HW #4 Interpolation

Name: Jian Wang **Fsuid: Jw09r**

0. Executive Summary

Generate the function of yield rate via the given eight points under the natural cubic spline algorithm and plot the yield curve and forward rate curve for twenty years.
Besides, using the existed spline curve to get the yield, discount factor and forward rate for seven years.
The result is that the yield curve I obtained from the cubic spline algorithm is continuous and the yield rate at 7th year is 0.0898962, forward rate at the 7th year is 0.103784,discount rate at the 7th year is 0.532979

I. Statement of Problem

Using the natural cubic spline algorithm to generate the yield curve and the forward rate curve under the eight given yield points. Our main tasks are as follows:

1) Implement the cubic spline algorithm. Test your implementation and show that it is working correctly.
2) Use a natural cubic spline to plot the yield curve for twenty years.
3) Use the natural cubic spline to compute and plot the forward rate curve over twenty years
4) What is the yield, discount factor and forward rate for seven years?

II. Description of the Mathematics

Natural cubic spline requires following condition:
On each interval $[t_{0,}t_1]$, $[t_{1,}t_2]$,…, $[t_{n-1}t_n]$, S(x) is given by a different cubic polynomial, and we have the same first derivative and second derivative at knot point from two sides.

Since S_i is a cubic polynomial on $[t_i, t_{i+1}]$, S'' is a linear function satisfying $S_i''(t_i) = Z_i$ and $S_i''(t_{i+1}) = Z_{i+1}$ and therefore is given by the straight line between Z_i and Z_{i+1} :

$$S_i''(x) = \frac{z_i}{h_i}(t_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - t_i)$$

Where $h_i = t_{i+1} - t_i$, if this is integrated twice, then the result is $S_i(x)$

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 + \frac{z_{i+1}}{6h_i}(x - t_i)^3 + (\frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6})(x - t_i) + (\frac{y_i}{h_i} - \frac{z_i h_i}{6})(t_{i+1} - x)$$

Where:

$$z_i = S_i''(t_i)$$

The linear system for $1 \le i \le n-1$ with $z_0 = 0$ and $Z_n = 0$ is symmetric, tridiagonal, diagonally dominant, and of the form

$$\begin{bmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & & \cdots & \cdots & \cdots & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \cdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \cdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}$$

Where

$h_i = t_{i+1} - t_i$

$u_i = 2(h_i + h_{i-1})$

$b_i = \frac{6}{h_i}(y_{i+1} - y_i)$

$v_i = b_i - b_{i-1}$

Calculated using the algorithm:

$$u_i = u_i - \frac{h_{i-1}^2}{u_{i-1}}$$

$$v_i = v_i - \frac{v_{i-1}h_{i-1}}{u_{i-1}}$$

We can get $z_i = \frac{v_i - h_i z_{i+1}}{u_i}$

Then we can get the $S_i\ (x)$ as follows:

$$S_i(x) = y_i + (x - t_i)\big[C_i + (x - t_i)[B_i + (x - t_i)A_i]\big]$$

Where

$$A_i = \frac{1}{6h_i}(z_{i+1} - z_i)$$

$$B_i = \frac{z_i}{2}$$

$$C_i = -\frac{h_i}{6} z_{i+1} - \frac{h_i}{3} z_i + \frac{1}{h_i}(y_{i+1} - y_i)$$

After we get the yield curve, from the formula given by the problem, we can get the value of forward rate f and discount factor D as follows:

$$f(t) = (t * S(t))' = S(t) + t * S(t)'$$

$$D(t) = e^{-t*S(t)}$$

III. Description of the Algorithm

The algorithm is as the follows:

$Input\ n, (ti), (yi)$
$For\ i = 0\ to\ n - 1\ do$
$hi\ =\ ti + 1 - ti$
$bi\ = 6(yi + 1 - yi)/hi$
$Enddo$
$u1 = 2(h0 + h1)$
$v1\ = b1 - b0$
$for\ i\ =\ 2\ to\ n - 1\ do$
$ui =\ 2(h_i + h_{i-1}) - h_{i-1}^2/u_{i-1}$
$v_i =\ b_i - b_{i-1} - h_{i-1}v_{i-1}/u_{i-1}$
End do
Zn=0
For i= n-1 to 1 step -1 do
$z_i = \dfrac{v_i - h_i z_{i+1}}{u_i}$
End do
$z_0 = 0$
Output (z_i)

For i= 0 to n-1 do
$A_i = \dfrac{1}{6h_i}(z_{i+1} - z_i)$
$B_i = \dfrac{z_i}{2}$
$C_i = -\dfrac{h_i}{6}z_{i+1} - \dfrac{h_i}{3}z_i + \dfrac{1}{h_i}(y_{i+1} - y_i)$
$S_i(x) = y_i + (x - t_i)[C_i + (x - t_i)[B_i + (x - t_i)A_i]]$
End do

IV. Results

From the formula $S_i(x) = y_i + (x - t_i)\left[C_i + (x - t_i)[B_i + (x - t_i)A_i]\right]$, we can know that

$S_i'(x) = C_i \; + 2B_i(x - t_i) + 3A_i(x - t_i)^2$
$S_i''(x) = 2B_i + 6A_i(x - t_i)$

This is used to generate the rate of the forward rate.

The Answer for Task One:

With the help of software VBA, I got the result as follows:

T(i)	Z(i)	Y(i)	A(i)	B(i)	C(i)
0.5	0	0.0552	-0.00077	0	0.009792
1.0	-0.0023	0.06	0.000132	-0.001149147	0.009217
2.0	-0.00151	0.0682	3.5E-05	-0.000752558	0.007315
4.0	-0.00109	0.0801	1.81E-05	-0.000542753	0.004725
5.0	-0.00098	0.0843	2.03E-05	-0.000488367	0.003694
10.0	-0.00037	0.0931	8.08E-06	-0.000183369	0.000335
15.0	-0.00012	0.0912	4.14E-06	-6.21578E-05	-0.00089
20.0	0	0.0857			

So the $S_i(x)\ is\ as\ follows$:

$S_0(x) = 0.0552 + (x - 0.5)\left[0.00972 + (x - 0.5)[0 + (x - 0.5) * -0.00077]\right]$
$S_1(x) = 0.0600 + (x - 1.0)\left[0.009217 + (x - 1.0)[-0.001149 + (x - 1.0) * -0.000132]\right]$
$S_2(x) = 0.0682 + (x - 2.0)\left[0.007315 + (x - 2.0)[-0.0007526 + (x - 2.0) * -0.000035]\right]$
$S_3(x) = 0.0801 + (x - 4.0)\left[0.004725 + (x - 4.0)[-0.0005428 + (x - 4.0) * -0.0000181]\right]$
$S_4(x) = 0.0843 + (x - 5.0)\left[0.003694 + (x - 5.0)[-0.000488 + (x - 5.0) * -0.0000203]\right]$
$S_5(x) = 0.0931 + (x - 10)\left[0.000335 + (x - 10.0)[-0.000183 + (x - 10.0) * 0.00000808]\right]$
$S_6(x) = 0.0912 + (x - 15)\left[-0.00089 + (x - 15.0)[-0.000062 + (x - 15.0) * 0.00000414]\right]$

To test if the cubic spline algorithm work correctly, I used two methods:

Firstly, I checked if the $S_i(x)$ had the same value with the true data at each knot, I got the results as follows:

i	Ti	Yi	Si(Ti)
0	0.5	0.0552	0.0552
1	1	0.06	0.06
2	2	0.0682	0.0682
3	4	0.0801	0.0801
4	5	0.0843	0.0843
5	10	0.0931	0.0931
6	15	0.0912	0.0912

The results showed that those $S_i(x)$ has no error at each knot.

Secondly, I tested if the following equations satisfy:

$S_{i-1}(t_i) = S_i(t_i)$
$S'_{i-1}(t_i) = S'_i(t_i)$
$S''_{i-1}(t_i) = S''_i(t_i)$

The results are as follows:

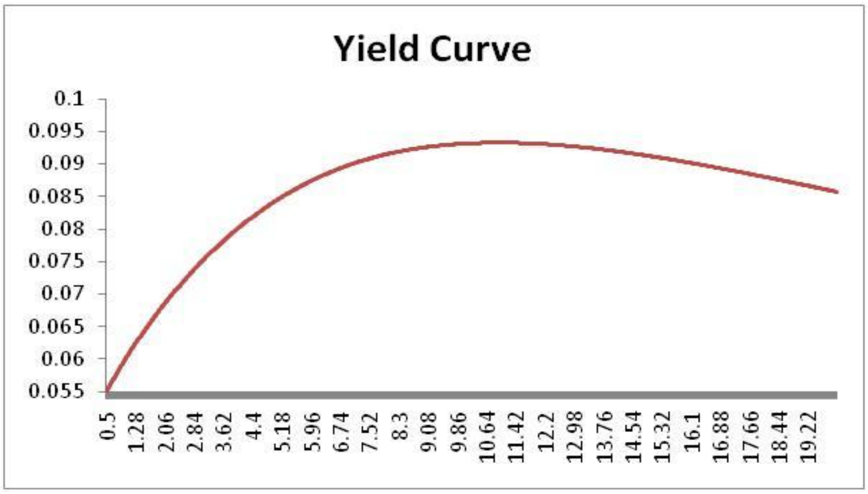
i	Ti	Yi	Si-1(Ti)	Si(Ti)	S'i-1(Ti)	S'i(Ti)	S''i-1(Ti)	S''i(Ti)
0	0.5	0.0552	Novalue	0.0552	Novalue	0.009791525	Novalue	0
1	1	0.06	0.06	0.06	0.009216951	0.009216951	-0.00229829	-0.00229829
2	2	0.0682	0.0682	0.0682	0.007315246	0.007315246	-0.00150512	-0.00150512
3	4	0.0801	0.0801	0.0801	0.004724624	0.004724624	-0.00108551	-0.00108551
4	5	0.0843	0.0843	0.0843	0.003693504	0.003693504	-0.00097673	-0.00097673
5	10	0.0931	0.0931	0.0931	0.000334826	0.000334826	-0.00036674	-0.00036674
6	15	0.0912	0.0912	0.0912	-0.00089281	-0.00089281	-0.00012432	-0.00012432

Which showed that the algorithm is continuous on each knot.

Therefore my results to generate $S_i(x)$ is correct and the method of natural cubic alpine algorithm works.

The answer of task two:

The figure of the yield curve is



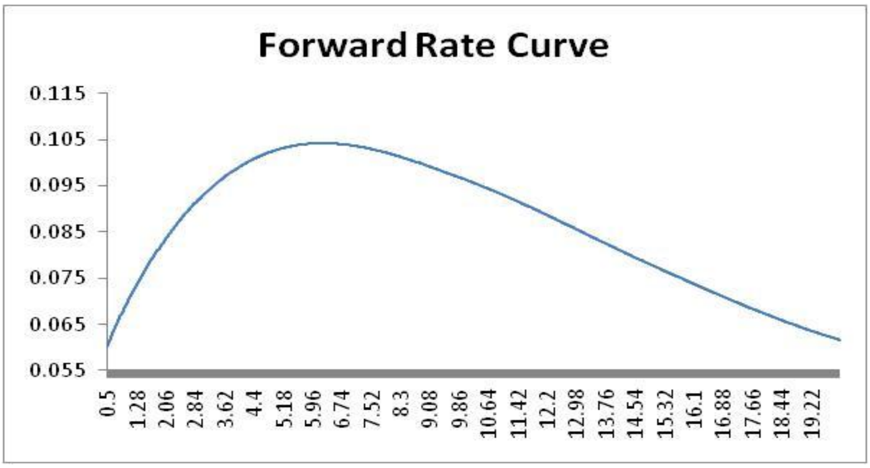
The answer of Task three:

The results over twenty years are as follows, I used the distance of half a year.

T(i)	Yield Rate	Forward rate	Discount rate
0.5	0.0552	0.060095762	0.9727774
1	0.06	0.069216951	0.941764534
1.5	0.064337713	0.076588139	0.908003932
2	0.0682	0.082830491	0.872493565
2.5	0.071673854	0.088146138	0.835951539
3	0.074797655	0.092542753	0.799001092
3.5	0.077597629	0.096125238	0.762165389
4	0.0801	0.098998497	0.725858636
4.5	0.08232889	0.101208495	0.690402854
5	0.0843	0.102767522	0.656061988
5.5	0.086027202	0.103739333	0.623036851
6	0.087525471	0.104192092	0.591464967
6.5	0.088810056	0.104186799	0.561431582
7	0.089896207	0.103784453	0.532978897
7.5	0.090799174	0.103046055	0.506113763
8	0.091534208	0.102032604	0.480814537
8.5	0.092116557	0.100805098	0.457037089
9	0.092561472	0.099424539	0.434719949
9.5	0.092884203	0.097951926	0.413788655
10	0.0931	0.096448257	0.394159354
10.5	0.093222581	0.094876514	0.375747803
11	0.093259538	0.093175171	0.358489856
11.5	0.093216931	0.09136847	0.342324997
12	0.093100822	0.089480653	0.327191431
12.5	0.092917271	0.087535963	0.313026983

13	0.092672338	0.085558642	0.299769813
13.5	0.092372084	0.083572932	0.287358956
14	0.092022569	0.081603075	0.27573474
14.5	0.091629854	0.079673314	0.264839079
15	0.0912	0.07780789	0.254615682
15.5	0.090738575	0.075984787	0.245012044
16	0.090249179	0.074174117	0.235985042
16.5	0.089734919	0.072388309	0.227495193
17	0.089198905	0.070639796	0.219504755
17.5	0.088644243	0.068941009	0.211977688
18	0.088074042	0.06730438	0.204879608
18.5	0.087491409	0.065742341	0.198177722
19	0.086899452	0.064267322	0.191840766
19.5	0.08630128	0.062891756	0.185838927
20	0.062325371	0.235228526	0.287507186

The Forward rate curve is



The discount rate curve is

The Answer of task Four:

The results for seven years are as follows:

Maturity(years)	1	2	3	4	5	6	7
Yield for different maturities	0.06	0.0682	0.0747977	0.0801	0.0843	0.0875255	0.0898962
Forward rate	0.069217	0.0828305	0.0925428	0.0989985	0.102768	0.104192	0.103784
Discount factor	0.941765	0.872494	0.799001	0.725859	0.656062	0.591465	0.532979

V. Conclusions

Firstly, the natural cubic spline algorithm gave us a good approximation of the Yield.

Secondly, the yield curve we generated is continuous and had the same value of the given data at each knot.

Thirdly, the result at the 7th year are as follows:

Maturity(years)	7
Yield for different maturities	0.0898962
Forward rate	0.103784
Discount factor	0.532979

VI Code Listing

```
Sub interpolation()

Dim rate(8), b(8), t(8), h(8), y(8), u(8), v(8), z(8), aa(8), bb(8), cc(8), x(1000), s(2000),
s1(2000), s2(2000) As Double

Dim maturity(8) As Double
Dim i As Integer

t(0) = 0.5
t(1) = 1
t(2) = 2
t(3) = 4
t(4) = 5
t(5) = 10
t(6) = 15
t(7) = 20
y(0) = 0.0552
y(1) = 0.06
y(2) = 0.0682
y(3) = 0.0801
y(4) = 0.0843
y(5) = 0.0931
y(6) = 0.0912
y(7) = 0.0857
For i = 0 To 6
h(i) = t(i + 1) - t(i)
b(i) = 6 * (y(i + 1) - y(i)) / h(i)
Next i
u(1) = 2 * (h(0) + h(1))
v(1) = b(1) - b(0)
For i = 2 To 6
u(i) = 2 * (h(i) + h(i - 1)) - h(i - 1) ^ 2 / u(i - 1)
v(i) = b(i) - b(i - 1) - h(i - 1) * v(i - 1) / u(i - 1)
Next i
z(7) = 0
For i = 6 To 1 Step -1
z(i) = (v(i) - h(i) * z(i + 1)) / u(i)
Next i
z(0) = 0
For i = 0 To 7
Worksheets("sheet1").Select
```

```
Worksheets("sheet1").Cells(i + 2, "a").Value = z(i)
Next i
For i = 0 To 6
aa(i) = 1 / (6 * h(i)) * (z(i + 1) - z(i))
bb(i) = z(i) / 2
cc(i) = (-h(i)) / 6 * z(i + 1) - h(i) / 3 * z(i) + 1 / h(i) * (y(i + 1) - y(i))
Worksheets("sheet1").Cells(i + 2, "b").Value = aa(i)
Worksheets("sheet1").Cells(i + 2, "c").Value = bb(i)
Worksheets("sheet1").Cells(i + 2, "d").Value = cc(i)
Next i
For i = 0 To 999
If i = 0 Then
x(i) = 0.5
Else
x(i) = x(i - 1) + 0.5
End If
Worksheets("sheet1").Cells(i + 2, "e").Value = x(i)
For j = 0 To 6
If x(i) >= t(j) And x(i) < t(j + 1) Then
s(i) = y(j) + (x(i) - t(j)) * (cc(j) + (x(i) - t(j)) * (bb(j) + (x(i) - t(j)) * aa(j)))
s1(i) = cc(j) + 2 * bb(j) * (x(i) - t(j)) + 3 * aa(j) * (x(i) - t(j)) ^ 2
s2(i) = 2 * bb(j) + 6 * aa(j) * (x(i) - t(j))
Worksheets("sheet1").Cells(i + 2, "f").Value = s(i)
Worksheets("sheet1").Cells(i + 2, "g").Value = s1(i)
Worksheets("sheet1").Cells(i + 2, "h").Value = s2(i)
End If
Next j
Next i
End Sub
```