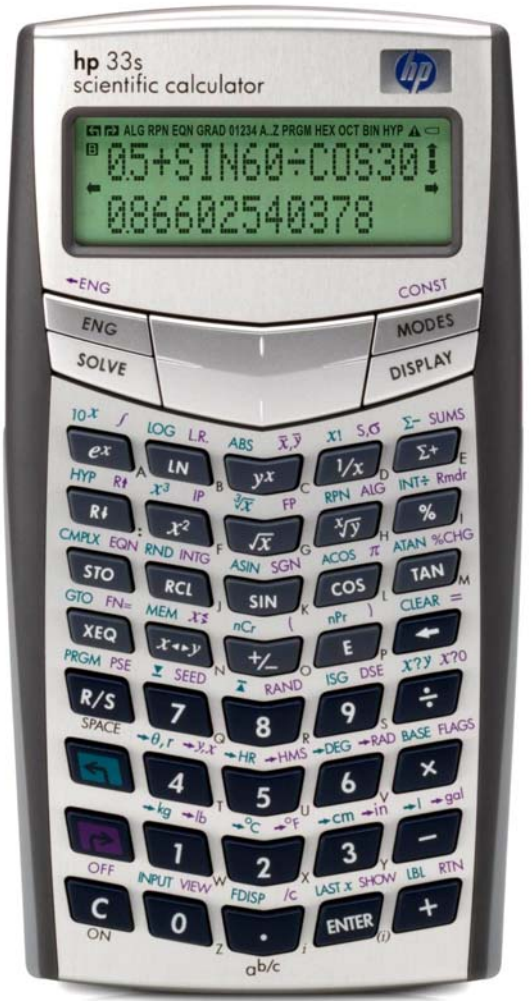
hp calculators



HP 33S Using the HP Solver – Part 2

Using the HP Solver

Practice Example: Solving an Expression in a Program

Practice Example: Solving a Program with Several Variables

Practice Example: Using the Solver in a Program

Using the HP Solver

Part 1 of this training aid showed examples of how the HP Solver can find roots of an expression **f**(x) and used those examples to show the main features of the Solver. All the examples used the Solver to find roots of expressions typed in the HP 33S Equation mode.

This second part provides further details of the HP Solver and shows how the Solver can be used with programs.

It is worth noting that the Solver can be used to:

Find exact roots.

Find approximate roots.

Find discontinuities that cross zero and find poles of functions.

Find non-zero minimum values of functions.

Find crossing points of pairs of functions.

The last is true because an expression such as:

$$g(x) = h(x)$$

where g(x) and h(x) are two functions of the variable x can be rewritten as

$$f(x) = g(x) - h(x) = 0$$

Then the root x of **f**(x) at which **f**(x) = 0 is the value of x at which g(x) and h(x) are equal.

Practice Example: Solving an Expression inside a Program

Each time an expression is to be solved, the user needs to select Equation mode. The HP 33S Equation mode can store a whole collection of different expressions. Therefore a user who wishes to solve one of several different expressions also needs to step through the list to the expression to be solved. It is quicker to store an expression in a program and solve that program by name. The special HP 33S key [FN=] is used to specify which program is to be solved.

Example 1: Solve for the roots of $X^2 - 4$ by putting it in a program.

Solution: First, enter a program that includes the expression $X^2 - 4$ as an equation. The program needs to be given a label that is not used by any other program. For this example, use the label T, for Test. If a program with label T already exists, a different label can be used. Press [↰] [PRGM] to enter Program mode. If necessary, step to a position in program memory where a new program can be entered, or else press [↰] [GTO] [.] [.] to go to the top of program memory. Then type [↰] [LBL] [T] to enter the program label. When [↰] [LBL] is pressed, the symbol A..Z appears at the top of the screen to remind the user to press one of the keys with letters to their lower right. Press [4] to enter the letter T.

In summary, press the keys:

[↰] [PRGM] [↰] [GTO] [.] [.] [↰] [LBL] [T]

Figure 1

Now type the expression $X^2 - 4$ in Equation mode. Press [↱] [EQN] to start Equation mode. Type [RCL] [X] [y^x] [2] [−] [4] [ENTER] to enter the expression. When [RCL] is pressed, the A..Z symbol is displayed again – press the [2] key, with the letter X at its lower right, to enter the symbol X. Finally press [↱] [RTN] to enter the Return command that ends the program.

In summary, press:

[↱] [EQN] [RCL] [X] [y^x] [2] [−] [4] [ENTER] [↱] [RTN]

Figure 2

The program is now stored. Press [↰] [PRGM] to leave program entry mode. Store a guess of 0 in X by pressing [0] [STO] [X]. Specify that the program which contains the function to solve is the program with the label "T". To do this, press [↱] [FN=] [T]. Press [SOLVE] [X] to solve function T for the variable X.

In summary, press the keys:

[↰] [PRGM] [0] [STO] [X] [↱] [FN=] [T] [SOLVE] [X]

Figure 3

Answer: The root is 2, as shown in Figure 3. Instead of looking for the equation in the list in Equation mode, it is possible to enter the expression as part of a program, then use [FN=] to identify the program to be solved. When only one expression is to be solved this has little advantage – it is particularly useful when several equations are to be solved in turn.

Note 1: If the equation is displayed when this program is run, and is not solved, then flag 10 has been set. When flag 10 is set, an equation is treated as a message to be displayed, not as an equation to be evaluated. This can be very useful, for example flag 10 can be set, a message typed in Equation mode can be displayed, then flag 10 can be cleared and the equation to be solved can be typed and solved. To make sure the equation above is solved and not displayed clear flag 10 by pressing the keys [↱] [FLAGS] [1] [.] [0].

Note 2: If the equation $X^2 = 4$ is solved in Equation mode, a direct solution is found, giving only the positive root 2. When the Solver is given a program to solve, it does not try to find a direct solution, so by choosing appropriate guesses you can find other solutions, -2 in this example, or solutions that are not principal values of inverse trig functions.

Practice Example: Solving a Program

There is no need for the expression to be typed in Equation mode. Equation mode has the advantage that it works the same in RPN mode and in algebraic mode, so a program such as that in Example 1 will work in either mode. Programs can do more though, for example they can include tests for special conditions, or "loops" that repeat an action several times. A very simple example will be used here, it is written so that it works both in RPN mode and in algebraic mode.

Example 2: Solve for the roots of $1/Y + 3Y^2 - 4$ by putting it in a program. To avoid a "divide by zero" error, make the program test if Y is zero, and if it is zero, replace it with a very small number. The example below does this; it uses label U for the program name, and the variables Y and Z. Different names can be used, but in that case the program checksum shown at the end will be different.

Solution: First, begin the program with a label. Following label T for the program above, this example will use label U. Instead of an expression to calculate the function, the program must now evaluate each part of the expression and calculate the result. Begin by recalling Y

Press the keys:

[↰] [PRGM] [↰] [GTO] [.] [.] [↰] [LBL] [U] [RCL] [Y]

Figure 4

Now calculate Y^2 and store it in register Z. To do this, press the following keys:

[x^2] [STO] [Z] [3] [STO] [×] [Z] [2] [STO] [−] [Z]

Figure 5

Then multiply register Z by 3. This uses "storage register arithmetic", a topic covered in detail in a separate training aid. Press the keys:

[3] [STO] [×] [Z]

Figure 6

Use another storage arithmetic command to subtract 4 from Z.

4 STO − Z

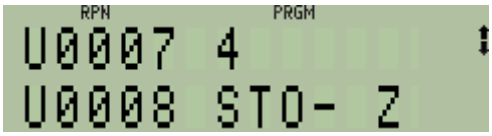


Figure 7

Now comes the tricky bit, which can not be done in Equation mode, recall Y again, and check if it is zero. To check if a number is zero, use the x?0 key to show a menu of possible tests. Then press 6 to select the sixth test in the menu, x=0?

RCL Y ▶ x?0 6 2

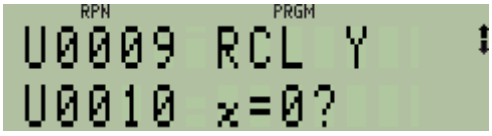


Figure 8

This test checks if "x", which is the number currently in use, is zero. If it is, replace it with the smallest possible non-zero number, so that 1/x will not cause a "divide by zero" error. This program step, entering 1E-499, is carried out only if the test is true. That follows the HP 33S program test rule "do if true", described in the training aid on programming. Then calculate 1/x, either for the number recalled from Y, or for the number that replaced it.

E +/− 4 9 9 1/x

Figure 9

To complete the calculation, recall the rest of the expression from register Z and add it to the current number, which is now 1/Y or 1E499 if Y was zero. This combined recall and addition is another register arithmetic command. Finally end the program with a Return step.

RCL + Z RTN

Figure 10

This program is much longer than the previous one. It can be worth checking if it has been typed correctly before using it. Press ▶ PRGM to leave Program mode, then ▶ MEM to display the Memory menu, and 2 to select the list of programs. Press the up or down keys below the screen until LBL U is shown. The program length is shown too.

Figure 11

Press [↗] [SHOW] to see the program checksum, which is 943E if the program has been typed correctly. If the program length or checksum are different, then either the program has been typed wrongly, or the program label or variable names are different, maybe because you were already using U, Y or Z for another purpose. Press the Cancel button at the lower left to cancel the above display.

At last, the program can be used. Store a zero in Y to make sure the special case will be tested, specify that the function to be solved is called U, and solve for Y. Press the keys:

[C] [0] [STO] [Y] [↗] [FN=] [U] [SOLVE] [Y]

Figure 12

Answer: The root is 1, as shown in Figure 12. This can be confirmed by checking the original expression, though it might have been difficult to deduce just by looking at the expression.

The program used the x=0? test to avoid the risk of a division by zero, though the Solver automatically tries to avoid this in Equation mode. The same method can be used to test for more difficult potential problems, for example the arc sine of a number larger than one. By using register arithmetic the program works both in RPN and in algebraic modes.

Practice Example: Solving a Program with Several Variables

If the Solver is used to solve a program with more than one variable, then the special [INPUT] key can be used to ask the user for values of known variables. Equation mode asks for unknown variables automatically, but in programs the INPUT command is used to do the same.

Example 3: Repeat the calculation of the formula for the volume of a cylindrical can, from Example 3 of the first part, but this time in a program. The expression to be used is $\pi r^2 h - V$ as in the first part. Give the program the name V, note that the program name V is independent of the variable name V used in the program.

See the next page for the program listing.

Solution: Press the following keys to enter the RPN mode program shown:

Keys to press	Step number	Program step	Explanation
[↰] [PRGM]			select program entry mode
[↰] [GTO] [.] [.]			go to top of program memory
[↰] [LBL] [V]	V0001	LBL V	put a label at the beginning, V for Volume
[↰] [INPUT] [V]	V0002	INPUT V	ask for the value of the variable V
[↰] [INPUT] [R]	V0003	INPUT R	ask for the value of R
[↰] [INPUT] [H]	V0004	INPUT H	ask for the value of H
[RCL] [R]	V0005	RCL R	get the value of R
[x^2]	V0006	x²	get the radius squared
[↱] [π]	V0007	π	
[×]	V0008	×	multiply π by R²
[RCL] [×] [H]	V0009	RCL× H	get the value of the cylinder height and multiply πR² by it
[RCL] [−] [V]	V0010	RCL- V	get the volume and subtract it from πR²H
[↱] [RTN]	V0011	RTN	end the program
[↰] [PRGM]			leave program entry mode

To make the program work in algebraic mode, exchange steps V0007 and V0008, and add an [ENTER] after V0009.

Use the program to calculate the volume of a can with radius 2 cm and height 10 cm. First select the program V for solving by pressing [FN=] [V], then solve for the variable V by pressing [SOLVE] [V]. The Solver recognizes that V is the unknown variable so it does not ask for the value of V. Instead, it skips the INPUT V step and immediately asks for the value of R.

Figure 13

Give the radius by typing [2] [R/S]. The Solver asks for the other variable, H. Press [1] [0] [R/S]. The Solver has now asked for all the INPUT values and begins to search for a solution for V. It will repeatedly use the program with different values of V but it will ignore the INPUT steps after the first time the program is used. When a solution is found, it is displayed:

Figure 14

The volume of a can with radius 2cm and height 10cm is about 125.66 cubic cm. Example 3 of part 1 asked what the radius should be if a volume of 100 cc is required and if the height is to stay at 10 cm. Store the new volume by pressing [1] [0] [0] [STO] [V]. Now that the required height and volume are already stored, there will be no need to type new values for them when INPUT asks.

To calculate the radius, press [SOLVE] [R]. When Solver asks for the value of H, just press [R/S].

Figure 15

Press R/S again when asked for H, and the Solver gives the required value for V:

Figure 16

Answer: As already shown in Example 3 of part 1, the cans should have a radius of 1.78 cm.

Note: If there is no need to ask for the values of variables, because they are always stored before the program is used, then there is no need to put INPUT steps into the program. It is enough to store the values and then to press SOLVE and the name of the unknown variable. This is the normal way that the Solver is used in programs as shown in the next example. To avoid prompting for values when an equation is used in a program, set flag 11 before the equation is used. To do this, press ↗ FLAGS 1 . 1. Flag 11 is automatically cleared by Solve, so it should be set inside any program that wants to avoid prompting.

Practice Example: Using the Solver in a Program

The Solve command can itself be used in programs. This allows Solve to be used in programs that solve an expression as part of a bigger task. For example, the above program could be used to find the radius of a can for a given volume, then the total surface area of the can could be calculated from that radius, to tell the user how much metal is needed to make such a can.

To use the Solver in a program do the following:

Write a program such as one of the above, to be used with the Solver.

If the program is to be fully automated, avoid INPUT steps, or set flag 11, as described in the note above.

Write a second program which includes a FN= step and a SOLVE step. The FN= must be followed by the name of the first program, and SOLVE must be followed by the name of the variable to solve for.

Store the known values of the variables not to be solved for, before the SOLVE step.

When the program runs, the SOLVE step does not display the root found. Instead it puts the results of Solve in the unknown variable and on the stack (or in the x1, x2, x3 menu in algebraic mode, these commands can be included in a program if the menu is shown and then ENTER is pressed) and acts as a "do if true" programming command. If an exact solution is found, the program step immediately after the SOLVE step is executed. If an exact solution is not found, the next step is skipped. By putting a GTO (go to) step immediately after SOLVE, the user can make the program continue from a different point if SOLVE was successful. Otherwise the program continues from the next step and can check the results returned by SOLVE and choose what to do next.

Example 4: Repeat the calculation of the formula for the volume of a cylindrical can, from Example 3, but this time using program V in another program. Find the radius required to give a volume of 200 cubic centimeters.

Solution: First change program V by deleting the INPUT steps. Select Program mode, use the cursor keys to step to line V0004, then use the ← to delete each of the lines V0004, V0003 and V0002. Then enter this new program:

Keys to press	Step number	Program step	Explanation
↑ GTO . .			go to top of program memory
↑ LBL W	W0001	LBL W	put a label at the beginning, W comes after V
2 0 0	W0002	200	type the number 200
STO V	W0003	STO V	store it as the known value of V
→ FN= V	W0004	FN=V	specify that V is the program to solve
SOLVE R	W0005	SOLVE R	start the Solver, looking for R to give a root of program V
→ RTN	W0006	RTN	end the program
→ RTN	W0007	RTN	a second RTN in case SOLVE skips the first
↑ PRGM			leave program entry mode

Run program W by pressing XEQ W. The program stores 200 in variable V, specifies that program V is the one to find a root for, then starts the Solver.

Figure 17

When the Solver has found a root it stops and allows the program to continue running. Note that there are two RTN steps after SOLVE R. This is in case the Solver does not find an exact answer. In that case the program would skip the first return, and continue from the next step after it. This would probably be the beginning of a new program if the second return step were not there. When the program stops running it just displays the final value, which is the required value of R in this case, but could be something calculated from R. Therefore R= is not displayed on the upper line of the screen.

Figure 18

Answer: For a volume of 200 cubic centimeters and a height of 10 cm, the can must have a radius of 2.52 cm, to two decimal places.

Program W works in RPN and in algebraic mode, but the correct version of program V, RPN or algebraic, must be used.

The Solver is a powerful HP 33S tool that can be used in many ways. See the HP 33S manual for further information, including Solver details in Appendix D.