

```
In [2]: import datetime as dt
def accrued_intest(par, c, m, t1, t2, t3, t4, rule):
    d1 = dt.datetime(t2[0], t2[1], t2[2]) - dt.datetime(t1[0], t1[1], t1[2])
    if rule == 'actual/actual':
        d2 = dt.datetime(t4[0], t4[1], t4[2]) - dt.datetime(t3[0], t3[1], t3[2])
    )
    interest = (d1.days / d2.days) * par * c
    elif rule == 'actual/360':
        interest = (d1.days / 360) * par * c
    else:
        interest = (d1.days / 365) * par * c
    return interest
```

```
In [3]: import numpy as np
def Bond_value(c, t, y):
    cashflow = []
    for i in np.arange(len(c)):
        cashflow.append(np.exp(-y[i] * t[i]) * c[i])
    return np.sum(cashflow)
```

```
In [4]: t_mature = dt.datetime(2028, 2, 1)
t_previous = dt.datetime(2018, 8, 1)
t_pricing = dt.datetime(2019, 1, 18)
t_next = dt.datetime(2019, 2, 1)
bond_par = 100
YTM = 0.031
coupon = 0.0385
m_coupon = 2
days_interest = (t_next - t_pricing).days
N = int((t_mature - t_pricing).days / 182.5) + 1
cashflow = np.ones(N) * bond_par * coupon / m_coupon
cashflow[-1] = 100 * coupon / m_coupon + bond_par
t_list = np.arange(N) / 2 + days_interest / 365
YTM_list = np.ones_like(t_list) * YTM
dirty_price = Bond_value( c = cashflow, y = YTM_list, t = t_list)
print("18Tresuary bond dirty price is: ", round(dirty_price, 6))
```

18Tresuary bond dirty price is: 107.453596

```
In [12]: bond_interest = accrued_intest(par = bond_par, c = coupon, m = m_coupon, t1 =
(2018, 8, 1), t2 = (2019, 1, 18), t3 = (2018, 8, 1), t4 = (2019, 2, 1), rule =
'actual/actual')
print("the 18 tresuary bond's accrued interest is: ", round(bond_interest, 6))
clean_price = dirty_price - bond_interest
print("the bond's clean price is:", round(clean_price, 6))
```

the 18 tresuary bond's accrued interest is: 1.793151