

Feb 14, 06 8:17 **random_walk.cpp** Page 1/2

```
// file: random_walk.cpp
//
// Program to illustrate random walks
//
// Programmer: Dick Furnstahl  furnstahl.1@osu.edu
//
// Revision history:
// 03/05/04  translated from random_walk.c (11/25/02)
// 02/19/05  added more comments and math.h
// 02/14/06  added output comment
//
// Notes:
// * implements method 2 from the list in section 6.10
//   of the Landau/Paez text.
// * random numbers are generated uniformly from a to b
// * uses the GSL random number functions
// * both the gsl_rng.h and gsl_randist.h header files are needed
// * the current version uses the gsl_rng_taus random number
//   generator.  There are many other choices (just change
//   the name in the gsl_rng_alloc statement).  See the GSL
//   manual for a list of generators and their properties.
//
//*****

// include files
#include <iostream>           // cout and cin
#include <iomanip>            // manipulators like setprecision
#include <fstream>           // file input and output
#include <cmath>
using namespace std;        // we need this when .h is omitted

#include <gsl/gsl_rng.h>     // GSL random number generators
#include <gsl/gsl_randist.h> // GSL random distributions

// function prototypes
extern unsigned long int random_seed (); // routine to generate a seed

//*****
int
main (void)
{
    int npts = 100;           // size of random walk
    unsigned long int seed;   // seed for random number generator

    double lower = -sqrt (2.); // lower limit of uniform range
    double upper = sqrt (2.);  // upper limit of uniform range

    double delta_x = 0.;     // uniform random number from [lower,upper]
    double delta_y = 0.;     // 2nd random number from [lower,upper]
    double x = 0.;          // current x
    double y = 0.;          // current y

    gsl_rng *rng_ptr;        // declare pointer to random number
                            // generator (rng)

    rng_ptr = gsl_rng_alloc (gsl_rng_taus); // allocate the rng
    cout << "Enter a long integer as a seed or 0 to generate one: ";
    cin >> seed;
    if (seed == 0)
    {
        // generate a seed from /dev/random
        seed = random_seed ();
    }

    cout << " Using " << seed << " to seed the RNG" << endl;
    gsl_rng_set (rng_ptr, seed); // seed the rng

    cout << "How many random numbers? ";
    cin >> npts;
}
```

Feb 14, 06 8:17 **random_walk.cpp** Page 2/2

```
// output file random_walk.dat holds a single walk
ofstream out;
out.open ("random_walk.dat");

x = y = 0.; // start at the origin
out << "# (x,y) coordinates of a random walk with " << npts << " points"
  << endl;
out << x << " " << y << endl; // output first point

// do the walk and output to a file
for (int i = 0; i < npts; i++)
{
    delta_x = gsl_ran_flat (rng_ptr, lower, upper);
    delta_y = gsl_ran_flat (rng_ptr, lower, upper);
    x += delta_x;
    y += delta_y;
    out << x << " " << y << endl; // output after the step is taken
}
cout << "Output " << npts << " random walk steps to random_walk.dat." << endl;

gsl_rng_free (rng_ptr); // free the random number generator
out.close ();          // close the output file

return (0);
}
```