AN EFFICIENT ALGORITHM FOR CALCULATING YIELDS AND DURATIONS

Delvin D. Hawley, Finance, University of Mississippi
Larry J. Johnson, Finance, University of Tulsa

ABSTRACT

This paper presents an algorithm for the computation of yields (or internal rates of return) which offers several important advantages over traditional computational algorithms. The method described here is extremely efficient in zeroing in on the correct discount rate or yield in a remarkably small number of iterations. It is particularly useful in the calculation of internal rates of return for projects with erratic cash flows.

Most financial researchers and professionals occasionally find it necessary to calculate accurate yields for a large sample of fixed-income securities. Others find it necessary to calculate internal rates of return for projects with many cash flows of unequal amounts and/or unequal timing.

Commercial spreadsheet software such as Lotus 1-2-3(1) lends some assistance in this area by providing pre-programmed functions which calculate the present value of a string of equally-spaced periodic cash flows of equal or unequal amounts given a discount rate supplied by the user.(2)

Even so, the calculation of yields or IRR's using these functions can involve a substantial amount of time and effort on the part of the user since they generally require the repeated trial-and-error substitution of discount rates followed by the recalculation of the formula until the correct rate is determined Thus, continuous intervention by the user is necessary. The process can be very slow and tedious when the task involves the calculation of yields for a large number of securities, especially when a high degree of accuracy is required in the calculated rates. Furthermore, typical spreadsheet functions of this type make no provision for situations involving uneven timing of project cash flows or, in the case of fixed income securities, fractional discounting periods and accrued interest.

Given these common shortcomings in spreadsheet software, most practitioners find it more convenient and expeditious to use a high-level computer language such as BASIC or Fortran to create a program to compute accurate yields or IRR's without continuous user intervention. The typical algorithm for such a program can best be described as a fixed-increment iterative search.

A general outline of this type of procedure is presented in Figure I in conjunction with a simple example. The example involves a $1000 par-value bond with an $80 annual coupon payment and 10-year maturity. The bond is currently trading at $920 (assuming that the next coupon payment is due in exactly one year and therefore that accrued interest is zero).(3) The user must set the fixed discount rate adjustment and the acceptable level of accuracy depending on the requirements of the current task.

Let's assume that we have set the acceptable level of accuracy at +/- $1 with respect to the current price. That is, if the trial discount rate gives a present value of future cash flows that is between $919 and $921, then it will be accepted as the yield of this bond. This level of accuracy will require a small fixed incremental adjustment, so let's assume that we have set the increment at 1 basis point. The user must also select an arbitrary starting point for the trial discount rate. In the example, this arbitrary starting point was set at 10%.

The steps involved in the fixed-increment iterative search procedure are shown in Figure I. In the first pass through the loop the trial discount rate (arbitrarily set by the user at 10%) is used to compute the present value (PV) of the 10 remaining coupon payments and the terminal payment. Given the example values, PV = $ 877.11 given a 10% discount rate. This computed present value is compared to the market value (P). If the difference between PV and P is greater than one dollar, then the trial discount rate will be changed by one basis point (the amount of the pre-set fixed increment). The present value calculation (Step 2) is then repeated using the new discount rate (9.99%) and the resulting PV is again compared to P (Step 3). This process continues until the calculated PV is within one dollar of the current market price. This would require 72 iterations in the example. The resulting yield is 9.27%, which gives a calculated PV that is fifty-four cents less than the bond's current market value. The computed yield is accurate to within one basis point.(4)

Although the fixed-increment iterative search procedure allows the user to calculate the yields for any number of bonds without intervention, the procedure is still slow and inefficient. If the sample size is of any appreciable magnitude, the time required for the yield calculations on a microcomputer can easily run to several hours. Furthermore, the amount of time required for calculation increases with the desired level of accuracy since greater accuracy requires a smaller incremental adjustment to the trial discount rate per iteration.(5)

This paper presents an algorithm for the computation of yields (or internal rates of return) which offers several important advantages over the fixed-increment iterative search procedure. First, it is much faster and more efficient. Second, the speed of the calculation is not reduced appreciably as the required level of accuracy increases. Third, the procedure can handle cash flows of uneven timing and/or unequal amounts without any reduction in speed or efficiency. Thus it is particularly useful in the calculation of the internal rate of return for projects with erratic cash flows.

The key to the speed and efficiency of this improved algorithm, (subsequently referred to as the proportional-increment iterative search procedure), is the method by which the trial discount rate is adjusted at the end of each iteration. Instead of using an arbitrary fixed increment, the proportional-increment procedure varies the magnitude of the discount rate adjustment as the calculated value of the cash flows (PV) and the market value (P) converge. When the error between PV and P is large, the incremental adjustment to the trial discount rate is proportionally large. When PV is very close to P, the incremental adjustment to the trial discount rate is proportionally small. The particular adjustment method described here is extremely efficient in zeroing in on the correct discount rate in a remarkably small number of iterations.(6)

Again, a simple example is helpful for describing the proportional-increment iterative search procedure. A general outline of the procedure is presented in Figure II along with the numerical results of each step as applied to the example data. The previous example inputs are retained here.

First, the present value of the future cash flows is computed using the arbitrarily selected starting discount rate of 10%, resulting in PV = $877.11. The bond's duration (D) is also computed using the current trial discount rate (DR) and the current calculated present value (PV) of the remaining cash flows, giving D = 6.882462 years.(7) The error between the calculated present value (PV) and the actual market value (P) is ($920-$877.11) = $42.89.

At this point, the fixed-increment procedure would have adjusted the trial discount rate by the pre-set fixed increment (1 basis point in the example) and then begun the second iteration. In contrast, the proportional adjustment procedure links the magnitude of the trial discount rate adjustment to the size of the error between P and PV. Thus, when a large (small) error is encountered, a large (small) adjustment in the trial discount rate will be made.

One way to accomplish this would be to simply view the error (P-PV) as a percentage of the market price (P) and link the magnitude of the discount rate adjustment to the relative magnitude of the error. While this would offer an

FIGURE I

Fixed-Increment Iterative Search Procedure for Yield-to-Maturity Calculations

Inputs: Market Price (P) = $920.00
Annual Coupon Payment (C) = $80
Terminal Payment (T) = $1000
Number of Payments Remaining (N) = 10
Fixed Increment per Iteration (I) = 1.0 Basis Point (set by the user as desired)
Acceptable Accuracy = +/- $1 on $1000 par value bond (set by the user as desired)

Step 1: Set trial Discount Rate (DR) at an arbitrary starting point, say 10%.

Step 2: Calculate the Present Value (PV) of the remaining coupon and terminal payments using the trial Discount Rate (DR).

Step 3: Compare the Calculated Present Value (PV) to the Market Price (P):

3a: If (P-PV) is greater than $1, then DR is decreased by the pre-set fixed increment (I) and Steps 2 and 3 are repeated.

If (P-PV) is less than -$1, then DR is increased by (I) and Steps 2 and 3 are repeated.

3b: If |P-PV| is less than or equal to $1, then the calculation is complete and the yield of the bond equals DR. The program will terminate or will begin processing the next bond in the sample.

For the first pass through the steps, the calculations would be:

With DR=10%, Step 2 calculates PV=$877.11 and Step 3 calculates (P-PV) = $42.89. Since (P-PV) is positive and greater than the acceptable level of accuracy, DR will be reduced by 1 Basis Point and Steps 2 and 3 will be repeated using the adjusted trial discount rate of 9.99%.

This process repeats until |P-PV| is within the acceptable range. In this example, the yield calculation would require a total of 72 iterations. The trial Discount Rate in the 72nd iteration would be 9.27% which would give PV = $919.46. Since |P-PV| is less than $1, the program would either terminate or return to Step 1 and begin processing the next bond in the sample.

FIGURE II

Proportional-Increment Iterative Search Procedure for Yield-to-Maturity Calculations

Inputs: Market Price (P) = $920.00
Annual Coupon Payment (C) = $80
Terminal Payment (T) = $1000
Number of Payments Remaining (N) = 10
Acceptable Accuracy = +/- $1 on $1000 par value bond (set by the user as desired)

Step 1: Set trial Discount Rate (DR) at an arbitrary starting point, say 10%.

Step 2: Calculate the Present Value (PV) of the remaining coupon and terminal payments using the trial Discount Rate (DR). Also calculate the duration (D) of the bond using the current value of DR as the discount rate and the current value of PV as the price.

Step 3: Compare the Calculated Present Value (PV) to the Market Price (P):

3a: If |P-PV| is greater than $1, then perform the following calculation to determine the new trial discount rate:

$$\text{(Adjusted R)} = \text{(Unadjusted R)} * \left[1 - \frac{(P-PV)}{D * PV} \right]$$

where: P = Current Market Price
PV = Present Value of remaining cash flows given the trial discount rate.
D = Duration
R = (1+DR) where DR is the trial discount rate expressed as a decimal.

Repeat Steps 2 and 3 using this adjusted rate.

3b: If |P-PV| is less than or equal to $1, then the calculation is complete and the yield of the bond equals DR. The program will terminate or will begin processing the next bond in the sample.

improvement over the typical fixed-increment procedure, it is possible to do even better by including a consideration of the magnitude of the error relative to the bond's duration.

Several theoretical and empirical papers have discussed the relationship between duration and price volatility with respect to a change in the yield of a fixed-income investment. For example, Hopewell and Kaufman(8) showed that the price volatility of a coupon bond is a linear function of duration and small changes in the yield structure:

[1] dP/P = -D * dr
where P = Market Price
D = Duration
r = Market Yield

That is, the sensitivity of a bond's price to small changes in the trial discount rate is inversely related to its duration. Consequently, a given change in the trial discount rate will produce a larger (smaller) change in the calculated present value of the remaining cash flows for a bond with a longer (shorter) duration. Thus, including a consideration of the bond's duration in the discount rate adjustment procedure serves to "fine-tune" the adjustment over the simple relative-error method suggested above.

A proportional adjustment which includes a consideration of both the relative magnitude of the error and the duration of the security is as follows:

[2] $$\text{(Adjusted R)} = \text{(Unadjusted R)} * \left[1 - \frac{(P-PV)}{D * PV} \right]$$

where: P = Current Market Price
PV = Present Value of remaining cash flows given the trial discount rate.
D = Duration
R = (1+DR) where DR is the trial discount rate expressed as a decimal.

In the example, the adjusted discount rate (R) which would be used in the first iteration (the second loop through the steps) would be:

1.10 * [1 - ((920-877.1087)/(6.99647 x 877.1087))] = 1.092312

This adjusted discount rate results in PV = $921.7836 and (P-PV) = -$1.7836. Thus, after only one iteration the error is less than two dollars on a market price of $920.

The second iteration, using 9.2612 as the adjusted discount rate, produces an error of less than two cents. Compare this with the 72 iterations necessary to produce a calculated yield with an error of 55 cents using the fixed-increment (1 basis point) procedure described previously!

For the example, the results of two iterations produce are summarized in Table I.

In addition to the obvious improvement in speed and efficiency, the proportional adjustment method also provides a significant improvement in the level of accuracy attainable in the yield calculation. When using the fixed-increment adjustment procedure, the accuracy of the calculated yield is limited to the size of the fixed increment, usually one basis point. However, the level of accuracy in the yield calculation using the proportional adjustment procedure is limited only by the computer hardware. Yields can be calculated accurately to within a hundredth of a basis point or

TABLE I
Results of First Two Iterations
of the Proportional Adjustment Method
using the Example Inputs

Iteration	Unadjusted R	PV	(P-PV)	D	Adjusted R
0	1.100000	877.1087	42.8913	6.99647	1.092312
1	1.092312	921.7836	-1.7836	7.04047	1.092612
2	1.092612	919.9828	0.0172	7.03879	

less and the calculation will still only involve a few iterations. One additional iteration in the example would produce a calculated yield that is accurate to the nearest hundredth of a basis point.(9)

While the preceding discussion has focused on coupon bonds, the proportional-adjustment iterative search procedure can easily be adapted to any situation requiring the computation of yields or rates of return. Unequal cash flows and uneven time intervals do not present a problem (assuming a reasonable level of programming skill). Consequently, this procedure is extremely useful in capital budgeting situations which require the computation of internal rates of return for complex cash flow scenarios.

In summary, the proportional-increment adjustment method presented in this paper offers several important advantages over the traditional fixed-increment adjustment procedure. It is many times faster and much more efficient. It offers a much greater level of accuracy in the resulting yields without a significant increase in the time required for computation. Finally, it is easily adaptable to a diverse set of decision environments which may involve uneven cash flows and unequal discounting periods.

For the convenience of the reader who wishes to take advantage of this procedure, an appendix to this paper is available from the authors which contains a general outline of the procedure including key program statements (in IBM BASIC). It can easily be adapted to other programming languages.

Footnotes

1. Lotus 1-2-3 is a trademark of Lotus Development, Inc. Similar functions are included in most other spreadsheet software packages.

2. Lotus 1-2-3 Release 2.0 goes beyond this with the @IRR function. The user supplies an initial guess for the IRR and the program executes an iterative search procedure similar to the one described in Figure I. While this is certainly an improvement over the capabilities of earlier versions, the calculation of yields or IRR's is still a slow and cumbersome process for large samples of bonds or for projects with cash flows of unequal amounts and uneven timing.

3. Of course, a reasonably adept programmer could extend the program to include a consideration if accrued interest and fractional discounting periods.

4. The actual yield on the bond in the example is 9.260914% (calculated with a financial calculator).

5. The time required for the calculation of yields for coupon bonds could be shortened by treating interim cash flows as an annuity. However, this shortcut is not applicable in situations involving cash flows which are of unequal amounts or which occur at uneven time intervals. In such cases, it is necessary to calculate the present value of each individual cash flow and then sum across all cash flows.

6. Lawrence Fisher. "An Algorithm for Finding Exact Rates of Return". Journal of Business 39(Jan 1966), 111-118. Presented an algorithm that was conceptually similar to the one described here, although the adjustment procedure was less efficient.

7. As defined originally by F. Macauley, the duration of a security is:

$$D = \frac{[\ CF_t * t\ ] \ /\ [\ (1 + r)^t]}{P}$$

where: D = duration; t = time in years; T = maturity in years; CF_t = cash flow at time t; r = yield to maturity; P = current price

Macauley, Frederick R., Some Theoretical Problems Suggested by Movements of Interest Rates, Bond Yields, and Stock Prices in the United States Since 1985, New York: National Bureau of Economic Research, 1938.

8. Hopewell, Michael H. and George G. Kaufman, "Bond Price Volatility and Term to Maturity: A Generalized Respecification"; American Economic Review 62(Sep 1973)749-753.

9. The adjusted discount rate (R) computed at the end of the second iteration would be 1.092609 which would give PV = $920.0008 in the third iteration, or an error of only $.0008. As noted in footnote 4, the actual yield on the example bond is 9.260914%.