Function coded: makeDepoCurve.m, getRateIntegral.m, makeFwdCurve.m, getFwdSpot.m

Testing scripts: testGetFwdSpot.m, testGetRateIntegral.m

The development of this derivatives pricing model is more than a project which improves my understanding of numerical methods, it’s also a valuable opportunity for me to work in a team and learn about the architecture and framework of a pricing model.

At first there was an issue for me in the original test of the getRateIntegral function. it had a precision problem that the result of my function was different from the original discount factors, about . After reviewing the lecture notes, I removed all the unnecessary division method and saved the result of some calculations, which improved the precision and efficiency of my code.

I also tried to look through the whole project and helped to improve other parts. By analyzing the Function Call Stack and the spent time of each functions, I found our program spent lots of time on the interpolation part, and the integral method we used also caused the precision problem. By applying the advanced built-in function instead of our code, we have succeeded in improving the spent time from 20s to 1s for project.m, and also solving the precision problem. I learned “Don't Reinvent the Wheel” from this lesson.

In addition, from this experience I have learned the importance and benefits of using GitHub for a team development. It is a powerful tool to help with code management. Besides, writing the testing scripts enables me to realize the importance of software testing. These scripts helped me to find out problems in my code. Finally, I have learned more about how to design the architecture and framework of a project, it’s like a work of art, and it’s the soul of a project. I have understanded this point in this project and C++ project.