Solutions to Homework 1 - Introduction

1 Lower bounded random walk.

A Simulation of a process realization. The function is already provided in the ECE440 class website; follow http://www.ece.rochester.edu/~gmateosb/ECE440/Homework/ssas_gambling.m
Note that functions must be saved in an m-file with the same name of the function.

```
% This function simulates a game in which the player begins
% with an initial wealth ($w_0), and bets amount $b every
% round (at each instant t). In each round, the player wins amount $b with
% probability p and loses $b with probability (1-p).
% The max_T variable indcates the maximum number of rounds
% which can be played (used to avoid infinite loops).
%
% The function returns a vector w with the history of the bets, a scalar
% t with the number of bets placed, and a scalar h to indicate whether
% or not the player went bankrupt. If the player didn't go bankrupt,
% t = max_t.
function [w, t, h] = ssas_gambling(w_0, b, p, max_t)

t = 1; % Time index.
h = 0; %Boolean to indicate bankruptcy.
w(t) = w_0;
while ( (w(t)>0) && (t<max_t) ) %perform loop until max_T is reached
                                %or player goes bankrupt.
   x = random('bino',1,p);      %draw random number determining
                                %bet's outcome.
                                %produces a 1 w.p. p and a 0 w.p. 1-p.
   if (x==1)
       w(t+1) = w(t) + b;       % If win = 1 then wealth increases by b.
   else
       w(t+1) = w(t) - b;       % If win = 0 wealth decreases by b.
   end
   t = t + 1;                   % Increase time and repeat.
end
if t < max_t
        h = 1;  %If the actual number of rounds is less than the max,
                %h=1 because the player went bankrupt.
end
end
```

Now we just call the function with the requested parameters and plot the results.

```
close all % Close figures.
clear     % Delete variables.
clc       % Clearing the screen.
% Simulation parameters
w_0 = 20;   % Initial wealth.
b = 1;      % Amount bet at every time.
max_t=1000; % Simulation will run for at most max_t iterations.
p_vector = [0.25 0.50 0.75];   % Vector of probabilities of winning.
```

```
figure; hold on; grid on; xlabel('bet index');
ylabel ('wealth (in $)'); axis([0,max_t,0,400])% create new figure.

for p=p_vector % p takes the values in p_vector one at each run of the for loop.
    [w, t, h] = ssas_gambling(w_0, b, p, max_t); % Run experiment;
                                                 % see file "ssas_gambling.m".
    plot(w); % plot the bet's history.
end
```

The resulting plot of the three process realizations is shown in Fig. 1.

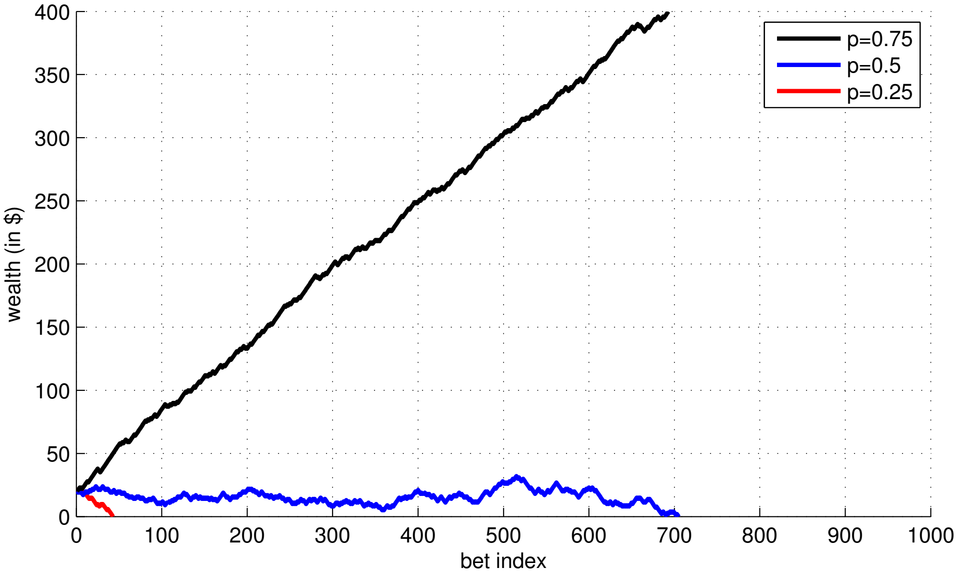

Fig. 1. Simulated processes for $w_0 = 20$ and $p = 0.25$, $p = 0.5$ and $p = 0.75$ (Part A)

B Probability of reaching home. When determining the number N of experiments to run, it is important to keep in mind that more experiments lead to greater accuracy, and the only drawback to increasing experiments is an increase in runtime. This function will begin with $N = 100$ experiments, but will rerun with $N = 1000$ experiments if too few players go bankrupt (which could be an indication of a not large enough sample size). The required function follows:

```
% Function estimating the probability of reaching home
% as a function of the winning probability 'p' and initial wealth 'w_0'
% The function makes sure there are least 10 home-reaching realizations or
% otherwise the simulation is performed 1000 times. This roughly prevents
% innacurate estimations based on too few samples.

function pr_r_h = pr_reaching_home(w_0, p)
b = 1;      % Amount bet at every time.
T=100;      % Simulation will run for at most max_t iterations.
N=100;      % Initial number of experiments.
i = 0       % Index.
nr_reaching_home=0; % Counter initialization.
while (nr_reaching_home<10) && (N<=1000)
    nr_reaching_home=0; % reset after checking condition
    for i=1:N
        [w, t, h] = ssas_gambling(w_0, b, p, T);
        % see file "ssas_gambling.m"
        if h ==1
           nr_reaching_home = nr_reaching_home + 1;
        end
    end
    pr_r_h = nr_reaching_home/N;  %Calculates the estimate for the
                                  %probability of the player reaching
                                  %home based on total runs of the
                                  %experiment (N) and number of times
                                  %the player reached home
    N=10*N; %updating N
end
```

When run in Matlab’s Command Window we find the estimated probability to be:

```
EDU>> pr_reaching_home(10, 0.55)
ans =
    0.0890
```

Thus, for this experiment $\hat{B}_N(0.55, 10) = 0.0890$. The player is very unlikely to go bankrupt.

An extremely faster and more concise version (but with a little Matlab twist, thus more involved) is provided below. Note that the following function does not call the *ssas-gambling* function. This is a smart way of calculating the same probability $\hat{B}_N(p, w_0)$. If you are eager to know how it works and if after consulting with Matlab help, you still have problem with it, bring it up during office hours (Hint: recall the digression on ”Gambling as LTI system with stochastic input” we discussed in class).

```
function pr_r_h = pr_reaching_home_fast(w_0, p)
    b = 1;      % Amount bet at every time.
    T=100; % Simulation will run for at most max_t iterations.
    N=1000; % number of simulation runs
    steps=b*(2*random(’bino’,1,p,[T,N])-1); % wins/losses
    w=w_0+cumsum(steps); % wealth
    pr_r_h=sum(sum(w<=0)>0)/N; % fraction of reaching home
end
```

C Probability of reaching home as a function of initial wealth. To determine $\hat{B}_N(p, w_0)$ as a function of wealth, we will vary w_0 between 1 and 20, set $p = 0.55$, and use the function `pr_reaching_home(w_0,p)` to obtain and then plot our results.

```
% Plot the probability of reaching home as a function of initial wealth
clear all; close all; clc;
p=0.55;
pr_rh=zeros(1,20); % initialization of the probabilities of reaching home

for i=1:20
    w_0=i; % initial wealth
    pr_rh(1,i) = pr_reaching_home(w_0, p);
end

figure; hold on; grid on; xlabel(’initial wealth’);
ylabel (’probability of reaching home (bankruptcy)’);
axis([0,20,0,1])% create new figure
plot(1:20,pr_rh)
```

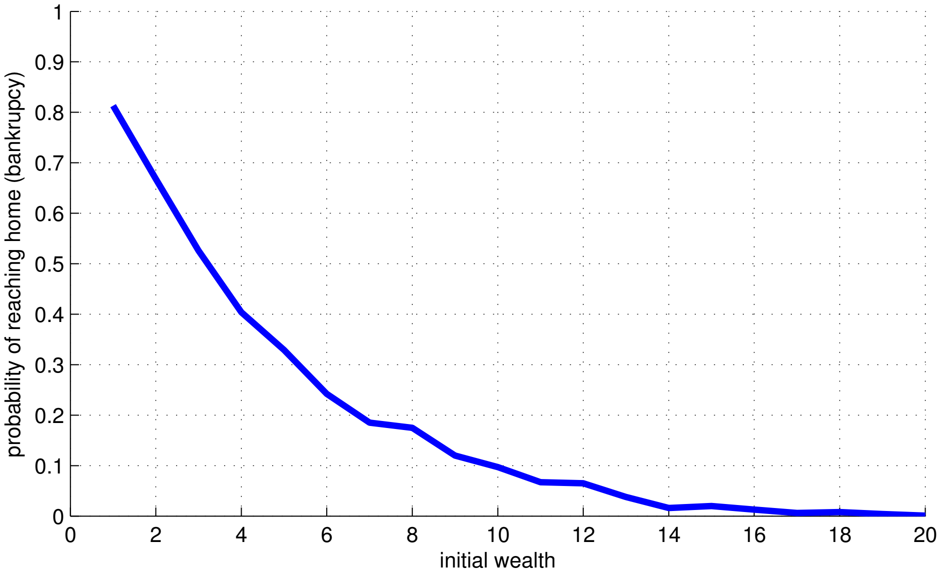

Fig. 2. Probability of reaching home as a function of initial wealth for $p = 0.55$ (Part C).

The result is shown in Fig. 2.

D Probability of reaching home as a function of p. To determine $\hat{B}_N(p, w_0)$ as a function of p, we will vary p between 0.3 and 0.7 using increments of 0.02, set $w_0 = 10$, and use the function `pr_reaching_home(w_0,p)` to obtain and then plot our results.

.

```
% plotting the probability of reaching home
% as a function of winning probability
clear all; close all; clc;
w_0=10; % initial wealth
p_vector=0.3:0.02:0.7; % vector of winning probabilities
pr_rh=zeros(1,length(p_vector)); % initialization of home-reaching
i=0; %index                      % probabilities
for p=p_vector
    i=i+1; % incrementing the index
    pr_rh(1,i) = pr_reaching_home(w_0, p);
end

figure; hold on; grid on; xlabel(’winning probability’);
ylabel (’probability of reaching home (bankruptcy)’);
axis([0,1,0,1])% create new figure

plot(p_vector,pr_rh)
```

The results with two different level of accuracies are provided in Figs. 3 and 4

We distinctly observe that for $p < 1/2$, the probability of bankruptcy is tending to 1, i.e., almost surely the patron loses all of its initial money. For $p > 1/2$ the probability of bankruptcy continually decreases to zero, i.e., with high probability, the patron never looses all of its money, which is to say, as p increases, the gambler can (with increasing probability,) win large amounts of money provided s/he stays in the game long enough. When p is very close to $1/2$, the probability of losing all money is extremely volatile. Gambling lesson: This game with $p < 0.5$ is extremely bad (the bankruptcy is guaranteed).

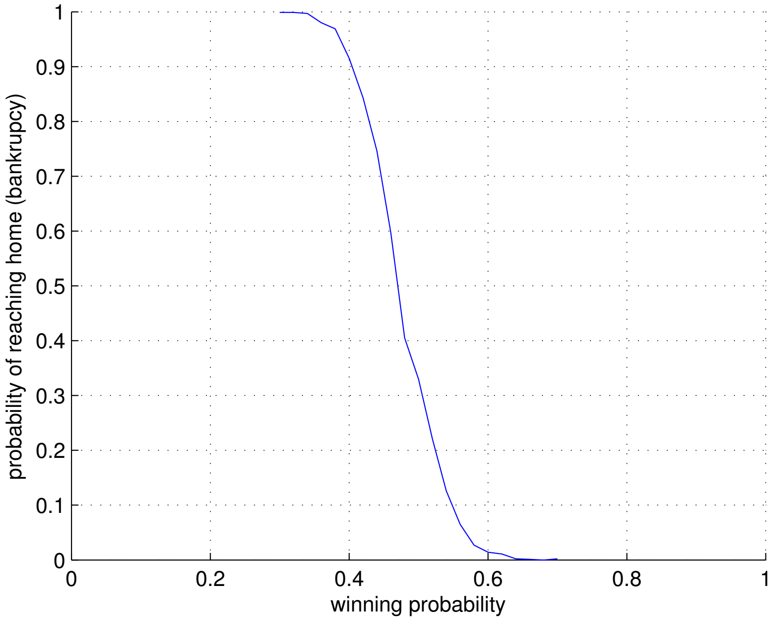

Fig. 3. Probability of reaching home (bankruptcy) as a function of winning probabilities for $w_0 = 10$. Here $T = 100$. (Part D)

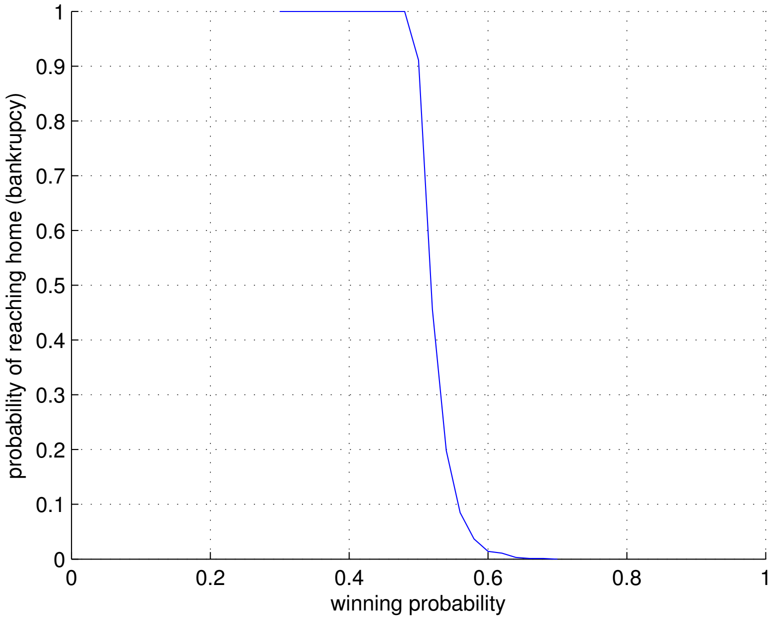

Fig. 4. Probability of reaching home (bankruptcy) as a function of winning probabilities for $w_0 = 10$. Here $T = 10000$. (Part D)

E Time to reach home. For fixed $p = 0.4$ and repeating the experiment $N = 1000$ times, we estimate the probability distribution of T_0 by constructing the histogram of observed values $\{t_0^n\}_{n=1}^N$ as discussed in class. The average value is estimated through sample averaging.

```
clear all; close all; clc;
```

```
p=0.4;
w_0=10;
b=1;
t_max=1000;
N=1000; % number of experiments
T_0=zeros(1,N); % initialization of the vector of home-touching times
for i=1:N
        [w, t, h] = ssas_gambling(w_0, b, p, t_max);
        % see file "ssas_gambling.m"
        T_0(1,i)=t;
end
figure; hold on; grid on; xlabel('T_0');
ylabel ('pdf');
[n,xout] = hist(T_0,25);
bar(xout,n/N)

average_T_0=sum(T_0)/N;
disp(['average T_0 = ',num2str(average_T_0)])
```

The outputs of the above Matlab script are as follows:
For $w_0 = 10$:

```
average T_0 = 50.874
```

For $w_0 = 20$:

```
average T_0 = 101.866
```

As expected, the average time to reach home increases with the initial wealth since it takes more time to deplete a larger pot. The obtained histogram plots are shown in Figs. 5 and 6.

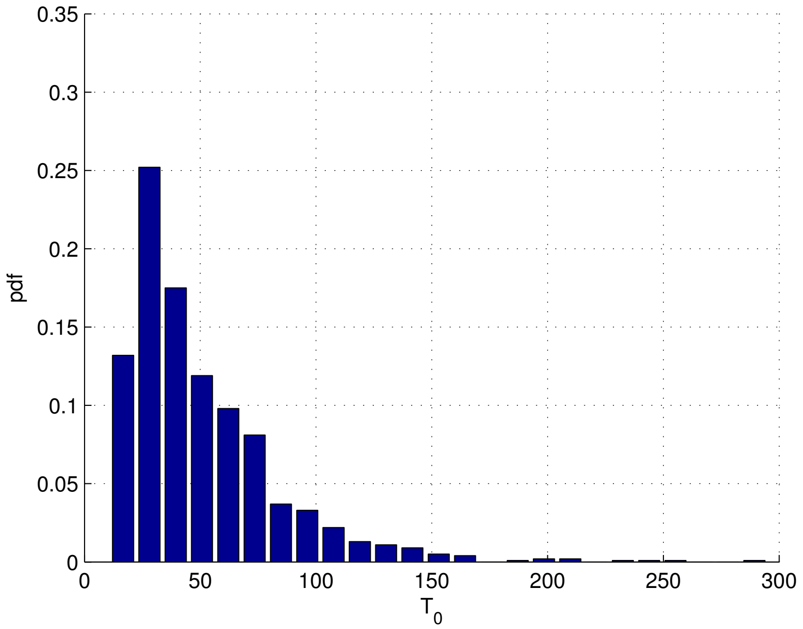

Fig. 5. Probability distribution of the home-reaching time (bankruptcy time) for $p = 0.4$ and $w_0 = 10$. Here $T = 1000$. (Part E).

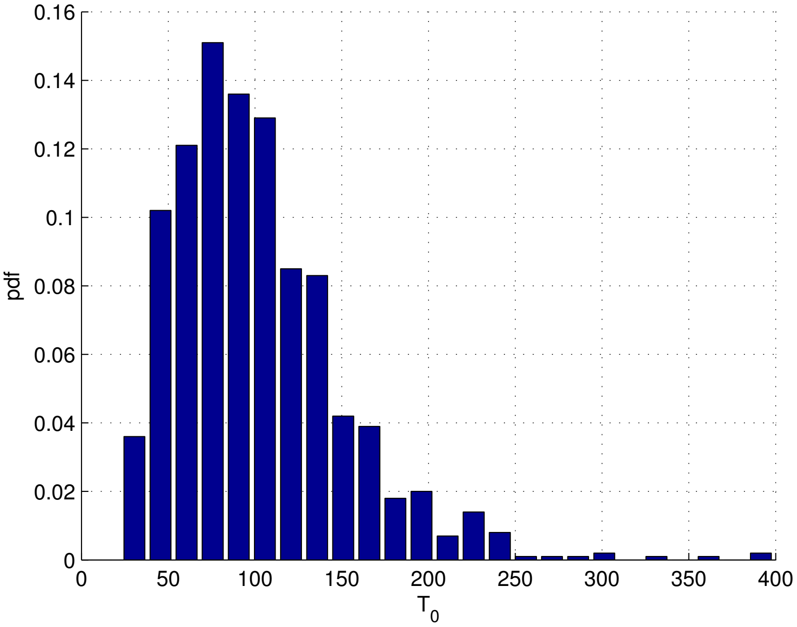

Fig. 6. Probability distribution of the home-reaching time (bankruptcy time) for $p = 0.4$ and $w_0 = 20$. Here $T = 1000$. (Part E).