Project 1

Ryan Sephton
ID:1524789

March 17, 2020

1 Forward Rates

To implement interest rate derivatives, we first need to formalise how we can calculate the forward rates that the derivatives will depend on. We start by invoking the principle of no-arbitrage, which states that any deposit of money held over an interval of time $t_1 - t_0$ at an interest rate $r_{[0,1]}$, which is then re-deposited for a further interval of time $t_2 - t_1$ at a forward rate $f_{[1,2]}$, must be equivalent to the same, time discounted, deposit being held at an interest rate $r_{[0,2]}$. Over these intervals, the deposits may be subject to compounding every ΔT days per year. Formalising this we may write,

$$\left[1 + r_{[0,1]}\frac{\Delta T}{365}\right]^{a(t_0,t_1)}\left[1 + f_{[1,2]}\frac{\Delta T}{365}\right]^{a(t_1,t_2)} = \left[1 + r_{[0,2]}\frac{\Delta T}{365}\right]^{a(t_0,t_2)}, \tag{1}$$

$$\therefore \left[1 + f_{[1,2]}\frac{\Delta T}{365}\right]^{a(t_1,t_2)} = \frac{\left[1 + r_{[0,2]}\frac{\Delta T}{365}\right]^{a(t_0,t_2)}}{\left[1 + r_{[0,1]}\frac{\Delta T}{365}\right]^{a(t_0,t_1)}}, \tag{2}$$

$$\therefore f_{[1,2]} = N\left[\left(\frac{\left[1 + \frac{r_{[0,2]}}{N}\right]^{a(t_0,t_2)}}{\left[1 + \frac{r_{[0,1]}}{N}\right]^{a(t_0,t_1)}}\right)^{\frac{1}{a(t_1,t_2)}} - 1\right]. \tag{3}$$

We now have an expression for the forward rate $f_{[1,2]}$ in terms of the corresponding interest rates and compounding frequency and we have introduced the number of compounding terms per year $N = \frac{365}{\Delta T}$, as well as the number of compounding terms over the interval $a(t_i, t_j) = \text{trunc}\frac{N(t_j - t_i)}{365}$. This formula reduces heavily, if we instead introduce 'price' notation P_i, which physically corresponds to the amount of money one would have at time t_i if a unit deposit was made at time t_0 at interest rate $r_{[0,i]}$. Using this we have,

$$P_i = \frac{1}{(1 + \frac{r_{[0,i]}}{N})^{a(t_0,t_i)}}, \tag{4}$$

hence we may write,

$$f_{[1,2]} = N\left[\left(\frac{P_1}{P_2}\right)^{\frac{1}{a(t_1,t_2)}} - 1\right]. \tag{5}$$

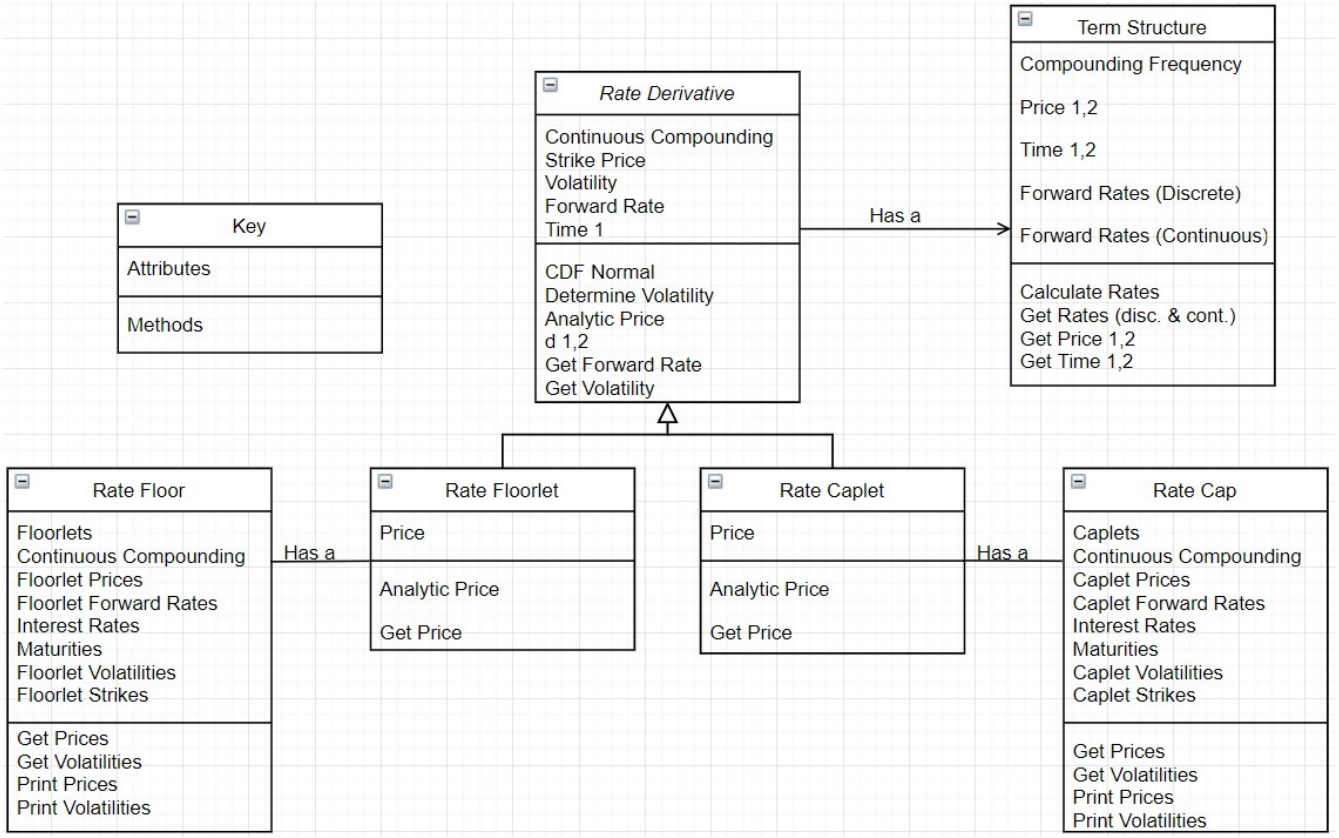
In the case of continuous compounding, we may take the limit $\lim_{N\to\infty}$ or alternatively $\lim_{\Delta T\to 0}$. Doing so and recalling the result

$$\lim_{N\to\infty}\left[1 + \frac{x}{n}\right]^n \equiv e^x, \tag{6}$$

we can now express (1) in the form

$$e^{r_{[0,1]}}e^{f_{[1,2]}(t_2 - t_1)} = e^{r_{[0,2]}}, \tag{7}$$

Figure 1: A structural representation of the project. Composition is shown with a 'has a' written above the connecting lines, whilst inheritance is shown with an arrow headed line and a vertical shift.



which can be solved for the forward rate once again, to yield

$$f_{[1,2]} = \frac{r_{[0,2]}t_2 - r_{[0,1]}t_1}{t_2 - t_1}, \tag{8}$$

$$f_{[1,2]} = \frac{\ln\left(\frac{P_1}{P_2}\right)}{t_2 - t_1}, \tag{9}$$

where in the last line we have re-introduced price notation, which is the calculation implemented in the code.

2 Program Structure

For this project, I have used a mixture of composition and inheritance, which can be seen in figure 1. The Rate Derivative class implements a pure virtual pricing function that is overwritten by both Rate Caplet and Rate Floorlet. This allows all of the calculations and numerical solvers to be written once only in the parent class and then the objective function will be updated based on which constructor is called at run time.

3 Determining the volatility of the options

When the user enters various data about the option including the fair price, strike, interest rates and corresponding times the program numerically determines the volatility of the option based on linearly interpolating the analytic price. The objective function $\hat{C}(v)$ used here is the fitted price, parametrised by the volatility v, minus the user submitted price. Linear interpolation then iterates through the closed interval [0,1], to find the volatility that corresponds to this price, within a user defined margin of error (tolerance). To start the

algorithm, the only requirement is that the objective function can be bounded from above and below i.e. $\hat{C}(v_1)\hat{C}(v_2) < 0$. After this we assume there is a solution, v^*, such that $\hat{C}(v^*) = 0$, hence we have,

$$\frac{\hat{C}(v_2) - \hat{C}(v^*)}{v_2 - v^*} = \frac{\hat{C}(v_1) - \hat{C}(v^*)}{v^* - v_1}, \tag{10}$$

$$\therefore v^* = \frac{v_2\hat{C}(v_1) + v_1\hat{C}(v_2)}{\hat{C}(v_1) + \hat{C}(v_2)}. \tag{11}$$

This can be applied recursively until the required convergence condition is met.

If the user enters a price corresponding to a negative volatility, then $\hat{C}(v_1)\hat{C}(v_2) > 0$ as the price manifold will be increasing on [0,1]. This provides a suitable stopping criterion to warn the user that the volatility is unphysical and return a warning.

4 Put-Call Parity for Caps and Floors (Scenario 1)

Put-call parity for caps and floors can be expressed as,

$$C(E) - P(E) = \Omega, \tag{12}$$

$$\Omega \equiv (f - E)e^{-rt_2}, \tag{13}$$

where C denotes the price of a cap with a strike price E and P denotes the equivalent floor. Ω here denotes the value of a swap (for more info see here) which can be evaluated by discounting the difference between the forward rate and the strike. Tables 1,2 and 3 show the results from scenario 1, which shows that put-call parity holds up until the 6th decimal place, at which point the precision of the implementation introduces some error

t_1	t_2	C(0.07)	P(0.07)	C(0.07) - P(0.07)	Ω (0.07)
91	182	0.00239907	0.00298349	-0.00058442	-0.000584423
182	273	0.00344498	0.00401938	-0.0005744	-0.000573993
273	365	0.0042048	0.00476924	-0.00056444	-0.000564439
365	456	0.00481815	0.00537291	-0.00055476	-0.000554758
456	547	0.00532173	0.00586698	-0.00054525	-0.000545242
547	638	0.0057509	0.00628679	-0.00053589	-0.000535890
638	730	0.00612102	0.00664761	-0.00052659	-0.000526599

Table 1: Put-Call parity results for caps and floors with a 0.07 strike value and a forward rate of 0.069395. If put-call parity is given, the 2 rightmost columns should be identical.

t_1	t_2	C(0.08)	P(0.08)	C(0.08) - P(0.08)	Ω (0.08)
91	182	0.00024967	0.010494	-0.01024433	-0.01024432
182	273	0.00082007	0.0108887	-0.01006863	-0.0100686
273	365	0.00138657	0.0112806	-0.00989403	-0.00989402
365	456	0.00191061	0.0116349	-0.00972429	-0.00972431
456	547	0.00237815	0.0119357	-0.00955755	-0.00955752
547	638	0.00280022	0.0121938	-0.00939358	-0.00939358
638	730	0.00318157	0.0124123	-0.00923073	-0.00923071

Table 2: Put-Call parity results for caps and floors with a 0.08 strike value and a forward rate of 0.069395. If put-call parity is given, the 2 rightmost columns should be identical.

t_1	t_2	C(0.09)	P(0.09)	C(0.09) - P(0.09)	Ω (0.09)
91	182	1.10E-05	0.0199152	-1.99E-02	-0.01990421
182	273	0.0001361	0.0196989	-1.96E-02	-0.01956281
273	365	0.00037008	0.0195937	-1.92E-02	-0.0192236
365	456	0.0006576	0.0195515	-1.89E-02	-0.01889387
456	547	0.00095907	0.0195289	-1.86E-02	-0.01856979
547	638	0.00126125	0.0195125	-1.83E-02	-0.01825128
638	730	0.00155571	0.0194905	-1.79E-02	-0.01793481

Table 3: Put-Call parity results for caps and floors with a 0.09 strike value and a forward rate of 0.069395. If put-call parity is given, the 2 rightmost columns should be identical.

5 Scenario 2

The option prices given by the black model are presented in table (4)- where P denotes a floorlet price and C a caplet price

t_1	t_2	C(0.059)	P(0.059)
91	182	0.00888234	5.50033e-05
182	273	0.0154565	6.83429e-05
273	365	0.0183971	9.80731e-05
365	456	0.0210227	0.000131901
456	547	0.0214058	0.000248479
547	638	0.0202763	0.000412912
638	730	0.0196652	0.000589884

Table 4: Caplet and Floorlet prices for scenario 2.

6 Scenario 3

For this question, I chose to return a NaN value for the implied volatility along with a warning to the console that this NaN value had occurred. Throwing an exception here would have been slightly inconvenient as if this program was a module for a larger system, a simple typo on the price may have occurred hence we do not wish to terminate any other operations that may not depend on this volatility (i.e. if this program was being used to determine the forward rate).

7 Bonus Question

The implied volatilities can be found in table (5).

t_1	t_2	σ
91	182	0.820643
182	273	0.728047
273	365	0.590326
365	456	0.660616
456	547	0.592719
547	638	0.631817
638	730	0.429952

Table 5: Caplet and Floorlet prices for scenario 2.