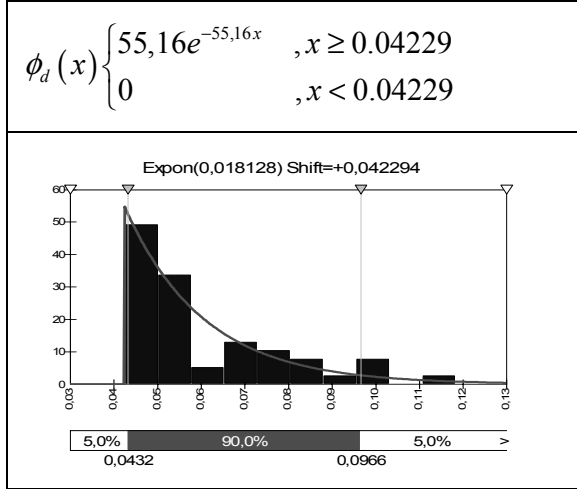


have to add this figure to the mean, which then becomes:  
mean = 0.01614 + 0.0422 = 0.05834. The variance obtained is: variance = 0.00026107.



**Table 4.** Jumps Down adjusted Exponential Distribution

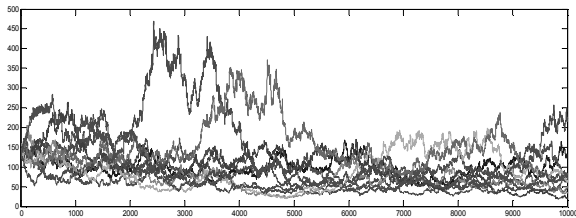
In the case of down jumps, we cannot generate an exponential distribution, as returns are now negative. We therefore need to convert the negative return distribution into a positive one in order to derive our distribution. When using it for the simulation, we thus turn to negative the result for the down jumps yielded by the distribution obtained. The actual result for the down jumps is shown.

The distribution derived from the optimization process is described on Table 4.

Having a mean = - 0.018128 - 0.042294 = -0.06042, and variance = 0.00032861. Graphically, the adjusted curve is in the Chart 8.

### Simulating the mean reversion motion with jumps

We use the above-stated formulas (1.13) and (1.14) to simulate the mean reversion process with the preceding parameters



**Chart 3.** Graphical output of the simulation of the mean reversion with exponential jumps model with 10,000 steps corresponding to 40 years.

The MATLAB code needed to simulate the process is set out in the appendix as MATLAB Code 1.

## Conclusions

The model laid out above allows us to derive the parameters directly from market data, and makes it possible to adjust them according to executives' expectations. All parameters used in the model can be directly and intuitively understood, and its comprehension requires just average mathematical and statistical know-how.

The model is capable of coherently replicating the shifts in commodity markets, allowing for situations such as the one we have analyzed here, which involves a jump up in prices, to fit in.

The process of feeding and updating the model with new market data is simple and can be implemented both on MATLAB-type programs and on simulation-based spreadsheet programs.

## APPENDIX

### MATLAB Code 1

```
T=10000; %Number of periods
dt=1; %Time interval
N=T/dt;
Speed=.000452; %Reversion Speed
Sigma=.0125236; %Volatility of the process
paths=10; %Number of paths
S0=142; %Initial Value
S = 129.68; %Expected mean value
LambdaU = 0.013014; %Jump up frequency
LambdaD = 0.015435; %Jump down frequency
Lambda = 0.02845; %Total jump frequency
MeanPhiU = 0.016148; %Mean jump up
ShiftPhiU = 0.042288; %Jump up scaling
VarPhiU = 0.00026107; %Jump up variance
MeanPhiD = -0.018128; %Mean jump down
ShiftPhiD = -0.042294; %Jump down scaling
VarPhiD = 0.00032861; %Jump down variance
MeanPhi = MeanPhiU + ShiftPhiU + MeanPhiD + ShiftPhiD; %Mean of Phi
VarPhi = VarPhiU + VarPhiD; %Variance of Phi
UniformU = find(unifrnd(0,1,N,paths)>LambdaU);
UniformD = find(unifrnd(0,1,N,paths)>LambdaD);
dQUp = exprnd(MeanPhiU, N,paths)+ ShiftPhiU;
dQUp(UniformU) = 0;
dQDown = -exprnd(-MeanPhiD, N,paths)+ ShiftPhiD;
dQDown(UniformD) = 0;
dQ = dQUp + dQDown;
Wiener=randn(N,paths);
SpeedDt=Speed*dt;
St=S0*ones(N,paths);
G1=exp(-SpeedDt);
G2=log(S+Lambda*MeanPhi*Speed)*(1-exp(-SpeedDt));
G3=Sigma*sqrt((1-exp(-2*SpeedDt))/(2*Speed));
G4=(1-exp(-2*SpeedDt))*((Sigma^2)+Lambda*VarPhi)/(4*Speed);
for i=1:N
    for j=1:paths
        St(i+1,j)=exp(log(St(i,j))*G1+G2+dQ(i,j)+G3*Wiener(i,j)-G4);
    end
end
```

## REFERENCES

Clellow, L., Strickland, C. 2000 "Energy Derivatives: Pricing and Risk Management", *Lacima Publications*