:00:00

#SBATCH --partition=standard

#SBATCH --account=lauram9

#SBATCG --output IBAMR2D.out

mpirun ./main2d input2d

Save your file and exit nano. Submit your job using the sbatch command:

sbatch script.sh

You can then check that the simulation is running by typing

squeue -unetid

A screen shot of a computer

Description automatically generated

-----------------------------------------------------

When submitting, it is important for to jot down the JOB ID#, and keep a good record of when the simulation was submitted as well as a little bit about the simulation you are running (i.e., what simulation are you running, what is the Re, what is the time-step, the viz-dump interval, etc).

1. All that is left is to wait! Sort of. There are a few things we can do while it’s running

We can continually check to see if the simulation is still running by typing *squeue -unetid*

1. We can also check other ways to see how the simulation is doing, i.e., how far it is into the simulation, etc. This time you MUST be in the folder containing the information for the specific simulation you are interested in.

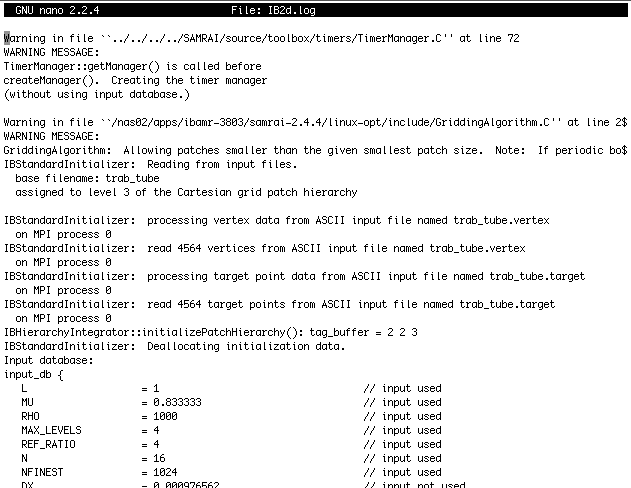
There are two ways to see how the simulation is doing. The easiest solution is to type *ls* and see that there should be a file called *IB2d.log*. This log file prints out information for each time-step such as what the current simulation time and time-step is, how many iterations it takes for the Stokes solver, the CFL number, etc. All you will need to keep track of is seeing which iteration and time-step it is on. 

**22.** Now to open the IB2d.log file we can use one of many editors. The editors I traditionally use are either *vi* or *nano*. I will illustrate an example of both and say the commands to scroll through the file with each editor now.

First let’s start with *nano*. To open the *IB2d.log* file with *nano* just type

nano IB2d.log

into the terminal window and hit *enter*. This will bring you to a screen that looks like the following,



Note that this is only part of the log file. It is probably pretty long…

You can scroll through the *IB2d.log* file using the ‘arrow’ keys on your keyboard. Doing this method you can essentially go character by character across a line or up and down one single line at a time.

Scrolling through the entire *log* file this way would take forever. Luckily for us there are very helpful commands in *nano* for scrolling and editing text (it is afterall a text editor.)

If you notice at the bottom of the editor there are keyboard shortcuts (useful commands) to use in *nano*. Macintosh HD:Users:nab4047:Desktop:Screen shot 2014-08-26 at 5.06.35 PM.png

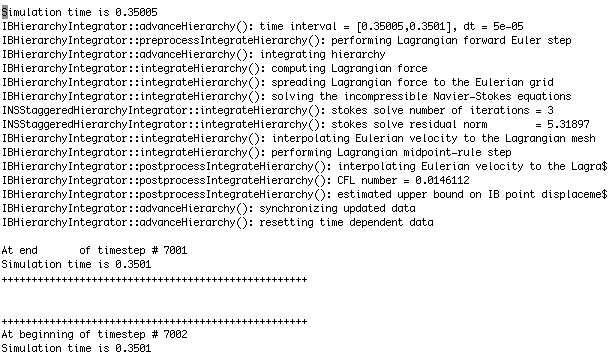
To scroll a page at the time you hit the following buttons on your keyboard together,

Ctrl + v : scroll down a page

Ctrl + y : scroll up a page

Ctrl + w + v : scroll to the bottom of the file

Ctrl + w + y : scroll to the top of the file

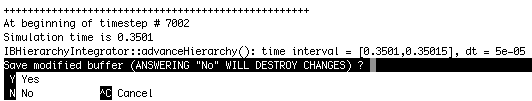
For instance if you want to scroll to the bottom of the file to see how far it’s gone you press *ctrl + w + v*  and then will probably see something similar to the following picture

Now the last thing we have to do in *nano* after we’ve seen how much of the simulation has finished running, is exit. I’ll make a quick note here to say we can of course change text files in *nano*, i.e., actually using the editor to, you know, edit something, but for our purposes here we just want to see what is in the *IB2d.log* file.

To exit simply press

*ctrl + x*

and you may be prompted by something that asks if you wish to save the modified buffer, as seen below,



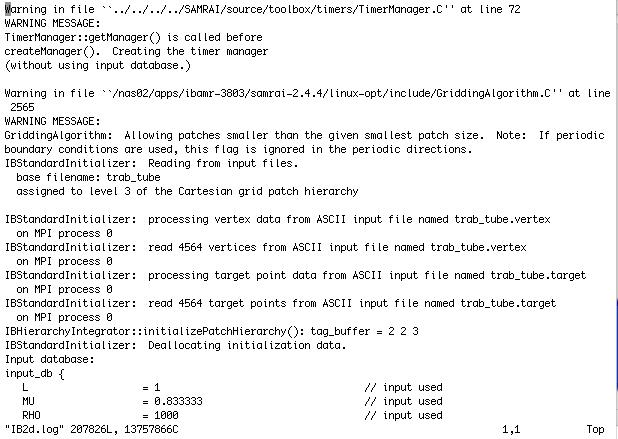
simply type ***n***to not save any changes you have made.

**24.** Likewise, we could also check the *IB2d.log* file using a different editor called ***vi*** or ***vim***. It is more of a personal preference to use whichever editor you feel more comfortable with.

To open the *IB2d.log* file with ***vi*** type in the terminal,

vi IB2d.log

and the following screen should appear (it may take a few moments for the editor to open the log file), which looks strikingly similar to the screen we saw in *nano.* IT SHOULD BE – it is the same log file after all.



The first thing you might note is that the ***vi*** editor does not list a bunch of keyboard shortcuts at the bottom. Now as has been described to me before by some computer science graduate students, ***vi*** and ***vim*** are the more cool editor to use because they are more cryptic, professional, and “cool”. Not sure what they mean by that, but luckily for us, there are a bunch of keyboard shortcuts – they just are not as conveniently located at the bottom of the editor such as is the case with *nano.*

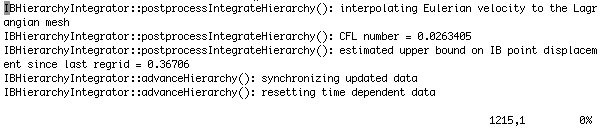
If you type a ‘ : ‘ followed by a number, i.e.,

:1215

and hit *enter* it will scroll to line 1215 in the file. If there are not 1215 lines in the file, it will scroll to the last line. This is how you can scroll to the bottom pretty easily, simply type “ : “ and a very large number after it.



Says what line your cursor is on and what character space it is on in the line



What % of the file your cursor is on

To make a change in ***vi*** you may have noticed you cannot simply open the file and start typing away. You must first tell the ***vi*** editor to go into *write* mode. To do this hit the *esc* key. You can now write all you wish.

To exit the file there are a few commands you can do

*esc* then *:wq* [exit with writing to the file]

*esc* then *:*q [exit without writing to the file if nothing was changed]

*esc* then *:q!* [exit without writing to the file, even if characters were changed but don’t want to save them]

**25.** Now that you can check on simulations to see how far they have run, you can write down the time and how far the simulation has run. Other than that, sit back, submit more jobs to the queue on the hpc, and wait to do the analysis on the simulations.

**Part II: Getting data off of the HPC**

1. Once the simulation has finished running you will want to copy all the data from the cluster to either your local machine or external hard drive (off of your local machine).

I want to stress TWO things at this point. The first is that you will want to copy all the data (basically the whole simulation folder) shortly after (at least within a week) of the simulation finishing.

The second thing is that you will want to store these files on an external hard drive. Most of these simulations typically will range in sizes between 500mb and 4gb of data or more. That will fill your local machine’s hard drive pretty fast.

You can use filezilla to download the files onto our computer.