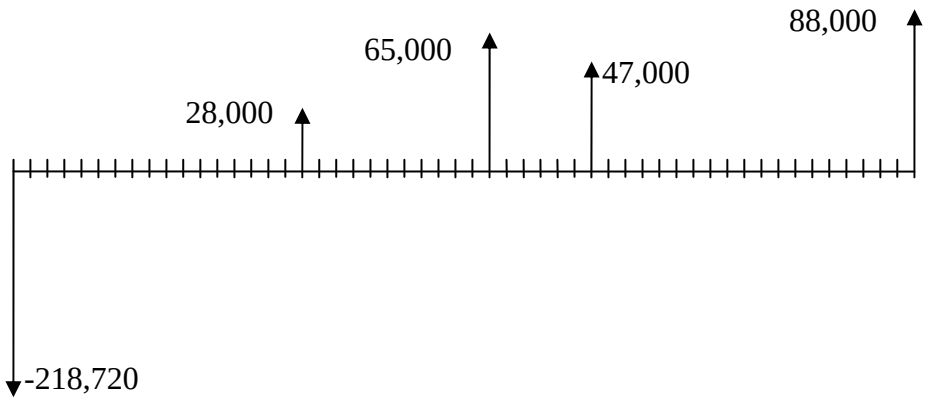
SOLVING IRR<0.4% WITH AN HP12C PLATINUM

Because of some complaints, it's been observed that the IRR routine for the HP12C Platinum miscalculates IRR values below 0.35%, while the NPV routine works perfectly fine. Based on these facts, a program listed below can be used as a workaround.

An example

Consider the following cash flow diagram.



The sequence to enter this data and to compute IRR in either the regular HP12C or the HP12C Platinum is:

Keystrokes	HP12C Display	HP12C Platinum Display
f CLEAR REG	0.00	0.00
218720 CHS g CFo	-218,720.00	-218,720.00
0 g CFj 16 g Nj	16.00	16.00
28000 g CFj	28,000.00	28,000.00
0 g CFj 10 g Nj	10.00	10.00
65000 g CFj	65,000.00	65,000.00
0 g CFj 5 g Nj	5.00	5.00
47000 g CFj	47,000.00	47,000.00
0 g CFj 18 g Nj	18.00	18.00
88000 g CFj	88,000.00	88,000.00
f IRR	0.11	1.191000-10
f CLEAR PREFIX	11.10498480	119.1000000

A full-precision IRR obtained with an HP17BII in this case is **0.111049848097**, meaning the HP12C Platinum returns a wrong value while the regular HP12C goes correct. In order to check the NPV in both calculators, each resulting IRR is used with each calculator to compute NPV:

Keystrokes	HP12C Display	HP12C Platinum Display
[i] [f] [NPV]	0.00001	9,280.00
[f] [CLEAR] [PREFIX]	7600000000	9280000000

In fact, when using the interest rate obtained with the HP12C Platinum to compute the NPV, we obtain 9,280.00 in both calculators. Also, if we use the IRR obtained with the HP12C as an interest rate, we obtain the correct NPV in the HP12C Platinum.

Keystrokes	HP12C Display	HP12C Platinum Display
.111049848 [i] [f] [NPV]	0.00001	0.00001
[f] [CLEAR] [PREFIX]	7600000000	7813670000

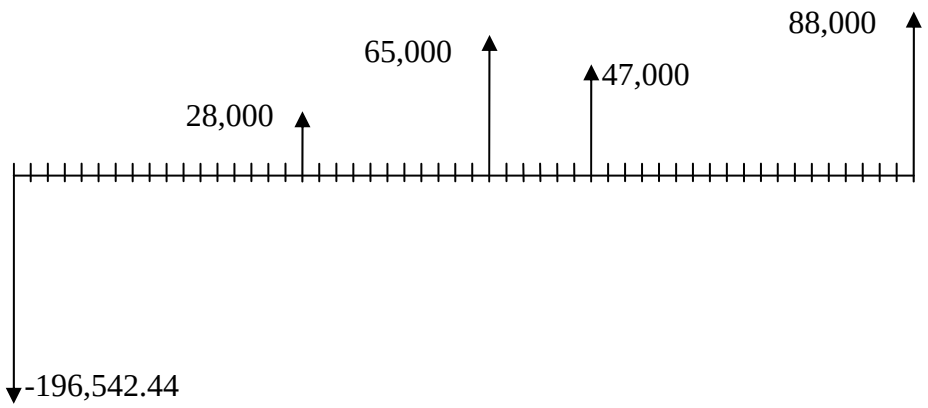
With the 10-digit IRR obtained above, the HP17BII returns **NPV=7.9710000×10^{-1}**, so the HP12C Platinum is actually closer than the HP12C. If we slightly change the conditions of the problem to achieve a 0.4% IRR, the actual NPV should be:

Keystrokes	HP12C Display	HP12C Platinum Display
.4 [i] [f] [NPV]	-22,177.56	-22,177.56
[f] [CLEAR] [PREFIX]	2217755985	2217755985

Incorporating this NPV to CF_0 and computing IRR again, it should lead us to the correct interest rate. (consider that the previous value is in the display):

Keystrokes	HP12C Display	HP12C Platinum Display
[STO] [−] [0] [f] [IRR]	0.40	0.40
[f] [CLEAR] [PREFIX]	399999993	399999993

Now the cash flow diagram for this problem is:



For any given cash flow when the IRR is above 0.4%, the HP12C Platinum internal routines compute it correctly. Also, the HP12C Platinum computes NPV closer to actual values than the HP12C when low interest rates are given.

A program to compute IRR below 0.4%

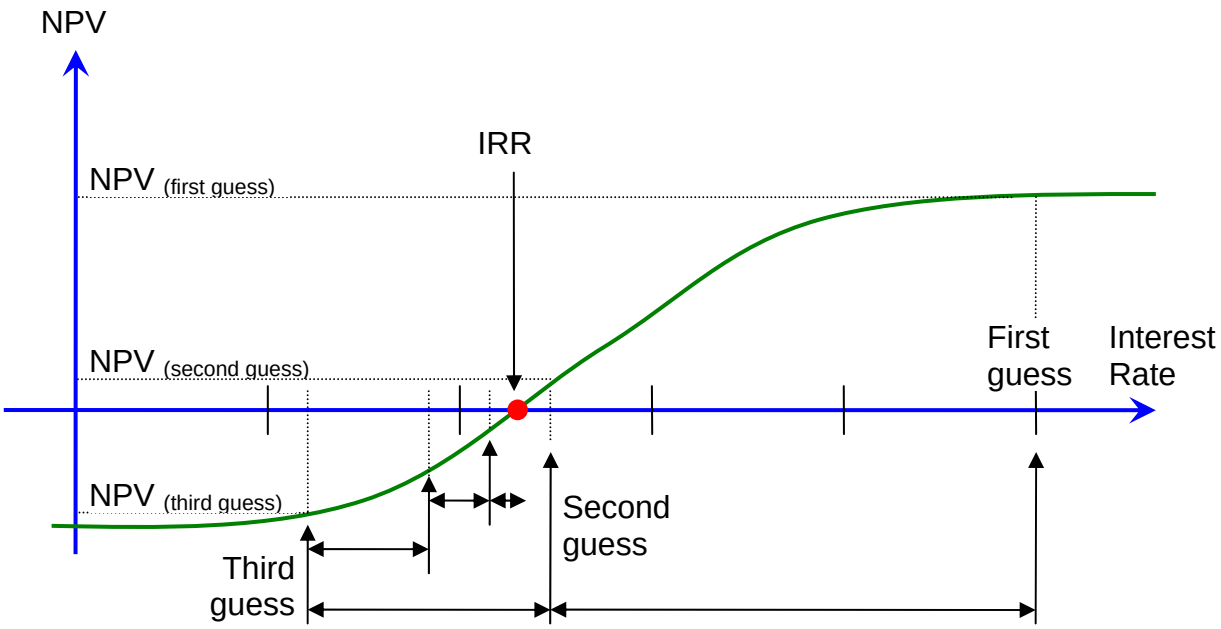
001	STO PMT	001, 44 14
002	f NPV	002, 42 13
003	ENTER	003, 36
004	RCL i	004, 45 12
005	2	005, 2
006	÷	006, 10
007	i	007, 12
008	f NPV	008, 42 13
009	f RND	009, 42 14
010	g x=0	010, 43 35
011	g GTO 029	011,43,33,029
012	X	012, 20
013	g LSTx	013, 43 40
014	x≷y	014, 34
015	ENTER	015, 36

016	g x^2	016, 43 20
017	g $\sqrt{x}$	017, 43 21
018	÷	018, 10
019	RCL i	019, 45 12
020	RCL PMT	020, 45 14
021	−	021, 30
022	2	022, 2
023	÷	023, 10
024	X	024, 20
025	RCL i	025, 45 12
026	STO PMT	026, 44 14
027	+	027, 40
028	g GTO 007	028,43,33,007
029	RCL i	029, 45 12

This program computes IRR between a given estimated value and zero.

Program description

The graphics below illustrates the program operation for a supposed IRR, below 0.35%



The following sequence describes the program operation.

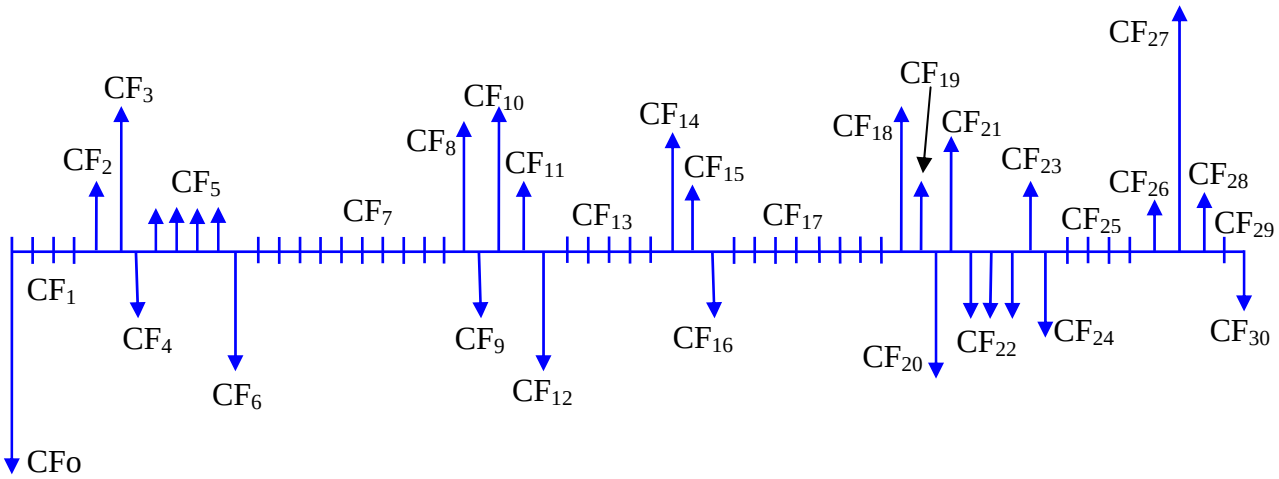
Step#	Comments
001 to 002	Considering a first guess for IRR (G1) of any value in the display and already stored as interest rate in [i], the program stores G1 in PMT (last guessed IRR) and computes NPV_{G1}.
003 to 006	The second guess (G2) is computed as (G1)/2.
007	Currently guessed IRR (Gn) is stored in [i]
008 to 011	NPV_{Gn} is computed and rounded to current display format. If The rounded resulting NPV_{Gn} is equal to zero, currently guessed IRR is valid and program execution jumps to step# 029.
012 to 018	If NPV_{Gn} is not equal to zero, the program tests if there was a change in signal from NPV_{Gn-1} to NPV_{Gn}. Either a 1 or a -1 (S) is obtained to signalize this change.
019 to 024	Δ_{IRR} is computed as: $\Delta_{IRR} = S \times \frac{G_n - G_{n-1}}{2}$
025 to 027	The newly guessed IRR is computed as: $NEW_{Gn} = G_n + \Delta_{IRR}$. Current G_n is now the last guessed IRR, and is stored in [PMT]
028	Program execution jumps to step# 007 where newly computed IRR is stored in [i]. The newly computed IRR is now G_n, and the program continues execution back in step #007 till the newly computed IRR causes NPV to be close enough to zero.
029	This program line is executed only when rounded NPV equals zero to recall G_n to the display. G_n is, then, IRR.

Using the program

To use the program after keying it in the calculator memory, simply key in the guessed interest rate (always greater than the expected IRR), press [i] and run the program from step # 000. If data from previous example is yet stored in calculator memory, please follow the next steps:

Keystrokes	Display	Comments
218720 [CHS]		
[STO] 0	-218,720.00	key in and stores original contents of CFo
[.] 5 [i]	0.50	key in and stores first guess for IRR
[g] [GTO] 000	0.50	positions in step # 000
[R/S]	running	starts program execution
	0.11	program stops and shows IRR
[f] CLEAR [PREFIX]	11.10498908	full precision IRR computed by the program

Let's take an example with more cash flow data.



CF0	-13197	CF1	0 N=3	CF2	1000	CF3	2800	CF4	-1000	CF5	550 N=4
CF6	-1800	CF7	0 N=10	CF8	2500	CF9	-1000	CF10	2800	CF11	1000
CF12	-1800	CF13	0 N=5	CF14	2000	CF15	1200	CF16	-1000	CF17	0 N=8
CF18	1800	CF19	1200	CF20	-1800	CF21	1200	CF22	-1000 N=3	CF23	1500
CF24	-1300	CF25	0 N=4	CF26	800	CF27	4500	CF28	1100	CF29	0
CF30	-950										

Cash-flow data from CF_1 to CF_{30} are taken from the movement of a private savings account. In order to create some additional cash-flow information and to force the HP12C Platinum to compute a low IRR, the data is first loaded in an HP17BII so NPV is computed for a given interest rate of 0.18%. The HP17BII returns **`NPV=13,197.82`**. So, the HP17BII is loaded with **CF_0`=-13,197.00`** and then it computes **`IRR%=0.18`** (full precision: **`1.80205188055E-1`**). Then the HP12C Platinum is loaded with the given cash-flow data.

Keystrokes	Display	Comments
f CLEAR REG	0.00	Initializes registers
13197 CHS g CFo	-13,197.00	stores CF_0= -13,197.00
0 g CFj 3 g Nj	3.00	stores CF_1= 0.00 and N_1= 3
1000 g CFj	1,000.00	stores CF_2= 1,000.00
...		stores all subsequent CF_j and N_j data

After storing all data, computing IRR with the HP12C Platinum leads to the wrong figure.

Keystrokes	HP12C Platinum Display
f IRR	1.351000-10
f CLEAR PREFIX	1351000000

Without changing anything, the NPV is now computed for this IRR:

Keystrokes	HP12C Platinum Display
f NPV	753.00
f CLEAR PREFIX	7530000000

Having the program to compute IRR with low values still stored in the HP12C Platinum, the keystrokes to use it and compare results are:

Keystrokes	Display	Comments
•5 i	0.50	key in and stores first guess for IRR
g GTO 000	0.50	positions in step # 000
R/S	running	starts program execution
	0.18	computed IRR (~ 3 min 35 sec)
f CLEAR PREFIX	1802062990	full precision IRR computed by the program

The percent change obtained when comparing the HP12C Platinum computed IRR and the one obtained with the HP17BII is approximately 0.00062%. After executing the program, the corresponding NPV is available either with RCL PV or g LSTx.

Keystrokes	HP12C Platinum Display
RCL PV	-0.004
f CLEAR PREFIX	4456484937

Remarks

- The program does not succeed finding an existing IRR if the first guess is between IRR and zero; although it is possible to add a checkpoint for such cases, the program runs forever if there is no IRR between first guess and zero;
- The program may not succeed finding an existing IRR between first guess and zero if there is an even number of possible IRR values between this first guess and zero.
- Using different first guesses may cause the program to find a solution faster;
- As it happens with f IRR, display format influences on program final results.

by Luiz Cláudio Vieira – UNIPAC - Brazil