#####################################################################################

### README OF CURVE MODULE

### AUTHOR: ALAN

### DATE: 2018/12/18

### CONTACTS: [SHAOLUN.DU@GMAIL.COM](mailto:SHAOLUN.DU@GMAIL.COM)

#####################################################################################

PART ZERO: HOW TO USE & MAINTAIN

#####################################################################################

1. LOCATION: “\*\*\*\Shared Projects\Portfolio\_Management\curve\”
2. TEST CASES: “\*\*\*\Shared Projects\Portfolio\_Management\curve\_test\”
3. HOT TO RUN:
   1. For stan alone module test please run “curve\_test” in TEST CASE lcation
   2. For portfolio level testing please make corresponding inputs and run “test\_portfolio” in TEST CASE of portfolio module

\*\*\* Will be detailed here after implementation \*\*\*

PART ONE: GENERAL OUTLINE

#####################################################################################

Curve module is used to generate and keep a record of generated curves. It has 5 different files and 1 object named “Curve\_Keeper”. This object calls “Boot\_Strapping” function which has three different ways of bootstrapping, they are: “boot\_strapping\_LIBOR”, ” boot\_strapping\_Zero” and “boot\_strapping\_OIS”. The reason of “boot\_strapping\_Zero” is BRL market instruments are given in zero rates. Also, OIS discounting is different from libor bootstrapping as listed “boot\_strapping\_OIS”. “Boot\_Strap\_Tools\_Func” incorporates all tools will be called in bootstrapping.

PART TWO: MODULE OUTLINE

#####################################################################################

Module objects outline:

This module has one object:

* Curve\_Keeper has methods: gen\_curve\_by\_date, gen\_fx\_dic\_li, gen\_curve, get\_yields\_curve, set\_curves, get\_other\_data, get\_raw\_data and get\_rates\_by\_type. Module takes inputs from data base: Yield\_Curve. Libor inputs from yield\_curve, FX inputs from fx\_curve, and curve settings from curve\_setting.
* NOTE: The most important thing here is the convention in the market instruments. With different currency market has different conventions. We also implement the shocking part in curve keeper, as mentioned it is actually PV01 and all KRPVs since we are shocking par swap rate based on market inputs. The shocking function based on linear interpolation on any given year point, for example, shocking year 3 will generate 1bps up in year 3 par swap rate also generate 0.5bps in both year 2 and year 4 par swap rate, the year 1 and year 5 swap rate should remain untouched.

PART THREE: KEY FUNCTIONS OUTLINE

#####################################################################################

The most important function is the bootstrapping function, (NOTE: Details on how to do bootstrapping is in a separate txt file named “How to do bootstrapping”.) Here I will only provide a brief review.

* boot\_strapping\_LIBOR( sdate, convention, instruments, Day\_Counter).

The convention parameter takes currency convention in market then this works as inputs for Day\_Counter. Instruments is a set of markets swap rates, cash rates and future rates.

* NOTE: Some of the market instruments do not have futures rates, in this case we will set the number of effective future rates into zero, please take a look in the corresponding function.