Physics 499
Computational Nuclear-Particle Physics
Spring 2013

Instructor: Peter Siegel, Room 8-215, Phone: 869-4029
email: pbsiegel@csupomona.edu
Web page: www.csupomona.edu/∼pbsiegel

Lecture Hours: Mon 4-5:50 in room 4-3-567 and online

Office Hours for this Course: Wed 4-5 in room 4-3-567 and online

Textbook: There is no specific textbook for the course. I will post my lecture notes online as well as the archived video sessions. There are a number of good texts on computational physics. A nice one I have used in the past is *Computer Simulation Methods: Applications to Physics Systems*, by Gould and Tobochnik

Grading:

Programming Projects	100%

Programming Projects:

There will be 5 programming projects assigned thoroughout the quarter. For each project you will be writting a computer program in ”C” and/or in the ROOT environment from CERN. I will try and design the assignment so that there is a basic part which I hope everyone will be able to complete, and a more challenging part.

Course Material

Computers are used in many ways in solving physics problems. The goal of this course is to teach you some numerical methods and give you experience of writing computer programs to solve problems in nuclear and particle physics. For those students who will be working at CERN this summer, the goal of the course is to give you a good introduction to the ROOT software used at CERN. Below is an initial outline of the numerical methods and physics topics that I plan to cover this quarter. These can vary depending upon the interest of the students.

Rough Course Outline Spring 2013 (Siegel)

Day	Date	Programming Topic	Physics Topic
Mon	April 1	Introduction	
Mon	April 8	Solving for the roots of a function	Λ-Hypernuclei
Mon	April 15	Differential Equations	Bound States in nuclei
Mon	April 22	Differential Equations	$q\bar{q}$ Resonances
Mon	April 29	Graphing	Meson-Nucleus Scattering
Mon	May 6	Numerical Integration	Born Approximation
Mon	May 13	χ^2 analysis	Meson-Nucleus Scattering
Mon	May 20	Random Numbers	resonances
Wed	May 29	χ^2 analysis	phase shift analysis
Mon	June 3	χ^2 analysis	resonances

The Ubuntu distributions are easy to install and use, and I recommend installing the latest version. On my homepage I have instructions on installing Ubuntu12.04. The latest version as of March 2013 is 12.04LTS. To install Ubuntu, download the Long Term Supported version (LTS) from
http://releases.ubuntu.com/12.04/. This link has instructions on how to setup Ubuntu on your computer. Basically you download the image file ubuntu-12.04.1-desktop-i386.iso Then burn this image to a CD. Restart your computer so that it boots off the CD. You will boot into Ubuntu 12.04. From here you can try it out, or install it to your computer. You can install it as a dual boot with the Windows you are running, or as the only operating system on your computer.

Installing ROOT on Ubuntu 12.04LTS

Before installing ROOT on Ubuntu12.04, you will need to have some programs installed. You can install these with the ubuntu software center icon, with apt-get, or with synaptic package manager. In the software center search on the following packages and make sure you have these packages installed:

Required:

1. subversion
2. dpkg-dev
3. make
4. g++
5. gcc
6. binutils
7. libx11-dev
8. libxpm-dev
9. libxft-dev
10. libxext-dev

The following are Optional:

gfortran, ncurses-dev, libpcre3-dev, xlibmesa-glu-dev, libglew1.5-dev, libftgl-dev, libmysqlclient-dev, libfftw3-dev, cfitsio-dev, graphviz-dev, libavahi-compat-libdnssd-dev, libldap-dev, python-dev, libxml2-dev, libssl-dev, libgsl0-dev.

Installing ROOT in Ubuntu

Open a terminal and you will be in your home directory. Download the source by typing in the terminal (in your home directory) the following command:

```
svn co http://root.cern.ch/svn/root/tags/v5-34-03 root
```

The ”svn co” command is a subversion command that will checkout (i.e. download) all the files needed for root while keeping the directories entact. The ”root” at the end makes a directory called root, where all the files are copied. Now checkout the root directory to see all the files that have been copied. Move to this directory by typing

```
cd root
ls
```

You should see a list of the files copied.

In the root directory, you should see the executable file ”configure”. Compile this file with:

```
./configure
```

This might take a while. Hopefully everything compiles. Problems might arise with linking libraries, but I did not have any problems with the version v5-34-03 and Ubuntu12.04LTS. Next type

```
make
```

Finally, you need to set some paths in the file .bashrc. The file .bashrc is a script file that runs whenever you open the terminal. Here you can set aliases, paths, etc. To setup the paths, check out the instructions on my homepage under ”rootinstall.html”. To edit the ”.bashrc” file using nano (or any other editor) to go your home directory and type:

```
    ls -a
```

You will see all the files in your home directory. You should see the file ”.bashrc”. Now edit the file ”.bashrc” and add the following lines:

```
export PATH=$ROOTSYS/bin:$PATH
export LD_LIBRARY_PATH=$ROOTSYS/lib:$LD_LIBRARY_PATH
cd root
. bin/thisroot.sh
cd
```

Note that there is a space between the ”.” and ”bin/thisroot.sh”. It will not work without the space.

You can test your root system by typing ”root” in the terminal. The CERN softward should boot up. To exit ROOT, just type ”.q”. To run a program just type ”.x filename”. I have appended some simple codes after lecture1 for you to try out. I’ll go over them shortly.

Some Useful Commands

To compile and run your code:
To compile your code: gcc ccode.c
You can also type: g++ ccode.c

To run this compiled version: ./a.out
Note: a.out is the default executable for unix.

To name your own executable: gcc -o name ccode.c
Note: to run the executable, just type ./name

The nano editor
To edit your code, type: nano ccode.c
Once in the editor, you can use the arrows to move around. You can type code, and delete with back-space. Some commands are:

control o: save the file without exiting
control w: exit and save the file
control k: cut a line (delete it and place it in the buffer)
control u: copy the buffer

Some unix commands

ls: list the files in the current directory
cd phy409: change to the directory phy409
cd ..: move up one directory
mkdir phy409: make a new directory called phy409
rmdir phy409: remove the directory called phy409
rm file.c: remove the file named file.c
mv file1.c file2.c: change the name of file1.c to file2.c
cp file1.c file2.c: copy file1.c to file2.c, note: now you will have two files the same, file1.c and file2.c
pwd: print working directory
man: the help manual, e.g. man ls: this will give you the manual on the command ls.

```
    //Test code for gcc
# include <stdio.h>
# include <stdlib.h>
# include <unistd.h>
# include <string.h>
# include <math.h>

int main()
{
double x, y, z1, z2;

printf(”Input x and y: \n”);
scanf(”%lf %lf”,&x,&y);
printf(”The value of x is %lf The value of y is %lf \n”,x,y);
z1=x+y;
z2=x*y;
printf(”The sum is %lf \n”,z1);
printf(”The product is %lf \n”,z2);
return (0);
}
```

```
    //Test Code for ROOT

void Testroot()
{
Double_t x, y, z1, z2;

cout ≪ ”Input x and y \n”;
cin ≫ x ≫ y;
cout ≪ ”The value of x is ” ≪ x ≪ ” The value of y is ” ≪ y ≪ endl;
z1=x+y;
z2=x*y;
cout ≪ ”The sum is ” ≪ z1 ≪ endl;
cout ≪ ”The product is ” ≪ z2 ≪ endl;
}
```