Table 1: Bash commands and what they mean

Command	**What it does**
`ls`	list contents of current directory
`ls -a`	show hidden files too
`ls -altr`	see the last changes made to the files in a directory
`mkdir directory`	make a new directory
`cd directory`	change directory
`cd ..`	go back a directory
`cd ../..`	go back two directories
`cd ˜`	go to your root
`pwd`	print working directory
`˜/`	means your root
`.`	means the current directory
`cp file/to/copy where/newName`	copy a file
`cp file/to/copy .`	copy a file to current directory without changing the name
`cp directory/* .`	copy all the files in a directory
	to the current directory
`cp -r directory new/directory`	copy a directory recursively
`rm file/to/remove`	remove a file
`rmdir directory`	remove a directory
`rm -rf directory`	blow away a directory permanently
`mv file/to/move where/newName`	moves or renames a file
`man command`	show the manual for a command
`cat file/one file/two > new_file`	concatonate two or more files into a new file
`history`	shows a history of your commands
`less file/to/see`	shows one page of a file
	space turns the page q quits
`head file/to/see`	see the first page of a file
`head -n 8 file/to/see`	see the first 8 lines of a file
`tail file/to/see`	see the last page of a file
`tail -n 10 file/to/see`	see the last 10 lines of a file
`grep keyword file/to/search`	search a file for a keyword and print all the lines with that keyword to the screen
`history | grep keyword`	search your history for a keyword
`grep keyword file/to/search | wc -l`	count the occurences of lines with a keyword
`command | less`	pipe the output of a command to less

Table 1 – *Continued on next page*

Table 1 – *Continued from previous page*

Command	**What it does**
`command >> file`	append the output of a command to a file
`command > file`	writes the output of the command to a file
`!command`	executes the most recent command that starts with the letters you typed
`echo something`	print something to the screen
`ls -altr`	see when files in the directory were last altered
`sed -i ’s/to replace/new phrase/’ file/to/search`	find and replace a phrase in a file
`grep -Rl keyword`	recursively search for a keyword and print the file it was found in
`awk ’!a[$0]++’ file/to/search`	get rid of duplicate lines
`echo "phrase" >> file/to/append`	append a phrase to a file

Table 2: Emacs commands and what they mean

Command	**What it does**
`emacs path/to/file`	enter emacs editor for existing file or creates new file with that name
ctrl+x ctrl+s	save file
ctrl+x ctrl+c y	save and quit a file
ctrl+x ctrl+c n	quit without saving
ctrl+w	cut a line (highlight line first)
ctrl+y	paste a line
ctrl+k	kills the contents of a line/cut a line
ctrl+k ctrl+k ...	kills however many lines–helpful to copy and paste blocks of code
ctrl+shift+-	undo
ctrl+u `3 command`	executes the command 3 times
ctrl+x ctrl+f	find and open a file (at the bottom of the screen)
ctrl+space	set marker
Add text to a block of code	
ctrl+space ctrl+x R	set marker set end of rectangle format as rectangle

Table 2 – *Continued on next page*

Table 2 – *Continued from previous page*

Command	**What it does**
T “text”	add text
or k	or delete

Table 3: Loops in bash

Command	**What it does**
`for i in {1..100}`	for 100 iterations
`do`	
`command $i`	do this thing(`$i`
`done`	references the index)
`if [ condition ]`	check the condition
`then`	if it’s true
`command`	do this
`else`	if it’s not
`command`	do this
`fi`	ends if statement
`if [ -e file ]`	check if a file exists
`for i in ‘ls -d */‘`	visit every directory in the current
`do cd $i`	directory
`command`	and execute some command
`cd .. done`	

Table 4: Example of a Bash loop

Command	**What it does**
`for i in {1..100}`	for 100 times
`do`	
`cd E.$i`	enter the directory named `E.#`
`if [! -e KPOINTS ]`	if KPOINTS doesn’t exist
`echo $i`	print the directory number
`getKPoints`	run the getKPoints script
`fi cd ..`	go back one directory
`done`	close loop

Table 5: Example of a Bash loop

Command	**What it does**
`for i in ‘ls -d */‘`	for all directories
`do`	
`cd $i`	enter the directory named
`if [! -f KPOINTS]`	if KPOINTS doesn’t exist
`then getKPoints`	run the getKPoints script
`echo $i`	print the directory number
`fi`	end if statement
`cd ..`	go back one directory
`done`	close loop