6.1.2 Simulazioni con generatore di numeri casuali

```
% Simulazione dell'andamento di un'azione soggetta al moto geometrico browniano
% mediante il generatore di numeri casuali.

function [f,s]=mgb(T,N,sigma,mu,nsim,iniz)

% iniz = prezzo iniziale

x=linspace(0,T,N+1);
dt=T/N;
for j=1:nsim+1
    s(1)=iniz;
for i=2:N+1
        s(i)=s(i-1)+s(i-1)*mu*dt+sigma*s(i-1)*randn*(dt^0.5);
end
    f(j)=s(end);
end
plot(x,s)
media=mean(f)

% Test:
% mgb(20,100,.2,.1,1000,30)
```

6.1.3 Generalizzazione del moto geometrico browniano

```
% Generalizzazione del moto geometrico browniano e caso di varianza
% infinita.
%
function [s,v,m]=mgb_var_infinita(So,N,nsim,T,a,b,c,al,be)
dt=T/N;
u=1+a*dt^al;
d=1+b*dt^al;
pr=0.5*(1+c*dt^be);
for j=1:nsim
    s(1,j)=So;
    for i=1:N
        if rand<pr
            s(i+1,j)=u*s(i,j);
        else
            s(i+1,j)=d*s(i,j);
        end
    end
end
v=var(s(N+1,:))
m=mean(s(N+1,:))
hist(s(N+1,:));
%
% Test:
%[s,v]=mgb_var_infinita(80,100,100,1,.4,.6,1,-1,1.5)
%[s,v,m]=mgb_var_infinita(80,100,100,1,0.2,5,1,-1,1)
%[s,v,m]=mgb_var_infinita(50,100,50,1,0.2,5,1,-1,1)

% Moto geometrico browniano; simulazioni successive implementate con
% generatore di numeri casuali e stima numerica di media e varianza.
%
function [t,S,M,V,m,v]=MGB_media(T,N,mu,si,So,nsim)
dt=T/N;
t=linspace(0,T,N+1);
for j=1:nsim
    S(1,j)=So;
```

```
    for i=1:N
        S(i+1,j)=S(i,j)*[1+mu*dt+si*randn*sqrt(dt)];
    end
end
M_num=mean(S(N+1,:))
V_num=var(S(N+1,:))
m_ex=So*exp((mu*T)+(si^2)*T)
v_ex=(So^2)*exp((2*mu*T)+T*(si^2))*(exp(T*(si^2))-1)
plot(t,S')
%[t,S]=MGB_media(.1,2300,.3,.4,80,100)
```

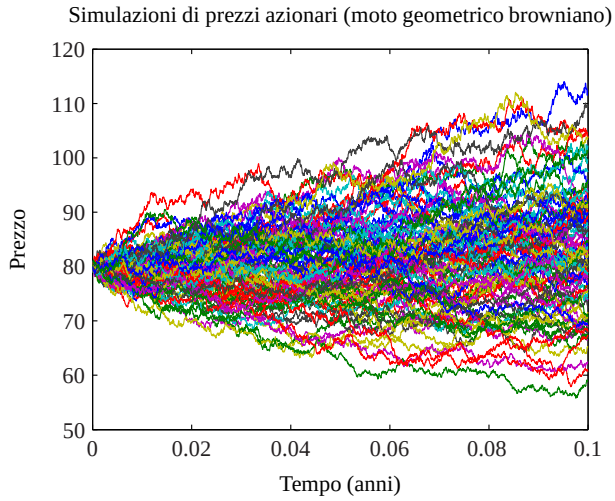

Figura 6.1: Simulazioni successive di cammini azionari con moto geometrico browniano.

6.2 Formula di Black e Scholes

```
% Formula di Black e Scholes per la determinazione del prezzo di un'opzione call.

function c=bsm_call(T,x,r,sigma,s_0)

% T = tempo di esercizio
% x = strike price
% r = tasso di interesse fisso
% sigma = volatilità
% s_0 = prezzo iniziale del sottostante

d1=(log(s_0./x)+(r+(sigma^2/2))*T)./(sigma*sqrt(T));
d2=(log(s_0./x)+(r-(sigma^2/2))*T)./(sigma*sqrt(T));
c=s_0*normcdf(d1)-x.*exp(-r*T).*normcdf(d2);

% Alternativamente, al fine di calcolare la distribuzione cumulativa,
% si può definire la funzione ausiliaria che segue e richiamarla al posto
% di normcdf.

%function N=cumu(x)
```

```
%N=0.5*erfc(-x/sqrt(2));

% c=s_0*cumu(d1)-x.*exp(-r*T).*cumu(d2);

% Test:
%x=linspace(80,120,40);
%a=bsm_call(.5,x,.05,.2,100);
%plot(x,a)
```

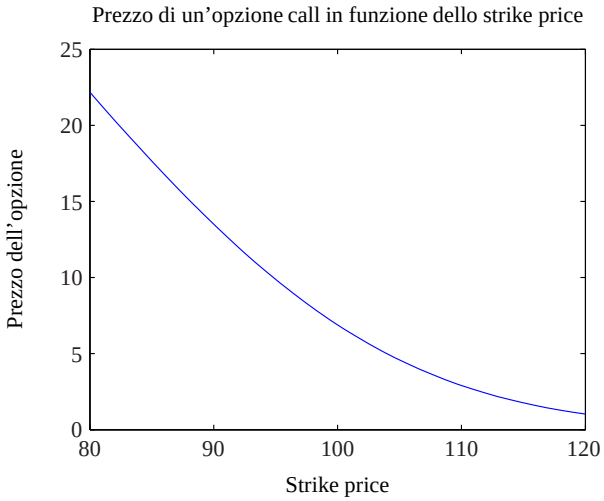

Figura 6.2: Grafico del prezzo di un'opzione *call*.

```
% Formula di Black e Scholes per la determinazione del prezzo di un'opzione put.

function c=bsm_put(T,x,r,sigma,s_0)

% T = tempo di esercizio
% x = strike price
% r = tasso di interesse fisso
% sigma = volatilità
% s_0 = prezzo iniziale del sottostante

d1=(log(s_0./x)+(r+(sigma^2/2))*T)./(sigma*sqrt(T));
d2=(log(s_0./x)+(r-(sigma^2/2))*T)./(sigma*sqrt(T));
c=x.*exp(-r*T).*normcdf(-d2)-s_0*normcdf(-d1);

% Alternativamente, al fine di calcolare la distribuzione cumulativa,
% si può definire la funzione ausiliaria che segue e richiamarla al posto
% di normcdf.

%function N=cumu(x)

%N=0.5*erfc(-x/sqrt(2));

% c=x.*exp(-r*T).*cumu(-d2)-s_0*cumu(-d1);

% Test:
%x=linspace(80,120,40);
%a=bsm_put(.5,x,.05,.2,100);
%plot(x,a)
```