**HPSC**

1. grep – for particular keyword search

Eg: pip list | grep jupyter

Returns list with jupyter as keyword

2. rm - remove files or directories

Eg: rm myfile

This will remove filename ‘myfile’

If myfile is directory then use below command to delete

rm myfile/ -rf

3. np.dot(A,x)

For matrix multiplication in python (take care of order of both matrices otherwise will give wrong answer)

4. A.T

For transpose of A in numpy array (matrix)

5. np.matrix (“1.,2.;2.,3.”)

Creates matrix in matlab style in python i.e. numpy.matrix

A.T is also applicable for transpose

Double AA is required to call matrix

Eg: AA #to print matrix named A

Eg: AA[0,1] for printing first element of first row of 2nd column

6. numpy.ones(m,n,o)

To create a matrix of rank 3 or 3 dimensional m, n, o elements in each direction with value of each element as 1

7. linalg

For linear algebra (in numpy)

Solve matrices AX=B

Use linalg.solve(A,B)

Type: linalg? #for more information about linalg in python

Linalg.eig(A) #for eigen value and vector of A

For actual rank of matrix, >numpy.linalg.matrix\_rank(matrixname)

For dimensional rank of matrix >np.rank(matrixname)

8. evals,evecs=linalg.eig(A)

For storing eigen values in evals and vector in evecs

9. from scipy.integrate import quad

For importing quad

Quad is for numerical integration

10. sympy

Library for symbolic integration (python)

11. lambda function

Eg: quad(lambda x: x\*\*2,0,2)

12. private variables accessible in python

Starts with double underscore and ends with double underscore

\_\_name\_\_

Timing

13. %timeit mysqrt(2.)

Also returns time to execute mysqrt

14. debug\_pdb

15. import pdb; pdb.set\_trace()

A python debugger package

Helps in tracing an error and tells why the error has occurred. We can trace up and down in the code and can find the values of various variables at any point of code.

16. we can also important inside user defined function but can not do import \*

17. run job on compute nodes not on head node for fast computing

18.

CPUHPC - for modelling and analysis on CPU architecture

GPUHPC – they are queue based system GPU

File system (storage) is common for both systems

19. HPSC Cluster Working

Head node will decide the serial based on requirement.

Go to HPC website for related commands, etc.

cp command for copy

rsync (confirm this command) command for just copying the which are changed

for login to hpc in terminal, type> ssh [s22033@10.8.1.19](mailto:s22033@10.8.1.19) followed by password

#see commands of sir for initial commands

>pip3 install –user –upgrade –proxy=http://10.8.0.1:8080 pip #to upgrade pip with proxy enabled

To create virtual environment

>pip3 install –user #to install packages to home

>pip3 install --user --upgrade --proxy=http://10.8.0.1:8080 virtualenv #to upgrade virtualenv #here –user will try to install in home otherwise it will install where python is installed

# for big packages install in local machine then transfer to cluster for fast working

>pip3 install –user –upgrade –proxy=http://10.8.0.1:8080 pip #to upgrade pip with proxy enabled

Go to virtual env

Install packages there such as python, numpy, etc.

For activating venv on compute node

Inside HPSC

>ping CPUHPC

>ping GPUHPC # for checking ping

>scl –list #to check installed packages

**Serial script #it containes commands to submit job using qsub command**

#!/bin/bash

#PBS -j oe

#specify queue name

#PBS -q serial

#to specify where output file will go

#PBS –o out.o

#to specify where error file will go

#PBS –e out.e

#to specify source to run code

#PBS –l nodes=1:ppn=1 #ppn is cores

#specify job name

#PBS -N  my\_serial\_job

#specify time required for job completion

#PBS -l walltime=10:00:00

#to locate where it is running

echo "Running on: "

cat ${PBS\_NODEFILE}  ##print information where job goes to

cat ${PBS\_NODEFILE} > machines.list  ##to redirect compute node info to machines.list file

echo

echo "Program Output begins: "

cd ${PBS\_O\_WORKDIR}  ##returns the path to current working directory

./a.out ##returns output in current folder

source ~/virtualenvs/venv/test/bin activate

python python\_scripy.py

To submit job

>qsub serialscriptfilename

* man PBS #for manual of PBS
* pbs nodes #for list of pbs nodes
* scp –r ~/pathof\_file\_to\_copy [s22033@10.8.1.19](mailto:s22033@10.8.1.19) path\_where\_file\_is\_to\_copy #**to copy a file to cluster**

20. Virtual env VS containers

Virtual envs are for python packages

Containers are like os installed

**FORTRAN**

Compiled language

! for comments

Indentation is just for code to look better

>implicit none !to make sure that variable type is only integer or means all variables must be explicitly declared

\* !to figure out best format

X=3.d !d is for double precision

X=3.e !e is for single precision

external :: f !to tell that f is a function

intent(in) :: x !intent(in) tells that x is going to passed as input and can not be modified

intent(out) :: z !can be modified

intent(inout) :: a,b !

dimesion(3) :: y,z ! declaration of array of rank 1 and size 3

y= (/2., 3.,4./) !way to assign array

reshape((/1,2, ), (/3,2/)) !to reshape array to 3 row and 2 columns

!reshape command do column wise reshape in fortran contrary to python

real (kind=8) :: x(3) !another way of declaring size 3 array