Investment Performance on the HP-12C

Tony Hutchins, #1049

Peter O. Dietz was an American investment analyst who wrote a book in 1966 about the investment performance of pension funds. He died in 1990 and since then his name has been associated with performance measures. In June 2006 www.hpmuseum.org was asked for a "Modified Dietz" program for the 12C. That is what lines 1-38 do below. For ***single*** modified Dietz returns it could finish at ***line 30***(see Appendix). For n=0 or 1 lines 39-61 do the so-called pure "time weighted" rate of return (***TWRR***), which does not involve any explicit time input at all. It does however require a market valuation just before (n=0) or just after (n=1) each and every cashflow point. For n=.5, the same lines do the "midpoint Dietz" method, also called the "original Dietz" method which is none other than the old "200I/(A+B-I)" method (see Datafile V24N5P33). The first 17 lines create a (daily product)/100 in R_2 - this ***could*** be used to calculate the interest amount charged on a bank loan: e.g. the interest may be i%/365*R_2 (RCL i RCL 2 × 365 ÷) or, input the interest amount and RCL 2 ÷ 365 × to calculate an i% p.a. The sign convention here is that deposits are positive, and withdrawals negative.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
f P/R		STO + 3	21-44 40 3	RCL n	43-45 11
f CLEAR PRGM	00-	g GTO 06	22-43,33 06	×	44- 20
RCL 0	01-45 0	R/S	23- 31	−	45- 30
0	02- 0	RCL 3	24-45 3	RCL FV	46-45 15
STO 1	03-44 1	−	25- 30	g LSTx	47-43 36
STO 2	04-44 2	STO + 3	26-44 40 3	−	48- 30
g DATE	05-43 16	RCL 2	27-45 2	Δ%	49- 24
R/S	06- 31	÷	28- 10	R/S	50- 31
0	07- 0	RCL 1	29-45 1	RCL i	51-45 12
g DATE	08-43 16	×	30- 20	%	52- 25
RCL 0	09-45 0	R/S	31- 31	g LSTx	53-43 36
x≷y	10- 34	RCL i	32-45 12	+	54- 40
STO 0	11-44 0	%	33- 25	+	55- 40
g ΔDYS	12-43 26	g LSTx	34-43 36	i	56- 12
g PSE	13-43 31	+	35- 40	RCL FV	57-45 15
STO + 1	14-44 40 1	+	36- 40	PV	58- 13
RCL 3	15-45 3	i	37- 12	RCL i	59-45 12
%	16- 25	g GTO 00	38-43,33 00	R/S	60- 31
STO + 2	17-44 40 2	RCL PV	39-45 13	g GTO 39	61-43,33 39
R/S	18- 31	RCL PMT	40-45 14	f P/R	
g x=0	19-43 35	+	41- 40	This should also work on a 12cp in RPN mode.	
g GTO 23	20-43,33 23	g LSTx	42-43 36		

In the following example data columns (3) and (5) show the less detailed data used in some methods. Fund managers cannot always provide the detail in column (4).

Date (D.MY) (1)	Cashflow (2)	MarketValue (3)	MarketValue (4)	Cashflow (5)
31.122005	0	1,000	1,000	
15.012006	100		1,010	
15.022006	150		1,120	450
15.032006	200		1,300	
31.032006	0	1,600	1,600	
15.042006	100		1,650	
15.052006	150		1,800	450
15.062006	200		1,900	
30.062006	0	2,000	2,000	

Market values here *exclude* cashflow on the same date so n=0 for the pure TWRR.

Program	Start Line	Data Cols.	Initialization. 0 [i]
Modified Dietz	0	1 ,2 and 3	31.122005[STO]0 1000[STO]3 [R/S]
Pure TWRR	39	2 and 4	0[n] 1000[PV] 0[PMT] 1010[FV]
Midpoint Dietz	39	3 and 5	0.5[n] 1000[PV] 450[PMT] 1600[FV]

Modified Dietz: R_1=accumulated days, R_3=balance. At an ***interest calculation point*** use a ***zero cashflow*** and input the market value at the next [R/S]. At the start of ***each*** new period the extra [R/S] is required for initialisation. As a check, pauses show the ***day of the week*** for each date, and the ***days between dates***. Bad date input [R/S] can be ***undone*** with [g][GTO]07, input previous date [R/S] [g][GTO]07.

Modified Dietz.[g][D.MY] 0[i] [f][PRGM]	**Pure TWRR** 0[n] 0[i]
31.122005[STO]0 1000[STO]3 [R/S]→31.12	1000[PV] 0[PMT] 1010[FV]
15.012006[R/S]→150.00 100[R/S]→100.00	[g][GTO]39 [R/S]→1.00 [R/S]→1.00
15.022006[R/S]→341.00 150[R/S]→150.00	100[PMT]1120[FV] [R/S]→0.90 [R/S]→1.91
15.032006[R/S]→350.00 200[R/S]→200.00	150[PMT]1300[FV] [R/S]→2.36 [R/S]→4.32
31.032006[R/S]→232.00 0[R/S]→0.00	200[PMT]1600[FV] [R/S]→6.67[R/S]→**11.27**
1600[R/S]→**12.58**[R/S]→12.58[R/S]→31.03	0[PMT]1650[FV] [R/S]→3.13 [R/S]→14.75
15.042006[R/S]→240.00 100[R/S]→100.00	100[PMT]1800[FV] [R/S]→2.86[R/S]→18.03
15.052006[R/S]→510.00 150[R/S]→150.00	150[PMT]1900[FV] [R/S]→-2.56[R/S]→15.00
15.062006[R/S]→573.50 200[R/S]→200.00	200[PMT]2000[FV] [R/S]→-4.76 [R/S]→**9.52**
30.062006[R/S]→307.50 0[R/S]→0.00	Note how the answers in bold (1 and 2 qtr. period returns) vary slightly for each method.
2000[R/S]→-2.79 [R/S]→**9.44**	

Midpoint Dietz: 0.5[n] 0[i] 1000[PV] 450[PMT] 1600[FV] [g][GTO]39[R/S]→ **12.24** [R/S] → 12.24. 450[PMT] 2000[FV] [R/S]→-2.74 [R/S]→**9.17**. Quick! ***Linking*** the Dietz ***MWRR*** (money weighted rates of return) creates a ***pseudo*** TWRR only. ***Pure*** TWRR is the ideal as it is a market like return on a single lump sum, and is perfect for ***comparison*** with ***market indices***, which were pioneered as performance benchmarks by Frank Russell, which was where Peter Dietz was working when he wrote his book. An *investor's* individual return also depends on cashflow timing which is precisely what the pure TWRR excludes.

What in *investor* wants to know is the IRR on *his* account as this can be compared with an alternative savings account return. Fund managers attempt to calculate a TWRR but GIPS (Global Investment Performance Standards) strangely still allows the Dietz MWRR to be linked to make a hybrid MW/TWRR. At the *investor* level however, the Dietz MWRR is a fairly reliable guess of the IRR - providing the fund does not more than say double or halve in size. The first 12 lines of the following code do a simple Dietz for n=.5. n is the fraction of the period when the cashflow is deemed to occur. E.g. .5[n] 10000[PV] 12000[FV] 1500[PMT] [R/S] →4.65%. n=0 corresponds to cashflow at the beginning of the period, and n=1 at the end. 0[n][R/S]→4.35%, 1[n][R/S]→5.00%. Actually this shows us *another way* - just use TVM, twice: 1[n] 12000[CHS][FV] [g][BEG] [i]→4.35 [g][END][i]→5.00 [+]2[÷] →4.67. This is close enough to the 4.65. 4.65 is theoretically the ***harmonic*** mean of the 5.00 and 4.35. These 8 lines convert X and Y to their harmonic mean in X and the arithmetic mean in Y: [%] [x≷y] [g][LSTx] [+] 2 [÷] [x≷y] [%T]. This way we see the *range* of returns (extreme "time weighted" returns) as well as the harmonic midpoint return. We have rhythm and harmony :-)

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
[f][P/R]		[RCL][i]	13-45 12	[RCL][PMT]	27-45 14
[f] CLEAR [PRGM]	00-	[RCL][PMT]	14-45 14	[RCL][n]	28-45 11
[RCL][PV]	01-45 13	[×]	15- 20	[×]	29- 20
[RCL][PMT]	02-45 14	[g][LSTx]	16-43 36	[−]	30- 30
[+]	03- 40	[RCL][PV]	17-45 13	[g][LSTx]	31-43 36
[RCL][n]	04-45 11	[+]	18- 40	[x≷y]	32- 34
[g][LSTx]	05-43 36	[RCL][i]	19-45 12	[RCL][PV]	33-45 13
[×]	06- 20	[%]	20- 25	[+]	34- 40
[−]	07- 30	[+]	21- 40	[RCL][i]	35-45 12
[RCL][FV]	08-45 15	[RCL][FV]	22-45 15	[%]	36- 25
[g][LSTx]	09-43 36	[−]	23- 30	[+]	37- 40
[−]	10- 30	[%T]	24- 23	[+]	38- 40
[Δ%]	11- 24	[g][GTO]00	25-43,33 00	[g][GTO]00	39-43,33 00
[g][GTO]00	12-43,33 00	[RCL][PMT]	26-45 14	[f][P/R]	

Sign conventions: PMT you deposit is positive. PV is positive (considered a deposit). FV is also positive here - but negative if TVM or IRR used *directly*.

1+i=(FV-f*PMT)/(PV+(1-f)*PMT), where f is the fraction in n. Only the first 12 lines are really necessary but it can be useful to resolve for f or FV:

f=((PV+PMT)*(1+i)-FV)/i/PMT solves for f, [g][GTO]13 [R/S]

FV=(PV+(1-f)*PMT)*(1+i)+f*PMT solves for FV, [g][GTO]26 [R/S]

If f is not a positive fraction then something is wrong with the input. Finally:

PV=(FV-PMT(1+(1-f)*i))/(1+i) & PMT=(FV-PV*(1+i))/(1+(1-f)*i).

Appendix: Two Modified Dietz Programs.

Date	Balance	Date	Deposit	Date	Withdl.	Date	Deposit
12.312004	$10,000	3.312005	$1,000	6.302005	-$500	9.302005	$1,000
12.312005	$12,000	From www.usatoday.com "Ask Matt" money column (Feb.06)					

The source of the above example is accompanied by a 12C IRR calculation.

First program

Works for ***any number*** of cashflows (the example above has only 3). The weekday is shown for each date - a 6 or 7 means a weekend (e.g. 12.312005), which would generally be suspect. Also the days traversed in each period are shown - see the example below. Bad date input, R/S can be immediately ***undone*** with g GTO 07, input ***previous date*** R/S g GTO 07, and now input the correct date and R/S.

Cash flows can be input out of order, however the display is more meaningful if they are input in strict order. The last date ***must*** however be input last (along with the zero cash flow signifying, to the program<G>, it is the last date). Here the final value is stored in FV as part of the initialisation.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
f P/R		x≷y	10- 34	STO + 3	21-44 40 3
f CLEAR PRGM	00-	STO 0	11-44 0	g GTO 06	22-43,33 06
RCL 0	01-45 0	g ΔDYS	12-43 26	RCL FV	23-45 15
0	02- 0	g PSE	13-43 31	RCL 3	24-45 3
STO 1	03-44 1	STO + 1	14-44 40 1	−	25- 30
STO 2	04-44 2	RCL 3	15-45 3	RCL 2	26-45 2
g DATE	05-43 16	%	16- 25	÷	27- 10
R/S	06- 31	STO + 2	17-44 40 2	RCL 1	28-45 1
0	07- 0	R/S	18- 31	×	29- 20
g DATE	08-43 16	g x=0	19-43 35	g GTO 00	30-43,33 00
RCL 0	09-45 0	g GTO 23	20-43,33 23	f P/R	

Items in quotes are displayed momentarily.

f PRGM g M.DY 12.312004 STO 0 10000 STO 3 12000 FV R/S→ "12,31,2004 5"

3.312005 R/S→"3,31,2005 4"	"90"	9,000.00	1000 R/S→1,000.00
6.302005 R/S→"6,30,2005 4"	"91"	10,010.00	500 CHS R/S→-500.00
9.302005 R/S→"9,30,2005 5"	"92"	9,660.00	1000 R/S→1,000.00
12.312005 R/S→"12,31,2005 6"	"92"	10,580.00	0 R/S→4.649682 (ans)
Total days→	365	39,250.00	← total interest at 1%/day

Only 4 numbered registers are used, and one financial register (FV).

RCL 0→12.312005, the final date. RCL 1→365, total of days traversed.

RCL 2→39,250.00 total of daily products (interest at the high rate of 1% per day).

RCL 3→11,500.00 the initial balance plus the total of the cash flows.

To get ready for the next period, first we could store the answer in i, and then:

RCL FV STO 3, input new final value FV R/S→"12,31,2005 6", and away we go.

Second program

All data is ***pre-stored*** and the number of cash flows is limited. On the 12C the 41 line program means 15 registers are free (refer Datafile V23N2pp9-10). Data for 7 dates can be stored (dates in R_0,R_2,...$R_{.2}$, corresponding amounts in R_1,R_3,...,$R_{.3}$=0) so we have room for 5 cash flows besides the opening and closing balances. A 99 line program would leave room for just one cash flow. The stored data is not changed so error correction is easy - just change the stored data and re-run.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
f P/R		x≷y	14- 34	+	29- 40
f CLEAR PRGM	00-	g ΔDYS	15-43 26	PMT	30- 14
CLx	01- 35	RCL PMT	16-45 14	g GTO 06	31-43,33 06
PV	02- 13	%	17- 25	RCL 0	32-45 0
STO n	03-44 11	RCL PV	18-45 13	RCL g CFj	33-45,43 14
RCL 1	04-45 1	+	19- 40	g ΔDYS	34-43 26
PMT	05- 14	PV	20- 13	RCL FV	35-45 15
RCL n	06-45 11	RCL n	21-45 11	RCL PMT	36-45 14
2	07- 2	4	22- 4	−	37- 30
+	08- 40	+	23- 40	×	38- 20
n	09- 11	n	24- 11	RCL PV	39-45 13
RCL g CFj	10-45,43 14	RCL g CFj	25-45,43 14	÷	40- 10
RCL g CFj	11-45,43 14	g x=0	26-43 35	g GTO 00	41-43,33 00
R↓	12- 33	g GTO 32	27-43,33 32	f P/R	
RCL g CFj	13-45,43 14	RCL PMT	28-45 14		

The example can be done as follows: f PRGM g M.DY

12.312004 g CFo 10000 g CFj	3.312005 g CFj 1000 g CFj
6.302005 g CFj 500 CHS g CFj	9.302005 g CFj 1000 g CFj
12.312005 g CFj 0 g CFj	12000 FV R/S →4.649682 (ans)

The "running" time here is 22 seconds on the 12C and only 4 seconds on the new 12cp! Total days are not stored. PV holds the daily product and PMT holds the accumulated balance. n is used to access the data. i is kept free. 3 lines could be added to show progress:0 g DATE after line 10, and g PSE after line 15. As before the last "cashflow" is input as zero to ***signal*** that the date is the last date and the final value is stored in FV. On the first 12cp (with under 240 program lines) we could store 13 cashflows for the period - on the new 12cp we can store 38 - useful for checking the monthly interest rate on a revolving mortgage which doubles as a cheque account, after a busy month. Cashflows on the same day can be aggregated or input separately with the same date. At last we have found a ***great application for the new 12cp!*** So, the above applies equally to a loan or a savings account where simple interest is used to calculate interest at each compounding point. The modern "Modified Dietz" is just a ***simple interest rate calculation***, something bankers have been doing for centuries! Such is the mystique of finance, knowing all 13 names for each process (after John Ball).

Second program *plus daily IRR calculation* for 12c platinum.

Matt Krantz had the great idea of making the 12C do the ***daily*** IRR, in addition to the daily simple return. The second program has been modified to store the daily simple return (in [PV]) and the total days (in [PMT]). For [IRR] the data must be in strict date order, with no duplicate dates - payments on the same day ***must*** be pre-totalled. If there are more than 99 days without a cashflow then a dummy tiny cashflow of E-99 (i.e. [EEX]99[CHS]) should be input to keep the sub periods under 100 days. Start with [f]CLEAR[REG] and after finding the simple return as before: just [g][GTO]048 [R/S]→0.012459(IRR). The original data is now ***all*** overwritten! The effective rate ***for the period*** is: 1[RCL][i][%][+][RCL][PMT][y^x]1[x≷y][Δ%]→4.652143%.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
[f][P/R]		[RCL][g][CFj]	028,45,43 14	[RCL][g][CFj]	057,45,43 14
[f]CLEAR[PRGM]	000,	[g][x=0]	029,43 35	[RCL][g][CFj]	058,45,43 14
[CLx]	001, 35	[g][GTO]035	030,43,33,035	[x≷y]	059, 34
[PV]	002, 13	[RCL][PMT]	031,45 14	[RCL][g][CFj]	060,45,43 14
[STO][n]	003,44 11	[+]	032, 40	[x≷y]	061, 34
[RCL] 1	004,45 1	[PMT]	033, 14	[g][ΔDYS]	062,43 26
[PMT]	005, 14	[g][GTO]006	034,43,33,006	[x≷y]	063, 34
[RCL][n]	006,45 11	[RCL] 0	035,45 0	[R↓]	064, 33
2	007, 2	[RCL][g][CFj]	036,45,43 14	[x≷y]	065, 34
[+]	008, 40	[g][ΔDYS]	037,43 26	[g][CFj]	066,43 14
[n]	009, 11	[RCL][FV]	038,45 15	[CLx]	067, 35
[RCL][g][CFj]	010,45,43 14	[RCL][PMT]	039,45 14	[g][CFj]	068,43 14
0	011, 0	[−]	040, 30	[R↓]	069, 33
[g][DATE]	012,43 16	[RCL][PV]	041,45 13	1	070, 1
[RCL][g][CFj]	013,45,43 14	[÷]	042, 10	[−]	071, 30
[R↓]	014, 33	[PV]	043, 13	[g][Nj]	072,43 15
[RCL][g][CFj]	015,45,43 14	[x≷y]	044, 34	[RCL][n]	073,45 11
[x≷y]	016, 34	[PMT]	045, 14	2	074, 2
[g][ΔDYS]	017,43 26	[×]	046, 20	[+]	075, 40
[g][PSE]	018,43 31	[g][GTO]000	047,43,33,000	[n]	076, 11
[RCL][PMT]	019,45 14	1	048, 1	[g][GTO]050	077,43,33,050
[%]	020, 25	[n]	049, 11	[RCL][g][CFj]	078,45,43 14
[RCL][PV]	021,45 13	[RCL][g][CFj]	050,45,43 14	[RCL][FV]	079,45 15
[+]	022, 40	[g][x=0]	051,43 35	[CHS]	080, 16
[PV]	023, 13	[g][GTO]078	052,43,33,078	[g][CFj]	081,43 14
[RCL][n]	024,45 11	[RCL][n]	053,45 11	[RCL][PV]	082,45 13
4	025, 4	2	054, 2	[RCL][g][R/S]	083,45,43 31
[+]	026, 40	[+]	055, 40	[g][GTO]000	084,43,33,000
[n]	027, 11	[n]	056, 11	[f][P/R]	The End.

[RCL][PV](0.012739)[RCL][PMT][×]→4.649682% recovers the original simple return. This program, with up to 38 cashflows is an ideal use for the new 12CPA .

Daily IRR calculation **for the HP-12C.**

We adopt a different paradigm - converting the timed cashflows ***directly into the IRR format*** with a program - lines 1-21 below. The FV is now considered as a withdrawal, signwise. Nice and short! It runs like this: f PRGM g M.DY

C.Flow		Date		Date with DOW	days	IRR n
12000	ENTER	12.312004	R/S →			0.00
1000	ENTER	3.312005	R/S →	"3,31,2005 4"	"90"	2.00
500 CHS	ENTER	6.302005	R/S →	"6,30,2005 4"	"91"	4.00
1000	ENTER	9.302005	R/S →	"9,30,2005 5"	"92"	6.00
12000 CHS	ENTER	12.312005	R/S →	"12,31,2005 6"	"92"	8.00

f IRR →0.012459 (25 sec. on 12C, 4 sec. on new 12cpt- but on the new 12cpt we need to either clear regs before starting or add 2 more lines to ensure Nj=1 when amounts are stored.) RCL PV RCL PMT g ΔDYS PMT →365, and as before, 1 RCL i % + RCL PMT y^x 1 x≷y Δ% →4.652143%. If there are more than 99 days without a cashflow then a ***zero*** cashflow can now be used. The dates are lost, converted to zero CFj with Nj in days, but we can still extract the modified Dietz ***directly from the IRR data*** with: g GTO 22 R/S →.012739 and RCL PMT × →4.649682% recovers the original simple return. This program can be enhanced to cope with repeating non-zero cashflows by inserting 8 lines after line 32 (new lines 33-40): g LSTx × g LSTx + 2 ÷ +, adding Nj(Nj+1)/2 to the previous PMT (a day count), before multiplying by CFj. Our examples don't require the 8 extra as CFj=0 when Nj>1. The final CFj needs to have Nj=1.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
f P/R		g Nj	17-43 15	+	35- 40
f CLEAR PRGM	00-	R↓	18- 33	RCL n	36-45 11
PV	01- 13	R↓	19- 33	1	37- 1
STO PMT	02-44 14	g CFj	20-43 14	+	38- 40
R↓	03- 33	g GTO 05	21-43,33 05	g x=0	39-43 35
g CFo	04-43 13	RCL n	22-45 11	g GTO 43	40-43,33 43
RCL n	05-45 11	PV	23- 13	R↓	41- 33
R/S	06- 31	RCL g CFj	24-45,43 14	g GTO 28	42-43,33 28
0	07- 0	CLx	25- 35	i	43- 12
g CFj	08-43 14	PMT	26- 14	RCL PV	44-45 13
g DATE	09-43 16	ENTER	27- 36	n	45- 11
RCL PMT	10-45 14	RCL PMT	28-45 14	f NPV	46-42 13
x≷y	11- 34	RCL PMT	29-45 14	CHS	47- 16
PMT	12- 14	RCL g Nj	30-45,43 15	%T	48- 23
g ΔDYS	13-43 26	+	31- 40	g GTO 00	49-43,33 00
g PSE	14-43 31	PMT	32- 14	f P/R	
1	15- 1	RCL g CFj	33-45,43 14		
−	16- 30	×	34- 20		

It is interesting how this gets a quick ***initial guess*** for the IRR. Here is how it looks, stand alone on the HP-12C, with the extra 8 lines. It is only 36 lines, and I found it quite hard to write. There are so few free resources once IRR takes over all numbered registers, including possibly [FV], and also uses [n]. Instead of accumulating the non-final [CFj] as before (I could have used i for that) I just do an NPV with i=0 to get the interest. PV is needed just to preserve [n], and [PMT] is used to remember the duration backwards from the last [CFj]. The daily product is kept in the stack - at line 9 the stack is full to the brim with essential data! [RCL][PMT] at the end to see the total number of periods involved.

Keystrokes	Display	Keystrokes	Display	Keystrokes	Display
[f][P/R]		[g][LSTx]	12-43 36	[+]	25- 40
[f] CLEAR [PRGM]	00-	[ENTER]	13- 36	[g][x=0]	26-43 35
[RCL][n]	01-45 11	[×]	14- 20	[g][GTO]30	27-43,33 30
[PV]	02- 13	[g][LSTx]	15-43 36	[R↓]	28- 33
[RCL][g][CFj]	03-45,43 14	[+]	16- 40	[g][GTO]07	29-43,33 07
[CLx]	04- 35	2	17- 2	[i]	30- 12
[PMT]	05- 14	[÷]	18- 10	[RCL][PV]	31-45 13
[ENTER]	06- 36	[+]	19- 40	[n]	32- 11
[RCL][PMT]	07-45 14	[RCL][g][CFj]	20-45,43 14	[f][NPV]	33-42 13
[RCL][PMT]	08-45 14	[×]	21- 20	[CHS]	34- 16
[RCL][g][Nj]	09-45,43 15	[+]	22- 40	[%T]	35- 23
[+]	10- 40	[RCL][n]	23-45 11	[g][GTO]00	36-43,33 00
[PMT]	11- 14	1	24- 1	[f][P/R]	

The results tend to be better with savings cash flows, rather than loans.

Savings example: 10 payments of $1,000 accumulate to $15,000. What is the IRR? We can do this with TVM: 10[n] 0[PV][g][BEG] 1000[PMT] 15000[CHS][FV][i]→7.26. With IRR: 1000[g][CFo] 10[g][Nj] 15000[CHS][g][CFj] [f][IRR]→7.26. We can also now just press [R/S] with the above program and get 9.09 in about 4 seconds. The IRR takes about 12 seconds.

Loan example: $7000, is repaid by 10 payments of $1,000. What is the IRR? With TVM: 10[n] 7000[CHS][PV] [g][END]1000[PMT] 0[FV][i]→7.07. With IRR: 7000[CHS][g][CFo] 1000[g][CFj] 9[g][Nj] 1000[g][CFj] [f][IRR]→7.07. Note how we had to split the final 1000 payment out separately for our program to work. This is its only peculiarity. It targets a *single* final value. [R/S]→12.00.

Perpetuity example: $5,000 is repaid by 10 payments of $1,000, plus a final repayment of $5,000. We know the answer<G>, but what is the IRR? With TVM: 10[n] 5000[CHS][PV] [g][END]1000[PMT] 5000[FV][i]→20.00. With IRR: 5000[CHS] [g][CFo] 1000[g][CFj] 9[g][Nj] 6000[g][CFj] [f][IRR]→20.00. But [R/S]→ 200.00. This extreme example is designed to show the limitations of the modified Dietz as a ***general*** initial guess for [i]. So, interpret the modified Dietz with ***great*** care. It is quite beyond me how the fund manager TWRR "standard" allows the Dietz to be linked to form a pseudo TWRR.