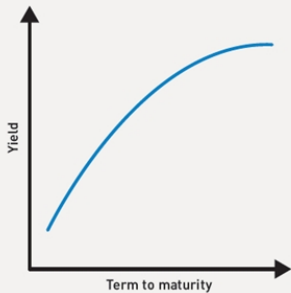


C++ fundamentals with use cases from finance

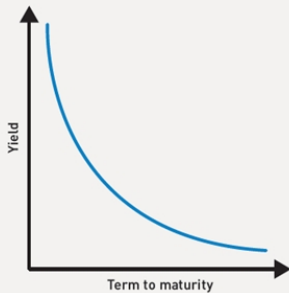
Tutorial 1: Interest Rate Curve Interpolation

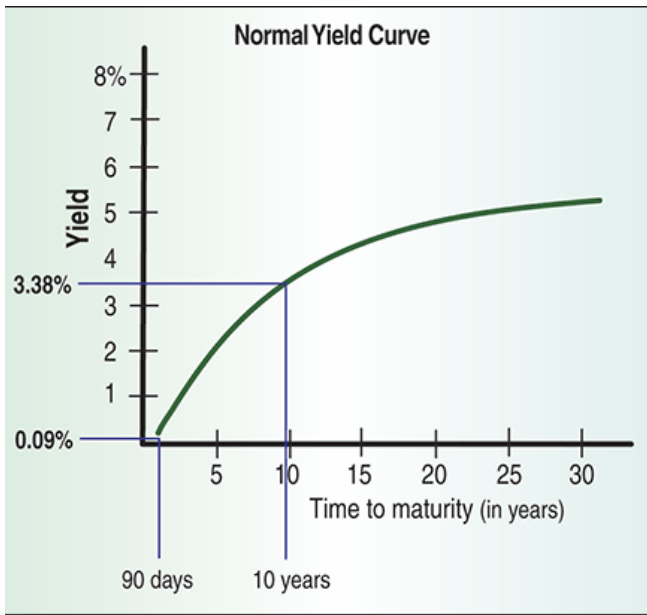
Ivan Zhdankin

Normal yield curve



Inverted yield curve



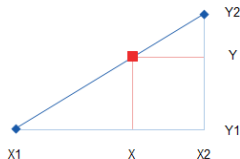


- Create variables of type *double*: *tenor1*, *tenor2*, *value1*, *value2*
- Using operator *cin* make the variables user defined, so that the user can input them from the screen
- Introduce testing for tenors to make sure that they are positive using operator *if*
- Print out on the screen all the tenors and all the values of the curve using *cout*

- Create a function that return maximum value between two tenors: the inputs are value1, value2
- Use *immediate if* operator for the function: "condition? return this if true : returns this if false"
- Write a code in *main* function to see on the screen the tenor with larger value

Part 3/5

- Implement "Linear" interpolation
- Assume now we have a bond which maturity is between two other benchmark bonds. However we do not know the yield of the bond, but we know the yield of the benchmark bonds. We would like to linearly interpolate the yield from benchmarks.
- Create function that returns *double* type and that take as arguments the following: *tenor1*, *tenor2*, *value1*, *value2*, *tenor*
- The function should return linearly interpolated value of the *tenor* using *tenor1*, *tenor2*, *value1*, *value2*



$$\frac{(X - X1)}{(X2 - X1)} = \frac{(Y - Y1)}{(Y2 - Y1)}$$

$$Y = Y1 + (X - X1) \frac{(Y2 - Y1)}{(X2 - X1)}$$

- Using operator *if* impose alerts for cases when it is impossible to interpolate, for example when *tenor1* > *tenor2*

- Implement "Flat" interpolation
- Assume now we would like to interpolate from the benchmark yields by taking the closest benchmark yield. That is the resulted yield of the new bond is going to be equal to one of the yields of the benchmarks depending on which benchmark expires closer to the maturity date of the new bond.
- Write function for this case: the function should have different name from the linear interpolation case and should return *double*, the inputs are *tenor1*, *value1*, *tenor2*, *value2* and the *tenor* that we need to interpolate

- Write function for interpolation of the values of the curve that implements two different interpolations: linear and "flat". Depending on the boolean flag the user should be able to chose between two options
- The function should return *double*, the inputs are *tenor1*, *value1*, *tenor2*, *value2*, *tenor* and boolean variable *linear*
- The function should return linear interpolation in case *linear* is *True*, or "flat" interpolation in case *linear* is *false*
- Write the declaration of all the functions and implementations of them in a separate header file (.h)