LULEÅ UNIVERSITY OF TECHNOLOGY
Hans Weber May 27, 2016

**Short note on the motion of a random walker (Wiener process)**
Notations follow the book 'Physics of Stochastic Processes' written by Mahnke, Kaupužs and Lubashevsky (MKL). The examples have been programed with matlab run in an octave environment (octave.org). Ocatave is freeware and matlab you have to buy. Both use the same syntax. Here the motion of a random walker is investigated. The position of the walker is denoted with $x(t)$ where $t$ is time. This is the case with no drift just random motion.
The steps of the walker are described accordingly:

Step 1. Define a starting position at $t = 0$ as $x(t = 0) = x_0$.

Step 2. Define the length in time of the walk $t \in [0, t_{max}]$ and a incremental time $\Delta t$ for new positions.

Step 3. Generate a random number $z$ (Gaussian distributed). In matlab there is a function *randn* for this. You may also use a 'Box–Muller' algorithm to generate $z$.

Step 4. Calculate $\Delta w = z\sqrt{\Delta t}$. This is eq. 5.153 (MKL).

Step 5. According to (MKL) eq 5.161 we have $dx(t) = adt + bdw(t)$ and with $a = 0$ and $b = 1\ \frac{\mathrm{m}}{\sqrt{\mathrm{s}}}$. We have for our discrete version the following relation (just keep track of dimensions):

$$\Delta x = b\Delta w = \Delta w \tag{1}$$

Step 6. Now assign new values to $x$ and $t$ according to

$$x_{i+1} = x_i + \Delta x \tag{2}$$
$$t = t + \Delta t \tag{3}$$

Step 7. Loop to step 3 until your condition in step 2 $t \in [0, t_{max}]$ is reached.

The steps above define our random walker. Now we turn to some results. First some attention is needed for the random number $z$ that are Gaussian distributed. They should fulfill the following requirements for the average and variance (eq. 5.159 in MKL):

$$\langle z \rangle = 0.0 \tag{4}$$
$$\langle z^2 \rangle = 1.0 \tag{5}$$
$$\tag{6}$$

if they do not the random numbers should be renormalized accordingly.
The boundary conditions used are

$$x_0 = 0.0 \text{ m} \tag{7}$$
$$t_{max} = 1000 \text{ s} \tag{8}$$
$$\Delta t = 0.1 \text{ s} \tag{9}$$

the last one is not really a boundary condition but a choice has to be made. In figure 1 four different realizations of a random walk are shown.

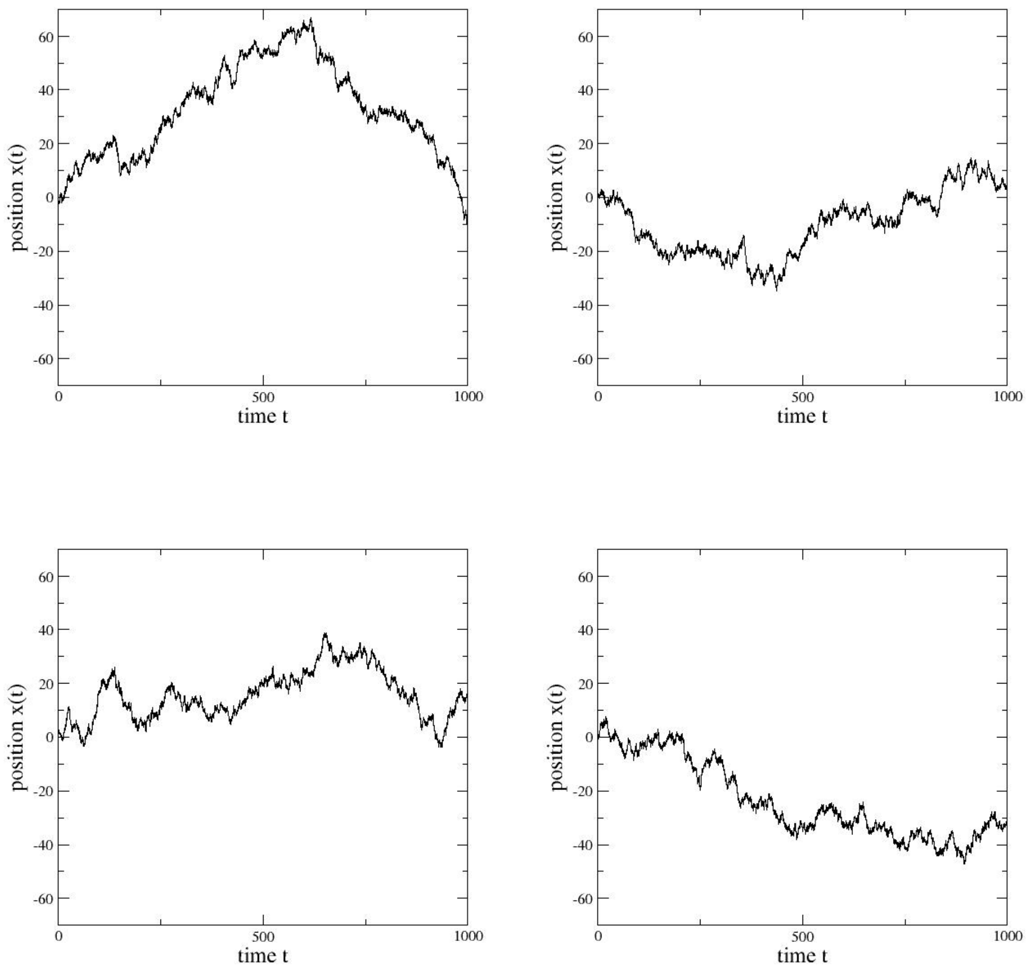

Figure 1: Four different realizations of a random walker. All walkers have starting position $x_0 = 0.0$ and run for $t_{max} = 1000$ s.

The data shown are for the time evolution of the walkers position $x(t)$. The total time of each run is $t_{max} = 1000$ s and as $\Delta t = 0.1$ each run consists of 10000 steps. At the far right in each time series as the walker reaches $t_{max}$ the walker is at position $x(t_{max})$. We now turn our attention to this final position. If we perform many realizations of random walkers and for each random walker we register its final position $x(t_{max})$. Now we can generate a distribution of these final positions. The final position forms a distribution where positions far away from the starting point are less likely and positions close to the starting point are more likely. the probability density of final positions can be shown as a histogram in the following way.

As the final positions are real numbers we will approximate them with the nearest integer and form a histogram of these final integer positions. If for a certain walker its final position is $x(t_{max}) = -28.8781963939354$ we will ad 1 to the histogram position '-29'.

The analyze this histogram consisting of the final positions of 20000 random walker. The sum of all histogram boxes is accordingly 20000 and to change this into a probability function. In figure 2 the results are shown as blue stars '*'.

It is also possible to evaluate the theoretical distribution $p(x, t)$ at time $t = t_{max}$

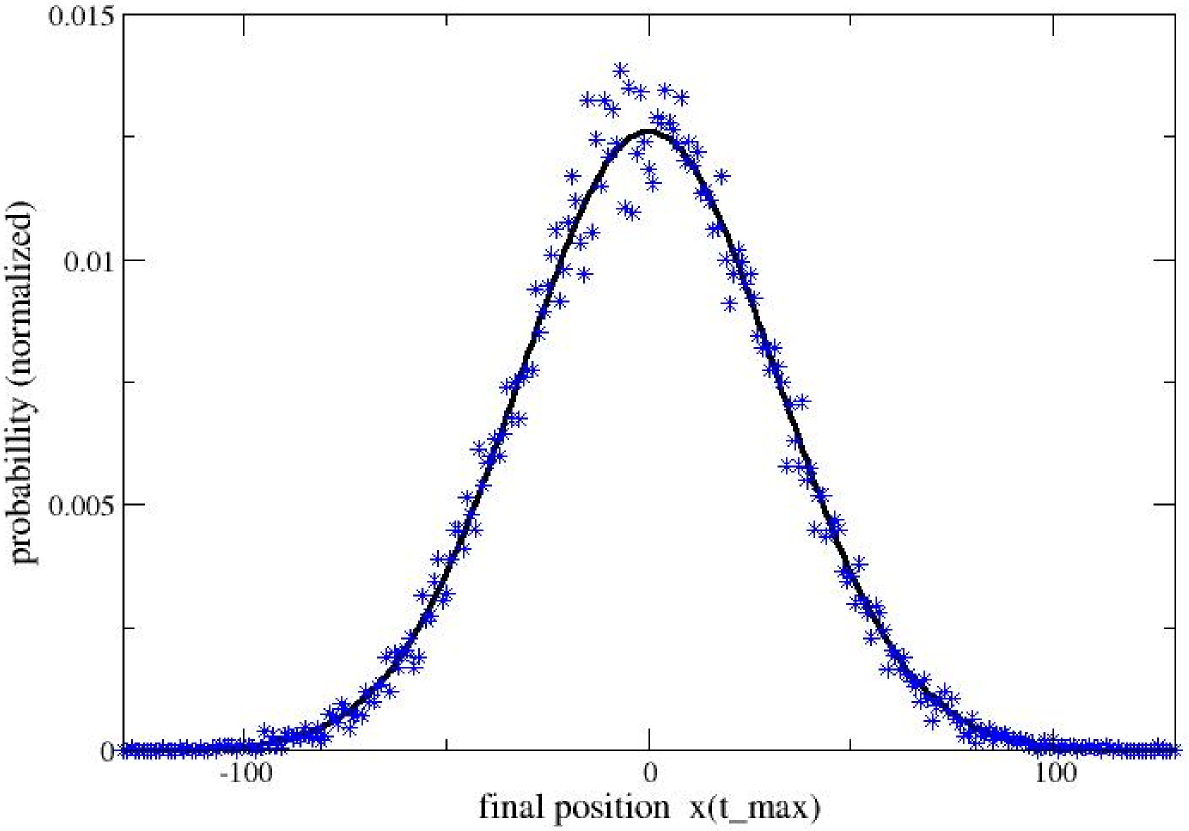

Figure 2: Histogram of the probability distribution of the final positions recorded for 20000 different walkers. The results for the walkers are marked with blue '*'. The solid line is the theoretical probability distribution function eq.(10).

according to eq. (10). This is eq. 5.162 in MKL with $x_0 = 0.0, a = 0.0$ and $b = 1.0$.

$$p(x,t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} \tag{10}$$

This function is shown in figure 2 as a solid black line.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Matlab program to visualize the stochastic process for one Random Walker
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
format long
clf ;

  X0 = 0.00 ;
  t_t(10001) = 0.0d0 ;
  X_t(10001) = 0.0d0 ;

  delta_t = 0.1 ;
  t_max = 1000.0 ;
  i_max = t_max / delta_t ;

  delta_W_t = 0.0d0 ;
  delta_x = 0.0d0 ;
  w_square = 0.0d0 ;
  xt = 0.0d0 ;

  for i = 1:i_max + 1
      Z = randn ;
      t_t(i) = i ;
      w_square = w_square + Z*Z ;
      delta_W_t = Z * sqrt(delta_t) ;
      delta_x = delta_W_t ;
      xt = xt + delta_x ;
      X_t(i) = xt ;
  end

  w_square = w_square/(double(i_max + 1))
  X_t(10001)

  t_t = t_t * delta_t ;

hold off
   plot (t_t,X_t)
   hold on

fprintf(1, '\n');

formatSpec = ' %6.3f  %9.6f \n';
fileID = fopen('timeseries.dat','w');
for i1=1: i_max + 1
  fprintf(fileID,formatSpec,t_t(i1),X_t(i1))
end
fclose(fileID);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Matlab program to visualize the stochastic process for 20000 random
% walkers as an end result a histogram is produced
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
format long

i_histo = 0 ;
i_histo_max  = 200 ;
histogram_x(401) = 0;
histogram_y(401) = 0;
for k = 1: 2*i_histo_max + 1
  histogram_x(k) = k - i_histo_max - 1 ;
end
fprintf(1, '\n');

clf ;

hold on
plt.p1 = plot(histogram_x, histogram_y,'bo','linewidth',3);
hold off
drawnow('expose')

% starts the loop to generate 20000 walkers
kmax = 20000 ;
for k=1: kmax

  X0 = 0.00 ;
  t_t(100001) = 0.0d0 ;
  X_t(100001) = 0.0d0 ;

  delta_t = 0.1 ;
  t_max = 1000.0 ;
  i_max = t_max / delta_t ;

  delta_W_t = 0.0d0 ;
  delta_x = 0.0d0 ;
  w_square = 0.0d0 ;
  xt = 0.0d0 ;

% loop for one walker
  for i = 1:i_max + 1
      Z = randn ;
      t_t(i) = i ;
      w_square = w_square + Z*Z ;
      delta_W_t = Z * sqrt(delta_t) ;
      delta_x = delta_W_t ;
      xt = xt + delta_x ;
      X_t(i) = xt ;
  end

  w_square = w_square/(double(i_max + 1))
  k
  X_t(10001)
  i_histo = round(X_t(10001))

  i_histo = i_histo_max + i_histo +1 ;
  histogram_y(i_histo)  = histogram_y(i_histo)  + 1;

   if ( (i_histo  > 0 ) && (i_histo  <=  2*i_histo_max + 1 ) )
       set(plt.p1, 'XData',histogram_x, 'YData',histogram_y);
   end
   drawnow('expose')

fprintf(1, '\n');

end


formatSpec = ' %6.3f  %9.6f \n';
fileID = fopen('histogram.dat','w');
for i1=1: 2*i_histo_max + 1
  fprintf(fileID,formatSpec,histogram_x(i1),histogram_y(i1))
end
fclose(fileID);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```