
```

# Step1: Given current market price p_0_market of zero-coupon bond with 6-month
maturity
#         Obtain r_0
#         Using: p_0*exp(r_0*0.5)=100
from scipy.optimize import fsolve
import math
from math import log, exp

p_0_market = 99.1338
r_0 = log(100 / p_0_market) / 0.5

# Step2: Given current market price p_1_market of zero-coupon bond with 1-year
maturity, knowing fixed annual vol and delta
#         Assuming interest rate tree follows Ho and Lee with varying theta_t with
initial guess theta_1=0.012
#         First get theta_1: by first obtaining r_1u, r_1d; then p_1u, p_1d,
p_1_model, next using p_1_model(97.8925)=p_1_market to obtain theta_1
#         Second: using theta_1 obtaining interest_Tree_1 and bond_Tree_1

def error1(theta):
    p_1_market = 97.8925
    vol = 0.0173
    delta = 0.5

    theta_1 = theta

    r_1u = r_0 + theta_1 * delta + vol * delta ** 0.5
    r_1d = r_0 + theta_1 * delta - vol * delta ** 0.5

    p_1u = 100 * exp(-r_1u * delta)
    p_1d = 100 * exp(-r_1d * delta)

    p_1_model = (1 / 2) * p_1u * exp(-r_0 * delta) + (1 / 2) * p_1d * exp(-r_0 *
delta)

    return (p_1_model - p_1_market) ** 2

theta_1_opt = fsolve(error1, 0.012)[0]
print(theta_1_opt)
print(error1(theta_1_opt))

# Interest Tree 1 with T=1
vol = 0.0173
delta = 0.5

r_1u = r_0 + theta_1_opt*delta + vol*delta**0.5
r_1d = r_0 + theta_1_opt*delta - vol*delta**0.5
print(r_1u, r_1d)

# Bond Price Tree 1 with T=1
p_1u = 100*exp(-r_1u*delta)
p_1d = 100*exp(-r_1d*delta)
print(p_1u, p_1d)

```

```

# Step3: Given current market price p_2_market of zero-coupon bond(96.1462) with
1.5-year maturity, knowing fixed annual vol and delta
# Assuming interest rate tree follows Ho and Lee with varying theta_t with
initial guess theta_2=0.017
# Knowing: r_1u, r_1d; First get: r_2uu, r_2dd, r_2ud(=r_2du); then get:
p_2uu, p_2dd, p_2ud(=p_2du), p_2u, p_2d, p_2_model; next: using p_2_model=p_2_market
to obtain theta_2
# last, using theta_2 obtaining interest_Tree_2 and bond_Tree_2
# Second: using theta_2 obtaining interest_Tree_2 and bond_Tree_2

def error2(theta):
    p_2_market = 96.1462
    vol = 0.0173
    delta = 0.5
    r_0 = 0.0173994664

    r_1u = 0.0374713845
    r_1d = 0.0130054899

    theta_2 = theta

    r_2uu = r_1u + theta_2*delta + vol*delta**0.5
    r_2ud = r_1u + theta_2*delta - vol*delta**0.5
    r_2dd = r_1d + theta_2*delta - vol*delta**0.5

    p_2uu = 100*exp(-r_2uu*delta)
    p_2ud = 100*exp(-r_2ud*delta)
    p_2dd = 100*exp(-r_2dd*delta)

    p_2u = (1/2)*p_2uu*exp(-r_1u*delta) + (1/2)*p_2ud*exp(-r_1u*delta)
    p_2d = (1/2)*p_2ud*exp(-r_1d*delta) + (1/2)*p_2dd*exp(-r_1d*delta)

    p_2_model = (1/2)*p_2u*exp(-r_0*delta) + (1/2)*p_2d*exp(-r_0*delta)

    return (p_2_model - p_2_market)**2

theta_2_opt = fsolve(error2, 0.017)[0]
print(theta_2_opt)
print(error2(theta_2_opt))

# Interest Tree 2 with T=1.5
r_1u = 0.0374713883196 # already known
r_1d = 0.0130054936905 # already known

r_2uu = r_1u + theta_2_opt*delta + vol*delta**0.5
r_2ud = r_1u + theta_2_opt*delta - vol*delta**0.5
r_2dd = r_1d + theta_2_opt*delta - vol*delta**0.5
print(r_2uu, r_2ud, r_2dd)

# Bond Price Tree 2 with T=1.5
p_2uu = 100*exp(-r_2uu*delta)
p_2ud = 100*exp(-r_2ud*delta)
p_2dd = 100*exp(-r_2dd*delta)

```

```

p_2u = (1/2)*p_2uu*exp(-r_1u*delta) + (1/2)*p_2ud*exp(-r_1u*delta)
p_2d = (1/2)*p_2ud*exp(-r_1d*delta) + (1/2)*p_2dd*exp(-r_1d*delta)
print(p_2uu, p_2ud, p_2dd, p_2u, p_2d)

# Step4: Given current market price p_3_market of zero-coupon bond(94.1011) with 2-
year maturity, knowing fixed annual vol and delta
#           Assuming interest rate tree follows Ho and Lee with varying
theta_t with initial guess theta_3=0.011
#           Knowing: r_1u, r_1d, r_2uu, r_2dd, r_2ud(=r_2du); First get:
r_3uuu, r_3ddd, r_3uud(=r_3duu), r_3ddu(=r_3dud); then get: p_3uuu, p_3ddd, p_3uud
(=p_3duu), p_3ddu(=p_3dud), p_3uu, p_3dd, p_3ud(=p_3du), p_3u, p_3d, p_3_model;
next: using p_3_model=p_3_market to obtain theta_3
#           Second: using theta_3 obtaining interest_Tree_3 and bond_Tree_3

def error3(theta):
    p_3_market = 94.1011
    vol = 0.0173
    delta = 0.5

    r_0 = 0.0173994664

    r_1u = 0.0374713845
    r_1d = 0.0130054899

    r_2uu = 0.0606155197
    r_2ud = 0.0361496251
    r_2dd = 0.0116837304

    theta_3 = theta

    r_3uuu = r_2uu + theta_3*delta + vol*delta**0.5
    r_3uud = r_2uu + theta_3*delta - vol*delta**0.5
    r_3ddu = r_2dd + theta_3*delta + vol*delta**0.5
    r_3ddd = r_2dd + theta_3*delta - vol*delta**0.5

    p_3uuu = 100*exp(-r_3uuu*delta)
    p_3uud = 100*exp(-r_3uud*delta)
    p_3ddu = 100*exp(-r_3ddu*delta)
    p_3ddd = 100*exp(-r_3ddd*delta)

    p_3uu = (1/2)*p_3uuu*exp(-r_2uu*delta) + (1/2)*p_3uud*exp(-r_2uu*delta)
    p_3ud = (1/2)*p_3uud*exp(-r_2ud*delta) + (1/2)*p_3ddu*exp(-r_2ud*delta)
    p_3dd = (1/2)*p_3ddu*exp(-r_2dd*delta) + (1/2)*p_3ddd*exp(-r_2dd*delta)

    p_3u = (1/2)*p_3uu*exp(-r_1u*delta) + (1/2)*p_3ud*exp(-r_1u*delta)
    p_3d = (1/2)*p_3ud*exp(-r_1d*delta) + (1/2)*p_3dd*exp(-r_1d*delta)

    p_3_model = (1/2)*p_3u*exp(-r_0*delta) + (1/2)*p_3d*exp(-r_0*delta)

    return (p_3_model - p_3_market)**2

theta_3_opt = fsolve(error3, 0.011)[0]
print(theta_3_opt)

```

```

print(error3)

# Interest Tree 3 with T=2
r_1u = 0.0374713883 # already known
r_1d = 0.0130054936 # already known
r_2uu = 0.0606155197 # already known
r_2ud = 0.0361496251 # already known
r_2dd = 0.0116837304 # already known

r_3uuu = r_2uu + theta_3_opt*delta + vol*delta**0.5
r_3uud = r_2uu + theta_3_opt*delta - vol*delta**0.5
r_3ddu = r_2dd + theta_3_opt*delta + vol*delta**0.5
r_3ddd = r_2dd + theta_3_opt*delta - vol*delta**0.5
print(r_3uuu, r_3uud, r_3ddu, r_3ddd)

# Bond Price Tree 3 with T=2
p_3uuu = 100*exp(-r_3uuu*delta)
p_3uud = 100*exp(-r_3uud*delta)
p_3ddu = 100*exp(-r_3ddu*delta)
p_3ddd = 100*exp(-r_3ddd*delta)

p_3uu = (1/2)*p_3uuu*exp(-r_2uu*delta) + (1/2)*p_3uud*exp(-r_2uu*delta)
p_3ud = (1/2)*p_3uud*exp(-r_2ud*delta) + (1/2)*p_3ddu*exp(-r_2ud*delta)
p_3dd = (1/2)*p_3ddu*exp(-r_2dd*delta) + (1/2)*p_3ddd*exp(-r_2dd*delta)

p_3u = (1/2)*p_3uu*exp(-r_1u*delta) + (1/2)*p_3ud*exp(-r_1u*delta)
p_3d = (1/2)*p_3ud*exp(-r_1d*delta) + (1/2)*p_3dd*exp(-r_1d*delta)
print(p_3uuu, p_3uud, p_3ddu, p_3ddd, p_3uu, p_3ud, p_3dd, p_3u, p_3d)

```