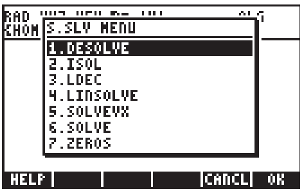
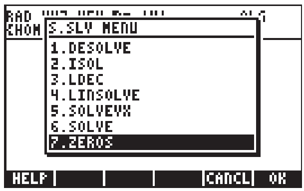
Chapter 6
Solution to single equations

In this chapter we feature those functions that the calculator provides for solving single equations of the form f(X) = 0. Associated with the 7 key there are two menus of equation-solving functions, the Symbolic SOLVer (↰ S.SLV), and the NUMerical SoLVer (↱ NUM.SLV). Following, we present some of the functions contained in these menus. Change CAS mode to Complex for these exercises (see Chapter 2).

Symbolic solution of algebraic equations

Here we describe some of the functions from the Symbolic Solver menu. Activate the menu by using the keystroke combination . With system flag 117 set to CHOOSE boxes, the following menu lists will be available:

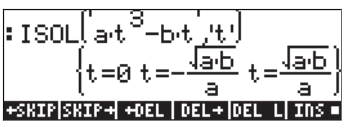




Functions DESOLVE and LDEC are used for the solution of differential equations, the subject of a different chapter, and therefore will not be presented here. Similarly, function LINSOLVE relates to the solution of multiple linear equations, and it will be presented in a different chapter. Functions ISOL and SOLVE can be used to solve for any unknown in a polynomial equation. Function SOLVEVX solves a polynomial equation where the unknown is the default CAS variable VX (typically set to 'X'). Finally, function ZEROS provides the zeros, or roots, of a polynomial. Entries for all the functions in the S.SLV menu, except ISOL, are available through the CAS help facility
(TOOL NXT HELP).

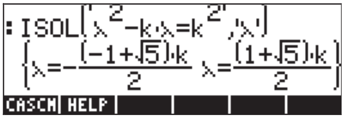
Function ISOL

Function ISOL(Equation, variable) will produce the solution(s) to *Equation* by isolating *variable*. For example, with the calculator set to ALG mode, to solve for *t* in the equation at^3-bt = 0 we can use the following:



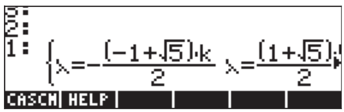
Using the RPN mode, the solution is accomplished by entering the equation in the stack, followed by the variable, before entering function ISOL. Right before the execution of ISOL, the RPN stack should look as in the figure to the left. After applying ISOL, the result is shown in the figure to the right:

The first argument in ISOL can be an expression, as shown above, or an equation. For example, in ALG mode, try:



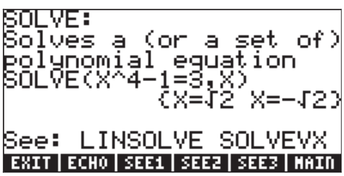
Note: To type the equal sign (=) in an equation, use ⮣ = (associated with the +/- key).

The same problem can be solved in RPN mode as illustrated below (figures show the RPN stack before and after the application of function ISOL):

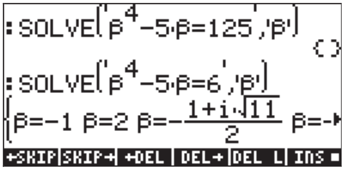


Function SOLVE

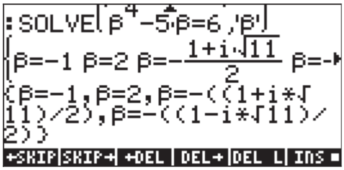
Function SOLVE has the same syntax as function ISOL, except that SOLVE can also be used to solve a set of polynomial equations. The help-facility entry for function SOLVE, with the solution to equation X^4 – 1 = 3 , is shown next:



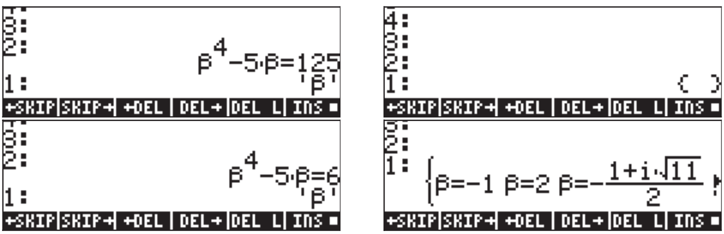
The following examples show the use of function SOLVE in ALG and RPN modes:



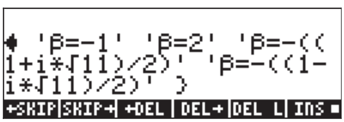
The screen shot shown above displays two solutions. In the first one, β^4-5β =125, SOLVE produces no solutions { }. In the second one, β^4 - 5β = 6, SOLVE produces four solutions, shown in the last output line. The very last solution is not visible because the result occupies more characters than the width of the calculator's screen. However, you can still see all the solutions by using the down arrow key (⌄), which triggers the line editor (this operation can be used to access any output line that is wider than the calculator's screen):



The corresponding RPN screens for these two examples, before and after the application of function SOLVE, are shown next:

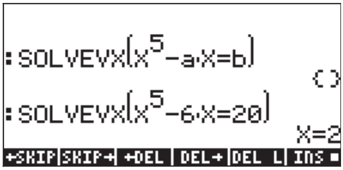


Use of the down-arrow key (⌄) in this mode will launch the line editor:



Function SOLVEVX

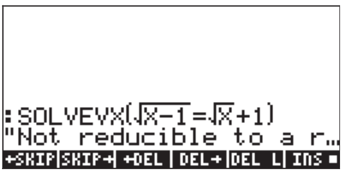
The function SOLVEVX solves an equation for the default CAS variable contained in the reserved variable name VX. By default, this variable is set to 'X'. Examples, using the ALG mode with VX = 'X', are shown below:

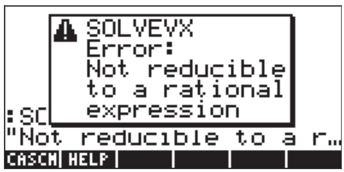


In the first case SOLVEVX could not find a solution. In the second case, SOLVEVX found a single solution, X = 2.

The following screens show the RPN stack for solving the two examples shown above (before and after application of SOLVEVX):

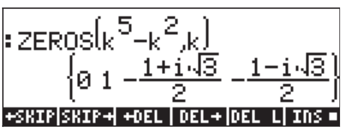
The equation used as argument for function SOLVEVX must be reducible to a rational expression. For example, the following equation will not processed by SOLVEVX:

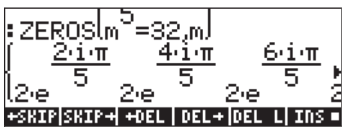




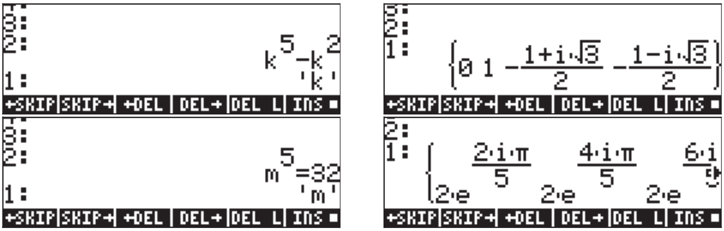
Function ZEROS

The function ZEROS finds the solutions of a polynomial equation, without showing their multiplicity. The function requires having as input the expression for the equation and the name of the variable to solve for. Examples in ALG mode are shown next:





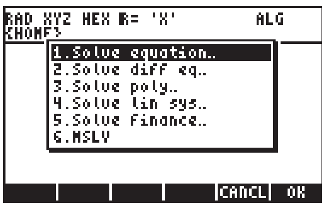
To use function ZEROS in RPN mode, enter first the polynomial expression, then the variable to solve for, and then function ZEROS. The following screen shots show the RPN stack before and after the application of ZEROS to the two examples above:



The Symbolic Solver functions presented above produce solutions to rational equations (mainly, polynomial equations). If the equation to be solved for has all numerical coefficients, a numerical solution is possible through the use of the Numerical Solver features of the calculator.

Numerical solver menu

The calculator provides a very powerful environment for the solution of single algebraic or transcendental equations. To access this environment we start the numerical solver (NUM.SLV) by using ⬚ NUM.SLV . This produces a drop-down menu that includes the following options:



Item *2. Solve diff eq..* is to be discussed in a later chapter on differential equations. Item *4. Solve lin sys..* will be discussed in a later Chapter on matrices. Item *6. MSLV* (Multiple equation SoLVer) will be presented in the next chapter. Following, we present applications of items *3. Solve poly..*, *5. Solve finance*, and *1. Solve equation..*, in that order. Appendix 1-A, at the end of Chapter 1, contains instructions on how to use input forms with examples for the numerical solver applications.

Notes:
1. Whenever you solve for a value in the NUM.SLV applications, the value solved for will be placed in the stack. This is useful if you need to keep that value available for other operations.
2. There will be one or more variables created whenever you activate some of the applications in the NUM.SLV menu.

Polynomial Equations

Using the *Solve poly…* option in the calculator's *SOLVE* environment you can:

(1) find the solutions to a polynomial equation;

(2) obtain the coefficients of the polynomial having a number of given roots;

(3) obtain an algebraic expression for the polynomial as a function of X.

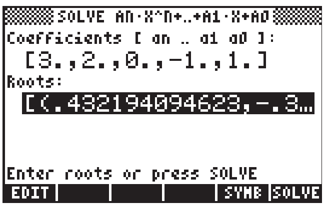
Finding the solutions to a polynomial equation

A polynomial equation is an equation of the form: $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0$. The fundamental theorem of algebra indicates that there are *n* solutions to any polynomial equation of order *n*. Some of the solutions could be complex numbers, nevertheless. As an example, solve the equation: $3s^4 + 2s^3 - s + 1 = 0$.

We want to place the coefficients of the equation in a vector $[a_n, a_{n-1}, a_1\ a_0]$. For this example, let's use the vector [3,2,0,-1,1]. To solve for this polynomial equation using the calculator, try the following:

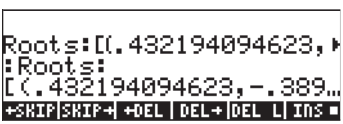
→ NUM.SLV ▼ ▼ OK	Select solve poly...
← [] 3 → , 2 → , 0	
→ , 1 +/- → , 1 OK	Enter vector of coefficients
SOLVE	Solve equation

The screen will show the solution as follows:



Press ENTER to return to stack. The stack will show the following results in ALG mode (the same result would be shown in RPN mode):

To see all the solutions, press the down-arrow key (▼) to trigger the line editor:



All the solutions are complex numbers: (0.432,-0.389), (0.432,0.389), (-0.766, 0.632), (-0.766, -0.632).

Note: Recall that complex numbers in the calculator are represented as ordered pairs, with the first number in the pair being the real part, and the second number, the imaginary part. For example, the number (0.432,-0.389), a complex number, will be written normally as 0.432 - 0.389*i*, where *i* is the imaginary unit, i.e., $i^2 = -1$.

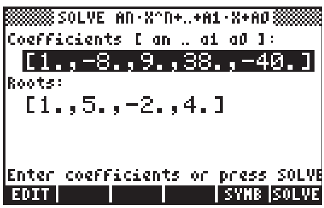
Note: The *fundamental theorem of algebra* indicates that there are *n* solutions for any polynomial equation of order *n*. There is another theorem of algebra that indicates that if one of the solutions to a polynomial equation with real coefficients is a complex number, then the conjugate of that number is also a solution. In other words, complex solutions to a polynomial equation with real coefficients come in pairs. That means that polynomial equations with real coefficients of odd order will have at least one real solution.

Generating polynomial coefficients given the polynomial's roots

Suppose you want to generate the polynomial whose roots are the numbers [1, 5, -2, 4]. To use the calculator for this purpose, follow these steps:

↱ NUM.SLV ▼ ▼ OK	Select solve poly...
▼ ↰ [] 1 ↱ , 5	
↱ , 2 +/- ↱ , 4 OK	Enter vector of roots
SOLVE	Solve for coefficients

Press ENTER to return to stack, the coefficients will be shown in the stack.



Press ▼ to trigger the line editor to see all the coefficients.

> **Note**: If you want to get a polynomial with real coefficients, but having complex roots, you must include the complex roots in pairs of conjugate numbers. To illustrate the point, generate a polynomial having the roots [1 (1,2) (1,-2)]. Verify that the resulting polynomial has only real coefficients. Also, try generating a polynomial with roots [1 (1,2) (-1,2)], and verify that the resulting polynomial will have complex coefficients.

Generating an algebraic expression for the polynomial

You can use the calculator to generate an algebraic expression for a polynomial given the coefficients or the roots of the polynomial. The resulting expression will be given in terms of the default CAS variable X. (The examples below shows how you can replace X with any other variable by using the function |.)

To generate the algebraic expression using the coefficients, try the following example. Assume that the polynomial coefficients are [1,5,-2,4]. Use the following keystrokes:

→ NUM.SLV ▼ ▼ OK	Select Solve poly…
← [] 1 → , 5	Enter vector of coefficients
→ , 2 +/- → , 4 OK	
▲ SYMB	Generate symbolic expression
ENTER	Return to stack.

The expression thus generated is shown in the stack as:

'X^3+5*X^2+-2*X+4'.

To generate the algebraic expression using the roots, try the following example. Assume that the polynomial roots are [1,3,-2,1]. Use the following keystrokes:

→ NUM.SLV ▼ ▼ OK	Select Solve poly…
▼ ← [] 1 → , 3	Enter vector of roots
→ , 2 +/- → , 1 OK	
▼ SYMB	Generate symbolic expression
ENTER	Return to stack.

The expression thus generated is shown in the stack as:'(X-1)*(X-3)*(X+2)*(X-1)'.

To expand the products, you can use the EXPAND command. The resulting expression is: 'X^4+-3*X^3+ -3*X^2+11*X-6'.

A different approach to obtaining an expression for the polynomial is to generate the coefficients first, then generate the algebraic expression with the coefficients highlighted. For example, for this case try:

→ NUM.SLV ▼ ▼ OK	Select Solve poly…
▼ ← [] 1 → , 3	Enter vector of roots
→ , 2 +/- → , 1 OK	
SOLVE	Solve for coefficients
▼ SYMB	Generate symbolic expression
ENTER	Return to stack.

The expression thus generated is shown in the stack as: 'X^4+-3*X^3+ -3*X^2+11*X+-6*X^0'. The coefficients are listed in stack level 2.

Financial calculations

The calculations in item *5. Solve finance..* in the Numerical Solver (*NUM.SLV*) are used for calculations of time value of money of interest in the discipline of engineering economics and other financial applications. This application can also be started by using the keystroke combination ← $ FINANCE (associated with the 9 key). Before discussing in detail the operation of this solving environment, we present some definitions needed to understand financial operations in the calculator.

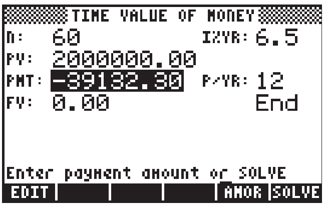
Definitions

Often, to develop projects, it is necessary to borrow money from a financial institution or from public funds. The amount of money borrowed is referred to as the *Present Value* (PV). This money is to be repaid through *n* periods (typically multiples or sub-multiples of a month) subject to an *annual interest rate* of I%YR. The *number of periods per year* (P/YR) is an integer number of periods in which the year will be divided for the purpose of repaying the loan money. Typical values of P/YR are 12 (one payment per month), 24 (payment twice a month), or 52 (weekly payments). The *payment*(PMT) is the amount that the borrower must pay to the lender at the beginning or end of each of the *n* periods of the loan. The *future value* of the money (FV) is the value that the borrowed amount of money will be worth at the end of *n* periods. Typically payment occurs at the end of each period, so that the borrower starts paying at the end of the first period, and pays the same fixed amount at the end of the second, third, etc., up to the end of the *n*-th period.

Example 1 – Calculating payment on a loan
If $2 million are borrowed at an annual interest rate of 6.5% to be repaid in 60 monthly payments, what should be the monthly payment? For the debt to be totally repaid in 60 months, the future values of the loan should be zero. So, for the purpose of using the financial calculation feature of the calculator we will use the following values: n = 60, I%YR = 6.5, PV = 2000000, FV = 0, P/YR = 12. To enter the data and solve for the payment, PMT, use:

← FINANCE	Start the financial calculation input form
60 OK	Enter n = 60
6.5 OK	Enter I%YR = 6.5 %
2000000 OK	Enter PV = 2,000,000 US$
▼	Skip PMT, since we will be solving for it
0 OK	Enter FV = 0, the option End is highlighted
▲ ◄ SOLVE	Highlight PMT and solve for it

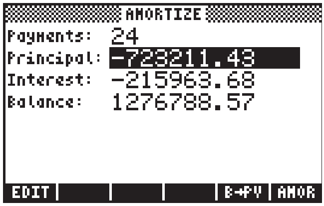
The solution screen will look like this:



The screen now shows the value of PMT as –39,132.30, i.e., the borrower must pay the lender US $ 39,132.30 at the end of each month for the next 60 months to repay the entire amount. The reason why the value of PMT turned out to be negative is because the calculator is looking at the money amounts from the point of view of the borrower. The borrower has + US $ 2,000,000.00 at time period t = 0, then he starts paying, i.e., adding -US $ 39132.30 at times t = 1, 2, …, 60. At t = 60, the net value in the hands of the borrower is zero. Now, if you take the value US $ 39,132.30 and multiply it by the 60 payments, the total paid back by the borrower is US $ 2,347,937.79. Thus, the lender makes a net profit of $ 347,937.79 in the 5 years that his money is used to finance the borrower's project.

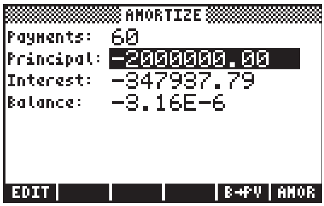
Example 2 – Calculating amortization of a loan
The same solution to the problem in Example 1 can be found by pressing AMOR, which is stands for AMORTIZATION. This option is used to calculate how much of the loan has been amortized at the end of a certain number of

payments. Suppose that we use 24 periods in the first line of the amortization screen, i.e., 2 4 OK. Then, press AMOR. You will get the following result:



This screen is interpreted as indicating that after 24 months of paying back the debt, the borrower has paid up US $ 723,211.43 into the principal amount borrowed, and US $ 215,963.68 of interest. The borrower still has to pay a balance of US $1,276,788.57 in the next 36 months.

Check what happens if you replace 60 in the *Payments:* entry in the amortization screen, then press OK AMOR. The screen now looks like this:



This means that at the end of 60 months the US $ 2,000,000.00 principal amount has been paid, together with US $ 347,937.79 of interest, with the balance being that the lender owes the borrower US $ 0.000316. Of course, the balance should be zero. The value shown in the screen above is simply round-off error resulting from the numerical solution.

Press ON or ENTER, twice, to return to normal calculator display.

Example 3 – Calculating payment with payments at beginning of period
Let's solve the same problem as in Examples 1 and 2, but using the option that payment occurs at the beginning of the payment period. Use:

← FINANCE	Start the financial calculation input form
60 OK	Enter n = 60
6.5 OK	Enter I%YR = 6.5 %
2000000 OK	Enter PV = 2,000,000 US$

▼	Skip PMT, since we will be solving for it
0 OK	Enter FV = 0, the option End is highlighted
CHOOS ▲ OK	Change payment option to *Begin*
▲ ◄ SOLVE	Highlight PMT and solve for it

The screen now shows the value of PMT as –38,921.47, i.e., the borrower must pay the lender US $ 38,921.48 at the <u>beginning</u> of each month for the next 60 months to repay the entire amount. Notice that the amount the borrower pays monthly, if paying at the beginning of each payment period, is slightly smaller than that paid at the end of each payment period. The reason for that difference is that the lender gets interest earnings from the payments from the beginning of the period, thus alleviating the burden on the lender.

Notes:

1. The financial calculator environment allows you to solve for any of the terms involved, i.e., n, I%YR, PV, FV, P/Y, given the remaining terms in the loan calculation. Just highlight the value you want to solve for, and press SOLVE. The result will be shown in the highlighted field.

2. The values calculated in the financial calculator environment are copied to the stack with their corresponding tag (identifying label).

Deleting the variables

When you use the financial calculator environment for the first time within the HOME directory, or any sub-directory, it will generate the variables n I%YR PV PMT PYR FV to store the corresponding terms in the calculations.. You can see the contents of these variables by using:

→ n → I%YR → PV → PMT → PYR → FV.

You can either keep these variables for future use, or use the PURGE function to erase them from your directory. <u>To erase all of the variables at once</u>, if using ALG mode, try the following:

TOOL PURGE VAR ← {}	Enter PURGE, prepare list of variables
' → n	Enter name of variable N
► → ,	Enter a comma
' → I%YR	Enter name of variable I%YR
► → ,	Enter a comma
' → PV	Enter name of variable PV
► → ,	Enter a comma
' → PMT	Enter name of variable PMT

▶ ↱ , Enter a comma
' ↱ PYR Enter name of variable PYR
▶ ↱ , Enter a comma
' ↱ FV . Enter name of variable FV
ENTER Execute PURGE command

The following two screen shots show the PURGE command for purging all the variables in the directory, and the result after executing the command.

In RPN mode, the command is executed by using:

VAR ↰ { } Prepare a list of variables to be purged
N Enter name of variable N
I%YR Enter name of variable I%YR
PV Enter name of variable PV
PMT Enter name of variable PMT
PYR Enter name of variable PYR
FV Enter name of variable FV
ENTER Enter list of variables in stack
TOOL PURGE Purge variables in list

Before the command PURGE is entered, the RPN stack will look like this:

Solving equations with one unknown through NUM.SLV

The calculator's NUM.SLV menu provides item *1. Solve equation..* solve different types of equations in a single variable, including non-linear algebraic and transcendental equations. For example, let's solve the equation: $e^x\text{-}\sin(\pi x/3) = 0$.

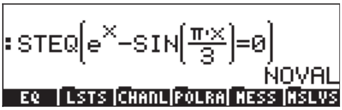
Simply enter the expression as an algebraic object and store it into variable EQ. The required keystrokes in ALG mode are the following:

' ← e^x ALPHA ← X ▶ — SIN ← π × ALPHA ← X ÷ 3 ▶ → = 0 ▶

STO▶ ALPHA E ALPHA Q ENTER

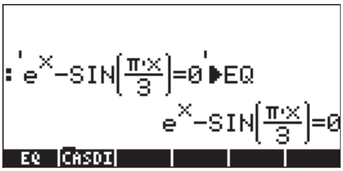
Function STEQ

Function STEQ, available through the command catalog, → CAT , will store its argument into variable EQ, e.g., in ALG mode:

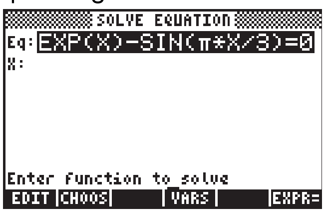


In RPN mode, enter the equation between apostrophes and activate command STEQ. Thus, function STEQ can be used as a shortcut to store an expression into variable EQ.

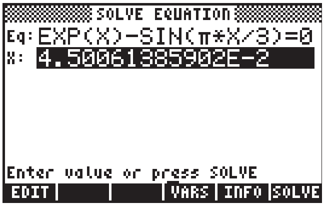
Press VAR to see the newly created EQ variable:



Then, enter the SOLVE environment and select *Solve equation…*, by using: → NUM.SLV OK. The corresponding screen will be shown as:



The equation we stored in variable EQ is already loaded in the *Eq* field in the SOLVE EQUATION input form. Also, a field labeled *x* is provided. To solve the equation all you need to do is highlight the field in front of X: by using ▼, and press SOLVE. The solution shown is X: 4.5006E-2:



This, however, is not the only possible solution for this equation. To obtain a negative solution, for example, enter a negative number in the X: field before solving the equation. Try 3 +/- OK ▼ SOLVE. The solution is now X: -3.045.

Solution procedure for Equation Solve...

The numerical solver for single-unknown equations works as follows:

- It lets the user type in or CHOOS an equation to solve.
- It creates an input form with input fields corresponding to all variables involved in equation stored in variable EQ.
- The user needs to enter values for all variables involved, except one.
- The user then highlights the field corresponding to the unknown for which to solve the equation, and presses SOLVE
- The user may force a solution by providing an initial guess for the solution in the appropriate input field before solving the equation.

The calculator uses a search algorithm to pinpoint an interval for which the function changes sign, which indicates the existence of a root or solution. It then utilizes a numerical method to converge into the solution.

The solution the calculator seeks is determined by the initial value present in the unknown input field. If no value is present, the calculator uses a default value of zero. Thus, you can search for more than one solution to an equation by changing the initial value in the unknown input field. Examples of the equations solutions are shown following.

Example 1 – Hooke's law for stress and strain

The equation to use is Hooke's law for the normal strain in the x-direction for a solid particle subjected to a state of stress given by

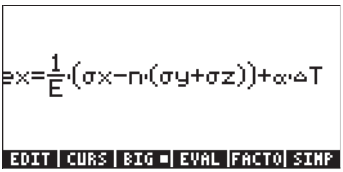
$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

The equation is $e_{xx} = \frac{1}{E}[\sigma_{xx} - n \cdot (\sigma_{yy} + \sigma_{zz})] + \alpha \cdot \Delta T$, here e_{xx} is the unit strain in the x-direction, σ_{xx}, σ_{yy}, and σ_{zz}, are the normal stresses on the particle in the directions of the x-, y-, and z-axes, *E* is Young's modulus or modulus of elasticity of the material, *n* is the Poisson ratio of the material, α is the thermal expansion coefficient of the material, and ΔT is a temperature increase.

Suppose that you are given the following data: σ_{xx}= 2500 psi, σ_{yy} =1200 psi, and σ_{zz} = 500 psi, E = 1200000 psi, *n* = 0.15, α = 0.00001/°F, ΔT = 60 °F. To calculate the strain e_{xx} use the following:

→ NUM.SLV OK	Access numerical solver to solve equations
→ EQW	Access the equation writer to enter equation

At this point follow the instructions from Chapter 2 on how to use the Equation Writer to build an equation. The equation to enter in the *Eq* field should look like this (notice that we use only one sub-index to refer to the variables, i.e., e_{xx} is translated as *ex*, etc. -- this is done to save typing time):

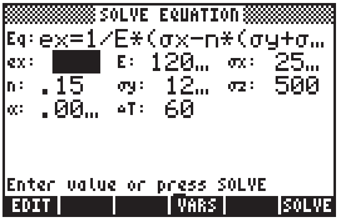


Use the following shortcuts for special characters:

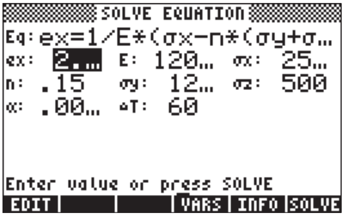
σ: ALPHA → S α: ALPHA → A Δ: ALPHA → C

and recall that lower-case letters are entered by using ALPHA ← before the letter key, thus, x is typed as ALPHA ← X .

Press ENTER to return to the solver screen. Enter the values proposed above into the corresponding fields, so that the solver screen looks like this:



With the *ex:* field highlighted, press SOLVE to solve for *ex*:



The solution can be seen from within the SOLVE EQUATION input form by pressing EDIT while the *ex:* field is highlighted. The resulting value is 2.470833333333E-3. Press OK to exit the EDIT feature.

Suppose that you now, want to determine the Young's modulus that will produce a strain of e_{xx} = 0.005 under the same state of stress, neglecting thermal expansion. In this case, you should enter a value of 0.005 in the *ex:* field, and a zero in the *ΔT:* field (with ΔT = 0, no thermal effects are included). To solve for E, highlight the *E:* field and press SOLVE. The result, seeing with the EDIT feature is, E = 449000 psi. Press SOLVE ENTER to return to normal display.

Notice that the results of the calculations performed within the numerical solver environment have been copied to the stack:

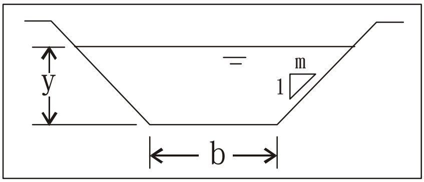
Also, you will see in your soft-menu key labels variables corresponding to those variables in the equation stored in EQ (press NXT to see all variables in your directory), i.e., variables *ex, ΔT, α, σz, σy, n, σx,* and *E.*

Example 2 – Specific energy in open channel flow

Specific energy in an open channel is defined as the energy per unit weight measured with respect to the channel bottom. Let E = specific energy, y = channel depth, V = flow velocity, g = acceleration of gravity, then we write

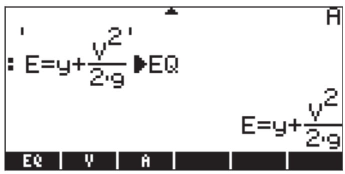
$$E = y + \frac{V^2}{2g}.$$

The flow velocity, in turn, is given by V = Q/A, where Q = water discharge, A = cross-sectional area. The area depends on the cross-section used, for example, for a trapezoidal cross-section, as shown in the figure below, A = (b+m·y) ·y, where b = bottom width, and m = side slope of cross section.

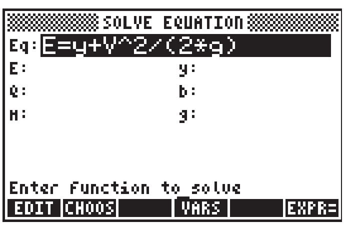


We can type in the equation for E as shown above and use auxiliary variables for A and V, so that the resulting input form will have fields for the fundamental variables y, Q, g, m, and b, as follows:

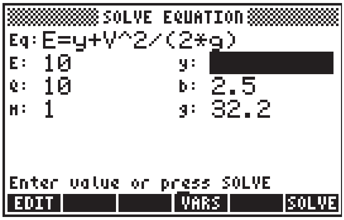
- First, create a sub-directory called SPEN (SPecific ENergy) and work within that sub-directory.
- Next, define the following variables:



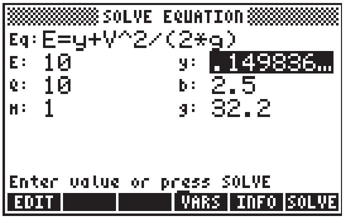
- Launch the numerical solver for solving equations: ↱ NUM.SLV OK. Notice that the input form contains entries for the variables y, Q, b, m, and g:



- Try the following input data: E = 10 ft, Q = 10 cfs (cubic feet per second), b = 2.5 ft, m = 1.0, g = 32.2 ft/s^2:

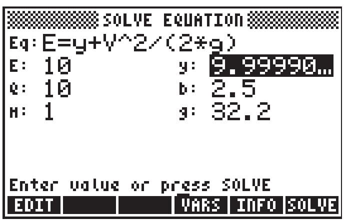


- Solve for y.



The result is 0.149836.., i.e., y = 0.149836.

- It is known, however, that there are actually two solutions available for y in the specific energy equation. The solution we just found corresponds to a numerical solution with an initial value of 0 (the default value for *y*, i.e., whenever the solution field is empty, the initial value is zero). To find the other solution, we need to enter a larger value of y, say 15, highlight the y input field and solve for y once more:



The result is now 9.99990, i.e., y = 9.99990 ft.

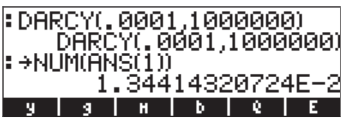
This example illustrates the use of auxiliary variables to write complicated equations. When NUM.SLV is activated, the substitutions implied by the auxiliary variables are implemented, and the input screen for the equation provides input field for the primitive or fundamental variables resulting from the substitutions. The example also illustrates an equation that has more than one solution, and how choosing the initial guess for the solution may produce those different solutions.

In the next example we will use the DARCY function for finding friction factors in pipelines. Thus, we define the function in the following frame.

Special function for pipe flow: DARCY (ε/D,Re)
The Darcy-Weisbach equation is used to calculate the energy loss (per unit weight), h_f, in a pipe flow through a pipe of diameter D, absolute roughness ε, and length L, when the flow velocity in the pipe is V. The equation is written as $h_f = f \cdot \frac{L}{D} \cdot \frac{V^2}{2g}$. The quantity *f* is known as the friction factor of the flow and it has been found to be a function of the relative roughness of the pipe, ε/D, and a (dimensionless) Reynolds number, Re. The Reynolds number is defined as Re = ρVD/μ = VD/ν, where ρ and μ are the density and dynamic viscosity of the fluid, respectively, and ν = μ/ρ is the kinematic viscosity of the fluid.

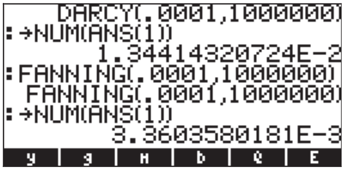
The calculator provides a function called DARCY that uses as input the relative roughness ε/D and the Reynolds number, in that order, to calculate the friction factor f. The function DARCY can be found through the command catalog:

For example, for ε/D = 0.0001, Re = 1000000, you can find the friction factor by using: DARCY(0.0001,1000000). In the following screen, the function →NUM () was used to obtain a numerical value of the function:

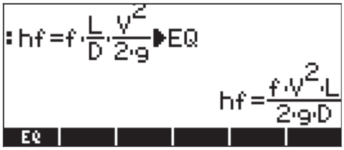


The result is f = DARCY(0.0001,1000000) = 0.01341…

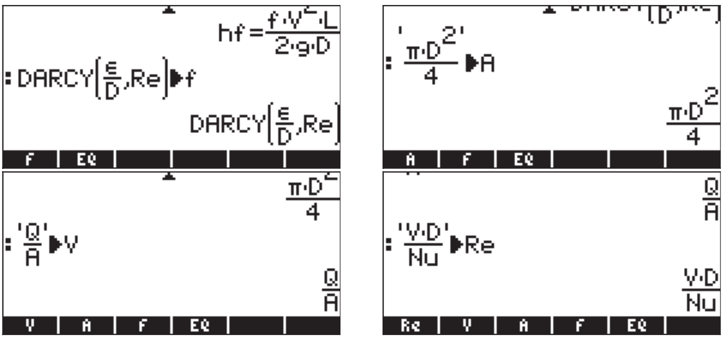
The function FANNING(ε/D,Re)
In aerodynamics applications a different friction factor, the Fanning friction factor, is used. The Fanning friction factor, f_F, is defined as 4 times the Darcy-Weisbach friction factor, *f*. The calculator also provides a function called FANNING that uses the same input as DARCY, i.e., ε/D and Re, and provides the FANNING friction factor. Check that FANNING(0.0001,1000000) = 0.003360358918 1s.



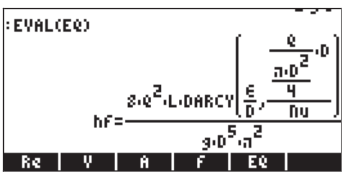
Example 3 – Flow in a pipe
You may want to create a separate sub-directory (PIPES) to try this example. The main equation governing flow in a pipe is, of course, the Darcy-Weisbach equation. Thus, type in the following equation into EQ:



Also, enter the following variables (f, A, V, Re):



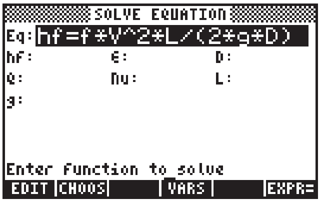
In this case we stored the main equation (Darcy-Weisbach equation) into EQ, and then replaced several of its variables by other expressions through the definition of variables f, A, V, and Re. To see the combined equation, use EVAL(EQ). In this example we changed the display setting so that we can see the entire equation in the screen:



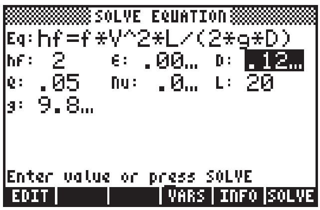
Thus, the equation we are solving, after combining the different variables in the directory, is:

$$h_f = \frac{8Q^2L}{\pi^2 gD^5} \cdot DARCY\left(\frac{\varepsilon}{D}, \frac{\dfrac{QD}{\pi D^2/4}}{Nu}\right)$$

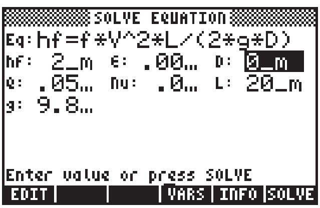
The combined equation has primitive variables: h_f, Q, L, g, D, ε, and Nu. Launch the numerical solver (→ NUM.SLV OK) to see the primitive variables listed in the SOLVE EQUATION input form:

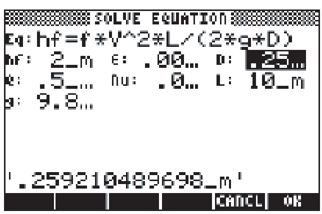


Suppose that we use the values hf = 2 m, ε = 0.00001 m, Q = 0.05 m^3/s, Nu = 0.000001 m^2/s, L = 20 m, and g = 9.806 m/s^2, find the diameter D. Enter the input values, and solve for D, The solution is: 0.12, i.e., D = 0.12 m.



If the equation is dimensionally consistent, you can add units to the input values, as shown in the figure below. However, you must add those units to the initial guess in the solution. Thus, in the example below we place 0_m in the D: field before solving the problem. The solution is shown in the screen to the right:



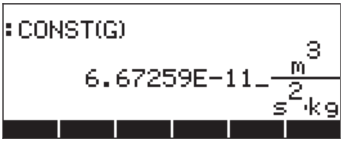


Press ENTER to return to normal calculator display. The solution for D will be listed in the stack.

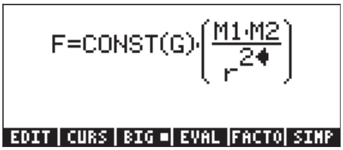
Example 4 – Universal gravitation

Newton's law of universal gravitation indicates that the magnitude of the attractive force between two bodies of masses m_1 and m_2 separated by a distance r is given by the equation $F = G \cdot \frac{M_1 \cdot M_2}{r^2}$.

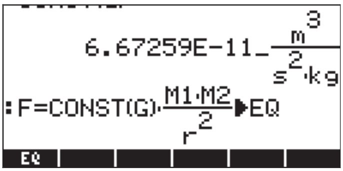
Here, G is the universal gravitational constant, whose value can be obtained through the use of the function CONST in the calculator by using:



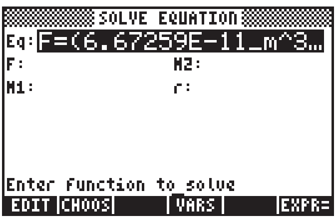
We can solve for any term in the equation (except G) by entering the equation as:



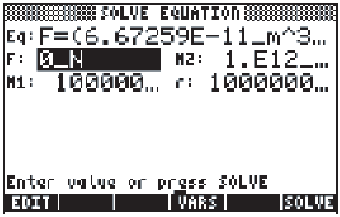
This equation is then stored in EQ:



Launching the numerical solver for this equation results in an input form containing input fields for F, G, m1, m2, and r.



Let's solve this problem using units with the following values for the known variables m1 = 1.0×10^6 kg, m2 = 1.0×10^{12} kg, r = 1.0×10^{11} m. Also, enter a value of 0_N in field F to ensure the proper solution using units in the calculator:

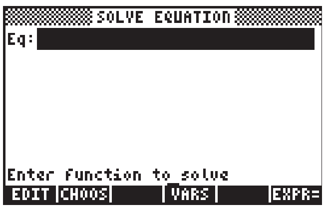


Solve for F, and press to return to normal calculator display. The solution is F : 6.67259E-15_N, or F = 6.67259×10^{-15} N.

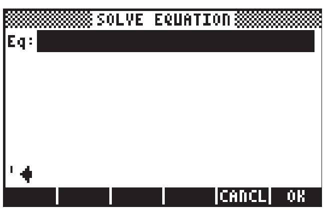
> **Note**: When using units in the numerical solver make sure that all the variables have the proper units, that the units are compatible, and that the equation is dimensionally homogeneous.

Different ways to enter equations into EQ

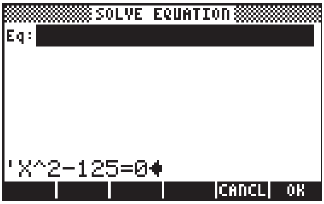
In all the examples shown above we have entered the equation to be solved directly into variable EQ before activating the numerical solver. You can actually type the equation to be solved directly into the solver after activating it by editing the contents of the EQ field in the numerical solver input form. If variable EQ has not been defined previously, when you launch the numerical solver (NUM.SLV OK), the EQ field will be highlighted:

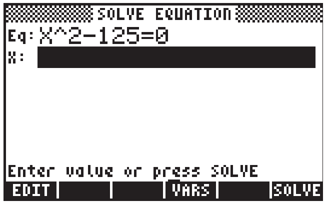


At this point you can either type a new equation by pressing EDIT. You will be provided with a set of apostrophes so that you can type the expression between them:



Type an equation, say X^2 - 125 = 0, directly on the stack, and press OK .

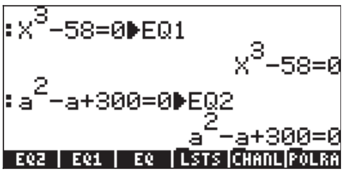




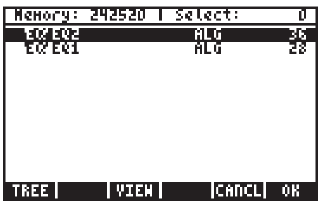
At this point the equation is ready for solution.

Alternatively, you can activate the equation writer after pressing EDIT to enter your equation. Press ENTER to return to the numerical solver screen.

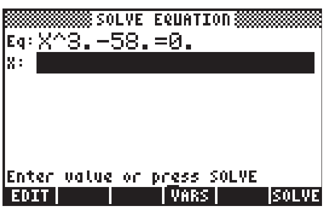
Another way to enter an equation into the EQ variable is to select a variable already existing in your directory to be entered into EQ. This means that your equation would have to have been stored in a variable name previously to activating the numerical solver. For example, suppose that we have entered the following equations into variables EQ1 and EQ2:



Now, launch the numerical solver (→ NUM.SLV OK, and highlight the EQ field. At this point press the CHOOS soft menu key. Use the up and down arrow keys (▲ ▼) to select, say, variable EQ1:



Press OK after selecting EQ1 to load into variable EQ in the solver. The new equation is ready to be solved.



The SOLVE soft menu

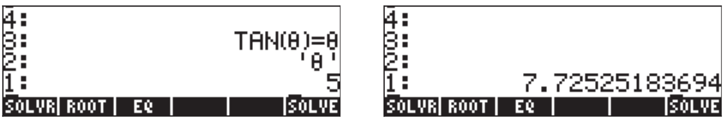
The SOLVE soft menu allows access to some of the numerical solver functions through the soft menu keys. To access this menu use in RPN mode: 74 MENU, or in ALG mode: MENU(74). Alternatively, you can use ↱(hold) 7 to activate the SOLVE soft menu. The sub-menus provided by the SOLVE soft menu are the following:

The ROOT sub-menu

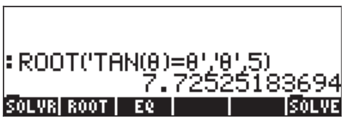
The ROOT sub-menu include the following functions and sub-menus:

Function ROOT

Function ROOT is used to solve an equation for a given variable with a starting guess value. In RPN mode the equation will be in stack level 3, while the variable name will be located in level 2, and the initial guess in level 1. The following figure shows the RPN stack before and after activating function ROOT:



In ALG mode, you would use ROOT('TAN(θ)=θ','θ',5) to activate function ROOT:



Variable EQ

The soft menu key EQ in this sub-menu is used as a reference to the variable EQ. Pressing this soft menu key is equivalent to using function RCEQ (ReCall EQ).

The SOLVR sub-menu

The SOLVR sub-menu activates the soft-menu solver for the equation currently stored in EQ. Some examples are shown next:

Example 1 - Solving the equation t^2-5t = -4
For example, if you store the equation 't^2-5*t=-4' into EQ, and press SOLVR, it will activate the following menu:

This result indicates that you can solve for a value of t for the equation listed at the top of the display. If you try, for example, ← [t], it will give you the result t: 1., after briefly flashing the message "Solving for t." There is a second root to this equation, which can be found by changing the value of t, before solving for it again. Do the following: 10 [t], then press ← [t]. The result is now, t: 4.0000000003. To verify this result, press the soft menu key labeled EXPR=, which evaluates the expression in EQ for the current value of t. The results in this case are:

To exit the SOLVR environment, press VAR. The access to the SOLVE menu is lost at this point, so you have to activate it once more as indicated earlier, to continue with the exercises below.

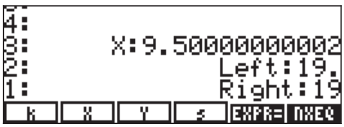
Example 2 - Solving the equation Q = at^2+bt
It is possible to store in EQ, an equation involving more than one variable, say, 'Q = at^2 + bt'. In this case, after activating the SOLVE soft menu, and pressing ROOT SOLVR, you will get the following screen:

Within this SOLVR environment you can provide values for any of the variables listed by entering the value in the stack and pressing the corresponding soft-menu keys. For example, say you enter the values Q = 14, a = 2, and b = 3. You would use: 14 [Q], 2 [a], 3 [b].
As variables Q, a, and b, get assigned numerical values, the assignments are listed in the upper left corner of the display. At this point we can solve for t, by using ← [t]. The result is t: 2. Pressing EXPR= shows the results:

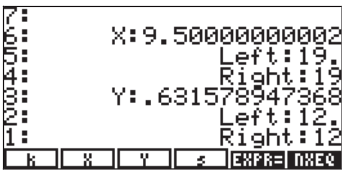
Example 3 - Solving two simultaneous equations, one at a time

You can also solve more than one equation by solving one equation at a time, and repeating the process until a solution is found. For example, if you enter the following list of equations into variable EQ: { ‘a*X+b*Y = c’, ‘k*X*Y=s’}, the keystroke sequence ROOT SOLVR, within the SOLVE soft menu, will produce the following screen:

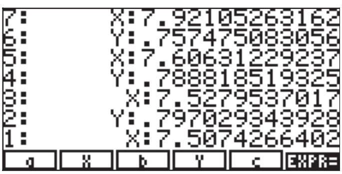
The first equation, namely, a*X + b*Y = c, will be listed in the top part of the display. You can enter values for the variables a, b, and c, say: 2 [a] 5 [b] 19 [c]. Also, since we can only solve one equation at a time, let’s enter a guess value for Y, say, 0 [Y], and solve for X, by using ← [X]. This gives the value, X: 9.4999…. To check the value of the equation at this point, press EXPR=. The results are: Left: 19, Right: 19. To solve the next equation, press NXT NEXQ. The screen shows the soft menu keys as:

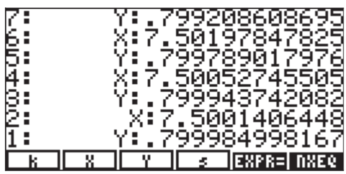


Say we enter the values k = 2, s = 12. Then solve for Y, and press EXPR=. The results are now, Y:



We then continue moving from the first to the second equation, back and forth, solving the first equation for X and the second for Y, until the values of X and Y converge to a solution. To move from equation to equation use NXEQ. To solve for X and Y use ← [X], and ← [Y], respectively. The following sequence of solutions is produced:





After solving the two equations, one at a time, we notice that, up to the third decimal, X is converging to a value of 7.500, while Y is converging to a value o 0.799.

Using units with the SOLVR sub-menu
These are some rules on the use of units with the SOLVR sub-menu:

- Entering a guess with units for a given variable, will introduce the use of those units in the solution.
- If a new guess is given without units, the units previously saved for that particular variable are used.
- To remove units enter a number without units in a list as the new guess, i.e., use the format { number }.
- A list of numbers can be given as a guess for a variable. In this case, the units takes the units used belong to the last number in the list. For example, entering { 1.41_ft 1_cm 1_m } indicates that meters (m) will be used for that variable.
- The expression used in the solution must have consistent units, or an error will result when trying to solve for a value.

The DIFFE sub-menu

The DIFFE sub-menu provides a number of functions for the numerical solution of differential equations. The functions provided are the following:

These functions are presented in detail in Chapter 16.

The POLY sub-menu

The POLY sub-menu performs operations on polynomials. The functions included are the following:

Function PROOT
This function is used to find the roots of a polynomial given a vector containing the polynomial coefficients in decreasing order of the powers of the independent variable. In other words, if the polynomial is $a_nx^n + a_{n-1}x^{n-1} + \ldots + a_2x^2 + a_1x + a_0$, the vector of coefficients should be entered as $[a_n, a_{n-1}, \ldots , a_2, a_1 , a_0]$. For example, the roots of the polynomial whose coefficients are [1, -5, 6] are [2, 3].

Function PCOEF

This function produces the coefficients [a_n, a_{n-1}, … , a_2, a_1 , a_0] of a polynomial $a_nx^n + a_{n-1}x^{n-1} + \ldots + a_2x^2 + a_1x + a_0$, given a vector of its roots [r_1, r_2, …, r_n]. For example, a vector whose roots are given by [-1, 2, 2, 1, 0], will produce the following coefficients: [1, -4, 3, 4, -4, 0]. The polynomial is $x^5 - 4x^4 + 3x^3 + 4x^2 - 4x$.

Function PEVAL

This function evaluates a polynomial, given a vector of its coefficients, [a_n, a_{n-1}, … , a_2, a_1 , a_0], and a value x_0, i.e., PEVAL calculates $a_nx_0^n + a_{n-1}x_0^{n-1} + \ldots + a_2x_0^2 + a_1x_0 + a_0$. For example, for coefficients [2, 3, -1, 2] and a value of 2, PEVAL returns the value 28.

The SYS sub-menu

The SYS sub-menu contains a listing of functions used to solve linear systems. The functions listed in this sub-menu are:

These functions are presented in detail in Chapter 11.

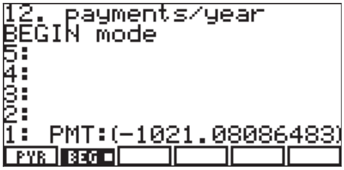
The TVM sub-menu

The TVM sub-menu contains functions for calculating Time Value of Money. This is an alternative way to solve FINANCE problems (see Chapter 6). The functions available are shown next:

The SOLVR sub-menu

The SOLVR sub-menu in the TVM sub-menu will launch the solver for solving TVM problems. For example, pressing SOLVR, at this point, will trigger the following screen:

As an exercise, try using the values n = 10, I%YR = 5.6, PV = 10000, and FV = 0, and enter ← [PMT] to find PMT = -1021.08…. Pressing NXT, produces the following screen:



Press VAR to exit the SOLVR environment. Find your way back to the TVM sub-menu within the SOLVE sub-menu to try the other functions available.

Function TVMROOT
This function requires as argument the name of one of the variables in the TVM problem. The function returns the solution for that variable, given that the other variables exist and have values stored previously. For example, having solved a TVM problem above, we can solve for, say, 'N', as follows: ['] ALPHA N ENTER TVMRO. The result is 10.

Function AMORT
This function takes a value representing a period of payment (between 0 and n) and returns the principal, interest, and balance for the values currently stored in the TVM variables. For example, with the data used earlier, if we activate function AMORT for a value of 10, we get:

Function BEG
If selected, the TMV calculations use payments at the beginning of each period. If deselected, the TMV calculations use payments at the end of each period.