**Brownian motion with drift - law of supremum**

Thomas Nguyen Hoang Nguyen

[1. Law of 2](#_Toc72154363)

[2. Law of supremum of Brownian motion with drift 3](#_Toc72154364)

[3. Law 5](#_Toc72154365)

[4. Law 7](#_Toc72154366)

[5. Law 8](#_Toc72154367)

[5.1. Solution 1 by integration 8](#_Toc72154368)

[5.2. Solution 2 by the reflection principle (not good) 9](#_Toc72154369)

[6. Law 10](#_Toc72154370)

[~~7.~~ ~~Pricing Barrier option Up and Out~~ 11](#_Toc72154371)

[8. Law of 12](#_Toc72154372)

[References 12](#_Toc72154373)

# Law of

Part A

Par B:

Part C:

Par D:

Part E: for , make a change of variable  , then deduce easily

Proof: example

Part B

* For : apply symmetric principle (reflection principle)
* For

Part C: Deduce easily from part B

# Law of supremum of Brownian motion with drift

Remark: I found the proof in “Nicolas Privault, Notes on Stochastic Finance” (page 336)

Let and we find the joint distribution of

Where is a Brownian motion in the probability .

Make a change of measure then

So,

Where is a Brownian motion in the probability

We know the joint distribution of (ref: [link](https://math.stackexchange.com/a/1055713/195378))

Hence

Besides, the expectation of is calculated as

Example :

Use [Seeger, 2007, Expectation Propagation for Exponential Families](https://www-users.cs.umn.edu/~baner029/Teaching/Fall07/papers/epexpfam.pdf)

Remark1: We can use Expectation Propagation to compute this expectation if we can eliminate this non-exponential term

**Idea 1** (this approximation is poor!): the term becomes very small when (we remind that , then ), so it suffice to approximate when with is small)

Let approximate the function

Some more accurate approximations can be done by denoting , then

These approximations are relatively accurate for given the fact that s in the range (…) (to be confirmed)

Code Mathematica:

f[x\_, n\_] := Sum[(-1)^(i + 1)/i\*(Exp[x] - 1)^i, {i, 1, n}]

Plot[{f[x, 3]\*Exp[-(1/2)\*x^2], x\*Exp[-(1/2) x^2]}, {x, 0, 1}]



The error of approximation is (poor)

**Idea 2** Approximate by for are quadratic functions. (Approximation by gaussian radical basis function <https://arxiv.org/pdf/1806.07705.pdf> or <https://www.sciencedirect.com/science/article/pii/S0898122116302413> )

*f[x\_] := x\*Exp[-(1/2) x^2]*

*Ndata = 200;*

*x = Table[5\*N[(i + 0.001)/Ndata], {i, 0, Ndata - 1}];*

*y = f[x];*

*data = Transpose[{x, y}];*

*GaussianRadialFunction[k\_, s\_, r\_] := Exp[-(k (r - s))^2];*

*bFuncs = Flatten[*

*Table[GaussianRadialFunction[k, s, r], {k, 0.01, 1, 0.5}, {s, 0, 1,*

*1}]];*

*lm = LinearModelFit[data, bFuncs, r];*

*Show[ListPlot[data, PlotStyle -> Pink], Plot[lm[x], {x, 0, 5}],*

*Frame -> True]*



This function is very “gaussian”

Remark 2: The precision of EP can be improved if we can get rid of (but if not, it’s not a big trouble)

Remark: What about this?

I think I can find the joint distribution of this and it is an integral of exponential function

# Law

Remark: need to see (Privault, Note of Stochastic calculus)

We want to find the joint distribution of , we have

We have

For the sake of simplicity, let denote

As and , we deduce that

Hence

We remind the joint distribution of

Then

Regarding the term

Finally,

The region of integration is expressed as follows

Remark: for , we have the region of integration is

And I think I can simplify the integral to get rid of

# Law

Then

The density function is

# Law

## Solution 1 by integration

For the sake of simplicity, let denote

As and , we deduce that

Hence

Make a change of variable and for simplicity, we ignore the sign :

## Solution 2 by the reflection principle (not good)

Just remind

Let denote the stopping time

Suppose . Denote

For the first one, apply the reflection principle

For the second one

We notice that, because then for .

Besides, is a Brownian motion, then for . By consequence,

Finally

# Law

As , we deduce that

Return back to the probability

Make a change of variable and for simplicity, we ignore the sign :

We deduce the density function of

Remark: the joint distribution of can be found easily as

….

Remark: we can simplify to some kind of . But in my opinion, this simplification is only utile for digital Barrier options. For Call, Put,… It better to use the formula . Because I guess we can obtain closed form solution from that.

Remark: Use the technique that I invented to be able to use the Expectation Propagation.

# ~~Pricing Barrier option Up and Out~~

The payoff of a *Bull up-and-out knock-out Auto-callable contract* is

With

If the stopping time is in then the product is cancelled immediately and we don’t need to wonder what happen after the time (in particular, the dynamic of the process after ). So, let us decompose the payoff as

NEED TO UNDERSTAND THE PRODUCT.

* What happen if : the condition is triggered. Option holder still needs to wait until the maturity for exercising the barrier option.

So, we need to know the law of

With

The price is

The inner integrand is the price of a *Bull up-and-out knock-out barrier option* and is equal to

Remark: There are still a lot of calculation. First, I need to make a change of measure to eliminate the drift. Second, obtain the closed form solution for a classical barrier option. Finally, put all the computation together and try to find a closed form solution.

Remark: A better solution is

* Approximate by gaussian functions with my technique
* can also be decompose as follows

Hence, we reduce the problem to the computation of gaussian integrals over hyper-rectangular region.

🡺 The problem is solve!

# Law of Bull up-and-out knock-out Auto-callable contract

We want to compute the probability of a *Bull up-and-out knock-out Auto-callable contract*

If for all, then it’s easy.

If for all, then

For each term , apply the reflection principle (this principle is apply only if )

Hence

# References

|  |  |
| --- | --- |
| [1] | P. Carr, "Two extensions to barrier option valuation," 1995. |
| [2] | J. P. Cunningham, P. Hennig and S. Lacoste-Julien, "Gaussian Probabilities and Expectation Propagation," 2011. |