

PHYS2305-Week 1a

S. Kunori (TTU)

28-Aug-2017

PHYS2305

- Goals
 - Develop the ability to write computer programs to solve physics problems that requires computers.
 - Code should be readable and re-usable
 - Learn some methods/algorithms to solve physics problems, e.g. numerical integration, massive data analysis, etc.
- OS: Operating System
 - Windows, MacOSX, **Linux**
- Language:
 - Fortran, C, **C++**, Python (numerical)
 - MATLAB, Mathematica (symbolic calculation)
- Code editor
 - vi, **emacs**, ... (or **your favorite**)

Additional SW:
GNU scientific library
GNU plot

PHYS 2305 Approximate Class Schedule

Week			
1	8/28-9/01	Introduction	
2	9/01-08	Numerical calculation (1)	no class 9/04
3	9/11-15	Numerical calculation (2)	
4	9/18-22	Numerical calculation (3)	
5	9/25-29	Monte Carlo simulation (1)	
6	10/02-06	Monte Carlo simulation (2)	
7	10/09-13	Monte Carlo simulation (3)	
8	10/16-20	Advanced method (1)	
9	10/23-27	Advanced method (2)	
10	10/20-11/03	Advance method (3)	
11	11/06-10	Advance method (4)	
12	11/13-17	final project	
13	11/20-24	final project	no class 11/22-26
14	11/27-12/01	final project	
15	12/04-06	final project	final project due 12/05, 06

Plan:

Lecture (short)

Lab session (mostly)

9 weekly homework

grade: 70%

(7 out of 9 count)

1 final project

grade: 30%

List of links: <https://www.dropbox.com/s/mn6tef90697qgl3/phys2305links.pdf?dl=0>

Syllabus: <https://www.dropbox.com/s/05veb4gemvs7edl/PHYS2305Syllabus-2017F-kunori.pdf?dl=0>

Phys2305 link page:

PHYS2305 COMPUTATION FOR THE PHYSICAL SCIENCES

Shuichi Kunori (shuichi.kunori@ttu.edu)

08/28/2017

url for this file

<https://www.dropbox.com/s/mn6tef90697qgl3/phys2305links.pdf?dl=0>

Syllabus

<https://www.dropbox.com/s/05veb4gemvs7edl/PHYS2305Syllabus-2017F-kunori.pdf?dl=0>

1 Slides and Assignments

Week 1 (8/28/2017)

Introduction to Linux and C++ (integer, float, double, for-loop, if-statement)

https://www.dropbox.com/s/q52or3k3iz0jyrn/sk_phys2305_week1a.pdf?dl=0

Assignment 1 (due 9/01/2017 for minimum C, a week later for upgrade)

Week 2 (9/05/2017)

UNIX Tutorial for Beginners

A beginners guide to the **Unix** and **Linux** operating system. Eight simple tutorials which cover the basics of UNIX / Linux commands.

Introduction to the UNIX Operating System

- What is UNIX?
- Files and processes
- The Directory Structure
- Starting an UNIX terminal

UNIX and Linux books

- If you wish to continue learning Unix, here is a [list of good Unix and Linux books](#), ranging from beginners to advanced.

Tutorial One

- Listing files and directories
- Making Directories
- Changing to a different Directory
- The directories . and ..
- Pathnames
- More about home directories and pathnames

Tutorial Two

- Copying Files
- Moving Files
- Removing Files and directories
- Displaying the contents of a file on the screen
- Searching the contents of a file

Tutorial Three

- Redirection
- Redirecting the Output
- Redirecting the Input
- Pipes

The screenshot shows the Cplusplus.com website. At the top, there is a search bar with a 'Go' button and navigation links for 'Tutorials' and 'C++ Language'. The main content area is titled 'C++ Language' and features a sub-section titled 'Introduction'. This section includes a list of topics such as Compilers, Basics of C++, Program structure, and Compound data types. To the right of the main content, there is a large image of a person riding a snowmobile through a snowy landscape with a dog running alongside.

C++ Language

These tutorials explain the C++ language from its have a practical orientation, with example program away.

Introduction

- Compilers

Basics of C++

- Structure of a program
- Variables and types
- Constants
- Operators
- Basic Input/Output

Program structure

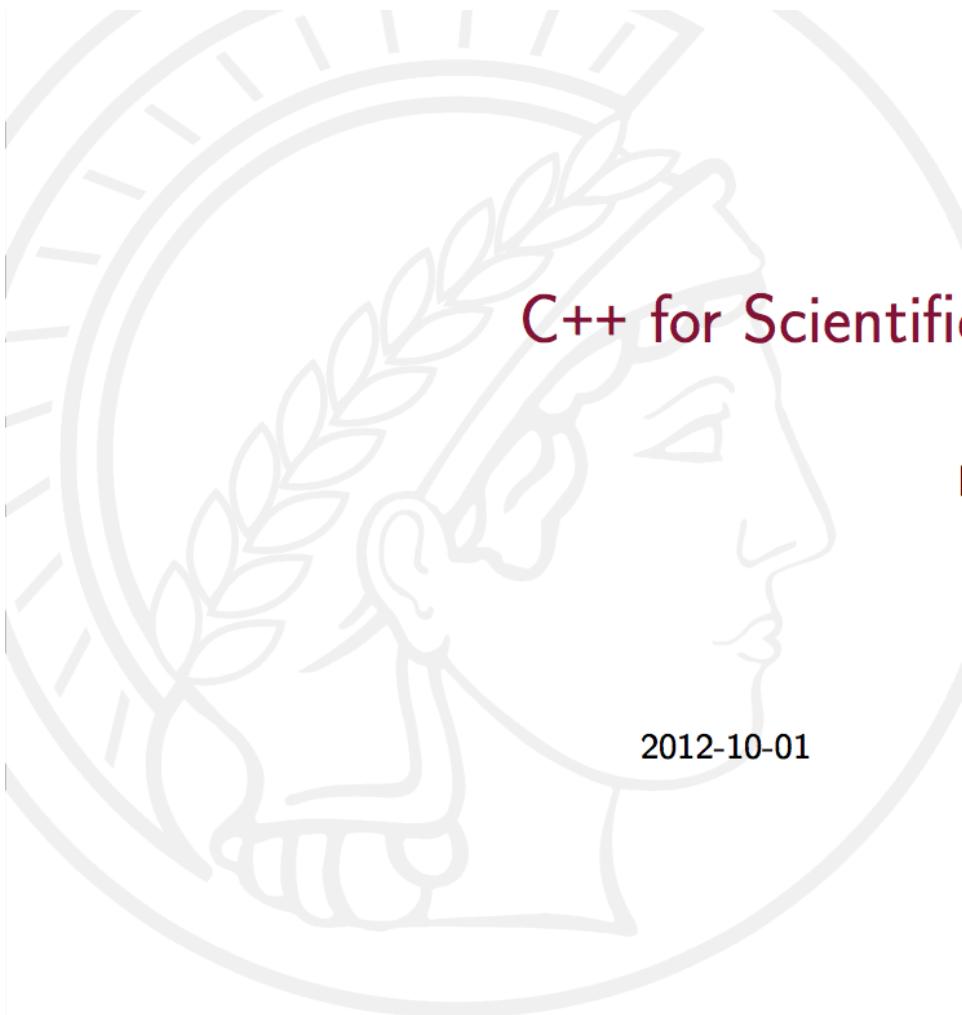
- Statements and flow control
- Functions
- Overloads and templates
- Name visibility

Compound data types

- Arrays
- Character sequences
- Pointers
- Dynamic memory

S2305 Week 1a

<https://www.dropbox.com/s/lnwzkbysar8mf93/cppscicomp.pdf?dl=0> (slides)



R. Kriemann, »C++ for Scientific Computing«



1/316

Editor: emacs

Google search results for "emacs tutorial".

Search bar: emacs tutorial

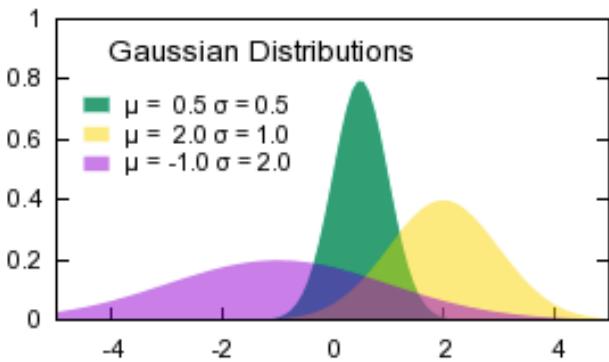
Filter: All

Results: About 455,000 results (0.43 seconds)

1. **Absolute Beginner's Guide to Emacs - Jessica Hamrick**
www.jesshamrick.com/2012/09/10/absolute-beginners-guide-to-emacs/ ▾
Sep 10, 2012 - I've been using Emacs () as my primary text editor for several years now. ... This tutorial is mainly for people who have primarily used GUI text ...

2. **A guided tour of Emacs - GNU**
<https://www.gnu.org/s/emacs/tour/> ▾
If you install Emacs first, you can follow along with the examples presented here. Whenever you decide to start using Emacs, you should take the **Emacs tutorial**.

Plotter: gnuplot



28-Aug-2017

Google search results for "gnuplot".

Search bar: gnuplot

Filter: All

Results: About 1,210,000 results (0.33 seconds)

1. **gnuplot homepage**
www.gnuplot.info/ ▾
Official website for gnuplot 4. Includes downloads, screenshots, documentation, and links to other help and tutorials.
Gnuplot download · Demos · Release Notes · Tutorials, learning, and help

2. **Gnuplot 4.2 Tutorial - Duke University**
people.duke.edu/~hpgavin/gnuplot.html ▾
Gnuplot is a free, command-driven, interactive, function and data plotting program. Pre-compiled executeables and source code for Gnuplot 4.2.4 may be ...

<http://farside.ph.utexas.edu/teaching/329/329.pdf>

Old and C, but good introduction

Computational Physics

Richard Fitzpatrick

Professor of Physics

The University of Texas at Austin

(example of contents)

3 Integration of ODEs

3.1	Introduction
3.2	Euler's Method
3.3	Numerical Errors
3.4	Numerical Instabilities
3.5	Runge-Kutta Methods

Advanced, python, \$109 (E-book)
Good examples of algorithms

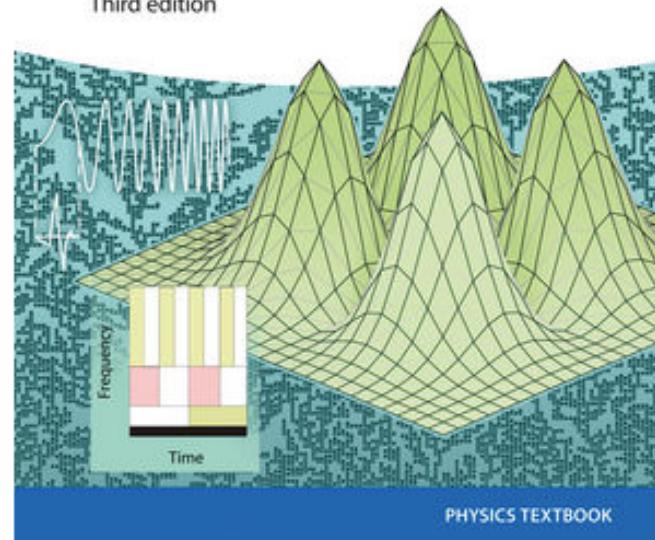


Rubin H. Landau, Manuel J. Páez
and Cristian C. Bordeianu

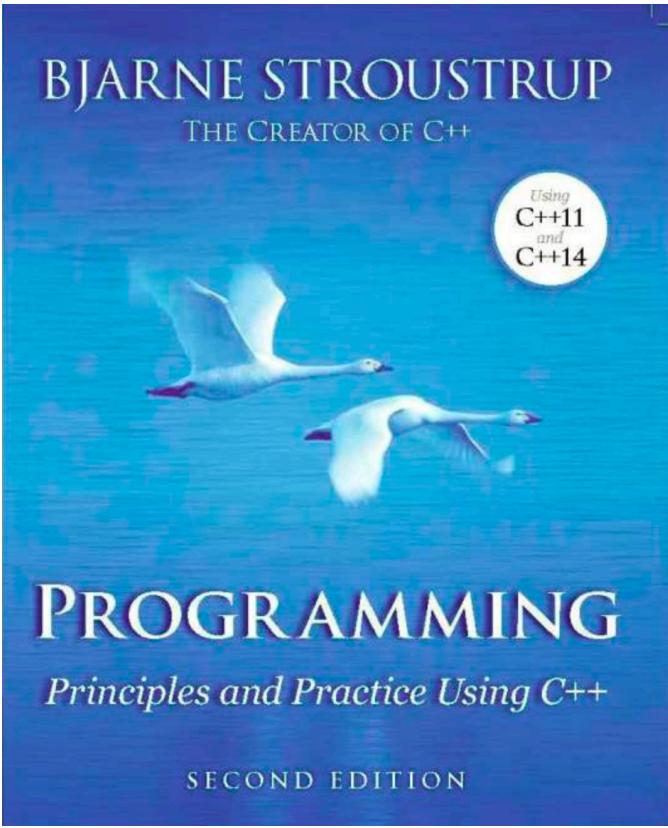
Computational Physics

Problem Solving with Python

Third edition

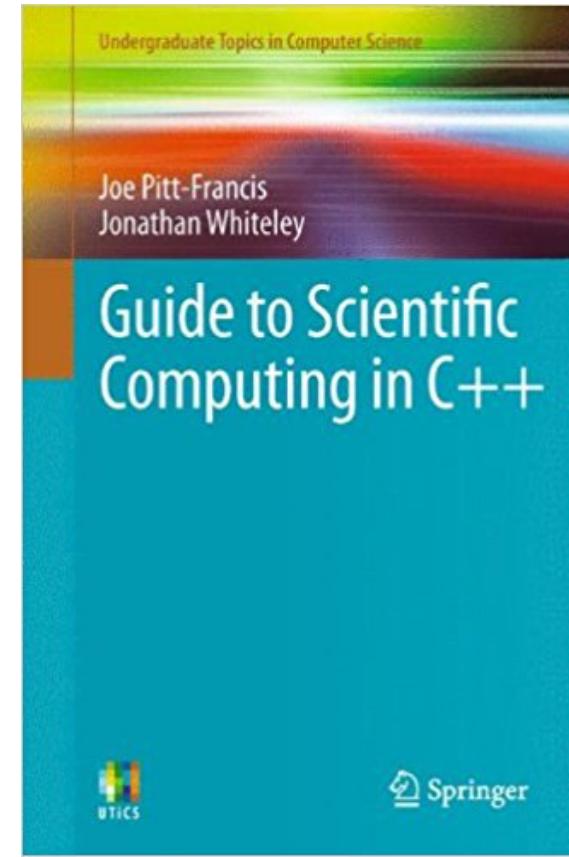


Textbook for PHYS4301/5322 (Dr. Igor Volobouev)



"This is an introduction to programming for people who has never programmed before. It will also be useful for people who have programmed a bit and want to improve their style and technique - or simply learn modern C++."

Addison-Wesley ISBN 978-0321-992789. May 2014.
Book \$57.99, eBook \$47.99 both \$80.99
1312 pages



"This easy-to-read textbook/reference presents an essential guide to object-oriented C++ programming for scientific computing."

Springer; 2012 edition (February 16, 2012)
\$35.96, 250 pages

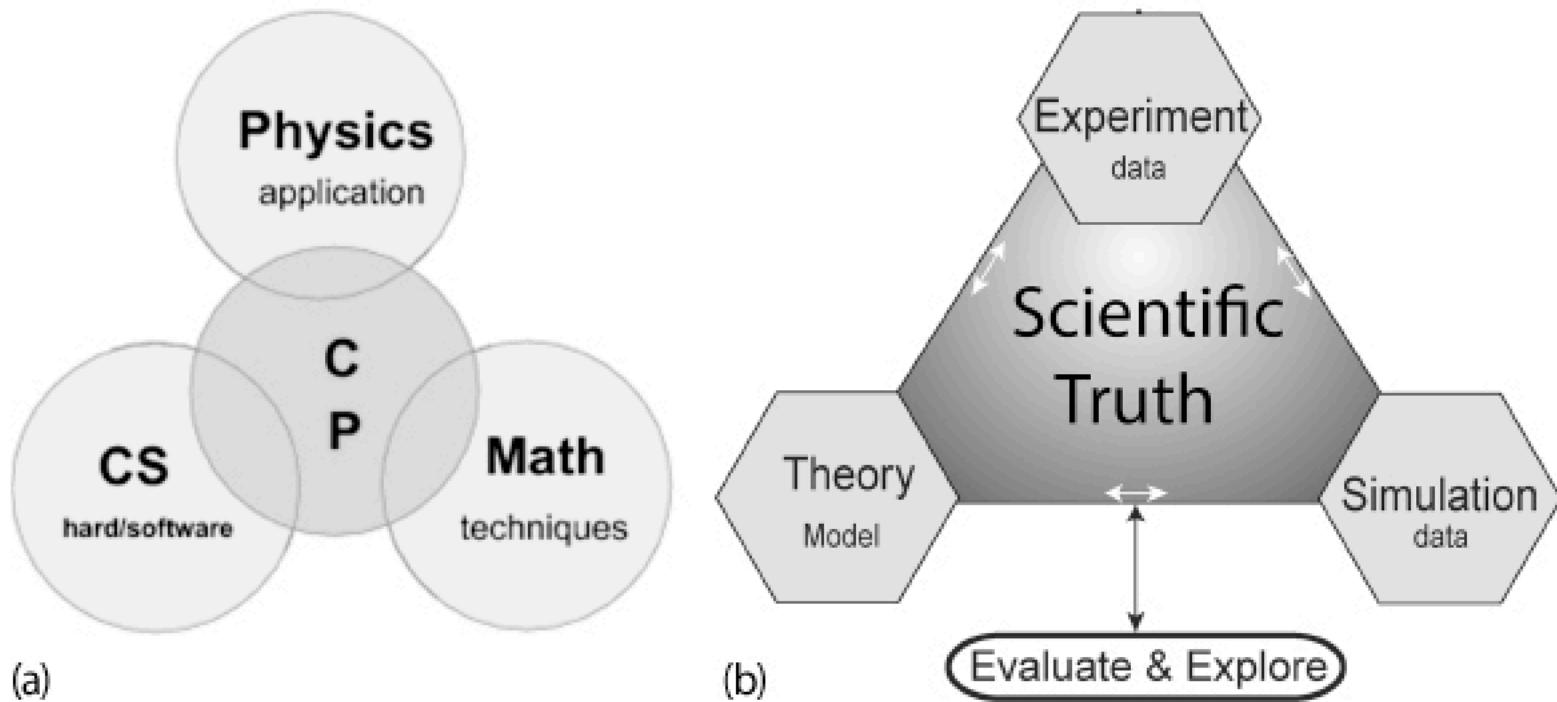
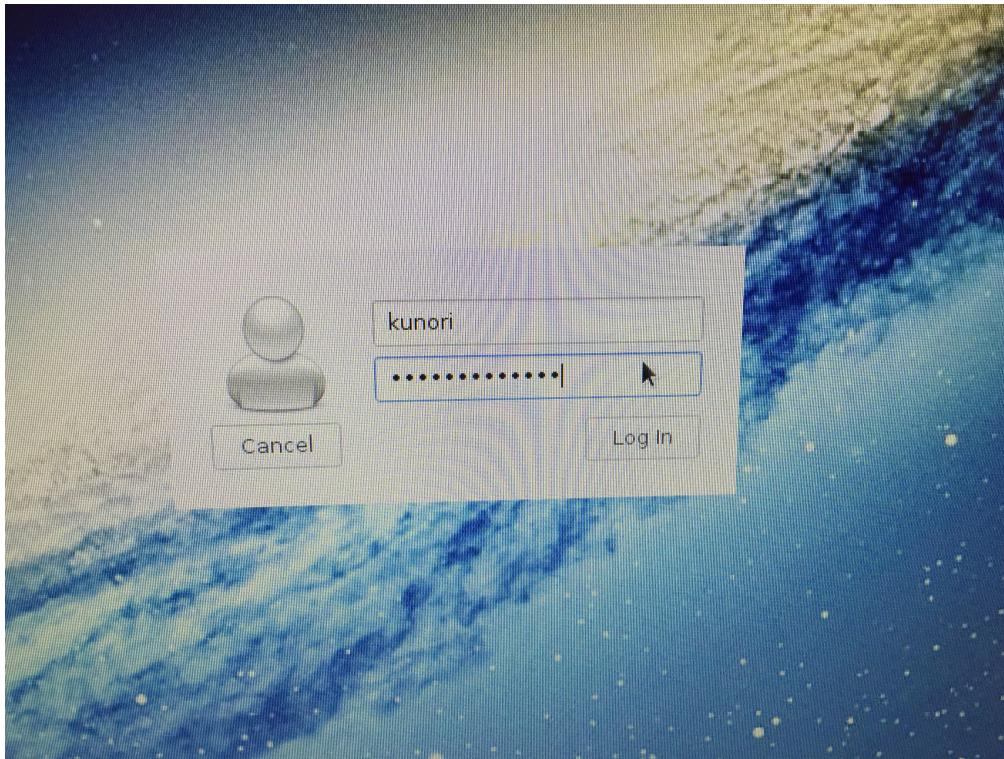


Figure 1.1 (a) A representation of the multi-disciplinary nature of computational physics as an overlap of physics, applied mathematics and computer science, and as a bridge among them. (b) Simulation has been added to experiment and theory as a basic approach in the search for scientific truth. Although this book focuses on simulation, we present it as part of the scientific process.

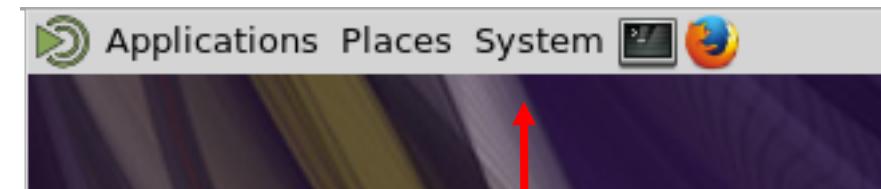
from Computational Physics, R. Landau

Let's start.

1) User name and password

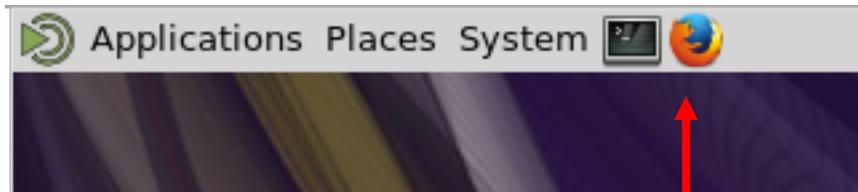


By the way, please logout at the end of each class before leaving.



- 1) System
- 2) Logout

0) Open Web browser (Firefox)



Firefox

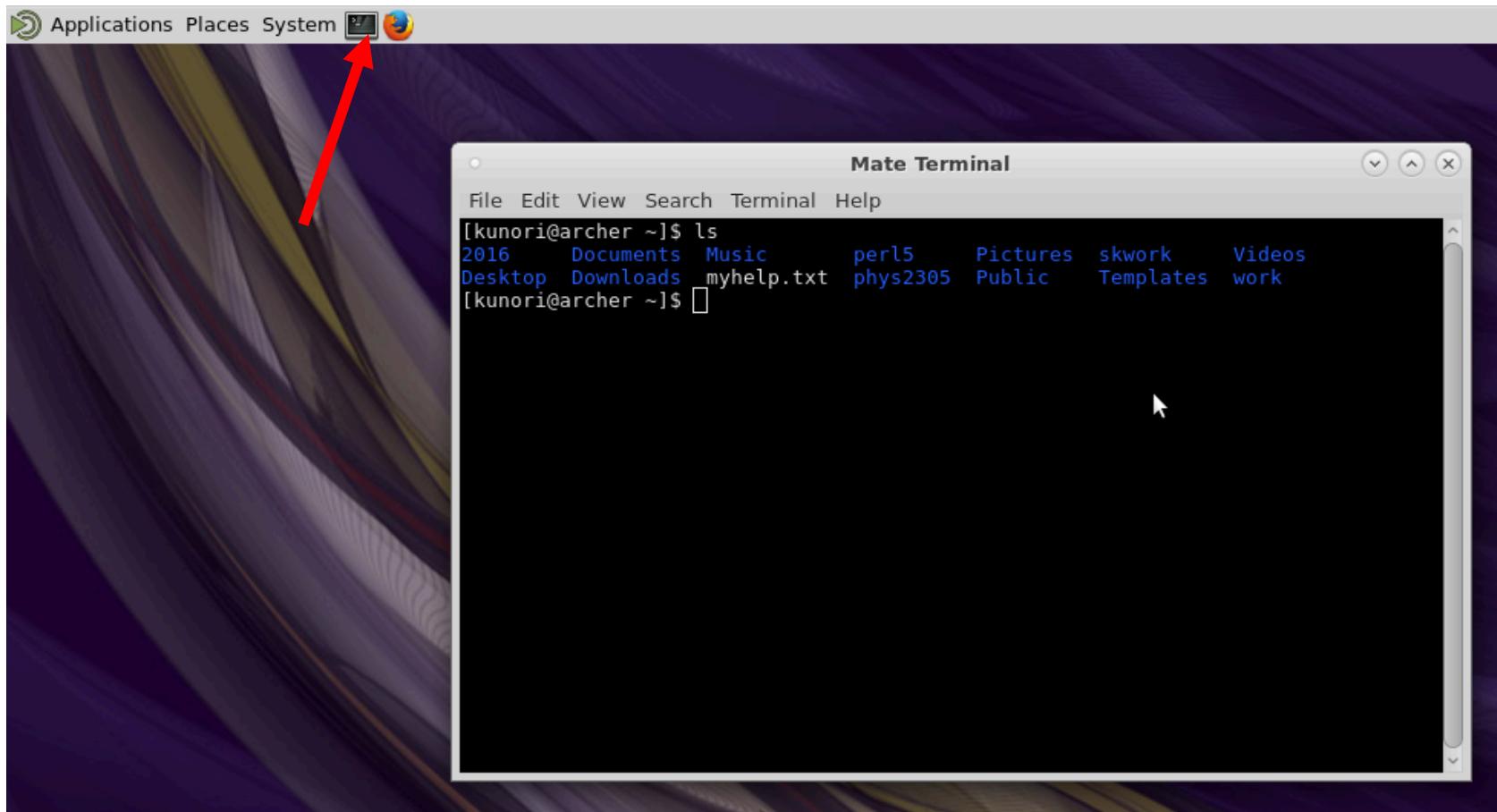
Accessing Kunori's link page in Dropbox

0.1) Type in the following link in the firefox url window.

<https://www.dropbox.com/s/mn6tef90697qgl3/phys2305links.pdf?dl=0>

to open the Phys2305 link file.

1) Open terminal



Try some linux commands:

pwd

ls

ls -l

whoami

(copy .emacs file from kunori)

cd

cp /home/kunori/.emacs .

ls -lra

(list all files in time-reverse order)

2) Create subdirectories and create a C++ program – helloworld.cc

```
1 //  
2 // g++ -std=c++11 helloworld.cc  
3 // ./a.out  
4 //  
5 #include <iostream>  
6  
7 int main()  
8 {  
9     std::cout << "Hello World!" << std::endl;  
10    return 0;  
11 }
```

“Print Screen” on keyboard for a screen shot.
A png file will be saved in Desktop.

- :**- helloworld.cc All L1 (C++/l Abbrev)

(in terminal window)

cd

mkdir phys2305 (make a new directory)

ls

cd phys2305 (change directory)

mkdir week1 (make a new subdirectory)

cd week1 (change directroy)

emacs helloworld.cc & (open eamcs editor)

(in emacs window)

#include <iostream>

int main()

{

std::cout << "Hello World!" << std::endl;

return 0;

}

== save code ==

(in terminal window)

g++ -std=c++11 helloworld.cc

./a.out

Output on terminal – just one line.

Hello World!

Warning: Do not copy source code from pdf images because of “hidden characters in pdf mess up .cc file. Example codes are available at kunori’s Dropbox through the link page” (at the bottom of the page).

<https://www.dropbox.com/s/mn6tef90697qgl3/phys2305links.pdf?dl=0>

3 Example Code

Week 1- helloworld, forloop, sinecurve.

<https://www.dropbox.com/s/ju51tu68doef153/helloworld.cc?dl=0>

<https://www.dropbox.com/s/dkdqr9zz1pnughh/forloop.cc?dl=0>

<https://www.dropbox.com/s/0v10uugbcuqm80w/sinecurve.cc?dl=0>

<https://www.dropbox.com/s/aw6cq0gfbtt913v/sineplot.cc?dl=0>

<https://www.dropbox.com/s/yiatic3zrn7qlx0/myplotSine.p?dl=0>

Week 2-

What's next?

Let's create our second program to try "for loop".

- 1) create a new program based on an existing program, i.e. helloworld.cc.

Copy helloworld.cc to a new .cc file.

```
[kunori@archer week1]$ cp helloworld.cc forloop.cc
```

- 2) Add code in forloop.cc

for-loop

Google:for loop C++ example

https://www.tutorialspoint.com/cplusplus/cpp_for_loop.htm

```
#include <iostream>
using namespace std;

int main () {
    // for loop execution
    for( int a = 10; a < 20; a = a + 1 ) {
        cout << "value of a: " << a << endl;
    }

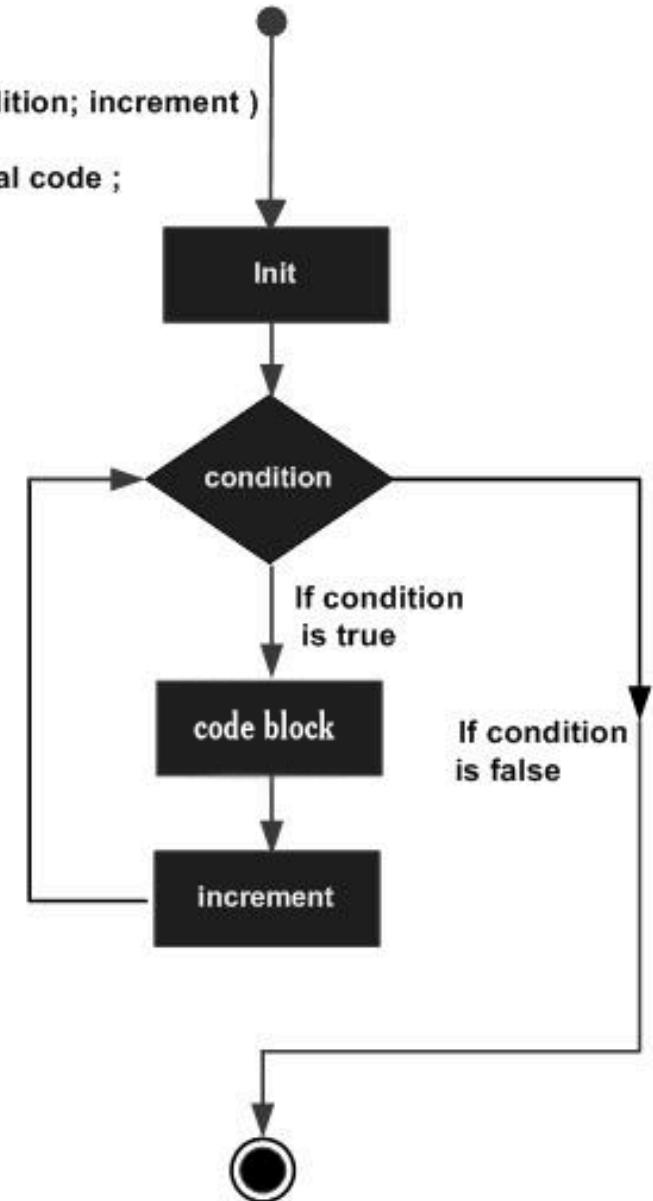
    return 0;
}
```

result:

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

Flow Diagram

```
for( init; condition; increment )
{
    conditional code ;
}
```



Next: Calculate $\sin(x)$ and then plot a sine curve using gnuplot.

Arithmetic operator

operator	description
+	addition
-	subtraction
*	multiplication
/	division
%	modulo

Google: "C++ math"

<http://www.cplusplus.com/reference/cmath/>

cos
sin
tan
acos
asin
atan
atan2

cosh
sinh
tanh
acosh C++11
asinh C++11
atanh C++11

exp
frexp
ldexp
log
log10

pow
sqrt

$7^3 \rightarrow \text{pow}(7.0, 3.0)$

About if statement and Operators.

Google search: “C++ if statement”

<http://www.cplusplus.com/doc/tutorial/control/>

```
if (x > 0)
    cout << "x is positive";
else if (x < 0)
    cout << "x is negative";
else
    cout << "x is 0";
```

Google: “C++ operators”

<http://www.cplusplus.com/doc/tutorial/operators/>

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Logical
operators

operator
&&

AND example: if(a>b && x==y) p=10.0;
OR example: if(a>b || x==y) q=5.0;

https://www.tutorialspoint.com/cplusplus/cpp_if_else_statement.htm

https://www.tutorialspoint.com/cplusplus/cpp_operators.htm

```

//  

// g++ -std=c++11 sinecurve.cc  

// ./a.out  

//  

#include <iostream>  

#include <math.h> // for sin(), cos() ← New  

using namespace std;  

int main()  

{  

    std::cout << "Hello World!" << std::endl;  

    double x=0.0;  

    double dx=0.1;  

    for( int i = 0; i<100; i=i+1) {  

        x=x+dx;  

        double y=sin(x);  

        cout<<" i "<<i<<" x "<<x<<" y "<<y<<endl;  

        if(y>-0.1 && y<0.1) {  

            cout<<" y is between -0.1 and 0.1: y="<<y<<endl;
        }
    }
    return 0;
}

```

[kunori@archer week1]\$./a.out
Hello World!

i 0	x 0.1	y 0.0998334
y is between -0.1 and 0.1: y=0.0998334		
i 1	x 0.2	y 0.198669
i 2	x 0.3	y 0.29552
i 3	x 0.4	y 0.389418
i 4	x 0.5	y 0.479426
i 5	x 0.6	y 0.564642
i 6	x 0.7	y 0.644218
i 7	x 0.8	y 0.717356
i 8	x 0.9	y 0.783327
i 9	x 1	y 0.841471
i 10	x 1.1	y 0.891207
i 11	x 1.2	y 0.932039
i 12	x 1.3	y 0.963558
i 13	x 1.4	y 0.98545
i 14	x 1.5	y 0.997495
i 15	x 1.6	y 0.999574

About function

Google search: C++ function
finds

<http://www.cplusplus.com/doc/tutorial/functions/>

```
1 // function example
2 #include <iostream>
3 using namespace std;
4
5 int addition (int a, int b)
6 {
7     int r;
8     r=a+b;
9     return r;
10}
11
12 int main ()
13 {
14     int z;
15     z = addition (5,3);
16     cout << "The result is " << z;
17 }
```

The result is 8

5) Write data to an output file (in text format)

X and y values in the output file

```
$ cat myoutfile.txt
```

0	0
0.1	0.0998334
0.2	0.198669
0.3	0.29552
0.4	0.389418
0.5	0.479426
0.6	0.564642
0.7	0.644218
0.8	0.717356
0.9	0.783327
1	0.841471
1.1	0.891207
1.2	0.932039
1.3	0.963558
1.4	0.98545
1.5	0.997495
1.6	0.999574
1.7	0.991665
1.8	0.973848
1.9	0.9463

Open file

```
#include <iostream>
#include <fstream> // for output file ← New
#include <math.h> // for sin(), cos()

using namespace std; // to omit std on cout...

// my own function
double myfunction(double x) { ← New
    double y=sin(x);
    return y;
}

int main()
{
    cout << "C++ code examples..."<<endl;

    // static sine curve...
    string fname="myoutfile.txt"; // define output file name
    ofstream outfile; // open output file...
    outfile.open(fname.c_str()); // C++ lib on archer is old,
    // outfile.open(fname); // new C++ lib use string.

    double xmin=0.0;
    double xmax=50.0;
    double dx=0.1;

    for (double x=xmin; x<xmax; x=x+dx) {
        double y=myfunction(x);
        outfile<< " " <<x<< " " <<y<<endl; ← New (write x, y to output file)
    }

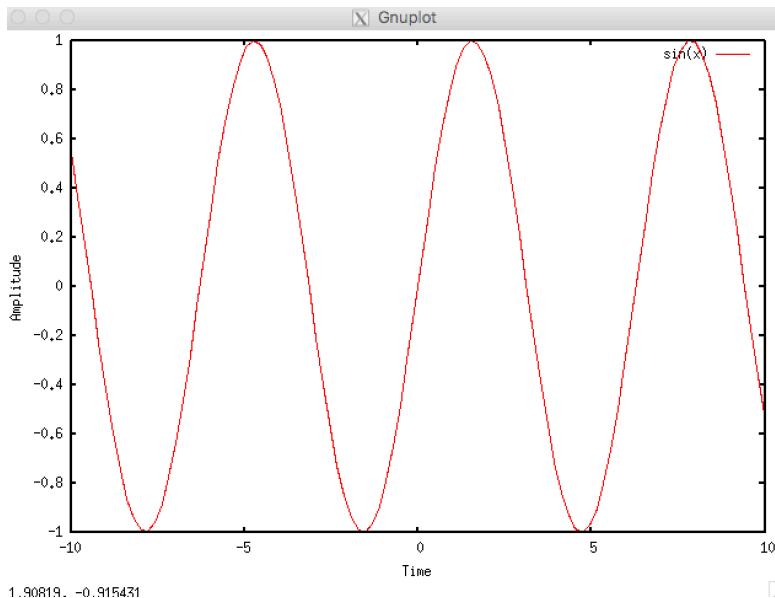
    outfile.close(); ← New (close output file)

    return 0;
}
```

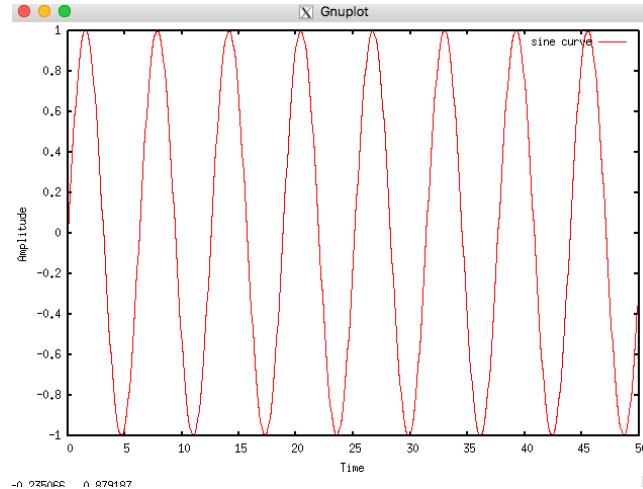
gnuplot

Brief tutorial: <http://people.duke.edu/~hpgavin/gnuplot.html>

gnuplot> plot sin(x)



gnuplot> load "myplotSine.p"



```
clear
# ##
#set terminal pdf
#set output "sinecurve.pdf"
# ##
# set terminal pngcairo
# set output "sinecurve.png"
#
set xlabel "Time"
set ylabel "Amplitude"
plot "myOutfile.txt" using 1:2 title "sine curve" with lines
```

myplotSine.p

are comment lines.

No homework for this week, but try those.

- 1) Plot a cosine curve plot instead of a sine curve plot.

copy the code for sin curve to a new file, i.e.

`cp sineplot.cc cosineplot.cc`

change “sin” to “cos”, and change the file name.

modify myplotSine.p to plot the new cosine curve, i.e.

`cp myplotSine.p myplotCosine.p`

- 2) Superimpose the sine and cosine curves.

`cp myplotSine.p myplotSineCosine.p`

- 3) Write the plot to a png file

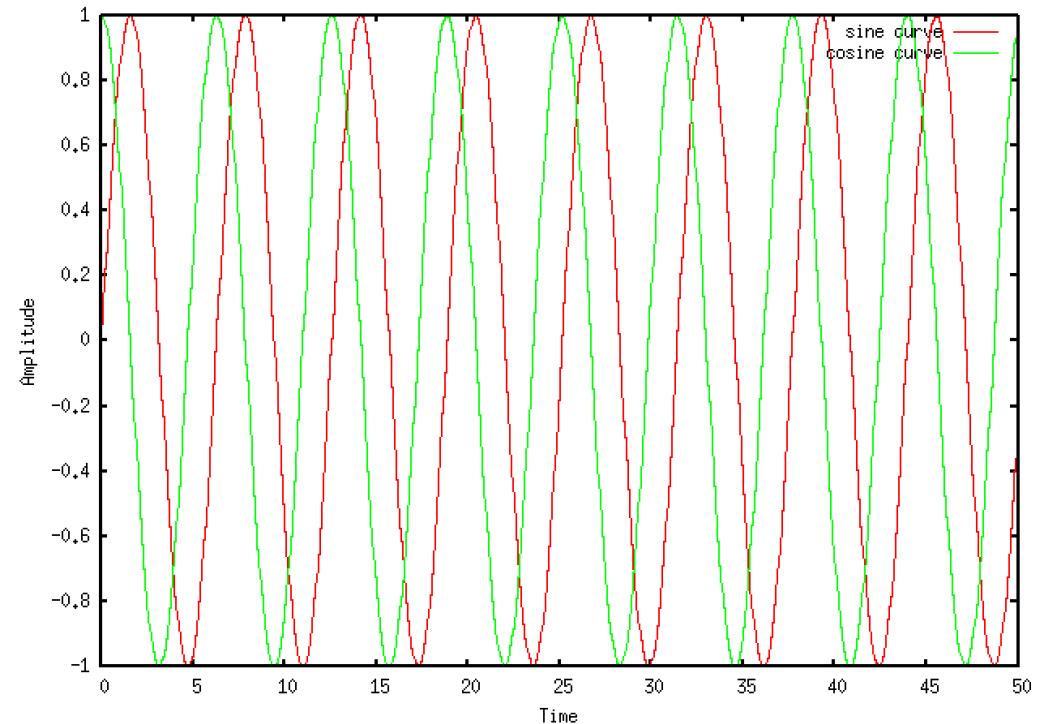
activate “set terminal” and “set output” in the file , i.e. myplotSineCosine.p, i.e

`set terminal pngcairo`

`set output "sinecurve.png"`

- 4) Display png file

`display sinecurve.png`



```
clear
# ##
#set terminal pdf
#set output "sinecurve.pdf"
# ##
set terminal pngcairo
set output "sinecurve.png"
#
set xlabel "Time"
set ylabel "Amplitude"
plot "myOutfile.txt" using 1:2 title "sine curve" with lines,\
"myOutCosine.txt" using 1:2 title "cosine curve" with lines
```