QFGB 8933 Homework #4

Jiayin Hu

Feb. 26 2019

1 Transition Density for Diffusions.

The function generating k-order Hermite polynomials.

```
function [temp]=Hermite(k)
% Hermite(1)=1
syms z

H{1}=sym('1');

for n=2:k
    H{n}=simplify(z*H{n-1}-diff(H{n-1},z));
end

temp=H{k};
```

The function calculate transition density function.

```
function pX = TransitionDensity(muX,sigmaX, K, J)
%% Transformation X(t) to Y(t)
syms a  b  c
syms xs ys zs
syms x  y  z
syms h t s

fX2Y=int(1/sigmaX,x);
fY2X=subs((finverse(fX2Y)), x,y);

%%  Drift and Diffusion for Y(t)
muY_temp=muX/sigmaX-sym('1')/sym('2')*diff(sigmaX,x,1);
muY=subs(muY_temp, x, fY2X);
muY=simplify(muY);

sigmaY=sym('1');

%% Transformation Y(t) to Z(t)
fY2Z=h^(-1/2)*(y-ys);
fZ2Y=h^(1/2)*z+ys;
```

```

%% Generating Beta
syms Htemp Expectation Beta_t
clear Beta_t Htemp Expectation
for k=1:K
     HTemp=subs(Hermite(k), z, fY2Z);
     Expectation=HTemp;

     for j=1:J
       HTemp=muY*diff(HTemp,y,1)+sym('1')/sym('2')*diff(HTemp, y, 2);
       Expectation=Expectation + h^j/factorial(j)*HTemp;
     end
     Beta_t{k}= sym('1')/factorial(k-1) * subs(Expectation, y, ys);
end

%%Geberating pZ With Loop
pZ=sym('0');

for m=1:K
  pZ=pZ+Beta_t{m}*Hermite(m);
end
findsym(pZ)

%% Generating pY pX
pZ=exp(-z^2/2)/sqrt(2*pi)*pZ;
pY=(h^(-1/2))*subs(pZ, z, fY2Z);
pX=(sigmaX^(-1))*subs(pY, y, fX2Y);
pX=subs(pX, ys, subs(fX2Y, x, xs));
pX=simplify(pX);
end
```

Check Vasicek Model.

```
syms a  b  c
syms xs ys zs
syms x  y  z
syms h t s

% Vasicek
muX=a*(b-x)
sigmaX=c
%sigmaX=c*sqrt(x)
K = 3
J = 4
density_v = TransitionDensity(muX, sigmaX, K, J);

%% transition density

g1=subs(density_v, {a,b,c,h,xs}, {1,1,2,1/250,1})

%%Exact Density for Vasicek
gamm=sigmaX*sqrt(1-exp(-2*a*h))
density_ve=(pi*gamm^2/a)^(-1/2)*exp( -(x-b-(xs-b)*exp(-a*h))^2 ...
```

```
    *a/(gamm^2) )
g2=subs(density_ve, {a,b,c,h,xs},{1,1,2,1/250,1})
g2=simplify(g2)
gDiff=g1-g2

% Plot
fig=figure
subplot(2,1,1)
ez1=fplot(g1,[0, 2], 'r-')
hold on
ez2=fplot(g2,[0, 2], 'b:')
legend('Transformation Density','Actual Density')
%% Plot Density Difference
subplot(2,1,2)
fplot(gDiff, [0,2])
legend('Difference')
```

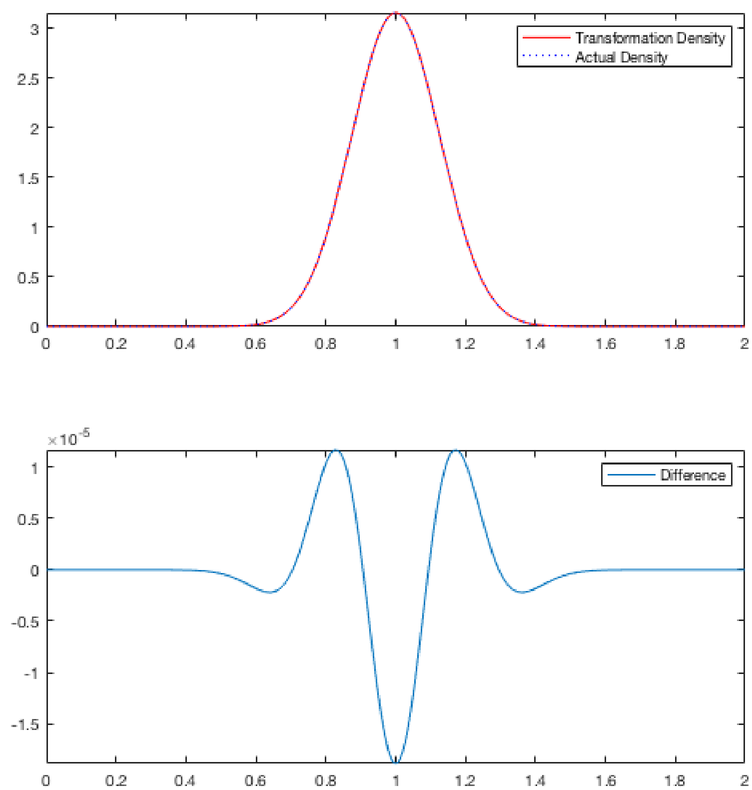

Figure 1: Vasicek Model with K=3, J=4

When change different K and J, the difference changes.

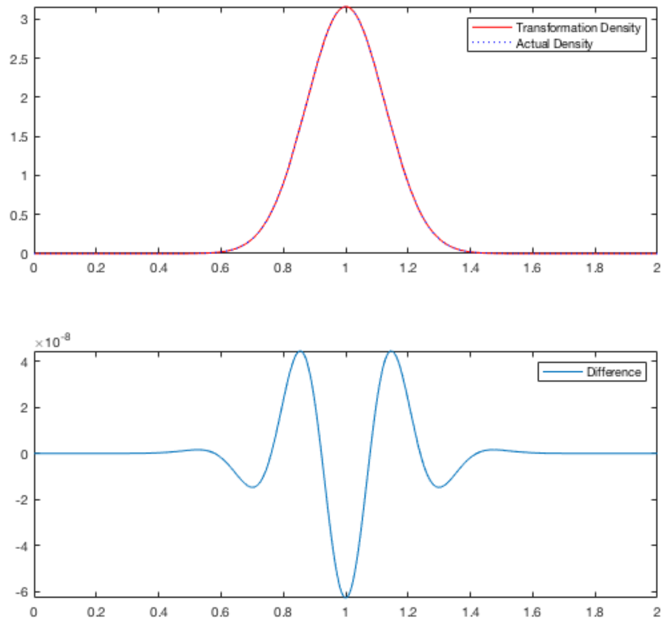

Figure 2: Vasicek Model with K=5, J=6

Using the same method checking Black-Scholes Model.

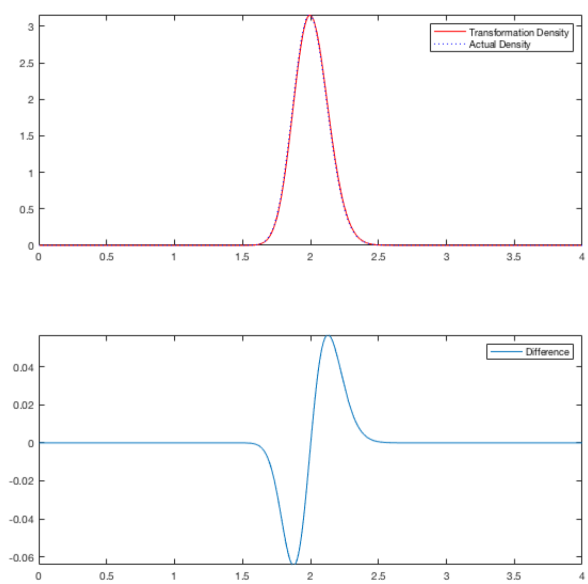

Figure 3: B-S Model with K=3, J=4

2 MLE for SDE and Empirical Applications

We select 1-month Treasury constant maturity as the dataset and choose Vasicek Model to fit the data. First, we plug h = 1/250 into the density functions.

```
% The transition density function of Vasicek Model with K=3 J=4.
density_v

density_v =

-(2251799813685248*exp(-(x - xs)^2/(2*c^2*h))*((((a*h*(a*b^2 - 2*a*b*xs ...
    - c^2 + a*xs^2))/(2*c^2) - (a^2*h^2*(3*a*b^2 - 6*a*b*xs - 2*c^2 + ...
    3*a*xs^2))/(6*c^2) + (a^3*h^3*(7*a*b^2 - 14*a*b*xs - 4*c^2 + ...
    7*a*xs^2))/(24*c^2))*(h*c^2 - x^2 + 2*x*xs - xs^2))/(c^2*h) + (a*(b ...
    - xs)*(x - xs)*(a^3*h^3 - 4*a^2*h^2 + 12*a*h - 24))/(24*c^2) - ...
    1))/(5644425081792261*c*h^(1/2))

f = subs(density_v, {h}, {1/250})

f =

-(2251799813685248*250^(1/2)*exp(-(125*(x - ...
    xs)^2)/c^2)*((250*((a^3*(7*a*b^2 - 14*a*b*xs - 4*c^2 + ...
    7*a*xs^2))/(375000000*c^2) - (a^2*(3*a*b^2 - 6*a*b*xs - 2*c^2 + ...
    3*a*xs^2))/(375000*c^2) + (a*(a*b^2 - 2*a*b*xs - c^2 + ...
    a*xs^2))/(500*c^2))*(c^2/250 - x^2 + 2*x*xs - xs^2))/c^2 + (a*(b - ...
    xs)*(x - xs)*(a^3/15625000 - a^2/15625 + (6*a)/125 - 24))/(24*c^2) - ...
    1))/(5644425081792261*c)

% Actual density function of Vasicek Model
gv=subs(density_ve, {h},{1/250})

gv =

exp((a*(x - b + exp(-a/250)*(b - xs))^2)/(c^2*(exp(-a/125) - ...
    1))))/(-(c^2*pi*(exp(-a/125) - 1))/a)^(1/2)
```

We still use the MLE functions we established before.

```
function [thetam, fval, sigma_hat_m, p_value] = mle(fm, Xm, Ym, theta0m)

LLm = @(theta) sum(log(fm(Xm, Ym, theta)));
[thetam, fval, ¬, ¬, ¬, h] = maximize(LLm, theta0m);

h_inv = inv(h);
sigma_hat_m = sqrt(diag(h_inv));
p_value = PValue(thetam, sigma_hat_m);

end
```

```
function [x, fval_neg, exitflag, output, grad, h] = maximize(fm, theta0)

%   f: the function will be maximazed
fun_neg = @(theta) -fm(theta);
[x, fval, exitflag, output, grad, h] = fminunc(fun_neg, theta0);
fval_neg = -fval;

end
```

```
function [p_value] = PValue(thetam, sigma_hat_m)

[m, n] = size(thetam);
for i = 1:1:n
    p_value(i) = 2 * (1-normcdf(abs(thetam(i)), 0, sigma_hat_m(i)));
end

end
```

Apply to new data.

```
data = rmmissing(RIFLGFCM01_NB);
n = length(data);
Xs_data = data(1:n-1,1);
X_data = data(2:n,1);
plot(data)
```

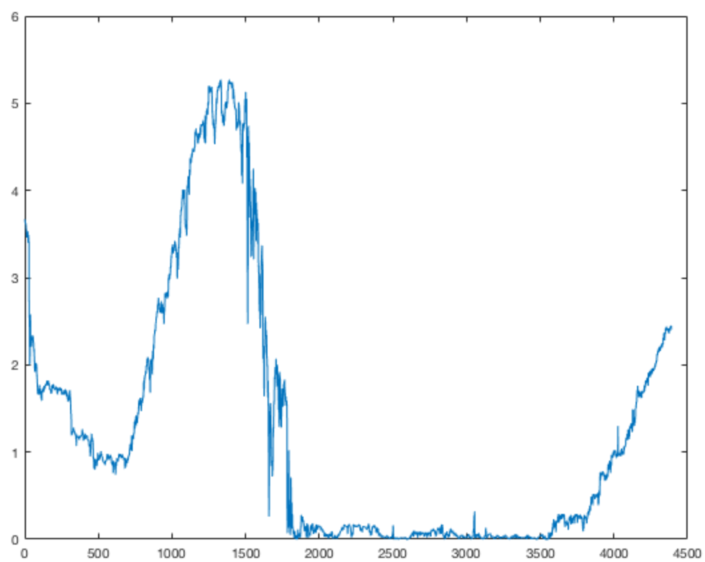

Figure 4:

```
% MLE using actual density function
gv = @(x, xs, theta) exp((theta(1)*(x - theta(2) + ...
    exp(-theta(1)/250)*(theta(2) - ...
    xs)).^2)/(theta(3).^2*(exp(-theta(1)/125) - ...
    1)))/(-(theta(3).^2*pi*(exp(-theta(1)/125) - 1))/theta(1)).^(1/2)
[y, fval, ste, p_value] = mle(gv, X_data, Xs_data, [1, 1, 1])

y =

    0.2863    1.0129    1.0257


fval =

   5.7815e+03


ste =

    0.1623
    0.8660
    0.0110

p_value =

    0.0776    0.2422         0

% MLE using transition density function
f = @(x, xs, theta) -(2251799813685248.*250.^(1/2).*exp(-(125.*(x - ...
    xs).^2)/theta(3).^2).*((250.*((theta(1).^3.*(7.*theta(1).*theta(2).^2 ...
    - 14.*theta(1).*theta(2).*xs - 4.*theta(3).^2 + ...
    7.*theta(1).*xs.^2))/(375000000.*theta(3).^2) - ...
    (theta(1).^2.*(3.*theta(1).*theta(2).^2 - 6.*theta(1).*theta(2).*xs ...
    - 2.*theta(3).^2 + 3.*theta(1).*xs.^2))/(375000.*theta(3).^2) + ...
    (theta(1).*(theta(1).*theta(2).^2 - 2.*theta(1).*theta(2).*xs - ...
    theta(3).^2 + ...
    theta(1).*xs.^2))/(500.*theta(3).^2)).*(theta(3).^2/250 - x.^2 + ...
    2.*x.*xs - xs.^2))/theta(3).^2 + (theta(1).*(theta(2) - xs).*(x - ...
    xs).*(theta(1).^3/15625000 - theta(1).^2/15625 + (6.*theta(1))/125 - ...
    24))/(24.*theta(3).^2) - 1))/(5644425081792261.*theta(3))
[y, fval, ste, p_value] = mle(f, X_data, Xs_data, [1, 1, 1])

y =

    0.2832    1.0315    1.0257


fval =

   5.7816e+03


ste =
```

```

    0.1600
    0.8619
    0.0110

p_value =

    0.0767    0.2314         0
```