PHYS2305-Week 3

S. Kunori (TTU)

11-Sep-2017

Last Week

Numerical equation solving

$f(x) = \sin(x)*x-1.0 = 0$

Numerical Differentiation

$d(\sin(x))/dx$

Numerical integration

Gaussian distribution

- three methods

Rectangle

Trapezoidal

Simpson’s

```
function f(x) {
    calculation
    return a value (y) to main
    }

main() {
    open output file

    for-loop {
        y=function(x)
        write x,y to output file
    }

    close output file
}
```

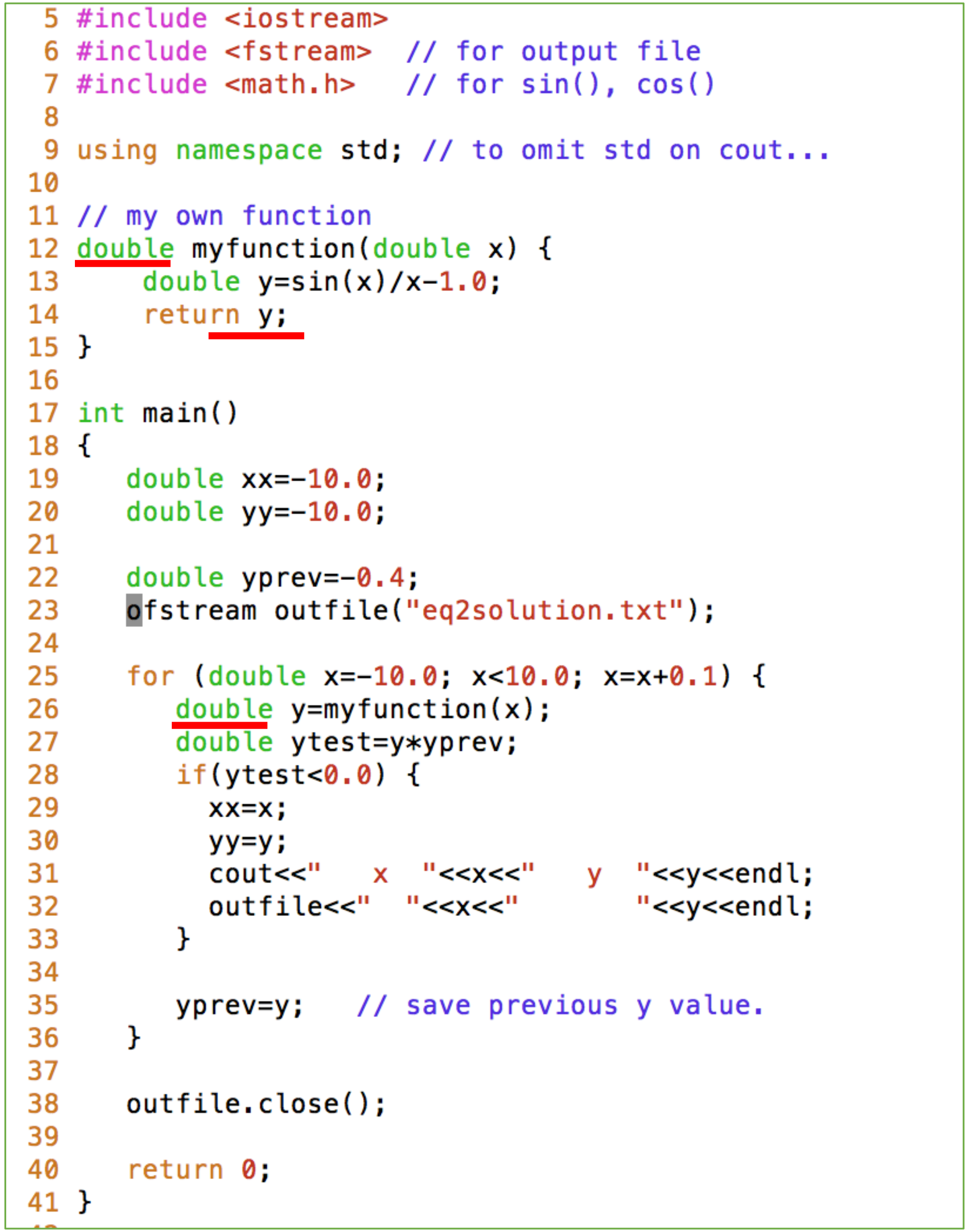
Variable (data) type of y is double.

Homework 1:

First submission: Mon 9/11 (required)
Second submission Mon 9/18 (for upgrade)
send plots and codes to shuichi.kunori@ttu.edu email subject: phys2305 homework-1

An example of function from week2.

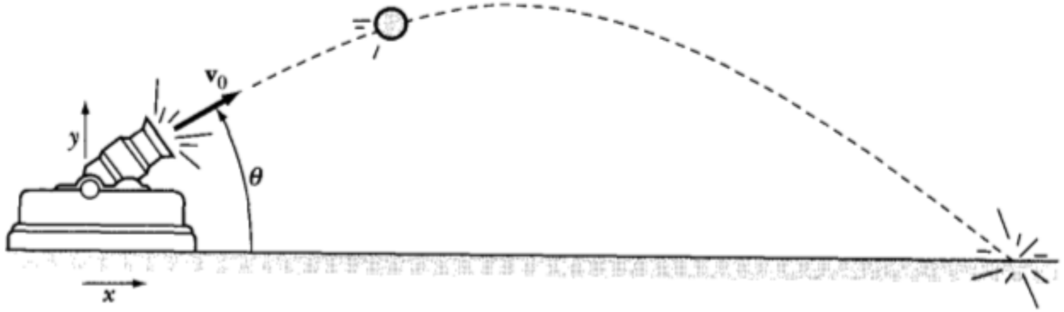
This function returns one value in double.



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

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

// my own function
double myfunction(double x) {
    double y=sin(x)/x-1.0;
    return y;
}

int main()
{
   double xx=-10.0;
   double yy=-10.0;

   double yprev=-0.4;
   ofstream outfile("eq2solution.txt");

   for (double x=-10.0; x<10.0; x=x+0.1) {
      double y=myfunction(x);
      double ytest=y*yprev;
      if(ytest<0.0) {
        xx=x;
        yy=y;
        cout<<"   x  "<<x<<"   y  "<<y<<endl;
        outfile<<"  "<<x<<"       "<<y<<endl;
      }

      yprev=y;  // save previous y value.
   }

   outfile.close();

   return 0;
}
```

Projectile and trajectory

1) Let’s plot trajectories of a particle (mass 2 kg) from muzzle angle (from 0 to 90 deg) with the initial velocity of 100 m/s. 2) Find the muzzle angle which gives the maximum range.



Position of the projectile at time t

$$x(t) = v0 * \cos(\theta) * t$$

$$y(t) = -0.5 * 9.8 * t * t \quad + \quad v0 * \sin(\theta) * t$$

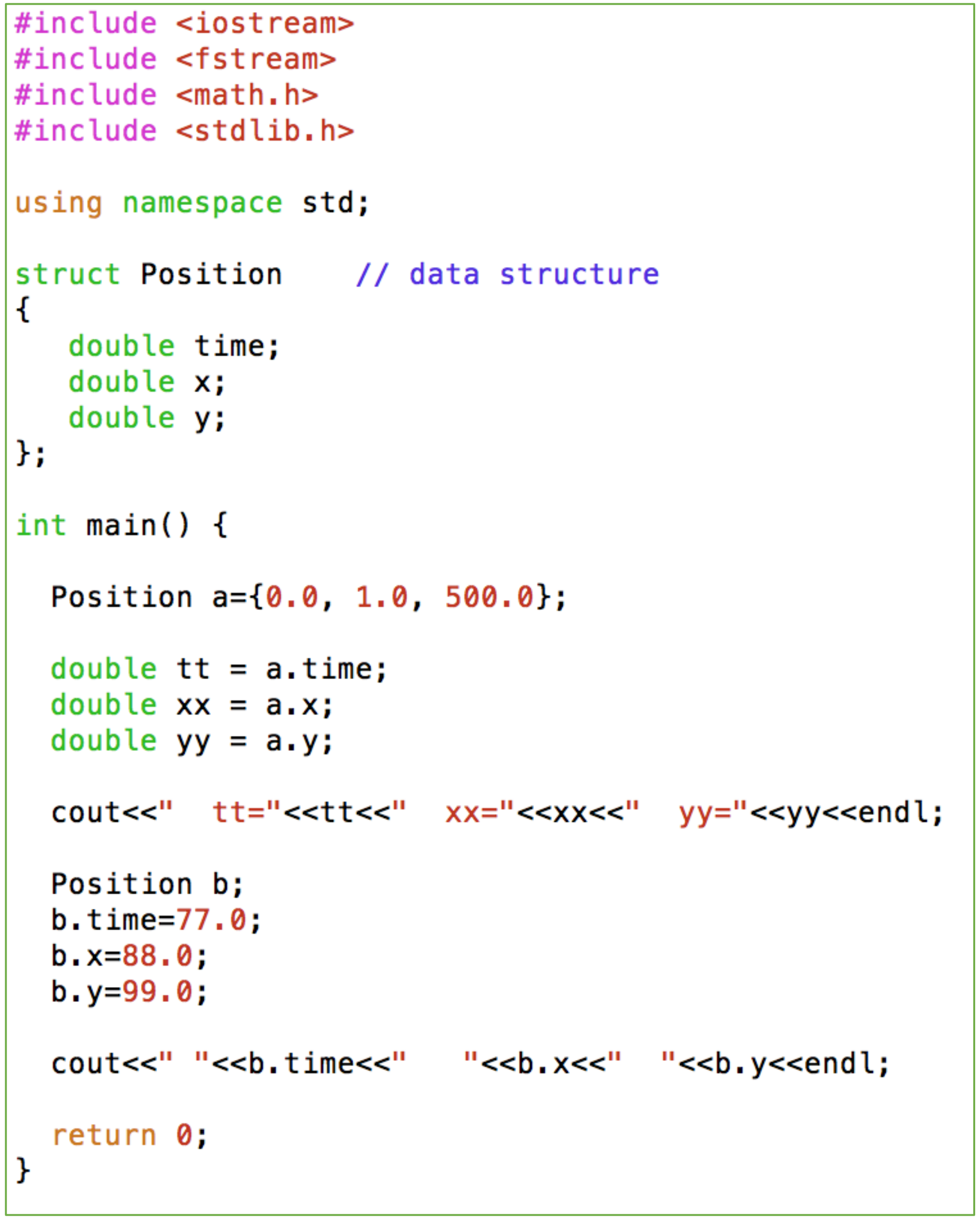
The projectile hits ground, when y(t) = 0.
the distance in x → range

```
function(t) {
    calculate x(t), y(t)
    return → [x, y] or [t,x,y]
    }

main() {
    open output file

    for-loop over time{
        [t,x,y]=function(t)
        write t, x, y to output file
}

    close output file
}
```

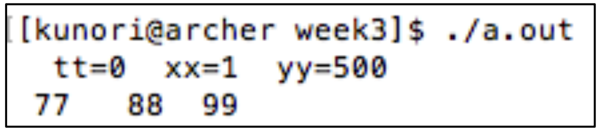
struct

define customized data type.

In this example, Position is a new data type, equivalent to double, int in standard C++ data type.



```
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdlib.h>

using namespace std;

struct Position    // data structure
{
   double time;
   double x;
   double y;
};

int main() {

  Position a={0.0, 1.0, 500.0};

  double tt = a.time;
  double xx = a.x;
  double yy = a.y;

  cout<<"  tt="<<tt<<"  xx="<<xx<<"  yy="<<yy<<endl;

  Position b;
  b.time=77.0;
  b.x=88.0;
  b.y=99.0;

  cout<<" "<<b.time<<"   "<<b.x<<"  "<<b.y<<endl;

  return 0;
}
```



```
[kunori@archer week3]$ ./a.out
  tt=0  xx=1  yy=500
 77   88  99
```

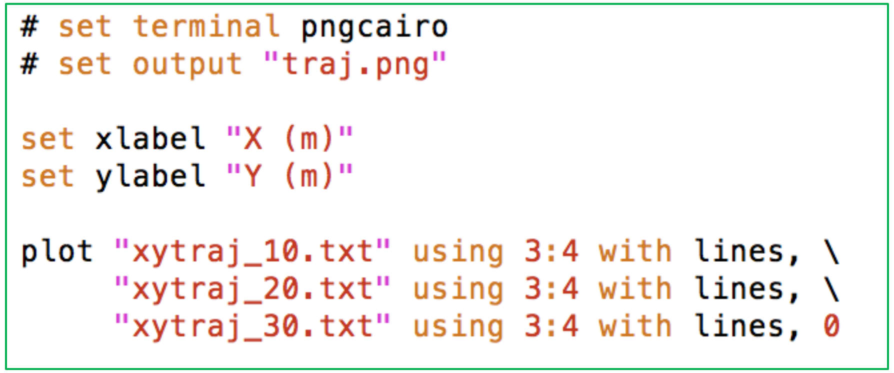
Trajectories for theta = 10, 20, 30 deg.

C++ code on the next page uses

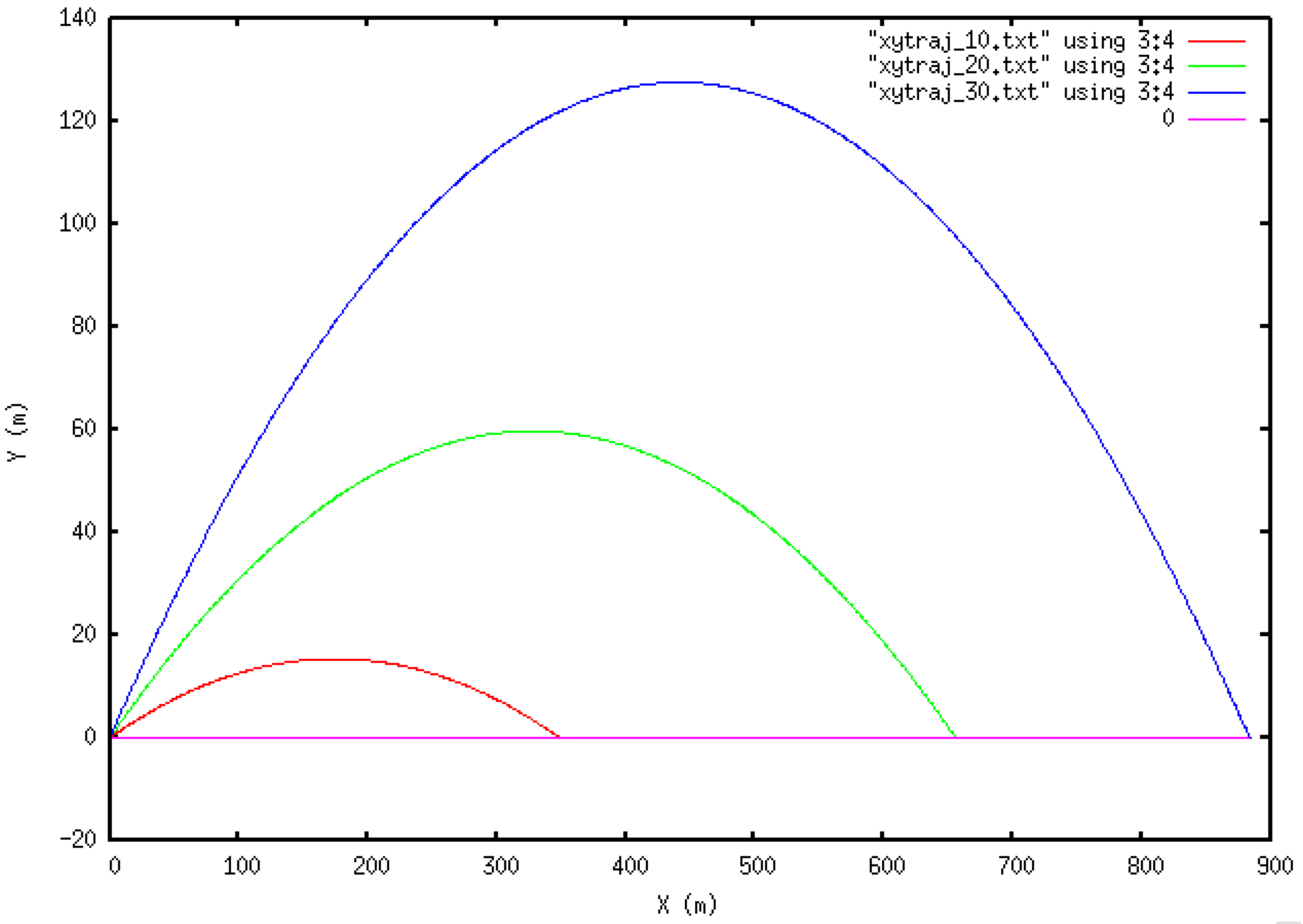
struct Position

to make this trajectory plot.

plotTraj1.p

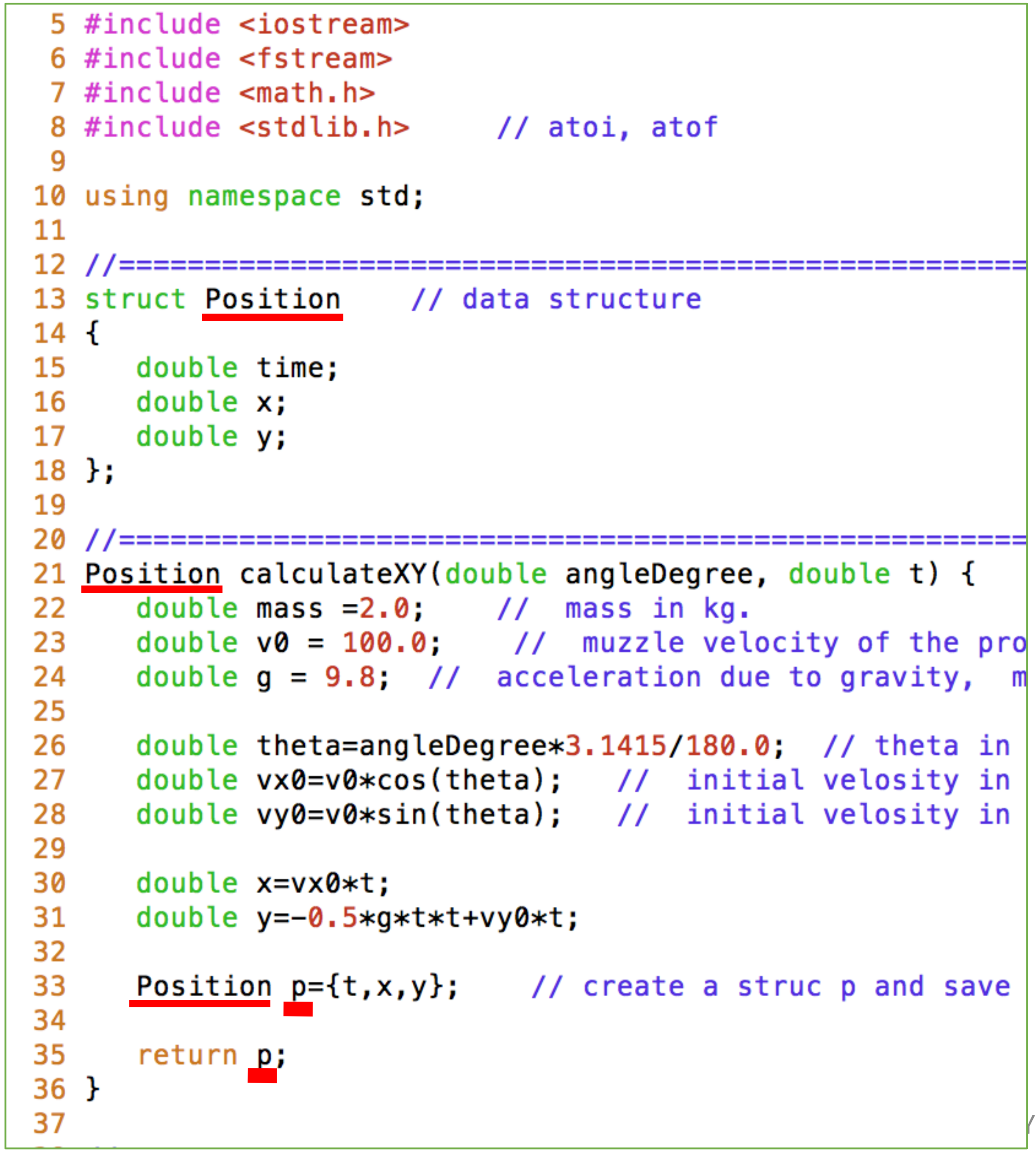


```
# set terminal pngcairo
# set output "traj.png"

set xlabel "X (m)"
set ylabel "Y (m)"

plot "xytraj_10.txt" using 3:4 with lines, \
     "xytraj_20.txt" using 3:4 with lines, \
     "xytraj_30.txt" using 3:4 with lines, 0
```

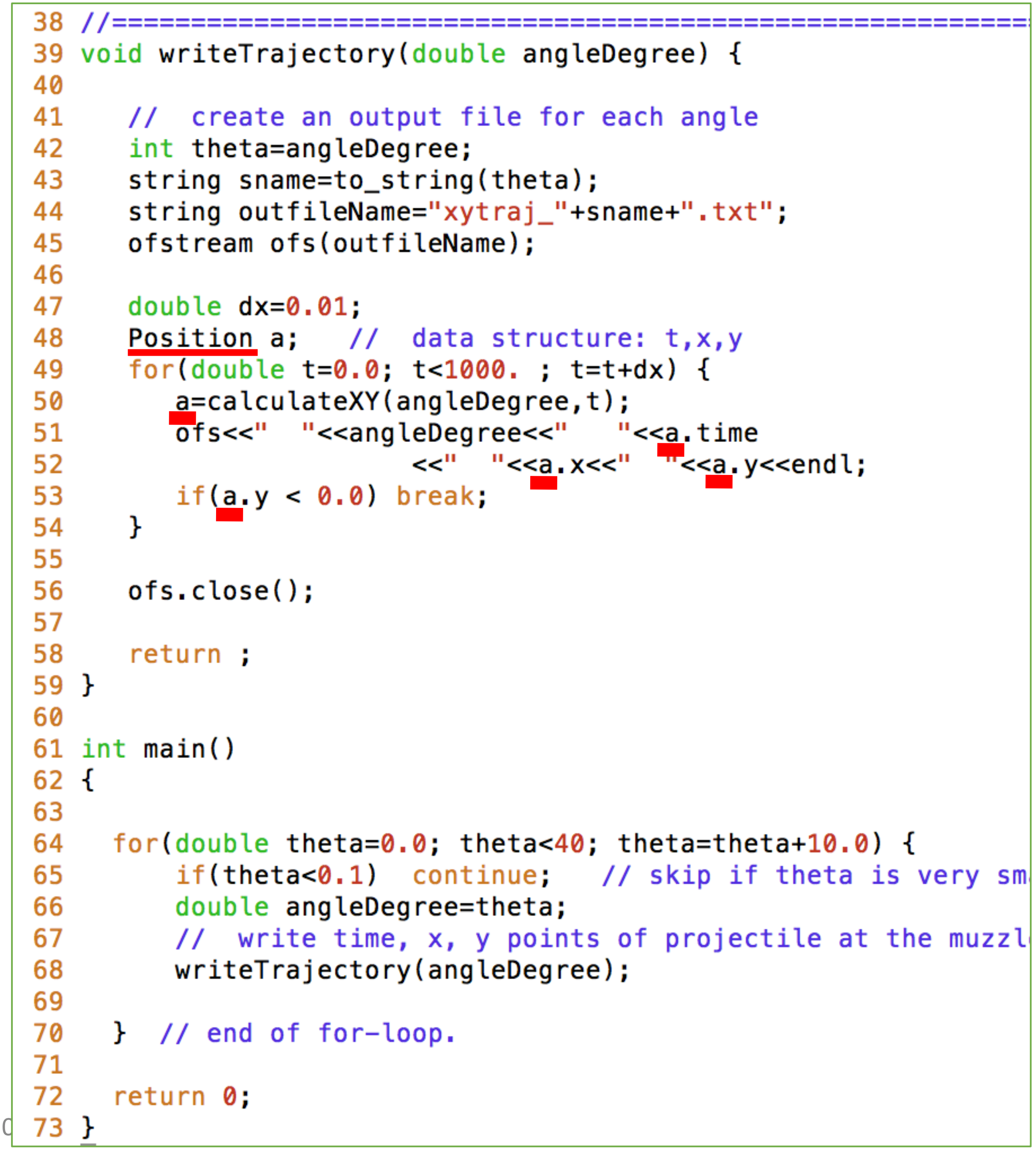


(Not good x-y aspect ratio- angles appear larger.)

trajectory1.cc



```cpp
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdlib.h>     // atoi, atof

using namespace std;

//==================================================================
struct Position    // data structure
{
   double time;
   double x;
   double y;
};

//==================================================================
Position calculateXY(double angleDegree, double t) {
   double mass =2.0;    //  mass in kg.
   double v0 = 100.0;    //  muzzle velocity of the pro
   double g = 9.8;  //  acceleration due to gravity,  m

   double theta=angleDegree*3.1415/180.0;  // theta in
   double vx0=v0*cos(theta);   //  initial velosity in
   double vy0=v0*sin(theta);   //  initial velosity in

   double x=vx0*t;
   double y=-0.5*g*t*t+vy0*t;

   Position p={t,x,y};    // create a struc p and save

   return p;
}

```



```cpp
//==================================================================
void writeTrajectory(double angleDegree) {

   //  create an output file for each angle
   int theta=angleDegree;
   string sname=to_string(theta);
   string outfileName="xytraj_"+sname+".txt";
   ofstream ofs(outfileName);

   double dx=0.01;
   Position a;   //  data structure: t,x,y
   for(double t=0.0; t<1000. ; t=t+dx) {
      a=calculateXY(angleDegree,t);
      ofs<<"  "<<angleDegree<<"   "<<a.time
                     <<"  "<<a.x<<"  "<<a.y<<endl;
      if(a.y < 0.0) break;
   }

   ofs.close();

   return ;
}

int main()
{

  for(double theta=0.0; theta<40; theta=theta+10.0) {
      if(theta<0.1)  continue;   // skip if theta is very sm
      double angleDegree=theta;
      //  write time, x, y points of projectile at the muzzl
      writeTrajectory(angleDegree);

  }  // end of for-loop.

  return 0;
}
```

Homework 2

First due: Mon. 9/18
Final due: Mon. 9/25 for upgrade
Email to shuichi.kunori@ttu.edu
Subject: phys2305 homework-2

C-course:

Make one of two pots on the right, i.e. trajectory (theta 10-80 deg) or range (1-89 deg)

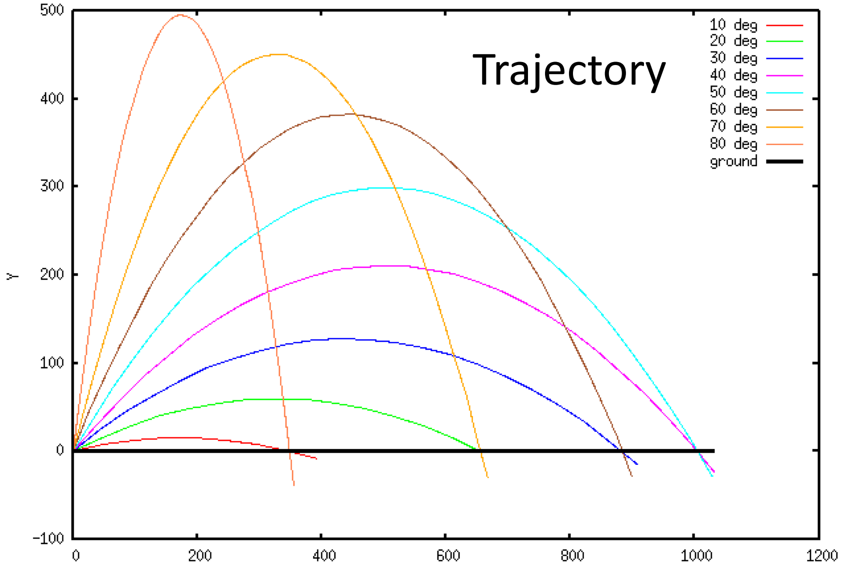
B-course:

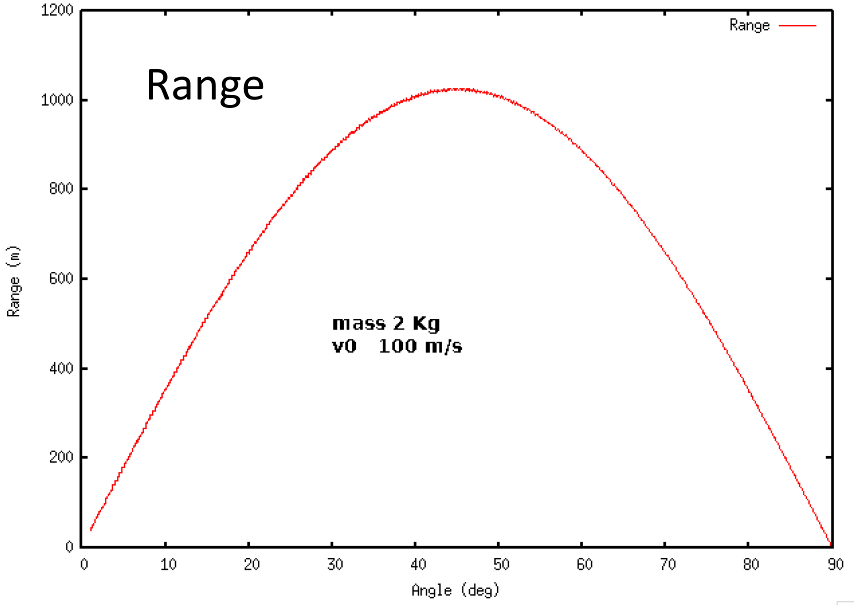
Make both plots on the right

A-course:

Anything beyond C and B, for example,

- Improvement of the appearance of those plots
    (eg larger fonts, good x-y aspect ratio)
- additional plots
    (eg. flight time vs muzzle angle)
- etc.


Trajectory
10 deg
20 deg
30 deg
40 deg
50 deg
60 deg
70 deg
80 deg
ground
Y


Range
Range
mass 2 Kg
v0 100 m/s
Range (m)
Angle (deg)