A Recursive Approach For Calculating The Effective Yield On Bonds When Settlement And/Or Expected Maturity Is Other Than An Interest Payment Date.

David Vicknair, (E-mail: David.Vicknaair@rockhurst.edu), Rockhurst University
Ricard E. Downing, (Deceased), Rockhurst University

ABSTRACT

Standard spreadsheet functions cannot compute bond yields if both the settlement and maturity dates fall on other than an interest payment date. Using reciprocal cell references, the RATE function is enhanced to overcome this limitation. An appendix extends the enhancement procedure to the IRR function. The procedure demonstrated has applications to a wide variety of financial transactions which are structured around an initial cash flow followed by an annuity and a lump sum, for example, a single advance loan with an odd first and/or last payment period and a terminal balloon payment.

INTRODUCTION

The conventional yield calculation in business school involves an initial cash inflow followed by a series of equal cash outflows in the form of an ordinary annuity and perhaps a terminal lump sum. These conventional flows could describe, for example, the yield to maturity on a U.S. corporate bond sold on an interest payment date or a single advance loan with payments beginning exactly one interest period after closing. Students can choose from several standardized functions in spreadsheet software and financial calculators and for calculating the effective yield when these simple conditions apply, for example, the IRR and RATE functions in Microsoft Excel and the IRR and I/Y functions in a Texas Instruments BA II Plus financial calculator.

However, in practice U.S corporate bonds are frequently sold and retired on other than an interest payment date. Likewise, term loans frequently have an odd (short or long) first or last payment period. In such circumstances, the standard functions cannot be used. Even the more powerful YIELD function is limited to bonds and cannot accommodate maturity on other than an interest payment date and requires a somewhat confusing transformation of data to a $100 par basis and the use of the arcane jargon of bond traders, such as quoted market price.

In this paper we develop and illustrate a simple recursive approach using the RATE and DAYS360 functions in Microsoft Excel for calculating the exact effective yield under a variety of transactions with odd first and/or last periods. Although our example and illustrations focus on U.S corporate bonds, the procedure we illustrate applies to a wide variety of common financial transactions with an unknown yield and which fit within the broad programming parameters of RATE. In an Appendix we extend the procedure to the IRR function, which has the advantage of being able to accommodate periodic, but unequal cash flows.

BOND YIELD ALGORITHMS

The effective periodic yield on a bond sold is the rate which equates the bond's interest and maturity or retirement-related cash flows with the net proceeds at issuance:[1]

$$0 = pv - \frac{pmt}{(1+y)^{1-\alpha}} - \frac{pmt}{(1+y)^{2-\alpha}} - \frac{pmt}{(1+y)^{n-\alpha}} - \frac{fv}{(1+y)^{n+\beta-\alpha}} \tag{1}$$

where,

pv   = actual price, net of incremental costs directly attributable to issuance
pmt   = periodic interest payment
y   = periodic effective yield
α = fraction of the current interest period elapsed prior to the settlement date
n   = number of interest payments remaining until the expected maturity date
fv   = expected cost to retire bonds on the expected maturity date[2]
β = fraction of the terminal interest period elapsed prior to maturity or expected retirement date

Business and accounting students are generally taught to think of the internal rate of return in terms of a net present value of zero as in equation (1). However, the effective periodic yield can also be expressed in terms of terminal values by multiplying equation (1) by $(1+r)^{n+\beta-\alpha}$ and simplifying:

$$0 = pv \times (1+y)^{n+\beta-\alpha} - pmt \times \left[\frac{(1+y)^{n+\beta-\alpha} - (1+y)^{\beta}}{y} \right] - fv \tag{2}$$

Despite the unconventional form of equation (2) it is of interest to this discussion because it is in the same form as the algorithm used by the RATE function in Excel to compute the yield.

ENHANCED RATE FUNCTION

The RATE function returns the periodic rate for an annuity. For a bond sold on an interest payment date, RATE takes on the following general form and algorithm substituting corresponding terms from equation (2):[3]

=RATE(y,n,-pmt,pv,-fv)   (3)

Observe that in its general form, RATE (equation 3) has no terms corresponding to α or β, that is, RATE implicitly assumes that both values are zero. However, we can cause RATE to solve equation (2) by substituting $pv \times (1+\theta)^{\beta-\alpha}$ for pv and $-pmt \times (1+\theta)^{\beta}$ for –pmt, where θ represents a reciprocal cell reference to the cell containing equation (3). Modified thusly, RATE takes on the following enhanced form:

=RATE(nper,-pmt*(1+θ)^β,pv*(1+θ)^(β-α),-fv)   (4)

Each circular cell reference in equation (4) incorporates an additional iterative calculation into the internal calculation conducted by RATE. Thus, the circular references, allow RATE to compute the values of y, $pv \times (1+y)^{n+\beta-\alpha}$ and $pmt \times [(1+y)^{\alpha} - (1+y)^{\beta}]/y$ simultaneously.

Recall that, except as required by a function's internal programming, by default Excel does not perform iterative calculations and so regards a circular reference as an error. To avoid Excel's error prompt, readers should disable the default setting immediately after opening the workbook and prior to entering any syntax containing a circular reference with the following keystrokes:[4]

1.   select <u>T</u>ools (from the Format toolbar),
2.   select <u>O</u>ptions (from the dropdown menu),
3.   select the Calculation tab and place a checkmark in the <u>I</u>teration checkbox.

Finally, an examination of the superscripts in equations (2) indicates that the correct yield cannot be obtained simply by entering a noninteger number of payments argument into RATE. For example, =RATE(n+β-α,-pmt,pv,-fv) results in the following algorithm, which by visual inspection is not equivalent to equation (2):

$$0 = pv \times (1+y)^{n+\beta-\alpha} + pmt \times \frac{(1+rate)^{n+\beta-\alpha} - 1}{y} + fv \tag{5}$$

AN EXAMPLE

To demonstrate equation (4) we adapt the following example from Rusbarsky and Vicknair (1999: 237) with the addition of issue costs and a deferred call provision:

A corporation issues semiannual bonds with a total par value of $1,000,000, dated April 1, 2007 and which mature in 25 years on April 1, 2032. The bonds pay interest equal to 4 percent of par value each April 1 and October 1 (an annual stated rate of 8 percent). The bonds are callable in after April 1, 2011 at an initial call premium of $80 per $1,000 par. The call premium declines to zero at the contractual maturity according to a fixed schedule. The total incremental, avoidable direct costs of the bond issue is $12,000. The bonds are expected to be called on December 13, 2021 when the call premium is $55 per $1,000 par, plus accrued interest.

Assuming settlement occurs on April 1, 2007 at a semiannual market rate of 5%, the pv of the bonds is =PV(5%,50,-40000,-1000000)-12,000, or $805,440.75. Correspondingly, if settlement occurs on May 16, 2007 at a semiannual market rate of 5.1%, α = 0.25 and the pv of the bonds is =PV(5%,50,40000,1000000)*(1+5%)^.25-12000, or $815,472.57.[5] Observe that by convention, the valuation of bonds assumes a 30 day month and 360 day year day counting convention and so the value of α is the number of 30/360 days between the April 1, 2007 dated date and the May 16, 2007 settlement date computed using the DAYS360 function divided by the 180, the total days in each semiannual period, or =DAYS360("4/1/2007","5/16/2007")/180 equals .25.

Table 1 shows the yield calculation for various values of α and β using an enhanced RATE function. Observe that fv takes on different values (e.g., maturity value or call value) depending on the scenario. Columns A and B compute the conventional yield to maturity which assumes that the bonds are held to their maturity date; Columns C and D compute the yield which assumes that the bonds are held the April 1, 2017 first call date; Columns E and F compute the yield to the December 13, 2013 expected call date.

Table 1: Yield Computations for Different Values of α and β Using Enhanced RATE Function						
	Yield to Maturity		Yield to First Call		Yield to Expected Call	
	A	B	C	D	E	F
Settlement	1-Apr-07	16-May-07	1-Apr-07	16-May-07	1-Apr-07	16-May-07
α	0	0.25	0	0.25	0	0.25
β	0	0	0	0	0.40	0.40
n	50	50	20	20	29	29
pmt	$ 40,000.00	$ 40,000.00	$ 40,000.00	$ 40,000.00	$ 40,000.00	$ 40,000.00
pv	$ 805,440.75	$ 800,286.78	$ 805,440.75	$ 800,286.78	$ 805,440.75	$ 800,286.78
fv	$1,000,000.00	$1,000,000.00	$1,080,000.00	$1,080,000.00	$1,071,000.00	$1,071,000.00
y	5.07872%	5.18131%	5.90264%	6.06949%	5.41826%	5.54605%
RATE Function for y:						
A	"=RATE(50,-40000,805440.75,-1000000)"					
B	"=RATE(50,-40000,800286.78*(1+θ)^-.25,-1000000)"					
C	"=RATE(20,-40000,805440.75,-1080000)"					
D	"=RATE(20,-40000,800286.78,-1080000*(1+θ)^.25)"					
E	"=RATE(20,-40000*(1+θ)^.40,805440.75*(1+θ)^.40,-1071000))"					
F	"=RATE(20,-40000*(1+θ)^.40,800286.78*(1+θ)^(.40-.25),-1080000)"					
Note: The Greek letter θ refers to a reciprocal cell reference to the cell containing the RATE function.						

Because Columns A and C assume that both settlement and maturity or call occur on an interest payment date, an unmodified RATE function is all that is required. In Columns B and D, the value of α is computed in the same way as discussed above. In Column E and F, the value of β is equal to the number of 30/360 days between October 1, 2021 which is the n=29 interest payment date and the December 13, 2021 expected call date divided by 180, the total number of days in each semiannual period, or =DAYS360("10/1/2021","12/13/2021")/180 equals .40. Significantly, the yields in Columns E and F cannot be computed using standard spreadsheet functions including YIELD.

CONCLUDING COMMENTS

This paper demonstrates an enhanced RATE function that will calculate the effective yield on bonds regardless of their settlement date or expected maturity date. Together, PV, FV, RATE (or IRR) and DAYS 360 constitute a complete set of bond valuation tools that share a common syntax. Instructors, by carefully planning settlement and expected maturity dates in their examples (for example by avoiding awkward fractional periods incorporating the end of February) can eliminate the need for precise day counting and so the need for DAYS360.

In contrast, "no frills" financial calculators can compute the present value of bonds sold with or without accrued interest; however, their internal programming is fixed and so they cannot find the effective yield when settlement and/or expected maturity fall between interest payment dates. Thus, their low cost and functionality comes at a loss of realism.

APPENDIX:
ENHANCED IRR FUNCTION

IRR returns the internal rate of return for an array of cash flows occurring at regular intervals and where the first cash flow in the array is assumed to occur at time equals zero and there is at least one sign change. The internal programming in IRR assumes that the value of both α and β is zero, and so would solve for the following periodic bond yield in which both pv and fv fall on an interest payment date.[6]

$$0 = pv - \frac{pmt}{(1+y)^1} - \frac{pmt}{(1+y)^2} - \frac{pmt}{(1+y)^n} - \frac{fv}{(1+y)^n} \tag{7}$$

However, when $\alpha>0$ and $\beta>0$, the spacing of pv and fv assumed by IRR is incorrect; pv is too far forward in time from n=0 by $(1+y)^{-\alpha}$ and fv is too far short in time from n+1 by $(1+y)^{(1-\beta)}$. We can correct these timing problems by multiplying pv and fv by recursive terms in much the same way as we did earlier with RATE:

$$0 = pv \times \frac{1}{(1+\theta)^\alpha} - \frac{pmt}{(1+y)^1} - \frac{pmt}{(1+y)^2} - \frac{pmt}{(1+y)^n} - \frac{fv}{(1+y)^n} \times \frac{1}{(1+\theta)^{(1-\beta)}} \tag{8}$$

As before, the greek term θ represents a reciprocal reference to the cell containing the IRR function and Excel's default setting regarding reciprocal references must be turned off.

Table 2 repeats the same yield calculations as Table 1, but using IRR. Observe that the corresponding yields in Table 1 and Table 2 are the same. An interesting artifact of using IRR can be seen, for example, in the values corresponding to n=0 and n=30 found in Column F of Table 2. The n=0 value was originally entered as =800286.78/(1+θ)^.25. Correspondingly, the n=30 value was originally entered as =1071000*(1+)^(1-.40). However, with the additional recursive terms, as IRR converged to the yield of 5.54605, it also simultaneously recalculated the values of pv and fv to their time adjusted the values. Thus, pv n=0 becomes \$789,560.02 [\$800286.78*(1+5.54605%)$^{.25}$] and fv at n=29+1 becomes \$-1,106,253.50 [\$-1,1071000×(1+5.54605%)$^{(1-.40)}$]. Thus, we could substitute these adjusted values for the corresponding values of pv (n=0) and fv (n=30) in an unmodified IRR function and obtain the same yield. The same would hold true for the values of pv (n=0) in Columns B, D and E.

Table 2:Yield Computations for Different Values of α and β Using Enhanced IRR Function					
	Yield to Maturity		**Yield to First Call**		**Yield to Expected Call**	
	A	B	C	D	E	F
Settlement	1-Apr-07	16-May-07	1-Apr-07	16-May-07	1-Apr-07	16-May-07
α	0	0.25	0	0.25	0	0.25
β	0	0	0	0	0.40	0.40
n	50	50	20	20	29	29
pmt	$40,000.00	$40,000.00	$40,000.00	$40,000.00	$40,000.00	$40,000.00
pv	$805,440.75	$800,286.78	$805,440.75	$800,286.78	$805,440.75	$800,286.78
fv	$1,000,000.00	$1,000,000.00	$1,080,000.00	$1,080,000.00	$1,071,000.00	$1,071,000.00
t=0	$805,440.75	$790,243.61	$805,440.75	$788,584.11	$805,440.75	$789,560.02
1	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
2	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
3	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
4	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
5	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
6	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
7	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
8	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
9	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
10	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
11	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
12	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
13	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
14	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
15	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
16	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
17	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
18	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
19	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)	(40,000.00)
20	(40,000.00)	(40,000.00)	(1,120,000.00)	(1,120,000.00)	(40,000.00)	(40,000.00)
21	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
22	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
23	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
24	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
25	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
26	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
27	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
28	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
29	(40,000.00)	(40,000.00)			(40,000.00)	(40,000.00)
30	(40,000.00)	(40,000.00)			(1,105,449.67)	(1,106,253.50)
31	(40,000.00)	(40,000.00)				
32	(40,000.00)	(40,000.00)				
33	(40,000.00)	(40,000.00)				
34	(40,000.00)	(40,000.00)				
35	(40,000.00)	(40,000.00)				
36	(40,000.00)	(40,000.00)				
37	(40,000.00)	(40,000.00)				
38	(40,000.00)	(40,000.00)				
39	(40,000.00)	(40,000.00)				
40	(40,000.00)	(40,000.00)				
41	(40,000.00)	(40,000.00)				
42	(40,000.00)	(40,000.00)				
43	(40,000.00)	(40,000.00)				
44	(40,000.00)	(40,000.00)				
45	(40,000.00)	(40,000.00)				
46	(40,000.00)	(40,000.00)				
47	(40,000.00)	(40,000.00)				
48	(40,000.00)	(40,000.00)				
49	(40,000.00)	(40,000.00)				
50	(1,040,000.00)	(1,040,000.00)				
IRR (y)	5.07872%	5.18131%	5.90264%	6.06949%	5.41826%	5.54605%
	10.41538%	10.63109%	12.15368%	12.50736%	11.13009%	11.39968%

BIBLIOGRAPHY

1.   Mayle, J. 1993. *Standard Securities Calculation Methods: Fixed Income securities Formulas for Price, Yield and Accrued Interest*, Volume 1. 3rd edition. New York, NY: Securities Industry Association.
2.   Rusbarsky, M. and D. B. Vicknair. 1999. Accounting for Bonds With Accrued Interest in Conformity With Brokers' Valuation Formulas. *Issues in Accounting Education* 14 (2) (May): 233-253.

ENDNOTES

[1] We adapt the bond pricing algorithm from Rusbarsky and Vicknair (1999):

$$pv = pmt \times \left[\frac{(1+y)^{\alpha} - (1+y)^{\alpha-n}}{y} \right] + fv \times (1+y)^{\alpha-n},$$ where pv, pmt, α, n, and y are defined per equation (1) and y

and fv are the periodic market determined yield and the maturity value, respectively. Using the PV function and substituting the terminology in equation (1) pv is computed as "=PV(y,n,-pmt,-fv)*(1+y)^α". Rusbarsky and Vicknair are interested in bond valuation and so distinguish between the present value of bonds sold without accrued interest (which they label V) and the present value of bonds sold with accrued interest (which they label V_{α}). For our purposes, such a distinction is unnecessary.

[2] FV can take on several values, including par value, first call price, or expected retirement cost.

[3] Equation (3) can be expressed mathematically as: $0 = pv \times (1+y)^n + pmt \times \dfrac{(1+y)^n - 1}{y} + fv$

[4] Alternatively the default can be toggled on and off by simultaneously pressing the "Ctrl", "T", "O", and "I" keys.

[5] Also =FV(5%,0.25,0,PV(5%,50,40000,1000000)-12000.

[6] Because equation (6) assumes that the last two terms fall on the same date, their sum is the last value.

NOTES