Advanced Calculus with FE Application: Quiz 2

Due on July 16, 2014

Advisor: Dan Stefanica

Weiyi Chen

Problem 1

The approximate values of the integral

$$\frac{1}{2} + \frac{1}{\sqrt{2\pi}} \int_0^t e^{-\frac{x^2}{2}} dx \tag{1}$$

using the Simpson's rules can be found in the table below:

No. Intervals	N(0.1)	N(0.5)	N(1.0)
4	0.539827837293	0.691462502398	0.841345406139
8	0.539827837278	0.69146246384	0.84134478715
16	0.539827837277	0.691462461434	0.841344748633
32		0.691462461284	0.841344746229
64		0.691462461275	0.841344746079
128		0.691462461274	0.841344746069
256			0.841344746069

The approximate values of the integral are

$$N(0.1) = 0.539827837277, N(0.5) = 0.691462461274, N(1.0) = 0.841344746069 \tag{2}$$

and are obtained for a 16 intervals partition, 128 intervals partition and 256 intervals partition, respectively, using Simpson's rule. Following is the python code as of the routine to compute above values

```
from math import *

def simpson(f, d_a, d_b, i_n):
    i_n *= 2
    d_h = (d_b - d_a) / i_n
    d_k, d_x = 0.0, d_a + d_h
    for i in range(1, i_n/2 + 1):
        d_k += 4*f(d_x)
        d_x += 2*d_h
    d_x = d_a + 2*d_h
    for i in range(1, i_n/2):
        d_k += 2*f(d_x)
        d_x += 2*d_h
    return (d_h/3)*(f(d_a)+f(d_b)+d_k)

def function(x): return exp(-x*x/2)

d_previous, d_current = 0.0, 0.0
for j in [0.1, 0.5, 1.0]:
    print "Compute N(", j, "):"
    for i in range(2,200):
        i_n = 2**i
        d_current = 0.5 + simpson(function, 0.0, j, i_n) / sqrt(2*pi)
        if abs(d_current-d_previous) < 10**(-12):
          print i_n, d_current
          break
        else:
          print i_n, d_current
          d_previous = d_current
```

Problem 2

Annual coupon bond

If the bond is an annual coupon bond, the value of a 19 months bond with coupon rate 4% and face value \$100 will be

$$\begin{aligned} B &= \sum_{t=7/12} 100Ce^{-tr(0,t)} + (100C+100)e^{-Tr(0,T)} \\ &= 4e^{-7/12r(0,7/12)} + 104e^{-19/12r(0,19/12)} \end{aligned} \tag{3}$$

The data below refers to the pseudocode from Table 2.5 for computing the bond price given the zero rate curve
Input: $n = 2$

$$\begin{aligned} t\_cash\_flow &= [7./12, 19./12] \\ v\_cash\_flow &= [4., 104.] \end{aligned} \tag{4}$$

The price of the bond is $B = 103.440082$.

Semiannual coupon bond

Input: $n = 4$

$$\begin{aligned} t\_cash\_flow &= [1./12, 7./12, 13./12, 19./12] \\ v\_cash\_flow &= [2., 2., 2., 102.] \end{aligned} \tag{5}$$

The price of the bond is $B = 103.495539$.

Quarterly coupon bond

Input: $n = 7$

$$\begin{aligned} t\_cash\_flow &= [1./12, 4./12, ..., 16./12, 19./12] \\ v\_cash\_flow &= [1., 1., ..., 101.] \end{aligned} \tag{6}$$

The price of the bond is $B = 102.518910$.
Attached is the python code to compute quarterly coupon bond as an example -

```python
from math import *

def r_2(time):
    return 0.02+time/(200.-time)

ls_cashflow = [1.,1.,1.,1.,1.,1.,101.]
ls_time = [1./12,4./12,7./12,10./12,13./12,16./12,19./12]
f_ret = 0.
for i in range(len(ls_cashflow)):
    f_ret += ls_cashflow[i]*exp(-ls_time[i]*r_2(ls_time[i]))
print f_ret
```

Problem 3

The price, duration, and convexity of the bond can be obtained from the yield y of the bond as follows:

$$
\begin{aligned}
B &= \sum_{t=[1/12,7/12,13/12]} 2\exp(-yt) + 102\exp(-yT) \\
D &= \frac{1}{B}\Big(\sum_{t=[1/12,7/12,13/12]} 2t\exp(-yt) + 102T\exp(-yT)\Big) \\
C &= \frac{1}{B}\Big(\sum_{t=[1/12,7/12,13/12]} 2t^2\exp(-yt) + 102T^2\exp(-yT)\Big)
\end{aligned}
\tag{7}
$$

where $T = 19/12$.
The data below refers to the pseudocode from Table 2.7 for computing the price, duration and convexity of a bond given the yield of the bond.
Input: $n = 4, y = 2.5\%$

$$
\begin{aligned}
t\_cash\_flow &= [1./12, 7./12, 13./12, 19./12] \\
v\_cash\_flow &= [2., 2., 2., 102.]
\end{aligned}
\tag{8}
$$

Output: bond price $B = 103.954808$, bond duration $D = 1.526212$, and bond convexity $C = 2.392899$.
Attached is the python code to compute bond price, duration and convexity -

```python
import numpy as np
from math import *

def bond_price(ls_cashflow, ls_time, f_yield):
    f_ret = 0.
    for i in range(len(ls_cashflow)):
        f_ret += ls_cashflow[i]*exp(-ls_time[i]*f_yield)
    return f_ret

def bond_duration(ls_cashflow, ls_time, f_yield):
    f_ret = 0.
    for i in range(len(ls_cashflow)):
        f_ret += ls_cashflow[i]*ls_time[i]*exp(-ls_time[i]*f_yield)
    return f_ret/bond_price(ls_cashflow, ls_time, f_yield)

def bond_convexity(ls_cashflow, ls_time, f_yield):
    f_ret = 0.
    for i in range(len(ls_cashflow)):
        f_ret += ls_cashflow[i]*ls_time[i]*ls_time[i]*exp(-ls_time[i]*f_yield)
    return f_ret/bond_price(ls_cashflow, ls_time, f_yield)

ls_cashflow = [2.,2.,2.,102]
ls_time = [1./12, 7./12, 13./12, 19./12]
f_yield = 0.025
print bond_price(ls_cashflow, ls_time, f_yield)
print bond_duration(ls_cashflow, ls_time, f_yield)
print bond_convexity(ls_cashflow, ls_time, f_yield)
```

Problem 4

(i)

The value of the fixed leg of a 30 months semiannual swap with a \$10 million notional with fixed rate 3% is

$$\begin{aligned} v_{fixed} &= \Big(\sum_{t=[.5,1,1.5,2]} 10 \times 0.03/2[1+0.5r(0,t)]^{-t/0.5}\Big) + 10 \times (1+0.03/2)(1+0.5r(0,T))^{-T/0.5} \\ &= \Big(\sum_{t=[.5,1,1.5,2]} 0.15[1+0.5r(0,t)]^{-t/0.5}\Big) + 10.15(1+0.5r(0,T))^{-T/0.5} \end{aligned} \tag{9}$$

where $T = 2.5$.
The data below refers to the pseudocode from Table 2.5 for computing the bond price given the zero rate curve
Input: $n = 5$

$$\begin{aligned} t\_cash\_flow &= [.5, 1, 1.5, 2, 2.5] \\ v\_cash\_flow &= [0.15, 0.15, 0.15, 0.15, 10.15] \end{aligned} \tag{10}$$

Output: the value is

$$v_{fixed} = 9.92141527551 \tag{11}$$

Therefore the value of the swap is

$$v_{swap} = v_{float} - v_{fixed} = 10. - 9.92141527551 = 0.0785847244 \tag{12}$$

million dollars, that is $78,584.72$ dollars.

Problem 4 continued on next page. . .

(ii)

The next floating payment happens at 1 month later as of $125,000. Then the value of the floating leg at $t = 1/12$ is the amount of notional adding the next floating payment, that is

$$v_{float}(t = \frac{1}{12}) = 0.125 + 10 = 10.125 \tag{13}$$

Then the current value of the floating leg is

$$v_{float}(t = 0) = v_{swap}(t) \times (1 + 0.5r(0,t))^{-t/0.5} = \$10.1038276352 \tag{14}$$

where $t = 1/12$.
Now we calculate the fixed leg as the way in (i), that is
Input: $n = 5$

$$\begin{aligned} t\_cash\_flow &= [1./12, 7./12, 13./12, 19./12, 25./12] \\ v\_cash\_flow &= [0.15, 0.15, 0.15, 0.15, 10.15] \end{aligned} \tag{15}$$

Output: the value of the fixed leg is

$$v_{fixed} = 10.0863823028 \tag{16}$$

Therefore the value of the swap is

$$v_{swap} = v_{float} - v_{fixed} = 0.0174453324728 \tag{17}$$

million dollars, that is $17,445.33$ dollars.