5.24 モンテカルロシミュレーション 分散減少法(2)　工事中

制御変量法 Control Variate

金融工学の分野でよく使われる分散減少のテクニックの１つである Control Varite はヨーロピアンオプションとアメリカンオプションの組み合わせで、アメリカンオプションの厳密解（解析解）に相当するものを求めようとするものです。それは、これから説明をする Control Variate のやり方のほんの一面でしかありません。具体的には、２組のよく似た動きをする（相関している）解または estimator が考えられているときに、一方の解には厳密解とシミュレーション解が１対である時、もう一方の解にはシミュレーション解は得ることは可能なのですが、厳密解を求めるには困難であったり、そもそも厳密解が存在しなかったりする時に用いられます。すなわち、よく見られる例では、

厳密解	**シミュレーション解**
ヨーロピアン	ヨーロピアン
？	アメリカン

のような対応関係になっていて、アメリカンオプションの厳密解に相当する部分の計算が直接困難である際に、この Control Variate の考え方を援用するわけです。もちろん、ヨーロピアンとアメリカンの組み合わせでなくとも、よく似た（相関する）解であれば、この関係は援用できるわけで、ヨーロピアンそのものと、ヨーロピアンの中でもそのバミューダンとの組み合わせも考えることができるでしょう。アメリカンの厳密解（解析解）がわかってれば、そのアメリカンの変形である別のなにかのオプションとの組み合わせも考えることができます。そもそも、アジアン（平均）オプションとそもそもの対数正規過程の組み合わせといった前レベルでの組み合わせももちろん可能となります。なぜなら、対数正規過程には、その平均が厳密解としてあらかじめパラメータ関係から既知であるためで、シミュレーションの平均と対をなしているからです。

通常金融工学関係の書物に載っているような記述の方法では、下記のように、

$$F_A = F_A^* + (F_B - F_B^*)$$

の関係において、Ａの方は例えばアメリカンオプションの解、Ｂの方は例えばヨーロピアンオプションの解である時、星印はシミュレーションでの解を示しています。そうであるときに、アメリカンオプションの厳密解に相当する解 F_A を求めようというものです。そうすることによって、分散がシミュレーションの解よりも減少することが確認されます。ただし、これは統計学や工学における Control Variate の方法の簡略形にすぎません。

厳密に表記するならば、Ｙを通常のシミュレーションでの estimator とするときに、その Control Variate による estimator は、

$$\hat{\theta}_{cv} = Y + c^*(Z - E(Z)) \qquad where \qquad c^* = -\frac{Cov(Y,Z)}{Var(Z)}$$

のような関係にあるとき、これを Control Variate 法と呼びます。$F_A = F_A^* + (F_B - F_B^*)$ の関係は、だいたいのところ c*＝－１としたときに Z－E(Z)の全体にマイナスをつけて逆転させた形であるＹ＋(E(Z)－Ｚ)を求めていることになります。Ｅ(Z)はヨーロピアンの解析解に相当して、Ｚはそのシミュレーション解。そして、Ｙはアメリカンのわかっている方のシミュレーション解に相当します。括弧の中で厳密な表記と位置関係が逆転していることに特に注意してください。なお、通常の近似解計算では c*を－1 と仮定してしまう、または別の考え方をすれば、ＹとＺが正に相関していれば通常 c*は絶対値で１以下のマイナスの値なので、近似値を求めるにには十分であるとすることもできます。ここで、その c*の値はどうやって計算されているかというと、Ｙ＋ｃ*（Ｚ－E(Z)）の分散計算をしてやります。今、E(Z)は定数（厳密解の計算の際も一定値）ですから、その分散も他の変数との共分散もともにゼロとなり、計算からは削除できます。Ｙ＋ｃ*Ｚの分散は、分散計算の初歩の関係より、定数はくくり出されて２乗の値になるので、

$$Var(\hat{\theta}_{cv}) = Var(Y) + 2cCov(Y,Z) + c^2 Var(Z)$$

となります。この分散の値を最小にするｃの値は、ｃに関しての１階の微分をゼロと置くことによって簡単に計算されます（２階の条件も自動的に満たされることが簡単にわかります)。すなわち、

$$c^* = -\frac{Cov(Y,Z)}{Var(Z)}$$

となります。そのうちでも Cov(Y,Z)と Var(Z)が等しいとき、この値は－１となります。これをもとの分散の式に代入しなおしてやると、

$$Var(\hat{\theta}_{cv}) = Var(Y) + 2\left(-\frac{Cov(Y,Z)}{Var(Z)}\right)Cov(Y,Z) + \left(-\frac{Cov(Y,Z)}{Var(Z)}\right)^2 Var(Z)$$

$$= Var(Y) - \frac{[Cov(Y,Z)]^2}{Var(Z)} < Var(Z) \qquad if \qquad Cov(Y,Z) \neq 0$$

となり、ＹとＺが全く相関しない場合を除きすべてのケースについて不等号は成り立ち、Control Variate によって分散減少が実現されます。

対数正規分布の平均を求める例

　ここでは、平均μ＝５標準偏差σ＝0.20 である正規分布にしたがう乱数を指数変換することによって、対数正規分布にしたがう乱数を作成します。ここでは仮に、元の正規分布の平均と分散（または標準偏差）は既知なのですが、対応する対数正規分布における平均は不明であるとします。もちろん、exp(５)がその分布の平均ではないことは、変換後の分布が正規分布でないことからも明らかです。そこで、今この対数正規分布における平均を Control Variate によって求めようというものです。また、指数変換されて対数正規分布にしたがうようになった乱数の平均の計算をしてやっても、この平均は Control Variate で求められた平均の分散よりも格段に小さくなることを示します。

下のプログラムでは、平均μ＝５標準偏差σ＝0.20 である正規分布にしたがう乱数をＺとし、その exponential をとった値をＹとします。Ｙは exp(Ｚ)なのですから、明らかに、ＺとＹは何らかに相関しているはずです。何らかの相関していれば、c*を求めることによって、推定値の分散が減少します。すなわち、一通りのシミュレーションを行なった結果の推定値も、より真の値に近い可能性があるということです。ここでは、求める推定値は対応する対数正規分布の平均値としています。

プログラム

```
new; cls;
mu=5;
sig=0.20;
p=10000;
n=10000;


times=1000;
m1=zeros(times,1);
m2=zeros(times,1);
i=1;
do while i<=times;
    m1[i]=LOGNORMcv(mu,sig,p,n);
    m2[i]=meanc(exp(mu+sig*rndn(n,1)));
    i=i+1;
endo;
print "THETAcv=" LOGNORMcv(mu,sig,p,n);
print/rd "Var of THETAcv=" vcx(m1);
print/rd "Var of E(Y)     =" vcx(m2);
```

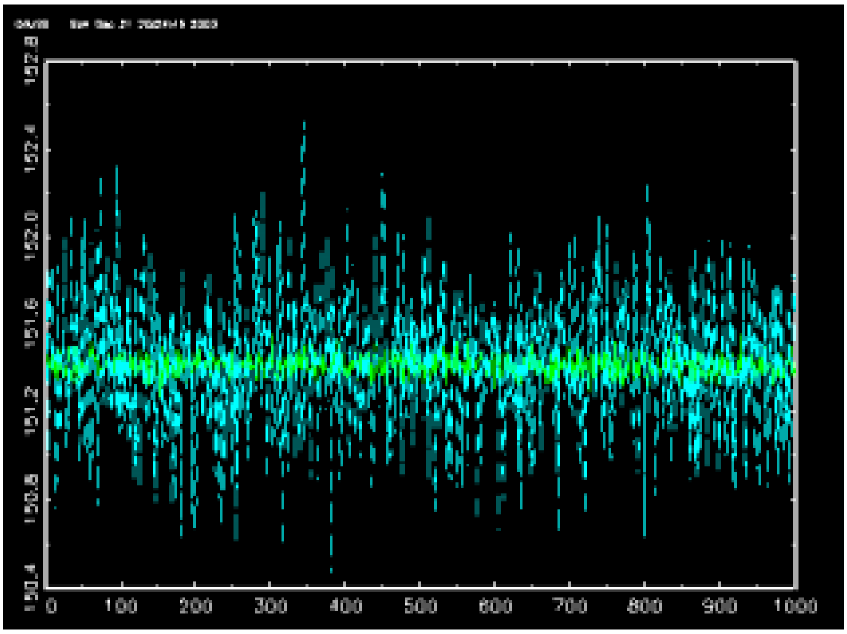
```
library pgraph;
graphset;
xy(seqa(1,1,times),m1~m2);


proc LOGNORMcv(mu,sig,p,n);
    local Z,VarZ,EZ,Y,cstar,X,THETAcv;
/* First, get c*. */
    Z=mu+sig*rndn(p,1);
    VarZ=sig^2;
    EZ=mu;
    Y=exp(Z);
    cstar=-1/(p-1)*((Y-meanc(Y))'(Z-meanc(Z)))/VarZ;
/* Then, simulate (Y,Z) again. */
    Z=mu+sig*rndn(n,1);
    Y=exp(Z);
    X=Y+cstar*(Z-EZ);
    THETAcv=meanc(X);
    retp(THETAcv);
endp;
```

結果表示

```
THETAcv=          151.41817
Var of THETAcv=        0.00194822
Var of E(Y)    =        0.09580495
```

なお、c*は－150 前後の値を取ることが print cstar;の 1 行を procedure の内部に置くことによってわかります。明らかに、Control Variate による推定値（この場合、対数正規分布にしたがう乱数の平均値）の分散は、直接 exponential をとることによって作成した乱数の平均の分散よりも小さいことが見てとれます。

グラフ表示



基本的なアルゴリズム

１）相関するＹとＺをｐ回分発生させる。

２）Cov(Y,Z)をもとめて、既知である Var(Z)の値とあわせて、

$$c^* = -\frac{Cov(Y,Z)}{Var(Z)}$$ を計算する。

３）再度、相関するＹとＺをｎ回分発生させる。

４）既知であるＥ(Z)とすでに計算された係数c*を使って、

$$X = Y + c^*(Z - E(Z))$$ を求める。

５）Ｘの平均をとり、求める$\hat{\theta}_{cv}$とする。必要であればＸの標準偏差を求め信頼区間を計算。

ただし、上の対数正規分布のケースでは、分布そのものが通常左右非対称で正規性を仮定できませんから、標準偏差をもとめて定数をかけることで、平均のプラスマイナスの値で信頼区間を求めることはふさわしくないでしょう。その場合には、Ｘ自体を小さい方から大きい方へ並べ替えた上で、その小さい方から数えて2.5%目と97.5%目の間が95%信頼区間であるとすればよいでしょう。

なお、この信頼区間とグラフの平均値の散らばりとは違います。上のグラフはｐ回のシミュレーションでｃ*を求め、ｎ回のメインのシミュレーションで推定値を求めるといった１セットのシミュレーションで求められた値をtimes回繰り返して、その散らばりを見ています。信頼区間は、１セットのシミュレーションの中でのＸの散らばりを見ていることになります。

逆関数法で標準正規分布から作成された指数分布の平均を求める例

正規分布を用いた例ではなくて、今度は０と１の間の一様乱数を用いた例についても見てみます。一番多用されるのは、逆関数法で何かの分布を作成し、その平均値を計算する際に用いられます。例えば、指数分布 $F(x)=1-\exp(-\lambda x)$ だたし$\lambda>0$ についてその分布の平均値がわからないものと仮定して、その値を Control Variate を用いて求めてみます。

プログラム

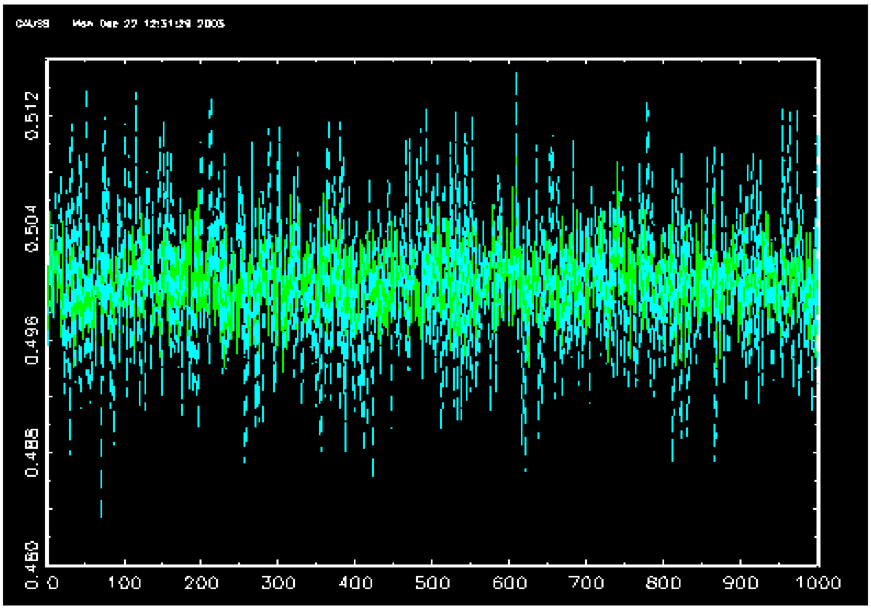
```
new; cls;
lambda=2;
p=10000;
n=10000;



times=1000;
m1=zeros(times,1);
m2=zeros(times,1);
i=1;
do while i<=times;
    m1[i]=EXPONcv(lambda,p,n);
    m2[i]=meanc(-ln(1-rndu(n,1))/lambda);
    i=i+1;
endo;
print "THETAcv=" EXPONcv(lambda,p,n);
print/rd "Var of THETAcv=" vcx(m1);
print/rd "Var of E(Y)     =" vcx(m2);
library pgraph;
graphset;
xy(seqa(1,1,times),m1~m2);



proc EXPONcv(lambda,p,n);
    local Z,VarZ,EZ,Y,cstar,X,THETAcv;
/* First, get c*. */
    Z=rndu(p,1);
    VarZ=1/12;
    EZ=1/2;
```

```
    Y=-ln(1-Z)/lambda;
    cstar=-1/(p-1)*((Y-meanc(Y))'(Z-meanc(Z)))/VarZ;
/* Then, simulate (Y,Z) again. */
    Z=rndu(n,1);
    Y=-ln(1-Z)/lambda;
    X=Y+cstar*(Z-EZ);
    THETAcv=meanc(X);
    retp(THETAcv);
endp;
```

画面表示

```
THETAcv=        0.49958576
Var of THETAcv=        0.00000642
Var of E(Y)    =        0.00002618
```

グラフ表示



上のように、Control Variate を用いたパラメータλ＝２の指数分布の平均は 0.5 のまわりに通常の逆関数法で作成された分布の平均よりも狭い範囲で求められているのがわかります。実際、この場合、統計上の理論値も 1/λ＝0.5 ですから、１セットのシミュレーションでも当たり外れなく推定値が求められていることがわかります。

手順は太字のところが違うだけで、前の例の時と基本的に同じですが、逆関数法との組み合わせのケースとしてあらためて手順を示すと以下のようになります。

基本的なアルゴリズム

１）相関するＹとＺをｐ回分発生させる。ここで、Ｚは０と１の間の一様乱数。Ｙは逆変換法でその一様乱数から変換された乱数。

２）Cov(Y,Z)をもとめて、既知である Var(Z)の値とあわせて、

$$c^* = -\frac{Cov(Y,Z)}{Var(Z)}$$

を計算する。ここでは、Var(Z)=1/12 とセットする。

３）再度、相関するＹとＺをｎ回分発生させる。

４）既知であるＥ(Z)＝1/2とすでに計算された係数c*を使って、

$$X = Y + c^*(Z - E(Z))$$

を求める。

５）Ｘの平均をとり、求める$\hat{\theta}_{cv}$とする。

Ｔ期における対数正規過程の平均を求める例

プログラム

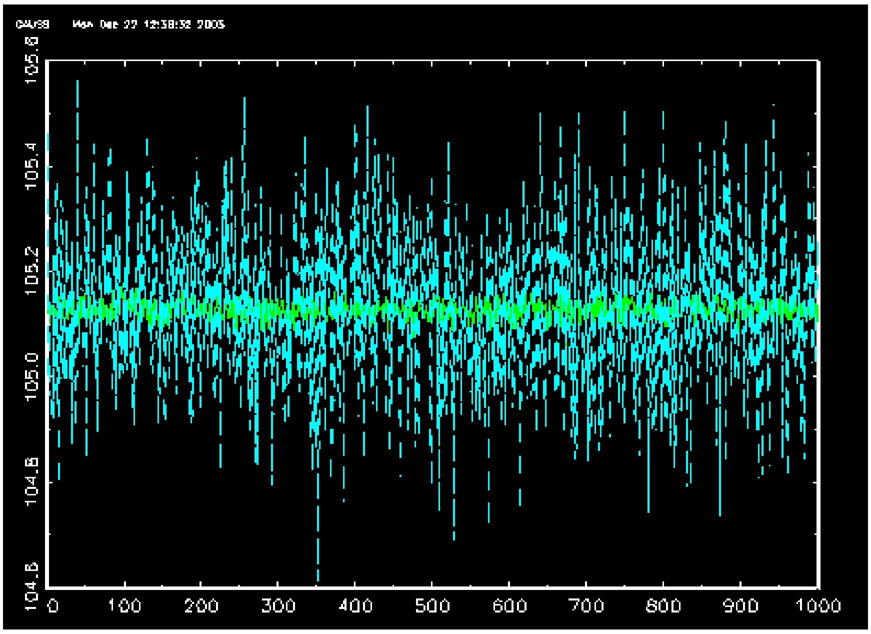
```
new; cls;
S0=100;
r=0.05;
sig=0.15;
T=1;
p=10000;
n=10000;



times=1000;
m1=zeros(times,1);
m2=zeros(times,1);
i=1;
do while i<=times;
    m1[i]=Scv(S0,r,sig,T,p,n);
    m2[i]=meanc( exp(ln(S0)+(r-sig^2/2)*T+sig*rndn(n,1)*sqrt(T)) );
    i=i+1;
endo;
print "THETAcv=" Scv(S0,r,sig,T,p,n);
print/rd "Var of THETAcv=" vcx(m1);
```

```
print/rd "Var of E(Y)     =" vcx(m2);
library pgraph;
graphset;
xy(seqa(1,1,times),m1~m2);


proc Scv(S0,r,sig,T,p,n);
    local Z,VarZ,EZ,Y,cstar,X,THETAcv;
/* First, get c*. */
    Z=rndn(p,1);
    VarZ=1;
    EZ=0;
    Y=exp(ln(S0)+(r-sig^2/2)*T+sig*Z*sqrt(T));            /* e=Z */
    cstar=-1/(p-1)*((Y-meanc(Y))'(Z-meanc(Z)))/VarZ;
/* Then, simulate (Y,Z) again. */
    Z=rndn(n,1);
    Y=exp(ln(S0)+(r-sig^2/2)*T+sig*Z*sqrt(T));
    X=Y+cstar*(Z-EZ);
    THETAcv=meanc(X);
    retp(THETAcv);
endp;
```

画面表示

```
THETAcv=         105.11849
Var of THETAcv=       0.00028500
Var of E(Y)    =       0.02367561
```

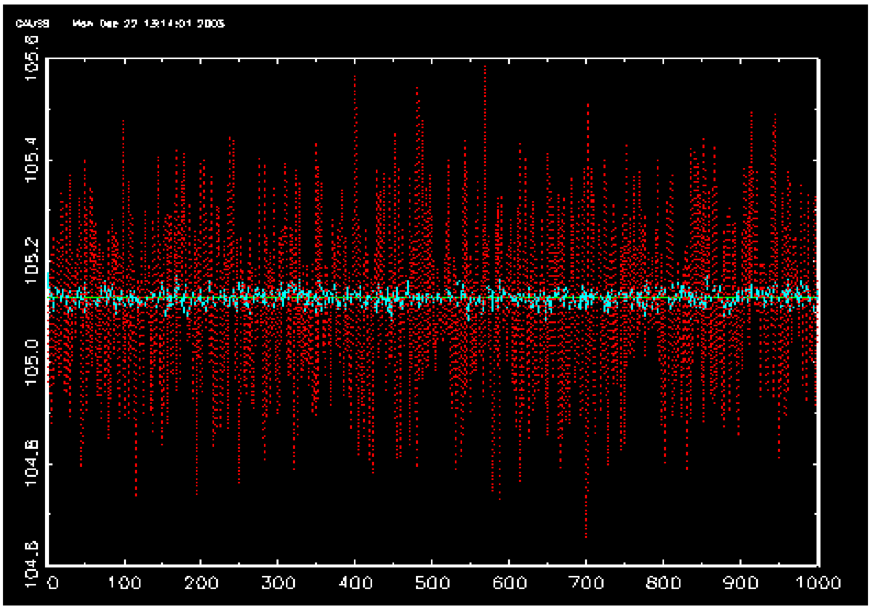
グラフ表示



もっとも、Ｔ期における対数正規過程の平均値はS0ｅrTですから、上のプログラムを

```
xy(seqa(1,1,times),S0*exp(r*T)*ones(times,1)~m1~m2);
```

というふうにグラフを描くｘｙ関数の箇所を変更してやれば、以下のグラフが描けます。



水色はControl Variateの結果。赤は通常のMonte Carlo解です。あきらかに、理論上の平均のまわりに、解が散らばっていていることがわかります。

ヨーロピアンコールオプション解を求める例

　さらに、今度はＴ期における対数正規過程の平均と分散がわかっているものと仮定して、それらの値を用いて、そのTruncateされた解の１つであるヨーロピアンコールオプションの解をControl Variateを用いて計算してみます。

プログラム（かなりの時間を所要）

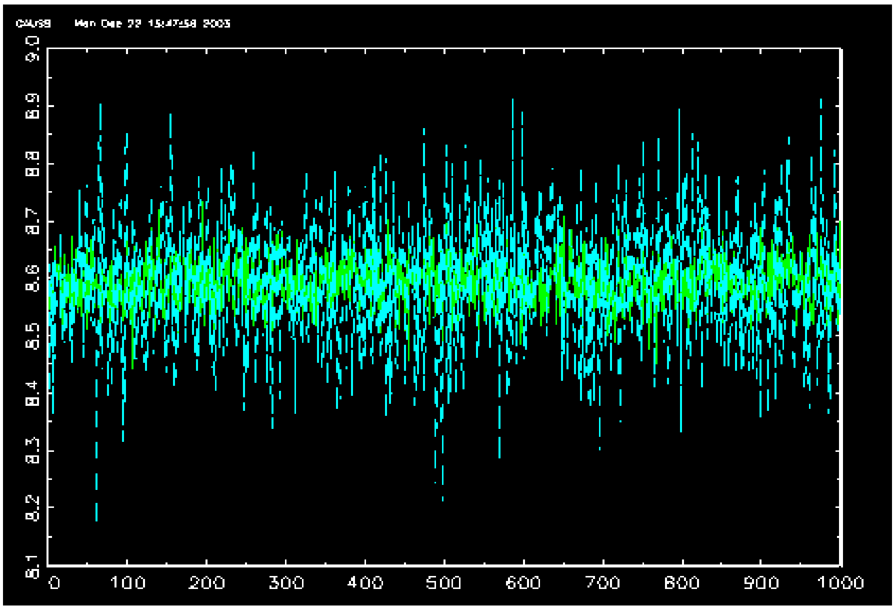
```
new; cls;
S0=100;
K=100;
r=0.05;
sig=0.15;
T=1;
p=10000;
n=10000;

times=1000;
m1=zeros(times,1);
m2=zeros(times,1);
i=1;
do while i<=times;
    m1[i]=Ccv(S0,K,r,sig,T,p,n);
    S=exp(ln(S0)+(r-sig^2/2)*T+sig*rndn(n,1)*sqrt(T));
    m2[i]=meanc( exp(-r*T)*callopt(S,K) );
    i=i+1;
endo;
print "THETAcv=" Ccv(S0,K,r,sig,T,p,n);
print/rd "Var of THETAcv=" vcx(m1);
print/rd "Var of E(Y)     =" vcx(m2);
library pgraph;
graphset;
xy(seqa(1,1,times),m1~m2);
```

```
proc Ccv(S0,K,r,sig,T,p,n);
    local Z,VarZ,EZ,Y,cstar,X,THETAcv;
/* First, get c*. */
    Z=exp(ln(S0)+(r-sig^2/2)*T+sig*rndn(p,1)*sqrt(T));
    VarZ=S0^2*exp(r*T)^2*(exp(sig^2*T)-1);
    EZ=S0*exp(r*T);
    Y=exp(-r*T)*callopt(Z,K);
    cstar=-1/(p-1)*((Y-meanc(Y))'(Z-meanc(Z)))/VarZ;
/* Then, simulate (Y,Z) again. */
    Z=exp(ln(S0)+(r-sig^2/2)*T+sig*rndn(p,1)*sqrt(T));
    Y=exp(-r*T)*callopt(Z,K);
    X=Y+cstar*(Z-EZ);
    THETAcv=meanc(X);
    retp(THETAcv);
endp;

proc callopt(S,K);
    local i,C;
    C=zeros(rows(S),1);
    i=1;
    do while i<=rows(S);
        C[i]=maxc(S[i]-K|0);
        i=i+1;
    endo;
    retp(C);
endp;
```

画面表示

```
THETAcv=        8.6155688
Var of THETAcv=       0.00174290
Var of E(Y)   =       0.01223894
```

グラフ表示



このように、あらゆるレベルのシミュレーションにおいて、２組のシミュレーションのうち一方の平均値（理論厳密値）がわかっていれば、負に相関してようが正に相関してようが、何らかの相関があれば、係数ｃ*を求めることを経て、より正確な推定値が得られる。すなわち、分散が減少する。あらゆると言ったのは、例えばAsian Optionの解を求める際に、もう一方のシミュレーションは標準正規分布でも構わないし、対数正規過程でも構わない。はたまた、Asian Optionの元になっている対数正規過程の平均でも構わないということである。いずれのケースでも、同じ乱数から２つのシミュレーションを行なえば、何らかの相関が認められるので、分散減少の効果があらわれる。

それでは、冒頭で示したような金融工学の教科書に載っているようなControl Variateの「公式」 $F_A = F_A^* + (F_B - F_B^*)$ とは何なのであろうか？

今、ｃ*＝－１と固定してやろう。そうすると、

$$\hat{\theta}_{cv} = Y + (E(Z) - Z)$$ となり両辺の分散をとると

$$\begin{aligned} Var(\hat{\theta}_{cv}) &= Var(Y + E(Z) - Z) \\ &= Var(Y) + Var(Z) - 2Cov(Y, Z) \\ &= 2Var(Y) - 2\rho_{YZ} Var(Y) \\ &= 2Var(Y)(1 - \rho_{YZ}) \qquad if \quad Var(Y) = Var(Z) \end{aligned}$$

となる。これは、c*=－１を代入した値と同じである。仮に、Var(Y)とVar(Z)が等しいと仮定できるなら、上のようなＹとＺの相関係数を含む式に書きなおすことができる。

また今、ＹとＺ自身もシミュレーションによる解であるあるとして、同様に両辺の分散をとってやれば、

$$Var(\hat{\theta}_{cv}) = Var(\bar{Y}) + Var(\bar{Z}) - 2Cov(\bar{Y}, \bar{Z})$$
$$= \frac{\sigma^2}{n} + \frac{\sigma^2}{n} - 2\rho_{\overline{YZ}}\frac{\sigma^2}{n} = 2(1-\rho_{\overline{YZ}})\frac{\sigma^2}{n}$$

となることから、ＹとＺの相関係数が 0.5 よりも上まわっていれば、すなわち、正に極めて相関しているなら、いつでもＹの推定値の分散σ²/ n よりも小さくなり分散減少が達成されるのである。この場合の条件は、ＹとＺの分散の値が一概には同じと言えないので、総じて論ずることはできないが、だいたいのところ、両者がよく相関していれば、ｃ*＝－１として固定してやっても分散減少が達成されるのである。これまで示してきた例は、そのほとんどが正方向によく相関している例なので、procedure 内部で、cstar=－1;と固定することによっても、分散減少効果があらわれることを確認してもらいたい。定数ｃ*を計算することなく、－１と固定して、$F_A = F_A^* + (F_B - F_B^*)$ と言っても差し支えがないのである。

アメリカンオプションの解を求める例

プログラム

```
new; cls;
n=5;
S0=50;
K=50;
r=0.10;
sig=0.40;
T=5/12;
call ameputcv(n,S0,K,r,sig,T);



proc ameputcv(n,S0,K,r,sig,T);
    local fAs,fBs,fB,fA;
    screen off;
    fAs=americanputhigher(n,S0,K,r,sig,T);
    fBs=europuthigher(n,S0,K,r,sig,T);
    fB=P(S0,K,r,sig,T);
    fA=fAs-fBs+fB;
    screen on;
    print "fA*=" fAs;
    print "fB*=" fBs;
    print "fB =" fB;
```

```
    print "fA(American Put Price)=" fA;
    print;
    retp(fA);
endp;

proc europuthigher(n,S0,K,r,sig,T);
    local delt,u,d,a,p,ma,mb,i,j;
    delt=T/n;
    u=exp(sig*sqrt(delt));
    d=exp(-sig*sqrt(delt));
    a=exp(r*delt);
    p=(a-d)/(u-d);
    print "== European Put Option on non-dividend-paying stock ==";
    print "delt=" delt delt*365 "days";
    print "u=" u;
    print "d=" d;
    print "a=" a;
    print "p=" p;
    print;
/* Last column */
    ma=miss(zeros(2*n+1,1),0);
    mb=miss(zeros(2*n+1,1),0);
    j=n+1;
    i=1;
    do while i<=2*j-1;
        mb[n+1-j+i]=maxc(K-S0*u^(j-1-(i-1)/2)*d^((i-1)/2)|0);
        i=i+2;
    endo;
/* Backward by 2 column vectors */
    j=n;
    do while j>=1;
        i=1;
        do while i<=2*j-1;
            ma[n+1-j+i]=exp(-r*delt)*(p*mb[n+1-j+i-1]+(1-p)*mb[n+1-j+i+1]);
            i=i+2;
        endo;
```

```
        mb=ma;
        j=j-1;
    endo;
    print/lz "f=" ma[n+1];
    print;
    retp(ma[n+1]);
endp;

proc americanputhigher(n,S0,K,r,sig,T);
    local delt,u,d,a,p,ma,mb,i,j;
    delt=T/n;
    u=exp(sig*sqrt(delt));
    d=exp(-sig*sqrt(delt));
    a=exp(r*delt);
    p=(a-d)/(u-d);
    print "== American Put Option on non-dividend-paying stock ==";
    print "delt=" delt delt*365 "days";
    print "u=" u;
    print "d=" d;
    print "a=" a;
    print "p=" p;
    print;
/* Last column */
    ma=miss(zeros(2*n+1,1),0);
    mb=miss(zeros(2*n+1,1),0);
    j=n+1;
    i=1;
    do while i<=2*j-1;
        mb[n+1-j+i]=maxc(K-S0*u^(j-1-(i-1)/2)*d^((i-1)/2)|0);
        i=i+2;
    endo;
/* Backward by 2 column vectors */
    j=n;
    do while j>=1;
        i=1;
        do while i<=2*j-1;
```

```
            ma[n+1-j+i]=maxc(K-S0*u^(j-1-(i-1)/2)*d^((i-1)/2) | exp(-r*delt)*(p*mb[n+1-
j+i-1]+(1-p)*mb[n+1-j+i+1]));
            i=i+2;
        endo;
        mb=ma;
        j=j-1;
    endo;
    print/lz "f=" ma[n+1];
    print;
    retp(ma[n+1]);
endp;

proc P(S0,K,r,sig,T);
    local d1,d2;
    d1=(ln(S0/K)+(r+sig^2/2)*T)./(sig*sqrt(T));
    d2=(ln(S0/K)+(r-sig^2/2)*T)./(sig*sqrt(T));
    retp( K.*exp(-r*T)*cdfn(-d2)-S0.*cdfn(-d1) );
endp;
```

画面表示

```
fA*=        4.4884585
fB*=        4.3190187
fB =        4.0759810
fA(American Put Price)=        4.2454208
```

他方、コールオプションの解は、次のようなことになります。

プログラム

```
new; cls;
n=5;
S0=50;
K=50;
r=0.10;
sig=0.40;
T=5/12;
call amecallcv(n,S0,K,r,sig,T);
```

```
proc amecallcv(n,S0,K,r,sig,T);
    local fAs,fBs,fB,fA;
    screen off;
    fAs=americancallhigher(n,S0,K,r,sig,T);
    fBs=eurocallhigher(n,S0,K,r,sig,T);
    fB=C(S0,K,r,sig,T);
    fA=fAs-fBs+fB;
    screen on;
    print "fA*=" fAs;
    print "fB*=" fBs;
    print "fB =" fB;
    print "fA(American Call Price)=" fA;
    print;
    retp(fA);
endp;

proc eurocallhigher(n,S0,K,r,sig,T);
    local delt,u,d,a,p,ma,mb,i,j;
    delt=T/n;
    u=exp(sig*sqrt(delt));
    d=exp(-sig*sqrt(delt));
    a=exp(r*delt);
    p=(a-d)/(u-d);
    print "== European Call Option on non-dividend-paying stock ==";
    print "delt=" delt delt*365 "days";
    print "u=" u;
    print "d=" d;
    print "a=" a;
    print "p=" p;
    print;
/* Last column */
    ma=miss(zeros(2*n+1,1),0);
    mb=miss(zeros(2*n+1,1),0);
    j=n+1;
    i=1;
    do while i<=2*j-1;
```

```
            mb[n+1-j+i]=maxc(S0*u^(j-1-(i-1)/2)*d^((i-1)/2)-K|0);
            i=i+2;
        endo;
/* Backward by 2 column vectors */
        j=n;
        do while j>=1;
            i=1;
            do while i<=2*j-1;
                ma[n+1-j+i]=exp(-r*delt)*(p*mb[n+1-j+i-1]+(1-p)*mb[n+1-j+i+1]);
                i=i+2;
            endo;
            mb=ma;
            j=j-1;
        endo;
        print/lz "f=" ma[n+1];
        print;
        retp(ma[n+1]);
endp;

proc americancallhigher(n,S0,K,r,sig,T);
        local delt,u,d,a,p,ma,mb,i,j;
        delt=T/n;
        u=exp(sig*sqrt(delt));
        d=exp(-sig*sqrt(delt));
        a=exp(r*delt);
        p=(a-d)/(u-d);
        print "== American Call Option on non-dividend-paying stock ==";
        print "delt=" delt delt*365 "days";
        print "u=" u;
        print "d=" d;
        print "a=" a;
        print "p=" p;
        print;
/* Last column */
        ma=miss(zeros(2*n+1,1),0);
        mb=miss(zeros(2*n+1,1),0);
```

```
    j=n+1;
    i=1;
    do while i<=2*j-1;
        mb[n+1-j+i]=maxc(S0*u^(j-1-(i-1)/2)*d^((i-1)/2)-K | 0);
        i=i+2;
    endo;
/* Backward by 2 column vectors */
    j=n;
    do while j>=1;
        i=1;
        do while i<=2*j-1;
            ma[n+1-j+i]=maxc(S0*u^(j-1-(i-1)/2)*d^((i-1)/2)-K | exp(-r*delt)*(p*mb[n+1-
j+i-1]+(1-p)*mb[n+1-j+i+1]));
            i=i+2;
        endo;
        mb=ma;
        j=j-1;
    endo;
    print/lz "f=" ma[n+1];
    print;
    retp(ma[n+1]);
endp;

proc C(S0,K,r,sig,T);
    local d1,d2;
    d1=(ln(S0/K)+(r+sig^2/2)*T)./(sig*sqrt(T));
    d2=(ln(S0/K)+(r-sig^2/2)*T)./(sig*sqrt(T));
    retp( S0.*cdfn(d1)-K*exp(-r*T).*cdfn(d2) );
endp;
```

画面表示

```
fA*=        6.3595459
fB*=        6.3595459
fB =        6.1165081
fA(American Call Price)=       6.1165081
```

このようなパラメータ関係であるとき、アメリカンコールオプションの２項ツリーによるシミュレーション解もヨーロピアンコールオプションの２項ツリーによるシミュレーション解も early exercise が生じないため同一になっています。そのようなケースでは、fA*とfB*は等しくなりその差は０でるから、結果として、$F_A = F_A^* + (F_B - F_B^*)$は$F_A = F_B$ということになります。すなわち、求める解は、ヨーロピアンコールオプションの厳密解F_Bと等しくなるということです。何もおもしろいところはなく、ヨーロピアンコールオプションの厳密解がそのままアメリカンの解となっているのです。