Structured Finance Course Paper: What Accumulators Are and How to Price Them

Zizhou Sang, 141292018

Abstract

Accumulator is a highly path dependent derivative structured financial product once enjoyed wide popularity in some Asian cities with its speculative nature and became especially controversial in the time of financial crisis 2008. Innovated with simple structure and embedded with barrier options, the accumulator itself serves as an engaging archetype for pricing research on structured finance. In this paper we chiefly studied Lam's[1] paper then made an original yet flawed implementation based on Matlab. The first section introduces what an accumulator is, followed by the pricing principle, and section three is a specific code realization of a numeric example.

1 What Accumulators Are

1.1 Introduction

In the year of 2008, a structured product called accumulator has aroused a storm of law-suits in Hong Kong and some other Asian cities. Dubbed as 'I kill you later', this unregulated product has made investors suffer exceptionally hard that period with some claim to had lost HK$ 1 billion. Larry Yung, chair of Citic Pacific, handed in his resignation after confirming a 15.5 billion HKD loss on currency accumulator against Australian and New Zealand dollars. So is this product really ferocious as above mentioned? If so why are people still buying it? All these lead to one question: what is an accumulator exactly?

Just like the South China Morning Post reported, the accumulator was 'once touted as a safe way to tap the bull market', it grants investors buying blue-chip stocks at a discounted price *every day* but stipulates to buy twice and even more times as much when the stock goes under the price, posing nearly unlimited downside risk which is scarcely told when buying. So in prosperity this product can be seemingly a effortless money-making tool, but just goes the saying ' there is no free lunch', when market turns the other side it corrupts at once into a financial black hole. Anyway the accumulator can just be accumulative high or low, joy or sadness. And after grasping the essence of it, our major instinctively calls for a pricing solution which constitutes the second part of this article.

1.2 Features of An Accumulator

We first introduce some basic characteristics of an accumulator:

Discount A typical accumulator contract obliges an investor to buy a quantity of a stock on specified observation days within the term of contract at a strike price K which is usually set at a discount of the original spot price S_0. K is called the strike price and remains a constant throughout the life of the contract.

Step-up The observation days are denoted by $t_1, \ldots, t_n$ where $t_i \leqslant N$ and N is the length of the contract term. Furthermore, if the closing price S_i on the i^{th} observation day is larger than or equal to K, the purchase quantity is fixed as Q. However if S_i is less than K, the purchase quantity is fixed as $2Q$ or can be other gearing ratio.

Knock-out The contract also has a knock-out feature in that if the closing price at any time within the contract life exceeds a barrier H, the contract will terminate automatically. An accumulator usually has a zero-cost structure meaning there is no premium payment for both parties.

There are some other details of accumulator to be noted for and later discussed:

1. Settlement of the accumulator can be done immediately or on a weekly or monthly basis. In practice, immediate settlement is not so common and normally under a monthly settlement, all stocks to be purchased in the same month will be cleared at the last day in that month.

2. The knock-out barrier H can be applied discretely or continuously. For discrete barriers they are checked at the end of each observation day while continuous barrier is throughout the period. To be specific, checking whether close price or the price of any moment has overstepped the bound exemplifies discrete and continuous barrier respectively.

2 How to Price Them

The general idea to price an accumulator can be split into two major branches: the *analytical* and *numerical* method . Given the path dependent nature of accumulator, the numerical method worked on the basis that fair price is the present value of expected return under risk-neutral random walk[2], and can be theoretically implemented through Monte Carlo simulation or binomial tree. The analytical method's core thinking was to decompose the accumulator to a combination of barrier options and then applying barrier options' pricing method. Hence theoretically, analytical pricing methods of accumulator is as much as barrier options', whose representatives are put forward by Reiner and Rubinstein (1991)[3], and later on there are series of amelioration done grounded on their research, including Lévy Process and Fourier cosine[4] etc. Next we focus on the analytical method using classical barrier solutions.

Analytical Method of Accumulator Pricing

In this subsection, we start from the simplest immediate settlement case where a typical accumulator can be directly decomposed, then consider the mapping between trading day t_i and settlement day T_i transiting to the delay settlement only by making a little modification on formula derived from immediate settlement. Then we move on to the discrete barrier, where there is a convenient and efficient test-proven approximation from continuous to discrete barrier based on previous scholars' work.

Immediate Settlement

We consider first an accumulator with n-day to expiration. Under immediate settlement, the pay-off at day $t_i (i \leqslant n)$ is given by:

$$\begin{cases} 0 & \max_{0 \leqslant \tau \leqslant t_i} \; S_\tau \geqslant H \\ S_{t_i} - K & \max_{0 \leqslant \tau \leqslant t_i} \; S_\tau < H, S_{t_i} \geqslant K \\ 2(S_{t_i} - K) & \max_{0 \leqslant \tau \leqslant t_i} \; S_\tau < H, S_{t_i} < K \end{cases}$$

The pay-off is equivalent to long one up-and-out barrier call option and short two up-and-out barrier put options with expiration time t_i . Hence, the fair price of an accumulator contract is given by:

$$V = \sum_{i=1}^{n} \{C_{uo}(t_i, K, H) - 2P_{uo}(t_i, K, H)\}$$

where C_{uo} represents the fair price of a up-and-out barrier call option and P_{uo} represents the fair price of a up-and-out put option. If we adopt the usual Black-Scholes assumptions of constant risk-free interest rate ($= r$), constant volatility ($=\sigma$) as well as a constant payout rate ($= q$) of the underlying asset, the terms C_{uo} and P_{uo} have closed form solutions. Rubinstein and Reiner (1991) studied binary options with continuously monitored barrier under the

Black-Sholes' framework and various kinds of binary barrier options can be priced. Using the results derived by Harrison(1985) and by Rubinstein and Reiner (1991), we are able to express C_{uo} and P_{uo} as below:

$$
\begin{aligned}
C_{uo}(K,H,t_i) = S_0 e^{-qt_i}\{N(x(t_i)) - N(x_1(t_i)) + (\frac{H}{S})^{2\lambda}[N(-y(t_i)) - N(-y_1(t_i))]\} \\
- e^{-rt_i}K\{N(x(t_i) - \sigma\sqrt{t_i}) - N(x_1(t_i) - \sigma\sqrt{t_i}) + \\
(\frac{H}{S})^{2\lambda-2}[N(-y(t_i) + \sigma\sqrt{t_i}) - N(-y_1(t_i) + \sigma\sqrt{t_i})]\} \quad (1)
\end{aligned}
$$

$$
\begin{aligned}
P_{uo}(K,H,t_i) = e^{-rt_i}K\{N(-x(t_i) + \sigma\sqrt{t_i}) - (\frac{H}{S})^{2\lambda-2}N(-y(t_i) + \sigma\sqrt{t_i})\} \\
- S_0 e^{-qt_i}\{N(-x(t_i)) - (H/S)^{2\lambda}N(-y(t_i))\} \quad (2)
\end{aligned}
$$

where $N(.)$ represents the cumulative standard normal distribution function and:

$$
x(t_i) = \frac{\log(S/K) + (\mu+\sigma^2)t_i}{\sigma\sqrt{t_i}}, x_1(t_i) = \frac{\log(S/H) + (\mu+\sigma^2)t_i}{\sigma\sqrt{t_i}}, \mu = r - q - \sigma^2/2
$$

$$
y(t_i) = \frac{\log(H^2/SK) + (\mu+\sigma^2)t_i}{\sigma\sqrt{t_i}}, y_1(t_i) = \frac{\log(H/S) + (\mu+\sigma^2)t_i}{\sigma\sqrt{t_i}}, \lambda = 1 + \mu/\sigma^2
$$

and we sum from day 1 to day n getting the fair value of the accumulator as follows

$$
\begin{aligned}
V = \sum_{i=1}^{n}\{Se^{-qt_i}[2 - N(x(t_i)) - N(x_1(t_i)) - (\frac{H}{S})^{2\lambda}N(-y(t_i)) - (\frac{H}{S})^{2\lambda}N(-y_1(t_i))] \\
- Ke^{-rt_i}[2 - N(x(t_i) - \sigma\sqrt{t_i}) - N(x_1(t_i) - \sigma\sqrt{t_i}) \\
- (\frac{H}{S})^{2\lambda-2}N(-y(t_i) + \sigma\sqrt{t_i}) - (\frac{H}{S})^{2\lambda-2}N(-y_1(t_i) + \sigma\sqrt{t_i})]\} \quad (3)
\end{aligned}
$$

Delay Settlement

In this subsection, we will price an accumulator with delayed settlement. We use $T_i (i = 1, 2, \ldots, n)$ denote the settlement day of certain observation day t_i which exists to be an one-to-one mapping. While $t_1 < t_2 < \ldots < t_n$ represent n different observation days, $T_1 \leqslant T_2 \leqslant \ldots \leqslant T_n$ may be equal. In fact, there are only m different values in $\{T_i; i = 1, 2, \ldots, n\}$ because there are only m different settlement days. The payoff at observation day t_i actually come on the settlement day T_i and can still be synthesized using long one up-and-out barrier call option and short two up-and-out barrier put options, we omit the details and the fair price of an accumulator contract with delay settlement is given by:

$$
V^{Delay} = \sum_{i=1}^{n}\{C_{uo}^F(t_i, K, H, T_i) - 2P_{uo}^F(t_i, K, H, T_i)\}
$$

Using density functions derived by Rubinstein and Reiner (1991), we can derive the values of the above barrier call and put options on forward contracts. Hence, the value of an accumulator with delay settlement is:

$$V^{Delay} = \sum_{i=1}^{n} \{ Se^{-qT_i}[2 - N(x(t_i)) - N(x_1(t_i)) - (\frac{H}{S})^{2\lambda} N(-y(t_i)) - (\frac{H}{S})^{2\lambda} N(-y_1(t_i))] - Ke^{-rT_i}[2 - N(x(t_i) - \sigma\sqrt{t_i}) - N(x_1(t_i) - \sigma\sqrt{t_i}) - (\frac{H}{S})^{2\lambda-2} N(-y(t_i) + \sigma\sqrt{t_i}) - (\frac{H}{S})^{2\lambda-2} N(-y_1(t_i) + \sigma\sqrt{t_i})] \} \quad (4)$$

where the functions $x(t_i), x_1(t_i)$ etc. are as defined in the last subsection. Note we only changed the discount factor from t_i to T_i in the delay settlement.

Discrete Barrier

Under a discrete barrier, an knock-out event happens only if the barrier is breached at market closes of day t_i. Since previous formulae are derived under a continuous barrier setting, they fail to deal with accumulator with a discrete barrier according to Cheuk and Vorst(1994)[5], whose research indicated will lead to remarkable pricing errors if left untreated.

We skip the development history and move directly to the approach widely used, the Broadie, Glasserman and Kou (1997)[6] approximation of discretely monitored barrier option values using continuous formulae with an appropriately shifted barrier. The correction term used to shift the barrier of an up-and-out barrier option is $e^{\beta\sigma\sqrt{T/m}}$ where σ is the underlying volatility, T is the tenor of the option, m is the number of barrier monitoring checkpoints throughout the tenor and β is a constant factor involving the zeta function and is given by $\beta = -\zeta(1/2)/\sqrt{2\pi} \approx 0.5826$. So the corrected barrier is $\widetilde{H} = He^{\beta\sigma\sqrt{T/m}}$. Under these notations, the value of an up-and-out barrier option will be approximated by:

$$V_{Discrete}(H) = V(\widetilde{H})$$

Intuitively, for up-and-out barrier if we want to use a discrete barrier as a perfect substitute for continuous barrier, we have to be more tolerant and heighten the barrier to maintain the same effect, otherwise a lower discrete barrier is much easier to attain since it's the way must through but it's not definitely pass the continuous barrier and result in inaccurate pricing. Moreover, the height is closely connected to the sampling frequency, if we are more lazy setting less checkpoints we must heighten the barrier more to ensure the there is no mistake, and if we set more checkpoints near to infinity, its limit should be the continuous barrier. If we use daily close price and specify 1 year 252 trading days as observation, in this case the correction term m is 252 and T is 1. We will see a pricing sample next section.

3 Pricing A Sample Accumulator

In this section, we study a numerical pricing example of an accumulator with a discrete barrier, a one year tenor ($T = 1$), daily observation and monthly settlement. Assume there are 21 trading days each month and the underlying security's spot price S_0 at day 0 is \$100, the accumulator contract's exercise price K is \$90 and the discrete barrier H is \$105, the contract stipulates to buy 1 stock each trading day and gearing ratio equals 2 (buy two stock per day if price goes below the strike price K). We further assume the risk-free rate 3%, underlying security's dividend payout rate is zero and volatility is 20%. In our sample contract, the notional amount equals $252 \times \$90 \times 1 = \$22680$. Next we will be using both analytical method and Monte Carlo simulation to price the sample accumulator on Matlab.

3.1 Analytical Method

Using formulae derived from last section, a program for calculating call and put up-and-out barrier options was written on Matlab[7], note that settlement is done on a monthly basis we have to map all observation day T within a month to that month Ts using floor function. The code is listed as below:

```
%UpOut.m
function [C,P]=UpOut(S0,K,r,T,sigma,H)
% returns call and put up-and-out barrier opt price
a=(H/S0)^(-1+2*r/sigma^2);
b=(H/S0)^(1+2*r/sigma^2);
d1=(log(S0/K)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2=(log(S0/K)+(r-sigma^2/2)*T)/(sigma*sqrt(T));
d3=(log(S0/H)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d4=(log(S0/H)+(r-sigma^2/2)*T)/(sigma*sqrt(T));
d5=(log(S0/H)-(r-sigma^2/2)*T)/(sigma*sqrt(T));
d6=(log(S0/H)-(r+sigma^2/2)*T)/(sigma*sqrt(T));
d7=(log(S0*K/H^2)-(r-sigma^2/2)*T)/(sigma*sqrt(T));
d8=(log(S0*K/H^2)-(r+sigma^2/2)*T)/(sigma*sqrt(T));

Ts=(floor((T*252-1)/21)+1)/12;

C=S0*(normcdf(d1)-normcdf(d3)...
-b*(normcdf(d6)-normcdf(d8)))...
-K*exp(-r*Ts)*(normcdf(d2)-normcdf(d4)...
-a*(normcdf(d5)-normcdf(d7)));
P=-S0*(1-normcdf(d1)-b*normcdf(d8))...
+K*exp(-r*Ts)*(1-normcdf(d2)-a*normcdf(d7));
```

Note that under a discrete barrier we have to take into account the correction of H to get the right answer, in our case:

$$\widetilde{H} = He^{\beta\sigma\sqrt{T/m}} \approx 105e^{0.5826\times0.2\times\sqrt{1/252}} \approx 105 \times 1.0074$$

Then we run the below Matlab code and the result is -84.6954, which suggests the investor is 'doomed' to an expectational loss of approximately \$84.7 from the \$22680 worth contract. Thus, the contract slant to the seller side.

```
sum=0;
for i=1:252
    [c,p]=UpOut(100,90,0.03,i/252,0.2,105*1.0074);
    sum=sum+c-2p;
end;
```

3.2 Monte Carlo Method

The core thinking of Monte Carlo method is fairly straightforward as previously discussed. After generating a sufficiently large amount of random walk path under given volatility and drift rate, we apply the rules of accumulator calculate the payoff of each path then average them, the procedure is listed as below, where *AssetPaths* is path-generating function and *NRepl* is the repeated times which we later choose to be 5 million, function *AccCalc* calculate the payoff of a specific path under given parameters:

```
%AccumulatorMC.m
function[P,CI]=AccumulatorMC(S0,K,r,mu,sigma,H,NSteps,NRepl)
Payoff=zeros(NRepl,1);
for i=1:NRepl  %NRepl is the number of repetition
    Path=AssetPaths(S0,mu,sigma,1,NSteps,1);  % ...
        generate a path
    Payoff(i)=AccCalc(Path,K,r,NSteps,H);  % ...
        calculate the path's worth
end;
[P,aux,CI]=normfit(Payoff) % returning average, CI ...
    is confidence interval
```

Program *AccCalc* and *AssetPaths* are listed as below, note we treated the delayed settlement using similar floor function to put trading in a specific month and discount them together.

```
%AccCalc.m
function payoff=AccCalc(Path,K,r,NSteps,H)
mpayoff=zeros(12,1);
for i=1:NSteps
```

```
        j=floor((i-1)/21)+1; % mapping day to the month ...
            it belongs
        if Path(i)≥H
            break;
        elseif  Path(i)≥K
            mpayoff(j)=mpayoff(j)+Path(i)-K;
        else
            mpayoff(j)=mpayoff(j)+2*(Path(i)-K);
        end;
    end;
    a=1:12;
    d=exp(-1/12*a*r);
    payoff=d*mpayoff; % monthly discount done in the ...
        form of matrix multiplication
```

And here is the random walk path generating function:

```
    %AssetPaths.m
    function ...
        SPaths=AssetPaths(S0,mu,sigma,T,NSteps,NRepl)
    %NRepl represents number of paths to be generated
    %NSteps represents number of steps each path
    SPaths = zeros(NRepl,1+NSteps);
    SPaths(:,1)=S0;% let every path begins with S0

    dt=T/NSteps;
    nudt=(mu-0.5*sigma^2)*dt;
    sidt=sigma*sqrt(dt);
    for i=1:NRepl
        for j=1:NSteps
        SPaths(i,j+1)=SPaths(i,j)*exp(nudt+sidt*randn);
        end;
    end;
```

However, the original author of this question did not give a specific μ, the drift term of the underlying asset necessary for generating asset paths, it raised question when we tried to substitute different μ into the program getting various results but the analytical method is independent of μ. We doubt there must be something we are missing and the above program may not be that reliable. Nevertheless, we still tried one example out which is done months ago by mistake assuming $\mu = r = 0.03$:

```
>>AccumulatorMC(100,90,0.03,0.03,0.2,105,252,5000000)
```

and the result was -89.0332, confidence interval CI is between -90.2532 and -87.8132.

4 Conclusion

In this so-called paper, we referred to previous scholars' work and wrote some trivial, problematic still code based on their findings. The process is mostly derivative, flawed and is by no means close to a real paper no matter in content or academic norm etc.

Nevertheless, there had been many first times in this paper, including my first Matlab program in Oct,5 2017, first practice of LaTeX article starting in Jan,27 2018, and this is my first try-to-be-academic homework written in English. The initial workload was insurmountable until luckily found some helpful resources from Internet, and eventually with greater difficulty comes greater sense of accomplishment.

Back to accumulators, personally I think it was extremity, bad luck and lack of risk awareness led to those disasters and not the financial innovation itself to be condemned. Besides the 'deconstructive' angle acquired from this case, hopefully I will continue to benefit from it as long as I bear in mind how hazardous it can be seeing only possible profit irrespective of risk.

At last I have to say I am very lucky indeed to be chosen for such an interesting topic and heartily grateful for Professor Yang, for the knowledge taught, reference provided and guidance kindly given!

References

[1] Kin Lam, LH Philip, and Ling Xin. Accumulator pricing. In *Computational Intelligence for Financial Engineering, 2009. CIFEr'09. IEEE Symposium on*, pages 72–79. IEEE, 2009.

[2] Paul Wilmott. Paul wilmott on quantitative finance. *Wiley John & Sons*, 2007.

[3] M. Rubinstein and E. Reiner. Unscrambling the binary code. In *Risk*, 1991.

[4] Deng Ding and Wenfei Wang. An accumulator pricing method based on fourier cosine series expansions. *International Journal of Financial Engineering*, 2(02):1550019, 2015.

[5] Terry H. F. Cheuk and Ton C. F Vorst. Complex barrier options. *Journal of Derivatives*, 4(1), 1998.

[6] Mark Broadie, Paul Glasserman, and Steven Kou. A continuity correction for discrete barrier options. *Mathematical Finance*, 7(4):325–349, 1997.

[7] Paolo Brandimarte. *Numerical Methods in Finance and Economics: A MATLAB®-Based Introduction.* Wiley Interscience,, 2006.