LULEÅ UNIVERSITY OF TECHNOLOGY May 31, 2016
Hans Weber

Short note on the motion of a geometric random walker

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). Octave is free ware and matlab you have to buy. Both use the same syntax. Here the motion of a **geometric** random walker is investigated. The position of the walker is denoted with $x(t)$ where t is time. This is the case with drift and random motion and in the second part a case with no drift just random motion is discussed.

The steps of the geometric 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 Δt 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). We note that dimensions for w is $\mathrm{s}^{\frac{1}{2}}$.

Step 5. According to (MKL) eq 5.166 we have $dx(t) = ax(t)dt + bx(t)dw(t)$. As one can see from this equation some care has to be taken about the choice of the starting position x_0. The choice used for the calculations here is $x_0 = 1.0$. One cannot choose $x_0 = 0.0$ as this will give no motion. Neither should x_0 be negative. In total two different choices for the parameters a and b are made. We have for our discrete version the following relation (just keep track of dimensions):

$$\Delta x = ax(t)\Delta t + bx(t)\Delta w \tag{1}$$

here dimensions for a is s^{-1} and for b is $\mathrm{s}^{-\frac{1}{2}}$ as the dimension of w is $\mathrm{s}^{\frac{1}{2}}$.

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

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

$$t_{i+1} = t_i + \Delta t \tag{3}$$

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 geometric walker. Now we turn to some results for two cases. First some attention is needed for the random numbers z that are Gaussian distributed. They should fulfil 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 renormalised accordingly.

The boundary conditions used are

$$x_0 = 1.0 \text{ m} \tag{7}$$
$$t_{max} = 10 \text{ s} \tag{8}$$
$$\Delta t = 0.001 \text{ s} \tag{9}$$

the last one is not really a boundary condition but a choice has to be made.

Results for $a = 1.0$ s^{-1} and $b = 0.5$ s$^{-\frac{1}{2}}$ (with drift).

A note of caution first. As we integrate out the model with increments in x given by eq. 1 one might think it could be possible to go from a $x(t) > 0$ to a negative $x(t)$. In principle a step in a Euler or Runge–Kutta integration would allow for this but if this happens the numerical integration is wrong as it gives an unphysical result and you should reduce Δt. Note that as $x(t)$ gets smaller also Δx get smaller according to eq. 1. It is also apparent from eq. 10 that x should not become negative.

The reason for this particular choice of a and b is that it is the one shown in MKL Figure 5.10 . In figure 1 four different realisations of a geometric random walk are shown.

The data shown are for the time evolution of the walkers position $x(t)$. The total time of each run is $t_{max} = 10$ s and as $\Delta t = 0.001$ s 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 realisations 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 positions form a distribution where positions far away from the starting point (drifting average) are less likely and positions close to the starting point (drifting average) are more likely. The probability density of final the 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 histogram consisting of the final positions of 10000 geometric random walkers is analysed in the following way. The sum of all histogram boxes is accordingly 10000 and to change this into a probability function the values in the histogram have to be divided by 10000 (as we have integer length of the boxes). This will transform the histogram into a probability density. The sum of all densities will equal 1.00 . In figure 2 the results are shown as blue stars '*'.

It is also possible to evaluate the theoretical distribution $p(x,t)$ at time $t = t_{max}$ according to eq. (10). This is eq. 5.175 in MKL. To compare with our results we set $x_0 = 1.0, a = 1.0$ and $b = 0.50$.

$$p(x,t) = \frac{1}{\sqrt{2\pi b^2 t}} \; \frac{1}{x} \; e^{-\frac{(\ln(\frac{x}{x_0}) - (a - \frac{b^2}{2})t)^2}{2b^2 t}} \tag{10}$$

This function is shown in figure 2 as a solid black line. A prominent feature of the probability density is that at about $x(t_{max}) \approx 25$ m there is a maximum. One can say there is a most probable route for our walker. As time evolves this maximum will get less prominent.

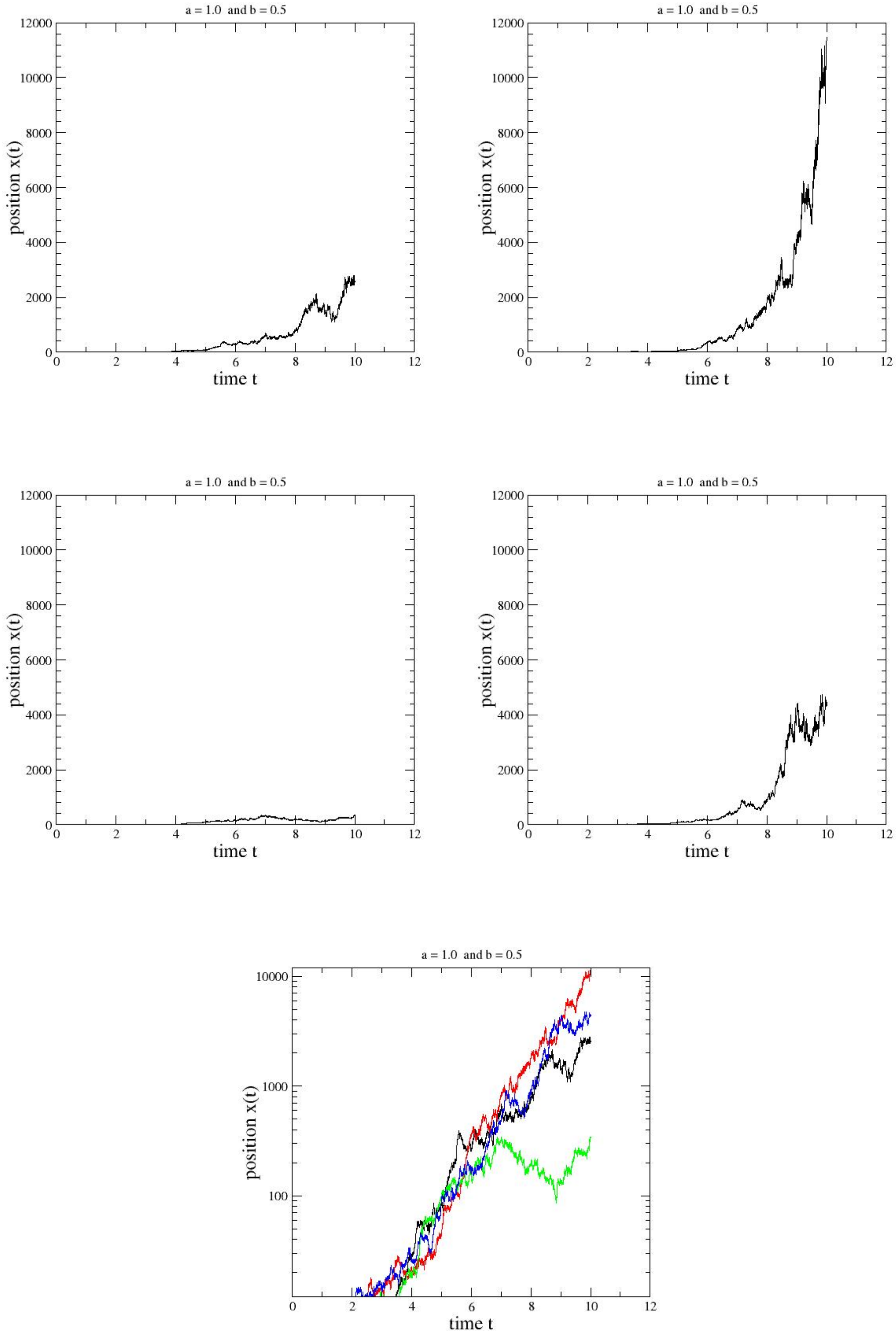

Figure 1: Four different realisations of a random walker. All walkers have starting position $x_0 = 1.0$ and run for $t_{max} = 10$ s. The parameters are $a = 1.0$ s^{-1} and $b = 0.5$ s$^{-\frac{1}{2}}$. The fifth figure shows the same data as the four above. It is only drawn with a logarithmic scale for the positions. The follow more or less straight lines in this scale indicating the exponential growth.

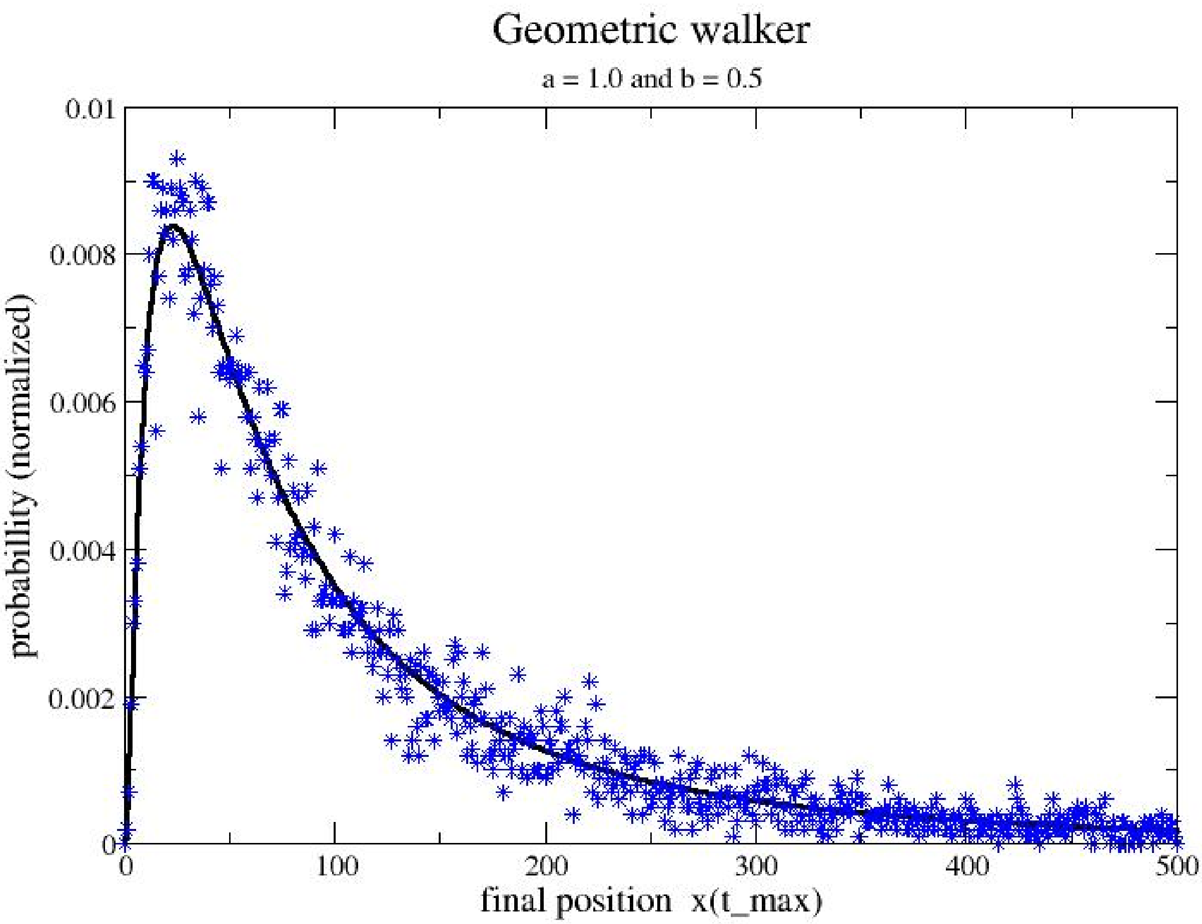

Figure 2: Histogram of the probability distribution of the final positions recorded for 10000 different walkers. The results for the walkers are marked with blue '*'. The solid line is the theoretical probability distribution function eq.(10).

Results for $a = 0.5\ \mathrm{s}^{-1}$ and $b = 1.0\ \mathrm{s}^{-\frac{1}{2}}$ (with no drift).

The reason for this particular choice of a and b is that eq. 10 contains a term $a - \frac{b^2}{2}$ and this particular choice makes this $a - \frac{b^2}{2} = 0$. The previous choice made $a - \frac{b^2}{2} > 0$ and hence a slow drift away to increasing x this was also easily seen in figure 1. The other possibility is $a - \frac{b^2}{2} < 0$ and for this case the distribution will not drift away from the vicinity of $x = 0$ with always $x(t) > 0$. This is also seen in eq. 10 where a negative x would be problematic for $\ln(x/x_0)$.

In figure 3 four different realisations of a geometric random walk are shown.

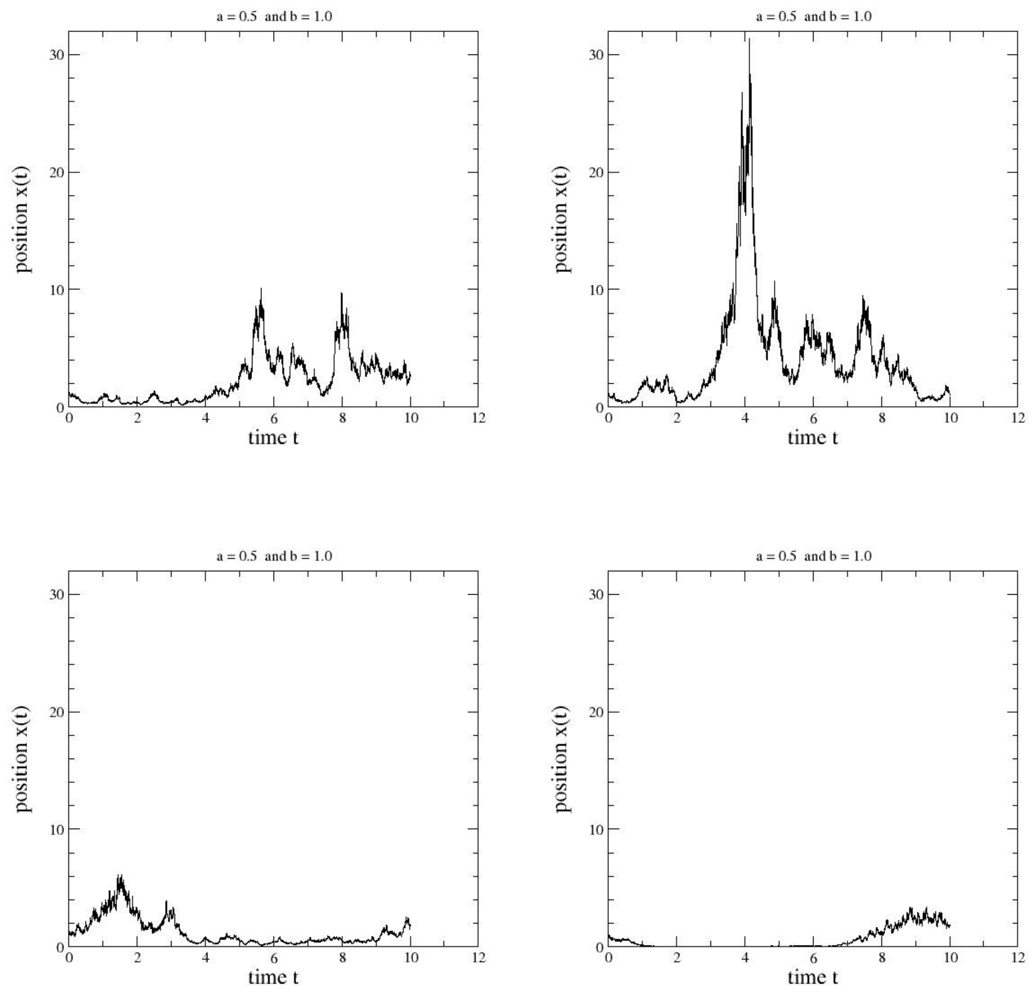

Figure 3: Four different realisations of a random walker. All walkers have starting position $x_0 = 1.0$ and run for $t_{max} = 10$ s. The parameters are $a = 0.5$ and $b = 1.0$

The data shown are for the time evolution of the walkers position $x(t)$. The total time of each run is $t_{max} = 10$ s and as $\Delta t = 0.001$ each run consists of 10000 steps.

If we compare the previous figure 1 to the last time series figure 3 we can clearly see a difference in appearance. The previous time series all show a tendency to increasing x whereas the last ones do not a feature very clear if we use a logarithmic axis. In agreement with eq. 10.

To analyse this histogram in figure 4 consisting of the final positions of 10000 geometric random walkers. The sum of all histogram boxes is accordingly 10000 and to change this

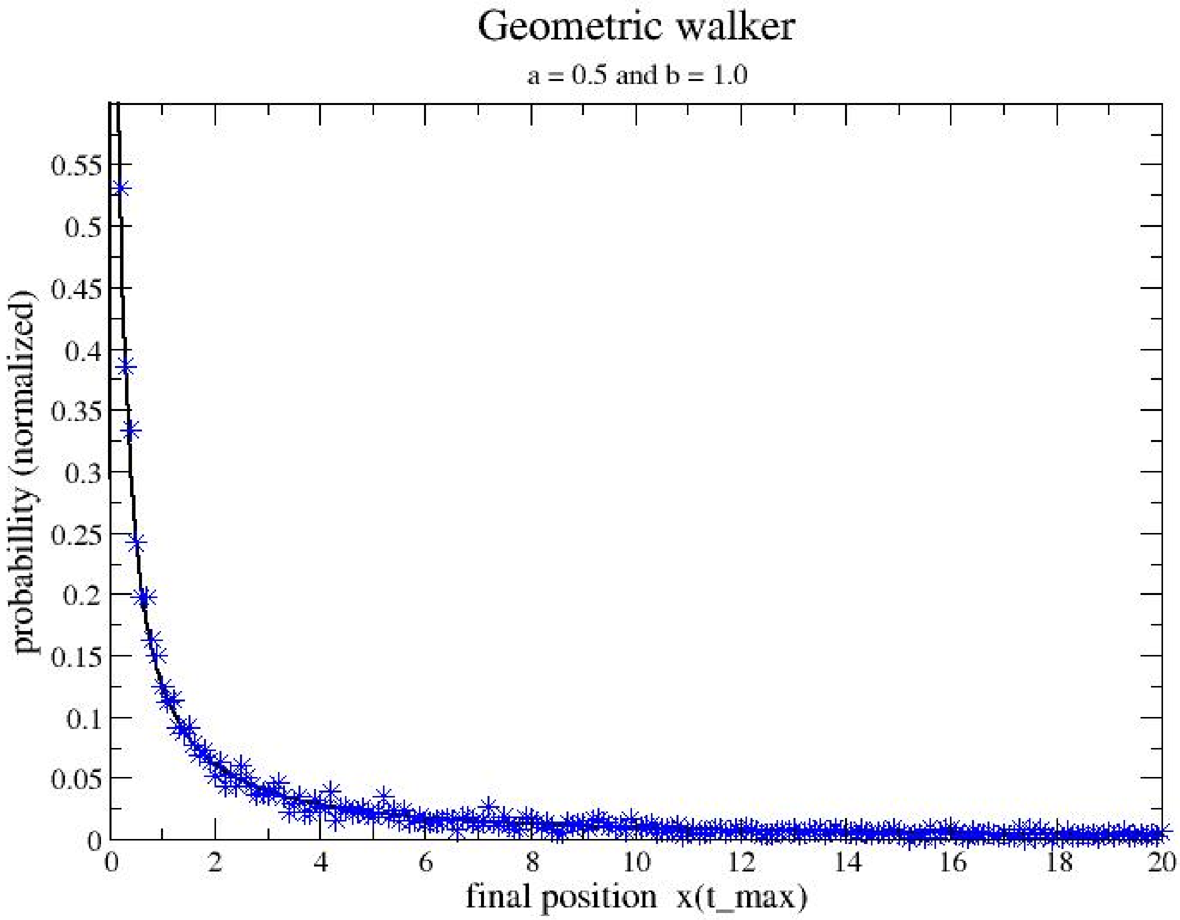

Figure 4: Histogram of the probability distribution of the final positions at $t = 10$ s recorded for 10000 different walkers. The results for the walkers are marked with blue '*'. All walkers started at $x_0 = 1.0$ and the parameters are $a = 0.5$ and $b = 1.0$ The solid line is the theoretical probability distribution function eq.(10).

into a probability function the values in the histogram have to be divided by 10000. (As a technical note just from the figure 4 one can see that the histogram is stretched by a factor 10 so in reality one should not divide by 10000 but with 1000 as the with a box is 0.1 m.) This will transform the histogram into a probability density. In figure 2 the results are shown as blue stars '*'. The solid black curve is the probability density function eq. 10 for this case.

The histogram in figure 4 is actually truncated as some of the geometric random walkers will reach quite far out in 10 seconds In total 31 random walkers managed to go beyond $x(t_{max}) = 4000$ m. As there is no drift in this configuration there is no maximum in the distribution either. as was the case in figure 2 where there is a maximum at about $x(t_{max}) \approx 25$ m.

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

clear
format long
clf ;

  t_t(10001) = 0.0d0 ;
  x_t(10001) = 0.0d0 ;
  delta_t = 0.0010 ;
  t_max = 10.0 ;
  i_max = t_max / delta_t ;
  delta_W_t = 0.0d0 ;
  delta_x = 0.0d0 ;
  w_square = 0.0d0 ;
  xt = 1.0d0 ;
%
%     a   b
%    _10_05_
%  a = 1.0d0 ;
%  b = 0.5d0 ;
%
%     a   b
%    _05_10_
  a = 0.5d0 ;
  b = 1.0d0 ;

  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 = a*xt*delta_t  + b*xt*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_05_10_x.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  = 40000 ;
histogram_x(40001) = 0;
histogram_y(40001) = 0;
for k = 1: i_histo_max + 1
  histogram_x(k) = 0.10d0*(k - 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 = 10000 ;
for k=1: kmax

  t_t(10001) = 0.0d0 ;
  X_t(10001) = 0.0d0 ;
  delta_t = 0.0010 ;
  t_max = 10.0 ;
  i_max = t_max / delta_t ;
  delta_W_t = 0.0d0 ;
  delta_x = 0.0d0 ;
  w_square = 0.0d0 ;
  xt = 1.0d0 ;
%
%     a   b
%    _10_05_
%  a = 1.0d0 ;
%  b = 0.5d0 ;
%
%     a   b
%    _05_10_
  a = 0.5d0 ;
  b = 1.0d0 ;


% 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 = a*xt*delta_t  + b*xt*delta_W_t ;
      xt = xt + delta_x ;
```

```
      X_t(i) = xt ;
      if  (xt  <  0.0d0 )
        echo " NEGATIVE  VALUES  "
        stop
      end

  end

  w_square = w_square/(double(i_max + 1))
  k
  X_t(10001)
  i_histo = round(10.0d0*X_t(10001))

  i_histo = i_histo +1 ;

  if  (i_histo  >  i_histo_max )
     i_histo  =  i_histo_max + 1
  end

  if ( (i_histo  > 0 ) && (i_histo  <=  i_histo_max + 1 ) )
     histogram_y(i_histo)  = histogram_y(i_histo)  + 1;
  end

   if ( (i_histo  > 0 ) && (i_histo  <=  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_05_10_10000_x.dat','w');
for i1=1: i_histo_max + 1
  fprintf(fileID,formatSpec,histogram_x(i1),histogram_y(i1))
end
fclose(fileID);

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