**Ideas Behind the Design:**

I created a base class, Option, and 2 derived class, which are EuropeanPutCallOption and PerpetualAmericanOption. This is because both derived class are options, and they share certain identical properties, member data, and member functions. In base class, it encapsulates member data like r, K, T, Underlaying Price, sig, and b. It also has a struct called OptionData to store a struct of parameters for initialization purpose, an enum class called OptionType, and another enum class called ParamType. These will be inherited by the derived class, since both European and American options will need these properties. The base class has 2 virtual functions, which are toggle() and Price(). This ensures all the derived option classes will at least have these functionalities, because these 2 functions are necessary for any specific option. The base class has 2 private functions, which are init() and copy(). This is because the derived class will not need to access these functions directly. They will delegate the base class to call these functions when initializing their own instances. For both EuropeanPutCallOption and PerpetualAmericanOption class, I created 6 constructors to make it more flexible when initialization. It is possible to pass in a struct of parameters, to directly pass in parameters, to pass in OptionType, or to pass in nothing to initialize instances. For pricing functions, I created 2 private functions for both derived class, which are Call Price and PutPrice. These 2 functions should not be accessed by the user. All the other public Price functions will use one of these 2 private functions to calculate the option price. I also defined global pricing functions. This is because sometimes the user may want to use a function to calculate prices directly, without creating an instance. Global functions can help to achieve this purpose. The 2 private pricing functions mentioned earlier will use this global Price function. For matrix pricer, I defined 2 versions of them. The first one accepts a matrix of size P\*N parameters, where P is the types of parameters (e.g. K, T, r, …) and N is the number of value of each parameter. It will return a matrix of size P\*N of corresponding option values. Each parameter is independent from others, meaning the function will hold other parameters unchanged while changing a single current parameter. The second version specifically calculate the relationship between 2 types of parameter. The input is a matrix of dimension 2 \* N, where N is the number of value of each parameters. The output is a matrix of N \* N, which consists the option value of different combinations between these 2 types of parameter. The Greek Matrix Pricer follows the same structure, where there are 2 versions of them, each emphasis on different purposes the user may want. For Greek functions, I also made both member and global functions, since the user may want to call it on an option object, or may want to call it directly without creating instances. All formula implementations are only inside these global functions, and member functions will simply use the global functions to calculate values. I also defined a generateMesh function inside the base class, so later on when we can use this function as long as we include the header file of the base class.

**A. European Option**

**Exact Solutions:**

**a) & b).**

European Call Option's parameters: T: 0.25, K: 65, Vol: 0.3, r: 0.08, S: 60, b: 0.08  
 Call Value (using Black-Scholes): 2.13337; Put Value (using Put-Call Parity): 5.84628  
 Put Value (using Black-Scholes): 5.84628; Call Value (using Put-Call Parity): 2.13337

European Call Option's parameters: T: 1, K: 100, Vol: 0.2, r: 0, S: 100, b: 0  
 Call Value (using Black-Scholes): 7.96557; Put Value (using Put-Call Parity): 7.96557  
 Put Value (using Black-Scholes): 7.96557; Call Value (using Put-Call Parity): 7.96557

European Call Option's parameters: T: 1, K: 10, Vol: 0.5, r: 0.12, S: 5, b: 0.12  
 Call Value (using Black-Scholes): 0.204058; Put Value (using Put-Call Parity): 4.07326  
 Put Value (using Black-Scholes): 4.07326; Call Value (using Put-Call Parity): 0.204058

European Call Option's parameters: T: 30, K: 100, Vol: 0.3, r: 0.08, S: 100, b: 0.08  
 Call Value (using Black-Scholes): 92.1757; Put Value (using Put-Call Parity): 1.2475  
 Put Value (using Black-Scholes): 1.2475; Call Value (using Put-Call Parity): 92.1757

All the values calculated using either the Black-Scholes function, or the Put-Call Parity, are all correct. These consistent values between BS and Put-Call Parity indicates the implementations are correct, and that plain European options satisfies this parity.

**c).**

Generate a mesh of Stock Prices and calculate option prices:

European Call Option's parameters: T: 0.25, K: 65, Vol: 0.3, r: 0.08, S: 0, b: 0.08

Stock Price = 10, European Call Option Value: 7.792e-36  
Stock Price = 11, European Call Option Value: 1.88964e-32  
Stock Price = 12, European Call Option Value: 1.64204e-29  
Stock Price = 13, European Call Option Value: 6.19303e-27  
Stock Price = 14, European Call Option Value: 1.17207e-24  
Stock Price = 15, European Call Option Value: 1.24529e-22  
Stock Price = 16, European Call Option Value: 8.11387e-21  
Stock Price = 17, European Call Option Value: 3.47944e-19  
Stock Price = 18, European Call Option Value: 1.03991e-17  
Stock Price = 19, European Call Option Value: 2.27042e-16  
Stock Price = 20, European Call Option Value: 3.76503e-15  
Stock Price = 21, European Call Option Value: 4.89964e-14  
Stock Price = 22, European Call Option Value: 5.1436e-13  
Stock Price = 23, European Call Option Value: 4.45934e-12  
Stock Price = 24, European Call Option Value: 3.25772e-11  
Stock Price = 25, European Call Option Value: 2.0405e-10  
Stock Price = 26, European Call Option Value: 1.11246e-09  
Stock Price = 27, European Call Option Value: 5.34891e-09  
Stock Price = 28, European Call Option Value: 2.29454e-08  
Stock Price = 29, European Call Option Value: 8.8713e-08  
Stock Price = 30, European Call Option Value: 3.11926e-07  
Stock Price = 31, European Call Option Value: 1.00545e-06  
Stock Price = 32, European Call Option Value: 2.99241e-06  
Stock Price = 33, European Call Option Value: 8.27565e-06  
Stock Price = 34, European Call Option Value: 2.13896e-05  
Stock Price = 35, European Call Option Value: 5.19359e-05  
Stock Price = 36, European Call Option Value: 0.000119024  
Stock Price = 37, European Call Option Value: 0.000258548  
Stock Price = 38, European Call Option Value: 0.000534403  
Stock Price = 39, European Call Option Value: 0.00105474  
Stock Price = 40, European Call Option Value: 0.00199417  
Stock Price = 41, European Call Option Value: 0.00362244  
Stock Price = 42, European Call Option Value: 0.00633924  
Stock Price = 43, European Call Option Value: 0.010714  
Stock Price = 44, European Call Option Value: 0.0175282  
Stock Price = 45, European Call Option Value: 0.0278174  
Stock Price = 46, European Call Option Value: 0.0429083  
Stock Price = 47, European Call Option Value: 0.0644467  
Stock Price = 48, European Call Option Value: 0.0944125  
Stock Price = 49, European Call Option Value: 0.135117  
Stock Price = 50, European Call Option Value: 0.189181

The mesh is generated by the global function in Option class. As we can see, the call option value increases when the stock price increases. This indicates that the implementation is correct.

**d).**

Testing on 2 versions of Matrix Pricer on variable T and Sig:  
  
Testing Matrix Pricer Version 1:  
European Call Option's parameters: T: 0.25, K: 65, Vol: 0.3, r: 0.08, S: 60, b: 0.08  
  
Varying S from 50 to 100: 0.189181 1.23614 4.11941 9.01336 15.2491 22.1011 29.1665 36.2916  
Varying K from 50 to 70: 11.3229 8.91421 6.77166 4.9542 3.48835 2.36479 1.54526 0.97498  
Varying r from 0.02 to 0.1: 2.16561 2.15943 2.15327 2.14713 2.141 2.13489 2.1288 2.12273  
Varying Sig from 0.1 to 0.8: 0.173097 1.04145 2.13337 3.29033 4.47244 5.66509 6.86147 8.05775  
Varying T from 0.25 to 1.05: 2.13337 3.04852 3.88708 4.66932 5.40789 6.11119 6.78514 7.43404  
Varying b from 0.02 to 0.1: 1.8173 1.8746 1.93324 1.99326 2.05465 2.11745 2.18166 2.2473  
  
Testing Matrix Pricer Version 2:  
European Call Option's parameters: T: 0.25, K: 65, Vol: 0.3, r: 0.08, S: 60, b: 0.08  
Row: Sig varying from 0.3 to 0.7  
Column: T varying from 0.25 to 1.05  
  
2.13337 3.29033 4.47244 5.66509 6.86147  
3.68335 5.28445 6.88985 8.49199 10.0866  
5.04346 6.97128 8.89048 10.7961 12.6843  
6.28224 8.47195 10.6438 12.7929 14.915  
7.43404 9.84251 12.2259 14.5782 16.8938

Here are the 2 versions of Matrix Pricer. They all takes a matrix as input and output a matrix of option values. Version 1 takes arbitrary number of types of parameters, and Version 2 specifically focus on the relationship between 2 types of parameters.

**Greeks & Option Sensitivities:**

**a).**

Testing Greeks Formulas:  
European Call Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 105, b: 0  
Call's Delta (using member function): 0.594629  
Call's Delta (using Global Function): 0.594629

European Put Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 105, b: 0  
Put's Delta (using member function): -0.356601  
Put's Delta (using Global Function): -0.356601

The values calculated are correct.

**b).**

Generate a mesh of Stock Prices and calculate Delta:  
European Call Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 0, b: 0  
Stock Price: 10, Call's Delta: 2.25551e-19  
Stock Price: 11, Call's Delta: 6.18174e-18  
Stock Price: 12, Call's Delta: 1.12536e-16  
Stock Price: 13, Call's Delta: 1.46673e-15  
Stock Price: 14, Call's Delta: 1.44882e-14  
Stock Price: 15, Call's Delta: 1.1336e-13  
Stock Price: 16, Call's Delta: 7.27491e-13  
Stock Price: 17, Call's Delta: 3.93787e-12  
Stock Price: 18, Call's Delta: 1.83921e-11  
Stock Price: 19, Call's Delta: 7.55191e-11  
Stock Price: 20, Call's Delta: 2.76878e-10  
Stock Price: 21, Call's Delta: 9.18317e-10  
Stock Price: 22, Call's Delta: 2.78591e-09  
Stock Price: 23, Call's Delta: 7.80377e-09  
Stock Price: 24, Call's Delta: 2.0348e-08  
Stock Price: 25, Call's Delta: 4.97346e-08  
Stock Price: 26, Call's Delta: 1.14647e-07  
Stock Price: 27, Call's Delta: 2.50577e-07  
Stock Price: 28, Call's Delta: 5.21714e-07  
Stock Price: 29, Call's Delta: 1.03903e-06  
Stock Price: 30, Call's Delta: 1.98667e-06  
Stock Price: 31, Call's Delta: 3.65877e-06  
Stock Price: 32, Call's Delta: 6.5092e-06  
Stock Price: 33, Call's Delta: 1.1216e-05  
Stock Price: 34, Call's Delta: 1.87624e-05  
Stock Price: 35, Call's Delta: 3.05351e-05  
Stock Price: 36, Call's Delta: 4.84406e-05  
Stock Price: 37, Call's Delta: 7.50367e-05  
Stock Price: 38, Call's Delta: 0.00011368  
Stock Price: 39, Call's Delta: 0.000168682  
Stock Price: 40, Call's Delta: 0.000245471  
Stock Price: 41, Call's Delta: 0.000350761  
Stock Price: 42, Call's Delta: 0.000492699  
Stock Price: 43, Call's Delta: 0.000681019  
Stock Price: 44, Call's Delta: 0.000927157  
Stock Price: 45, Call's Delta: 0.00124435  
Stock Price: 46, Call's Delta: 0.00164771  
Stock Price: 47, Call's Delta: 0.00215423  
Stock Price: 48, Call's Delta: 0.00278276  
Stock Price: 49, Call's Delta: 0.00355398  
Stock Price: 50, Call's Delta: 0.00449025

Here we also used the global function in Option class to generate the mesh.

**c).**

Testing Greeks Matrix Pricer:  
  
Test Greek Matrix Pricer Version 1:  
European Call Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 105, b: 0  
Varying T from 0.5 to 5: 0.594629 0.557693 0.530158 0.504647 0.480137 0.456437 0.433532 0.411443  
Varying Sig from 0.36 to 0.8: 0.594629 0.592366 0.592766 0.594912 0.598257 0.602456 0.607279 0.612564  
Varying K from 100 to 110: 0.594629 0.574362 0.554086 0.533861 0.513744 0.493791 0.474052 0.454576  
Varying r from 0.1 to 0.5: 0.594629 0.57788 0.561602 0.545784 0.530411 0.51547 0.500951 0.486841  
  
Test Greek Matrix Pricer Version 2:  
European Call Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 105, b: 0  
Row: Sig varying from 0.36 to 0.8  
Column: T varying from 0.5 to 5  
0.594629 0.592366 0.592766 0.594912 0.598257 0.602456 0.607279 0.612564  
0.557693 0.56267 0.569255 0.576856 0.585119 0.593818 0.602799 0.611953  
0.530158 0.538591 0.548145 0.55836 0.568958 0.579759 0.590636 0.601499  
0.504647 0.51512 0.526392 0.538079 0.549947 0.56184 0.573646 0.585285  
0.480137 0.491872 0.504164 0.516681 0.529214 0.541622 0.553807 0.565693  
0.456437 0.468941 0.481807 0.494739 0.507549 0.520107 0.532324 0.544129  
0.433532 0.446466 0.4596 0.472664 0.485488 0.497952 0.509972 0.521486  
0.411443 0.424565 0.437747 0.450745 0.4634 0.475602 0.487274 0.498361

As mentioned above, we also have 2 versions of Greek Matrix Pricer, each focusing on different perspectives.

**d).**

Testing Approximate Delta:  
European Call Option's parameters: T: 0.5, K: 100, Vol: 0.36, r: 0.1, S: 105, b: 0  
  
Exact Delta: 0.594629  
h: 0.01, Approx.Delta (member function): 0.594629, Approx.Delta (global function):0.594629  
h: 0.2098, Approx.Delta (member function): 0.594627, Approx.Delta (global function):0.594627  
h: 0.4096, Approx.Delta (member function): 0.594621, Approx.Delta (global function):0.594621  
h: 0.6094, Approx.Delta (member function): 0.594611, Approx.Delta (global function):0.594611  
h: 0.8092, Approx.Delta (member function): 0.594597, Approx.Delta (global function):0.594597  
h: 1.009, Approx.Delta (member function): 0.59458, Approx.Delta (global function):0.59458  
h: 1.2088, Approx.Delta (member function): 0.594558, Approx.Delta (global function):0.594558  
h: 1.4086, Approx.Delta (member function): 0.594533, Approx.Delta (global function):0.594533  
h: 1.6084, Approx.Delta (member function): 0.594504, Approx.Delta (global function):0.594504  
h: 1.8082, Approx.Delta (member function): 0.594471, Approx.Delta (global function):0.594471  
h: 2.008, Approx.Delta (member function): 0.594434, Approx.Delta (global function):0.594434  
h: 2.2078, Approx.Delta (member function): 0.594394, Approx.Delta (global function):0.594394  
h: 2.4076, Approx.Delta (member function): 0.594349, Approx.Delta (global function):0.594349  
h: 2.6074, Approx.Delta (member function): 0.594301, Approx.Delta (global function):0.594301  
h: 2.8072, Approx.Delta (member function): 0.594249, Approx.Delta (global function):0.594249  
h: 3.007, Approx.Delta (member function): 0.594193, Approx.Delta (global function):0.594193  
h: 3.2068, Approx.Delta (member function): 0.594134, Approx.Delta (global function):0.594134  
h: 3.4066, Approx.Delta (member function): 0.59407, Approx.Delta (global function):0.59407  
h: 3.6064, Approx.Delta (member function): 0.594003, Approx.Delta (global function):0.594003  
h: 3.8062, Approx.Delta (member function): 0.593932, Approx.Delta (global function):0.593932  
h: 4.006, Approx.Delta (member function): 0.593857, Approx.Delta (global function):0.593857  
h: 4.2058, Approx.Delta (member function): 0.593779, Approx.Delta (global function):0.593779  
h: 4.4056, Approx.Delta (member function): 0.593696, Approx.Delta (global function):0.593696  
h: 4.6054, Approx.Delta (member function): 0.59361, Approx.Delta (global function):0.59361  
h: 4.8052, Approx.Delta (member function): 0.593521, Approx.Delta (global function):0.593521  
h: 5.005, Approx.Delta (member function): 0.593427, Approx.Delta (global function):0.593427  
h: 5.2048, Approx.Delta (member function): 0.59333, Approx.Delta (global function):0.59333  
h: 5.4046, Approx.Delta (member function): 0.593229, Approx.Delta (global function):0.593229  
h: 5.6044, Approx.Delta (member function): 0.593124, Approx.Delta (global function):0.593124  
h: 5.8042, Approx.Delta (member function): 0.593016, Approx.Delta (global function):0.593016  
h: 6.004, Approx.Delta (member function): 0.592904, Approx.Delta (global function):0.592904  
h: 6.2038, Approx.Delta (member function): 0.592789, Approx.Delta (global function):0.592789  
h: 6.4036, Approx.Delta (member function): 0.592669, Approx.Delta (global function):0.592669  
h: 6.6034, Approx.Delta (member function): 0.592546, Approx.Delta (global function):0.592546  
h: 6.8032, Approx.Delta (member function): 0.59242, Approx.Delta (global function):0.59242  
h: 7.003, Approx.Delta (member function): 0.59229, Approx.Delta (global function):0.59229  
h: 7.2028, Approx.Delta (member function): 0.592156, Approx.Delta (global function):0.592156  
h: 7.4026, Approx.Delta (member function): 0.592019, Approx.Delta (global function):0.592019  
h: 7.6024, Approx.Delta (member function): 0.591878, Approx.Delta (global function):0.591878  
h: 7.8022, Approx.Delta (member function): 0.591734, Approx.Delta (global function):0.591734  
h: 8.002, Approx.Delta (member function): 0.591586, Approx.Delta (global function):0.591586  
h: 8.2018, Approx.Delta (member function): 0.591435, Approx.Delta (global function):0.591435  
h: 8.4016, Approx.Delta (member function): 0.59128, Approx.Delta (global function):0.59128  
h: 8.6014, Approx.Delta (member function): 0.591122, Approx.Delta (global function):0.591122  
h: 8.8012, Approx.Delta (member function): 0.59096, Approx.Delta (global function):0.59096  
h: 9.001, Approx.Delta (member function): 0.590795, Approx.Delta (global function):0.590795  
h: 9.2008, Approx.Delta (member function): 0.590626, Approx.Delta (global function):0.590626  
h: 9.4006, Approx.Delta (member function): 0.590454, Approx.Delta (global function):0.590454  
h: 9.6004, Approx.Delta (member function): 0.590279, Approx.Delta (global function):0.590279  
h: 9.8002, Approx.Delta (member function): 0.5901, Approx.Delta (global function):0.5901  
h: 10, Approx.Delta (member function): 0.589918, Approx.Delta (global function):0.589918  
  
  
Exact Gamma: 0.0134936  
h: 0.01, Approx.Gamma (member function): 0.0134936, Approx.Gamma (global function):0.0134936  
h: 0.2098, Approx.Gamma (member function): 0.0134936, Approx.Gamma (global function):0.0134936  
h: 0.4096, Approx.Gamma (member function): 0.0134935, Approx.Gamma (global function):0.0134935  
h: 0.6094, Approx.Gamma (member function): 0.0134933, Approx.Gamma (global function):0.0134933  
h: 0.8092, Approx.Gamma (member function): 0.0134931, Approx.Gamma (global function):0.0134931  
h: 1.009, Approx.Gamma (member function): 0.0134928, Approx.Gamma (global function):0.0134928  
h: 1.2088, Approx.Gamma (member function): 0.0134924, Approx.Gamma (global function):0.0134924  
h: 1.4086, Approx.Gamma (member function): 0.013492, Approx.Gamma (global function):0.013492  
h: 1.6084, Approx.Gamma (member function): 0.0134915, Approx.Gamma (global function):0.0134915  
h: 1.8082, Approx.Gamma (member function): 0.0134909, Approx.Gamma (global function):0.0134909  
h: 2.008, Approx.Gamma (member function): 0.0134903, Approx.Gamma (global function):0.0134903  
h: 2.2078, Approx.Gamma (member function): 0.0134896, Approx.Gamma (global function):0.0134896  
h: 2.4076, Approx.Gamma (member function): 0.0134888, Approx.Gamma (global function):0.0134888  
h: 2.6074, Approx.Gamma (member function): 0.013488, Approx.Gamma (global function):0.013488  
h: 2.8072, Approx.Gamma (member function): 0.0134871, Approx.Gamma (global function):0.0134871  
h: 3.007, Approx.Gamma (member function): 0.0134862, Approx.Gamma (global function):0.0134862  
h: 3.2068, Approx.Gamma (member function): 0.0134851, Approx.Gamma (global function):0.0134851  
h: 3.4066, Approx.Gamma (member function): 0.013484, Approx.Gamma (global function):0.013484  
h: 3.6064, Approx.Gamma (member function): 0.0134829, Approx.Gamma (global function):0.0134829  
h: 3.8062, Approx.Gamma (member function): 0.0134816, Approx.Gamma (global function):0.0134816  
h: 4.006, Approx.Gamma (member function): 0.0134803, Approx.Gamma (global function):0.0134803  
h: 4.2058, Approx.Gamma (member function): 0.013479, Approx.Gamma (global function):0.013479  
h: 4.4056, Approx.Gamma (member function): 0.0134776, Approx.Gamma (global function):0.0134776  
h: 4.6054, Approx.Gamma (member function): 0.0134761, Approx.Gamma (global function):0.0134761  
h: 4.8052, Approx.Gamma (member function): 0.0134745, Approx.Gamma (global function):0.0134745  
h: 5.005, Approx.Gamma (member function): 0.0134729, Approx.Gamma (global function):0.0134729  
h: 5.2048, Approx.Gamma (member function): 0.0134712, Approx.Gamma (global function):0.0134712  
h: 5.4046, Approx.Gamma (member function): 0.0134694, Approx.Gamma (global function):0.0134694  
h: 5.6044, Approx.Gamma (member function): 0.0134676, Approx.Gamma (global function):0.0134676  
h: 5.8042, Approx.Gamma (member function): 0.0134657, Approx.Gamma (global function):0.0134657  
h: 6.004, Approx.Gamma (member function): 0.0134637, Approx.Gamma (global function):0.0134637  
h: 6.2038, Approx.Gamma (member function): 0.0134617, Approx.Gamma (global function):0.0134617  
h: 6.4036, Approx.Gamma (member function): 0.0134595, Approx.Gamma (global function):0.0134595  
h: 6.6034, Approx.Gamma (member function): 0.0134574, Approx.Gamma (global function):0.0134574  
h: 6.8032, Approx.Gamma (member function): 0.0134551, Approx.Gamma (global function):0.0134551  
h: 7.003, Approx.Gamma (member function): 0.0134528, Approx.Gamma (global function):0.0134528  
h: 7.2028, Approx.Gamma (member function): 0.0134504, Approx.Gamma (global function):0.0134504  
h: 7.4026, Approx.Gamma (member function): 0.013448, Approx.Gamma (global function):0.013448  
h: 7.6024, Approx.Gamma (member function): 0.0134455, Approx.Gamma (global function):0.0134455  
h: 7.8022, Approx.Gamma (member function): 0.0134429, Approx.Gamma (global function):0.0134429  
h: 8.002, Approx.Gamma (member function): 0.0134402, Approx.Gamma (global function):0.0134402  
h: 8.2018, Approx.Gamma (member function): 0.0134375, Approx.Gamma (global function):0.0134375  
h: 8.4016, Approx.Gamma (member function): 0.0134347, Approx.Gamma (global function):0.0134347  
h: 8.6014, Approx.Gamma (member function): 0.0134319, Approx.Gamma (global function):0.0134319  
h: 8.8012, Approx.Gamma (member function): 0.0134289, Approx.Gamma (global function):0.0134289  
h: 9.001, Approx.Gamma (member function): 0.0134259, Approx.Gamma (global function):0.0134259  
h: 9.2008, Approx.Gamma (member function): 0.0134228, Approx.Gamma (global function):0.0134228  
h: 9.4006, Approx.Gamma (member function): 0.0134197, Approx.Gamma (global function):0.0134197  
h: 9.6004, Approx.Gamma (member function): 0.0134165, Approx.Gamma (global function):0.0134165  
h: 9.8002, Approx.Gamma (member function): 0.0134132, Approx.Gamma (global function):0.0134132  
h: 10, Approx.Gamma (member function): 0.0134098, Approx.Gamma (global function):0.0134098

I used both the member function and global function to calculate Approximate Delta/Gamma. The results indicate that when the stepsize is small, the value is the same as the exact value. However, when stepsize gets larger and larger, the value will start to deviate from the exact result.

**B. Perpetual American Option**

**a).** The formulas were implemented in the PerpetualAmericanOption derived class.

**b).**

Testing Perpetual American Option Pricing formula:  
American Call Option's parameters: K: 100, Vol: 0.1, r: 0.1, S: 110, b: 0.02  
American Call Option Value(Using member function): 18.5035  
American Call Option Value(Using global function): 18.5035  
  
American Put Option's parameters: K: 100, Vol: 0.1, r: 0.1, S: 110, b: 0.02  
American Put Option Value(Using member function): 3.03106  
American Put Option Value(Using global function): 3.03106

The results calculate matches with the pdf result. This shows that the formula implemented are correct. Special cases like when y1 = 1 or y2 = 0 are also handled in the pricing function.

**c).**

Generate a mesh of Stock Price and calculate Option value:  
American Call Option's parameters: K: 100, Vol: 0.1, r: 0.1, S: 0, b: 0.02  
Stock Price: 10, American Call's value: 0.00826235  
Stock Price: 11, American Call's value: 0.011227  
Stock Price: 12, American Call's value: 0.0148535  
Stock Price: 13, American Call's value: 0.0192158  
Stock Price: 14, American Call's value: 0.0243891  
Stock Price: 15, American Call's value: 0.03045  
Stock Price: 16, American Call's value: 0.0374762  
Stock Price: 17, American Call's value: 0.0455465  
Stock Price: 18, American Call's value: 0.054741  
Stock Price: 19, American Call's value: 0.0651405  
Stock Price: 20, American Call's value: 0.076827  
Stock Price: 21, American Call's value: 0.0898835  
Stock Price: 22, American Call's value: 0.104394  
Stock Price: 23, American Call's value: 0.120442  
Stock Price: 24, American Call's value: 0.138115  
Stock Price: 25, American Call's value: 0.157497  
Stock Price: 26, American Call's value: 0.178677  
Stock Price: 27, American Call's value: 0.201742  
Stock Price: 28, American Call's value: 0.226781  
Stock Price: 29, American Call's value: 0.253883  
Stock Price: 30, American Call's value: 0.283138  
Stock Price: 31, American Call's value: 0.314637  
Stock Price: 32, American Call's value: 0.348471  
Stock Price: 33, American Call's value: 0.384732  
Stock Price: 34, American Call's value: 0.423512  
Stock Price: 35, American Call's value: 0.464906  
Stock Price: 36, American Call's value: 0.509007  
Stock Price: 37, American Call's value: 0.555908  
Stock Price: 38, American Call's value: 0.605706  
Stock Price: 39, American Call's value: 0.658495  
Stock Price: 40, American Call's value: 0.714373  
Stock Price: 41, American Call's value: 0.773434  
Stock Price: 42, American Call's value: 0.835777  
Stock Price: 43, American Call's value: 0.901499  
Stock Price: 44, American Call's value: 0.970699  
Stock Price: 45, American Call's value: 1.04347  
Stock Price: 46, American Call's value: 1.11993  
Stock Price: 47, American Call's value: 1.20015  
Stock Price: 48, American Call's value: 1.28425  
Stock Price: 49, American Call's value: 1.37233  
Stock Price: 50, American Call's value: 1.46448

I generate the mesh using the global function defined in the Option class. We observe that when stock price increases, the option value increases as well. This implies the results are correct.

**d).**

Testing Matrix Pricer Version 1:  
American Call Option's parameters: K: 100, Vol: 0.1, r: 0.1, S: 110, b: 0.02  
  
Varying S from 50 to 110: 1.46448 2.43636 3.78054 5.5643 7.85679 10.7289 14.2532 18.5035  
Varying K from 50 to 100: 86.027 63.9833 49.279 39.0138 31.5828 26.042 21.8076 18.5035  
Varying r from 0.02 to 0.1: 110 48.2521 34.7327 28.2459 24.3751 21.7826 19.916 18.5035  
Varying Sig from 0.1 to 0.8: 18.5035 26.5984 34.6188 42.1321 48.9929 55.158 60.6398 65.4822  
Varying b from 0.02 to 0.1: 18.5035 23.0977 28.9856 36.4052 45.8253 58.1941 75.7681 110  
  
Testing Matrix Pricer Version 2:  
American Call Option's parameters: K: 50, Vol: 0.1, r: 0.1, S: 110, b: 0.02  
Row: K varying from 50 to 110  
Column: Sig varying from 0.1 to 0.5  
  
86.027 48.0865 30.3461 20.732 14.9791  
62.654 45.3009 35.0463 28.3394 23.6424  
60.0309 48.7401 41.3307 36.0581 32.0952  
61.8669 53.494 47.6788 43.3471 39.9642  
65.0119 58.4102 53.6643 50.0294 47.1237

Here are the 2 versions of matrix pricer. They focus on different perspectives or preferences the user may have. We observe in Version 1, when K = 100, the price matches the exact value in the pdf. In Version 2, the value when K = 50 matches Version 1’s value when K = 50. This indicates the functionality is correct.