Group A&B: Exact Pricing Method

✧ **Output from execution of code.**

A. Exact Solutions of One-Factor Plain Options

a) Batch1 to Batch4 option pricing:

```
Batch1:
Call Price: 2.13337 Put Price: 5.84628
Batch2
Call Price: 7.96557 Put Price: 7.96557
Batch3
Call Price: 0.204058 Put Price: 4.07326
Batch4
Call Price: 92.1757 Put Price: 1.2475
```

b) Two ways for each Batch to check put call parity.
First line in each Batch is using the put-call parity with put(call), calculate call(put) price.
Second line is using the BS option price result to validate Put-Call parity.

```
Put/Call Parity:
Batch1:
With call price: 2.13337, Put price calculated is: 5.84628
Put/Call Parity Satisfied
Batch2:
With call price: 7.96557, Put price calculated is: 7.96557
Put/Call Parity Satisfied
Batch3:
With Put price: 4.07326, Call price calculated is: 0.204058
Put/Call Parity Satisfied
Batch4:
With Put price: 1.2475, Call price calculated is: 92.1757
Put/Call Parity Satisfied
```

c) Vector stock price input
This part stock price is increased from 55 to 65 by an increment of 1. Other parameters stay the same as Batch1, where option type is set as a call. When S = 60, the call option price is 2.13337, verified with Batch 1 call price.

```
Batch 1 call option price with increasing S from 55 to 65:
0.76652 0.965684 1.19971 1.47106 1.78175 2.13337 2.52699 2.96317 3.44196 3.96293
```

d) Matrix input
Matrix follows the vector sequence S, T, r, b, K, sig, option type.
S vector ranges from 55 to 65 by an increment 1.
T ranges from 0.2 to 0.3 by an increment of 0.01.
Sig ranges from 0.25 to 0.35 by an increment of 0.01.

r, b, K, option type are vectors that has the same content as answer c.
When S, T, sig are the same as Batch1, verified call option price 2.13337.

```
Option Price with Matrix Inputs:
0.270294 0.479128 0.769125 1.1432 1.5996 2.13337 2.73773 3.40518 4.12824 4.89985
```

Option Sensitivities

a) Call/Put delta when parameters set as question required

```
Senesitivity a:
Call delta: 0.594629 Put delta:-0.356601
```

b) Stock vector input. The vector ranges from 96 to 110 by an increment of 1.
Using other parameters same as a), call delta is showed below.
When stock price is 105, a call delta is verified as 0.594629.

```
Sensitivity b:
Delta with stock increasing from 96 to 110, increment 1
0.463062 0.478508 0.493791 0.50889 0.523785 0.538459 0.552894 0.567076 0.580992 0.594629 0.607976 0.621025 0.633767 0.64
6196
```

c) With the same matrix input from exercise d) of previous section, calculating call option delta and option gamma.

```
Sensitivity c:
Delta with matrix input
0.0976313 0.146795 0.201922 0.259667 0.317207 0.372483 0.424191 0.471643 0.514603 0.553132
Gamma with matrix input
0.0280438 0.0344435 0.0390069 0.0416196 0.0425062 0.0420428 0.0406251 0.0386 0.0362398 0.0337425
```

d) Delta/Gamma calculated by formulae and numerical delta/gamma with shock size 1 and 0.01. Input parameters are same from Batch4, option type is set as put.

```
Sensitivity d: Numerical Delta
BS formula Delta:-0.0112394 , Gamma:0.000179578
ShockSize 1 Delta: -0.0112402 , Gamma:0.000179589
ShockSize 0.01 Delta: -0.0112394 , Gamma:0.000179578
```

B. Perpetual American Option

b) Call/Put prices are verified.

```
Perpetual American b:
Perpetual American Option Call Price: 18.5035 Put Price: 3.03106
```

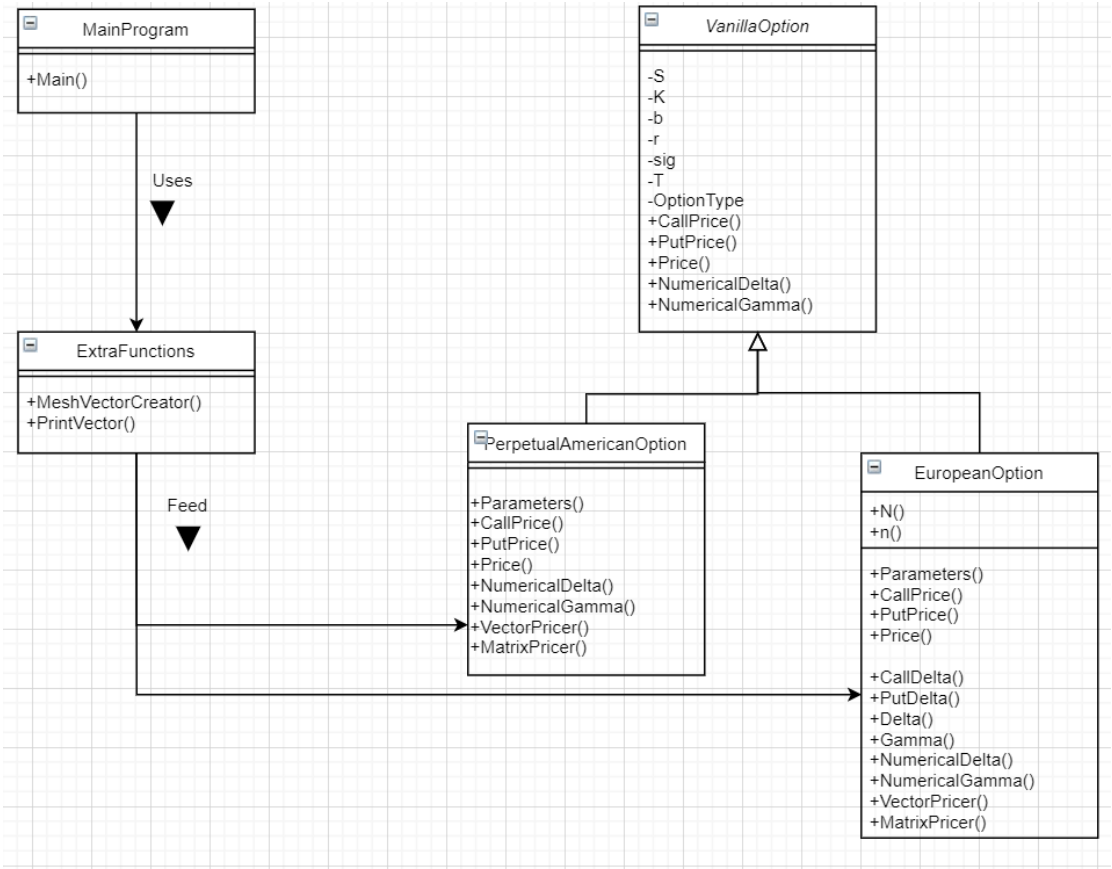
c) Vector input of S ranged from 105 to 115. Other parameters stay the same as b) in this part. Calculating put price for perpetual American put option. Put price is verified 3.03106 when S=110.

```
Perpetual American c: increasing underlying stock vector input
4.04761 3.81598 3.5996 3.39733 3.20813 3.03106 2.86523 2.70985 2.56416 2.42748
```

d) Matrix input. The matrix is in the order of vectors of S, r, b, K, sig and option type. Using the stock vector created from c) and other parts remain the same.
Put price is calculated as following. Verified with part c)

```
Perpetual American d: matrix input
4.04761 3.81598 3.5996 3.39733 3.20813 3.03106 2.86523 2.70985 2.56416 2.42748
```

✧ **Code Design**



✧ Inheritance Structure:
EuropeanOption and PerpetualAmericanOption are inherited from VanillaOption. VanillaOption has a datamember - double OptionType. 0 stands for call, all number else for put.

1. EuropeanOption inherits all the data member and member function from VanillaOption.
   Private:
   - N() is the CDF that return the cumulative probability with d1

calculated from data members.

- n() is the PDF that return the probability density with d1 calculated from data members.

Public:

- Parameters() modifies data member, input data are in the order (S, T, r, b, K, sig, OptionType)
- CallPrice() uses BS formulae to calculate call option premium with all data member other than OptionType. PutPrice() uses the same logic as CallPrice(). Price() acknowledges OptionType and determine either CallPrice() or PutPrice() to use.
- Delta(), CallDelta(), PutDelta() uses the same logic as Price().
- Gamma are the same for calls and puts.
- Numerical delta and gamma functions are designed to shock member data S up and down by h. Shocked option price are generated by Price() function (numerical functions acknowledge OptionType). After calculating numerical delta and gamma by Taylor Expansion formulae, set member data to before shock level in case messing up future usage.
- VectorPricer() takes in a stock price vector, and a metric index and return a vector corresponding to the metric index(0-price,1-delta, else gamma). Note after using this function, parameters are not the same. This function can be further expanded to take any parameter vector, adding a parameter index.
- MatrixPricer() takes in matrix. The matrix is a vector of vector. The encapsulated vectors are in the order of S, T, r, b, K, sig, OptionType. Note after using this function, parameters are not the same. Metric index logic is the same as VectorPricer().

2. PerpetualAmericanOption inherits all member functions and all data member expect for maturity time T.

   Public:

   - Parameters() modifies data member, input data are in the order (S, r, b, K, sig, OptionType)
   - Price(), CallPrice(), PutPrice() use different formulae but the same logic as in EuropeanOption.
   - NumericalDelta() and NumericalGamma() use the same logic and formulae as in EuropeanOption.
   - VectorPricer() takes in a vector of S and return the option price vector.
   - MatrixPricer() takes in a matrix(in the order of S, r, b, K, sig, OptionType) and return the option price vector.

✧ ExtraFunctions contain two global functions

1. MeshVectorCreator() takes in a low number(lbound), a high number (ubound), and a increment of h. It creates a vector starting from lbound, ending in

ubound, with increment of h. User should note this function doesn't apply to h=0 for now. Function converges when last number of vector is within (-0.000001+ubound, ubound), user should be aware of input digit length.

2. PrintVector() prints the input vector, elements separate by ",".