The Fundamental Financial Equation.

By Henrik Vestermark (hve@hvks.com)

Abstract:

The Fundamental Financial Equation links the 5 variables. Present value (PV), Future value (FV), period Payment (PMT), number of periods (NP), and the interest rate (IR) together in this equation:

$$\left(PV + \frac{PMT(1 + cIR)}{IR}\right)((1 + IR)^{NP} - 1) + PV + FV = 0$$

That can be useful in several scenarios including mortgage calculation. This paper highlights the equation and devises a method for how to calculate all the other variables when only 4 of the 5 variables are known, Including the Interest rate based on Newton's iteration.

Contents

Introduction to the Fundamental Financial Equation

The Fundamental Financial equation is a financial equation that links the Present Value (PV) of a loan, investment, starting balance, etc., with the Future Value (FV), with a periodical payment (PMT), with the number of periods (NP) to an interest rate of (IR) through the equation:

$$\left(PV + \frac{PMT(1+cIR)}{IR}\right)((1+IR)^{NP} - 1) + PV + FV = 0 \qquad (1)$$

The variables are:

- PV – Present Value of a starting balance of a bank account, Mortgage, Investment, etc.
- FV – Future Value of the above starting balance
- NP – Number of periods. Can be Years, quarters, months, or daily.
- PMT – Periodic payment
- IR – Interest rate per period.
- c – Is a constant that is 1 for interest added at the beginning of a period and 0 for the end of the period.

The PV, FV, and PMT can either be 0, positive, or negative. The sign is used to either indicate taking it out of the account (positive) or negative for paying into the account. The result is the same regardless as long as it is done consistently. So reversing the sign of PF, FV, and PMT yield the same result.

The above formula can also be written as:

$$PV(1+IR)^{NP} + PMT\ (1+cIR)((1+IR)^{NP} - 1)/IR + FV = 0 \qquad (2)$$

Or

$$\frac{PV * IR(1+IR)^{NP} + PMT\ (1+cIR)((1+IR)^{NP} - 1) + FV * IR)}{IR} = 0$$

And other variations. But the basic underline equation is all the same.

Now by knowing 4 out of the 5 variables you can always calculate the 5th variable.

To make it easy we create the following entity:

$$A = (1+IR)^{NP} - 1 \qquad (3)$$

And

$$B = \frac{1+c*IR}{IR} = \frac{1}{IR} + c \tag{4}$$

Substituting (3) and (4) above into (1) you get:

$$PV + (PMT * B) * A + PV + FV = 0 \tag{5}$$

Which solves each variable and gives the following equations:

$$PV = -\frac{FV+A*PMT*B}{A+1}\ Conditions\ IR <> -1 \text{ (-100\%)} \tag{6}$$

$$FV = -(PV + A(PV + PMT * B)) \tag{7}$$

$$PMT = -\frac{FV+PV(A+1)}{A*B}\ Conditions\ IR <> -1\ (-100\%)\ and\ c = 1 \tag{8}$$

$$NP = \frac{\log\left(\frac{PMT*B-FV}{PMT*B+PV}\right)}{\log(1+IR)} = \frac{\log(PMT*B-FV)-\log(PMT*B+PV)}{\log(1+IR)} \tag{9}$$

And then unfortunately we would need to resort to iteration to find the IR based on our fundamental financial equation. More about that later on.

Looking at (1) we see that the only troubling part is the division with IR when IR=0.we will take a closer look at that in special case 1 below.

Looking at (6) we see that we divide with A+1 however $A + 1 = (1 + IR)^{NP} - 1 + 1 = (1 + IR)^{NP}$ which is always <> 0 except for IR=-1. In (8) we have a similar problem since we divide by A*B meaning that neither A nor B can be zero. A is zero for IR=0 since $A = (1 + IR)^{NP} - 1 = (1 + 0)^{NP} - 1 = 0.$ B can be zero only for IR=-1 and c=1. For IR=0 it is undefined but that was already stated that (3) only works for IR<>0.

Lastly, in (9) we have that neither log(1+IR)<>0 nor $\log\left(\frac{PMT*B-FV}{PMT*B+PV}\right) <> 0$. In the first case, we have that IR=0 yields log(1+0)=0 and for IR<=-1 log is undefined since the argument to the log() is <=0 where the log() function is undefined at least in the real domain. In the second case, we need $\frac{PMT*B-FV}{PMT*B+PV} > 0$ otherwise NP can't be calculated.

Special case 1 (IR=0, PMT <> 0):

Now let's consider a special case where IR=0 and PMT is not 0 and we quite easily find

$$PV + NP * PMT + FV = 0 \tag{10}$$

Which solves for PV, FV, PMT, and NP give:

$$PV = -(FV + NP * PMT) \tag{11}$$

$$FV = -(PV + NR * PMT) \tag{12}$$

$$PMT = -\frac{PV+FV}{NP} \tag{13}$$

$$NP = -\frac{PV+FV}{PMT} \tag{14}$$

Special case 2 (IR<>0, PMT=0)

Another special case is where IR is not equal to 0 and PMT is 0:

$$(PV(1 + IR)^{NP} - 1) + PV + FV = 0 \tag{15}$$

Solves for PV, FV, NP, and IR were given using the substation at (3):

$$PV = -\frac{FV}{1+A} \tag{16}$$

$$FV = -PV(1 + A) \tag{17}$$

$$NP = \frac{\log\left(-\frac{FV}{PV}\right)}{\log(1+IR)} \tag{18}$$

$$IR = \left(\frac{FV}{PV}\right)^{\frac{1}{NP}} - 1;\ Condition: PV * FV < 0) \tag{19}$$

Special case 3 (when IR=0 and PMT=0)

This case is not relevant since you have PV+FV=0 and regardless of the number of periods you will also have that PV=-FV or FV=-PV

Interest Compounding & Period Payment Frequency

The interest rate IR is usually specified in annual interest rate in %. However as you have seen from your mortgage these are not compounding on an annually basic, but rather on a monthly, quarterly, or even on even continuous compounding rate. We have also learned from the period payment PMT that typically your mortgage is done every month, but other payment rates can also be seen as quarterly, half-year, etc. so we will introduce two extra variables Periodic Payment frequency (PF) and Interest Compounding frequency (CF) and then distinguish between continuous compounding and discrete compounding with the frequency of CF.

To convert from the nominal interest (IN) or annual interest rate to the IR in our formula (sometimes it is called the effective interest rate) we use the conversion below:

Continuous compounding:

$$IR = \exp\left(\frac{IN}{PF}\right) - 1 \tag{20}$$

Where IN is the nominal or annual rate of interest.

For discrete compounding we need to take both the Compounding frequency CF and Payment frequency PF into account:

$$IR = \left(1 + \frac{IN}{CF}\right)^{\frac{CF}{PF}} - 1 \tag{21}$$

And to convert back to the nominal interest IN we get:
Continuous compounding:

$$IN = \ln((1 + IR)^{PF}) \tag{22}$$

And for discrete compounding:

$$IN = CF((1 + IR)^{\frac{PF}{CF}} - 1) \tag{23}$$

Example for IR calculation

Let's see how it works out using an example. Assuming you use discrete compounding and a frequency for both CF and PF as monthly and a nominal interest rate IN=6%. Since there are 12 months in a year you have:

$$PF = 12, CF = 12, IN = 6\% = 0.06$$

Using (21) you get:

$$IR = \left(1 + \frac{0.04}{12}\right)^{\frac{12}{12}} - 1 = 0.005$$

And to convert it back using

$$IR = 0.005 = 12, CF = 12, PF = 12$$

And discrete compounding (23) you get:

$$IN = 12\left((1 + 0.005)^{\frac{12}{12}} - 1\right) = 0.06 = 6\%$$

Another example of using continuous compounding is:

$$PF = 12, IN = 8\% = 0.08$$

You get:

$$IR = \exp\left(\frac{0.08}{12}\right) - 1 = 0.0067$$

And to convert it back using (22) you get:

$$IN = \ln((1 + 0.0067)^{12}) = 0.08 = 8\%$$

Example of Mortgage calculation of the monthly payment

What is your monthly payment on a fixed 30-years mortgage of $200,000 at an annual nominal rate of 6%? Since it is compounding monthly and the interest rate is accrued monthly at the end we have

$$PF = 12, CF = 12, IN = 6\%, c = 0, PV = 200{,}000, FV = 0, NP = 30 * 12 = 360$$

We have in the previous example already calculated the IR=0.005
Using the formula (8): $PMT = -\frac{FV + PV(A+1)}{A * B}$ where A is given by (3) and B in (4) we get:

$$A = (1 + 0.005)^{360} - 1 = 5.0225$$

$$B = \frac{1 + 0 * 0.005}{0.005} = 200$$

$$PMT = -\frac{0 + 200{,}000(5.0225 + 1)}{5.0225 * 200} = -1199.10$$

The monthly payment PMT is $1,199.10.

If we change it to 20 years mortgage instead of 30 years we get.

$$NP = 20 * 12 = 240$$

$$A = (1 + 0.005)^{240} - 1 = 2.3102$$

$$B = \frac{1+0*0.005}{0.005} = 200$$

$$PMT = -\frac{0+200{,}000(2.3102+1)}{2.3102*200} = -1432.86$$

,

The new monthly payment is $1432.86

Example of Mortgage calculation with a balloon payment at the end

Same example as the last one however instead of paying a monthly payment of $1,432.86 over 20 years you have negotiated a lower monthly rate of $1,300 and instead have to pay a balloon payment at the end after 20 years. What is the value of the balloon payment at the end?

.

$$PF = 12, CF = 12, IN = 6\%, c = 0, PV = 200{,}000, PMT = -1300, NP = 20 * 12 = 240$$

FV is given by (7) as $FV = -(PV + A(PV + PMT * B))$
A & B has previously been found to be A=2.3102 and B=200 so we have:

$$FV = -\big(200{,}000 + 2.3102(200{,}000 - 1300 * 200)\big) = -61{,}388$$

The minus sign indicates that you owe the mortgage company $61,388 after 20 years. (The correct result is $61,387.73 but we only used 4 digits in the calculation of A).

Example of Mortgage calculation with number of periods

What is your monthly payment on a fixed 30-years mortgage of $200,000 at an annual nominal rate of 6%? Since it is compounding monthly and the interest rate is accrued monthly at the end we have

$$PF = 12, CF = 12, IN = 6\%, c = 0, PV = 200{,}000, FV = 0, NP = 30 * 12 = 360$$

We have in a previous example calculated the monthly payment to be $1199.10

Now, what happens if we instead increase our monthly payment to $1300?

We have in the previous example already calculated the IR=0.005 and the new PMT is -1300.

Using the formula (9): $NP = \frac{\log\left(\frac{PMT*B-FV}{PMT*B+PV}\right)}{\log(1+IR)}$ and B is given in (4) we get:

$$B = \frac{1+0*0.005}{0.005} = 200$$

$$NP = \frac{\log\left(\frac{-1300*200-0}{-1300*200+200000}\right)}{\log(1+0.005)} = \frac{\log(4.3333)}{\log(1.005)} = 294$$

The new number of periods is now 294 months or a shortening of the 30 years fixed mortgage with 5½ years.

Finding the nominal interest rate per year through iteration

This is the most interesting chapter and the reason I wrote this paper. In our fundamental equation:

$$\left(PV + \frac{PMT(1+cIR)}{IR}\right)((1+IR)^{NP} - 1) + PV + FV = 0$$

There is no way to find the interest IR as an equation and instead to solve the equation for IR we need to resort to the classic Newton iterations. Newton Iterations is defined as the number of repeated uses of the below iterations.

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)} \text{ Or } x_{n+1} = x_n - d_n \; where \; d_n = \frac{f(x)}{f'(x)}$$

Now before we can use this we first need to rework our equation so it will be easier to apply the Newton iteration. Substitute x for IR and name IR=i we get:

$$f(i) = \left(PV + \frac{PMT(1+ci)}{i}\right)((1+i)^{NP} - 1) + PV + FV$$

Which can also be rewritten into an alternative form as:

$$f(i) = \frac{PMT(1+ci)(1+i)^{NP} - PMT(1+ci) + iFV + iPV(1+i)^{NP}}{i}$$

Now with f(i) above we can find the derivate as:

$$f'(i) = \frac{((1+i)^{NP}-1)(-PMT(1+ci)) + ni(1+i)^{(NP-1)}\ (PMT(1+ci)+iPV)}{i^2}$$

We therefore get:

$$i_{n+1} = i_n + \frac{\dfrac{PMT(1+ci)(1+i)^{NP} - PMT(1+ci) + iFV + iPV(1+i)^{NP}}{i_n}}{\dfrac{((1+i)^{NP}-1)(-P\quad(1+ci)) + n\ (1+i)^{(NP-1)}\ (PMT(1+ci)+iPV)}{{i_n}^2}}$$

Or

$$i_{n+1} = i_n + i_n \frac{PMT(1+ci)(1+i)^{NP} - PMT(1+ci) + iFV + iPV(1+i)^{NP}}{((1+i)^{NP}-1)(-PMT(1+ci)) + ni(1+i)^{(NP-1)}\ (PMT(1+ci)+iPV)}$$

To see how it works in real life we can try to find interest in the following scenario.

Example: A 30-year maturity bond with a face value of $1000. The coupon rate of 8% annually. It sells for $900. Interest is accrued at the end of the period. We therefore have

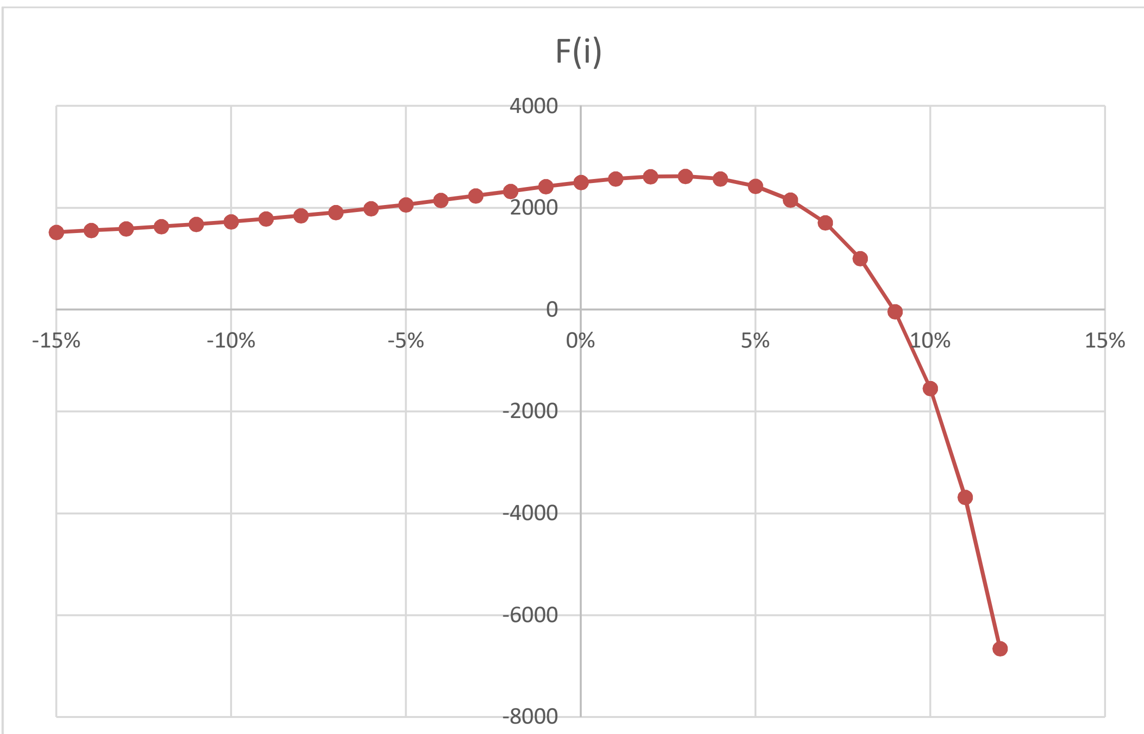
$$NP = 30, FV = 1000, PV = -900, PMT = 8\%\ of\ \$1000 = 80,$$

Start with a guess of i_0=1% and get the following iterations.

n	I	F(i)	F'(i)	dx
0	1.000%	2,569.73	5,970	4.30E-01
1	-42.0426%	1,190.28	453	2.63E+00
2	-305.0355%	(2,096,943,689,911.92)	30,662,207,805,072	-6.84E-02
3	-298.1966%	(758,378,102,955.59)	11,471,817,147,157	-6.61E-02
4	-291.5858%	(274,274,032,389.87)	4,292,013,875,073	-6.39E-02
5	-285.1955%	(99,193,562,252.03)	1,605,795,098,231	-6.18E-02
6	-279.0183%	(35,874,195,371.97)	600,785,234,349	-5.97E-02
7	-273.0470%	(12,974,204,330.23)	224,775,250,484	-5.77E-02
8	-267.2750%	(4,692,228,676.37)	84,096,488,526	-5.58E-02
9	-261.6954%	(1,696,982,831.08)	31,463,527,011	-5.39E-02
10	-256.3019%	(613,727,370.38)	11,771,646,881	-5.21E-02
11	-251.0883%	(221,959,165.01)	4,404,205,991	-5.04E-02
12	-246.0486%	(80,273,022.90)	1,647,780,736	-4.87E-02
13	-241.1770%	(29,031,096.04)	616,502,583	-4.71E-02
14	-236.4680%	(10,499,047.87)	230,664,295	-4.55E-02
15	-231.9163%	(3,796,787.67)	86,308,448	-4.40E-02
16	-227.5172%	(1,372,863.36)	32,299,962	-4.25E-02
17	-223.2669%	(496,232.55)	12,093,726	-4.10E-02
18	-219.1637%	(179,192.48)	4,534,159	-3.95E-02
19	-215.2116%	(64,533.12)	1,706,196	-3.78E-02

20	-211.4293%	(23,067.51)	648,544	-3.56E-02
21	-207.8725%	(8,076.40)	253,329	-3.19E-02
22	-204.6844%	(2,669.13)	106,212	-2.51E-02
23	-202.1714%	(750.25)	52,536	-1.43E-02
24	-200.7433%	(133.82)	34,945	-3.83E-03
25	-200.3604%	(7.11)	31,296	-2.27E-04
26	-200.3377%	(0.02)	31,092	-7.48E-07
27	-200.3376%	(0.00)	31,091	-8.08E-12
28	-200.3376%	(0.00)	31,091	-1.17E-16

And after 28 iterations we end up with a solution for IR= -200.33%, while the correct solution is 8.97%. What exactly happens here?
If we take a look at the f(i) between -15% to 12% we get the following plot of the function f(i):



With a starting guess of i_0=1%, we see that the tangent of f'(i) is positive and we get a huge step towards the left and our next iteration starting point is i_1=-42%. Since f'(i) at that point is also positive (see previous table) we get another huge step towards left as we end up with i_2=-305%, now f(i_2) turns negative while f'(i_2) remains positive and we now move towards the right to i_3=-298%. This continues until i_{28} where we get f(i_{28})=0 and the iteration stops at i_{28}=-200.33%. A starting guess of i_0=1% was not desirable since we didn't get to the right solution for i.

Now let's see what happens if we instead choose a starting guess of i_0=3%.

n	I	F(i)	F'(i)	dx
0	3.000%	2,621.50	(1,970)	-1.33E+00
1	136.0842%	(130,892,094,140,252.00)	(1,670,010,893,194,660)	7.84E-02
2	128.2464%	(47,333,112,076,215.30)	(624,880,411,164,816)	7.57E-02
3	120.6717%	(17,116,241,452,970.80)	(233,820,804,449,557)	7.32E-02
4	113.3514%	(6,189,302,659,306.10)	(87,494,317,620,514)	7.07E-02
5	106.2775%	(2,238,014,131,122.31)	(32,740,802,164,227)	6.84E-02
6	99.4419%	(809,224,331,324.50)	(12,252,212,171,840)	6.60E-02
7	92.8372%	(292,587,963,047.92)	(4,585,210,252,243)	6.38E-02
8	86.4561%	(105,784,176,555.85)	(1,716,044,588,858)	6.16E-02
9	80.2916%	(38,243,312,317.04)	(642,286,972,280)	5.95E-02
10	74.3374%	(13,824,597,902.70)	(240,419,607,864)	5.75E-02
11	68.5872%	(4,996,896,752.82)	(90,004,295,247)	5.55E-02
12	63.0354%	(1,805,856,746.36)	(33,699,733,930)	5.36E-02
13	57.6767%	(652,497,404.21)	(12,620,727,483)	5.17E-02
14	52.5067%	(235,697,010.71)	(4,727,956,311)	4.99E-02
15	47.5215%	(85,105,954.67)	(1,771,939,795)	4.80E-02
16	42.7185%	(30,712,839.27)	(664,504,313)	4.62E-02
17	38.0966%	(11,074,152.25)	(249,437,881)	4.44E-02
18	33.6569%	(3,987,693.84)	(93,775,190)	4.25E-02
19	29.4045%	(1,432,775.58)	(35,344,404)	4.05E-02
20	25.3508%	(512,818.14)	(13,382,081)	3.83E-02
21	21.5187%	(182,228.85)	(5,110,634)	3.57E-02
22	17.9530%	(63,817.70)	(1,986,378)	3.21E-02
23	14.7402%	(21,646.29)	(802,037)	2.70E-02
24	12.0413%	(6,802.04)	(352,742)	1.93E-02
25	10.1130%	(1,751.38)	(186,348)	9.40E-03
26	9.1731%	(260.15)	(133,446)	1.95E-03
27	8.9782%	(9.09)	(124,213)	7.32E-05
28	8.9709%	(0.01)	(123,877)	9.93E-08
29	8.9708%	(0.00)	(123,876)	1.83E-13

This time we get to the right solution after 29 iterations at i_{29}=8.97%, but not without a wild swing of interest I, going from 3% to 136% at the first iteration and slowly going back to the correct solution after 29 iterations. This is also very classic behavior since we observed that f(i) goes towards a larger negative number and the tangent (f'(i)) goes close to parallel with f(x) and therefore it only procedure small steps back towards the right solution.

I like this example since it highlights two potential issues with the newton iteration:

1) First a poorly chosen starting point can set us off in the wrong direction resulting in a false solution to the problem.

2) Controlling the step size is needed to avoid wild swing and slow convergence to the right solution

To cope with this we modify the newton iteration by applying the following safeguards:

1) We determine which direction the iteration is going by calculating NP_0 using an interest of 0. If $NP_0<NP$) then the direction is towards positive interest. If on the other hand $NP_0>NP$ then the direction is toward negative interest.
2) We then chose i_0 using some approximate value and ensure if the direction is less than 0 then the starting value should also be less than 0 otherwise positive interest is the starting point.
3) If a step size put us in the wrong direction we simply ignore the iterations step and multiply the original start guess with 2. If the next step also put us in the wrong direction we now multiply the original start guess with 4, then 8, etc. until we are moving in the right direction.
4) If a step size $di_n = \frac{f(i_n)}{f'(i_n)}$ results in an increase of the function value $f(i_{n+1})$: $f(i_{n+1}) > f(i_n)$ then we continue to reduce the step $di_n = di_n/2$ until $f(i_{n+1}) <= f(i_n)$ or we have done it 3 times in a row. In this case, we stop and use the new i_{n+1} as the new starting point for the next iteration.

This algorithm works pretty efficiently and we get the correct solution after only 7 iterations as seen below:

n	I	Comment
0	1.6800%	
1	3.3640%	Safeguards rule 1. Doubling the initial guess of 1.68%
2	10.9550%	Safeguards rule 2. Reduced step size with a factor of 8
3	9.5152%	
4	9.0211%	
5	8.9713%	
6	8.9708%	
7	8.9708%	

I have seen many financial calculators that sometimes go wrong in the iteration process for finding the interest rate. With these safeguards mentioned above you will get a lot more robust and accurate solution. All the above algorithm has been implemented in the financial calculator found at http://www.hvks.com/Numerical/webfinance.html

Reference

1. GnuCash – Chapter 8.4 Calculation https://www.gnucash.org/docs/v2.6/C/gnucash-guide/loans_calcs1.html Sep 19, 2016
2. P. Lutus – Finance Calculator http://arachnoid.com/finance/index.html Version 4.5 2015.11.23
3. http://www.hvks.com/Numerical/webfinance.html, version 2.3 Sep 19, 2016