Chapter 9
Vectors

This Chapter provides examples of entering and operating with vectors, both mathematical vectors of many elements, as well as physical vectors of 2 and 3 components.

Definitions

From a mathematical point of view, a vector is an array of 2 or more elements arranged into a row or a column. These will be referred to as *row* and *column vectors*. Examples are shown below:

$$v = \begin{bmatrix} -1 \\ 3 \\ 6 \end{bmatrix}, \quad u = [1, -3, 5, 2]$$

Physical vectors have two or three components and can be used to represent physical quantities such as position, velocity, acceleration, forces, moments, linear and angular momentum, angular velocity and acceleration, etc. Referring to a Cartesian coordinate system (x,y,z), there exists unit vectors **i**, **j**, **k** associated with each coordinate direction, such that a physical vector **A** can be written in terms of its components A_x, A_y, A_z, as **A** = A_x**i** + A_y**j** + A_z**k**. Alternative notation for this vector are: **A** = [A_x, A_y, A_z], **A** = (A_x, A_y, A_z), or **A** = < A_x, A_y, A_z >. A two dimensional version of this vector will be written as **A** = A_x**i** + A_y**j**, **A** = [A_x, A_y], **A** = (A_x, A_y), or **A** = < A_x, A_y >. Since in the calculator vectors are written between brackets [ ], we will choose the notation **A** = [A_x, A_y, A_z] or **A** = [A_x, A_y, A_z], to refer to two- and three-dimensional vectors from now on. The magnitude of a vector **A** is defined as |**A**| = $\sqrt{A_x^2 + A_y^2 + A_z^2}$. A unit vector in the direction of vector **A**, is defined as $\mathbf{e}_A$ = **A**/|**A**|. Vectors can be multiplied by a scalar, e.g., k**A** = [kA_x, kA_y, kA_z]. Physically, the vector k**A** is parallel to vector **A**, if k>0, or anti-parallel to vector **A**, if k<0. The negative of a vector is defined as –**A** = (–1)**A** = [–A_x, –A_y, –A_z]. Division by as scalar can be interpreted as a multiplication, i.e., **A**/k = (1/k)·**A**. Addition and subtraction of vectors are defined as **A**±**B** = [$A_x \pm B_x$, $A_y \pm B_y$, $A_z \pm B_y$], where **B** is the vector **B** = [B_x, B_y, B_z].
There are two definitions of products of physical vectors, a scalar or internal product (the dot product) and a vector or external product (the cross product). The dot product produces a scalar value defined as **A•B** = |**A**||**B**|cos(θ),

where θ is the angle between the two vectors. The cross product produces a vector **A**×**B** whose magnitude is |**A**×**B**| = |**A**||**B**|sin(θ), and its direction is given by the so-called right-hand rule (consult a textbook on Math, Physics, or Mechanics to see this operation illustrated graphically). In terms of Cartesian components, **A•B** = $A_xB_x+A_yB_y+A_zB_z$, and **A**×**B** = $[A_yB_z-A_zB_y, A_zB_x-A_xB_z, A_xB_y-A_yB_x]$. The angle between two vectors can be found from the definition of the dot product as cos(θ) = **A•B**/|**A**||**B**|= **e**$_A$•**e**$_B$. Thus, if two vectors **A** and **B** are perpendicular (θ = 90^0 = $\pi/2^{rad}$), **A•B** = 0.

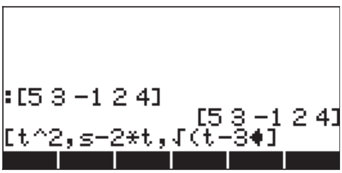
Entering vectors

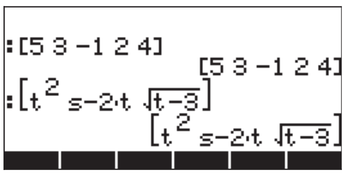
In the calculator, vectors are represented by a sequence of numbers enclosed between brackets, and typically entered as row vectors. The brackets are generated in the calculator by the keystroke combination ← [] , associated with the × key. The following are examples of vectors in the calculator:

`[3.5, 2.2, -1.3, 5.6, 2.3]`	A general row vector
`[1.5,-2.2]`	A 2-D vector
`[3,-1,2]`	A 3-D vector
`['t','t^2','SIN(t)']`	A vector of algebraics

Typing vectors in the stack

With the calculator in ALG mode, a vector is typed into the stack by opening a set of brackets (← []) and typing the components or elements of the vector separated by commas (→ ,). The screen shots below show the entering of a numerical vector followed by an algebraic vector. The figure to the left shows the algebraic vector before pressing ←. The figure to the right shows the calculator's screen after entering the algebraic vector:





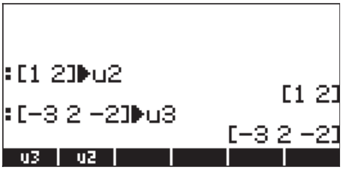
In RPN mode, you can enter a vector in the stack by opening a set of brackets and typing the vector components or elements separated by either commas (→ ,) or spaces (SPC). Notice that after pressing ENTER , in either mode, the calculator shows the vector elements separated by spaces.

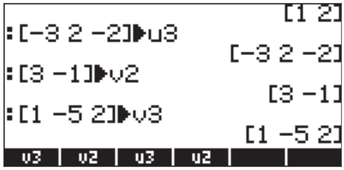
Storing vectors into variables

Vectors can be stored into variables. The screen shots below show the vectors

$\mathbf{u}_2$ = [1, 2], $\mathbf{u}_3$ = [-3, 2, -2], $\mathbf{v}_2$ = [3,-1], $\mathbf{v}_3$ = [1, -5, 2]

stored into variables u2, u3, v2, and v3, respectively. First, in ALG mode:





Then, in RPN mode (before pressing STO▶, repeatedly):

Using the Matrix Writer (MTRW) to enter vectors

Vectors can also be entered by using the Matrix Writer ← MTRW (third key in the fourth row of keys from the top of the keyboard). This command generates a species of spreadsheet corresponding to rows and columns of a matrix (Details on using the Matrix Writer to enter matrices will be presented in a subsequent chapter). For a vector we are interested in filling only elements in the top row. By default, the cell in the top row and first column is selected. At the bottom of the spreadsheet you will find the following soft menu keys:

EDIT VEC■ ←WID WID→ GO→■ GO↓

The EDIT key is used to edit the contents of a selected cell in the Matrix Writer.

The VEC key, when selected, will produce a vector, as opposite to a matrix of one row and many columns.

Vectors vs. matrices
To see the VEC key in action, try the following exercises:
(1) Launch the Matrix Writer (← MTRW). With VEC■ and GO→■ selected, enter 3 ENTER 5 ENTER 2 ENTER ENTER . This produces [3. 5. 2.]. (In RPN mode, you can use the following keystroke sequence to produce the same result: 3 SPC 5 SPC 2 ENTER ENTER).
(2) With VEC deselected and GO→■ selected,, enter 3 SPC 5 SPC 2 ENTER ENTER . This produces [[3. 5. 2.]].

Although these two results differ only in the number of brackets used, for the calculator they represent different mathematical objects. The first one is a vector with three elements, and the second one a matrix with one row and three columns. There are differences in the way that mathematical operations take place on a vector as opposite to a matrix. Therefore, for the time being, keep the soft menu key VEC■ selected while using the Matrix Writer.

The ←WID key is used to decrease the width of the columns in the spreadsheet. Press this key a couple of times to see the column width decrease in your Matrix Writer.
The WID→ key is used to increase the width of the columns in the spreadsheet. Press this key a couple of times to see the column width increase in your Matrix Writer.
The GO→■ key, when selected, automatically selects the next cell to the right of the current cell when you press ENTER . This option is selected by default.
The GO↓ key, when selected, automatically selects the next cell below the current cell when you press ENTER .

Moving to the right vs. moving down in the Matrix Writer
Activate the Matrix Writer and enter 3 ENTER 5 ENTER 2 ENTER ENTER with the GO→■ key selected (default). Next, enter the same sequence of numbers with the GO↓■ key selected to see the difference. In the first case you entered a vector of three elements. In the second case you entered a matrix of three rows and one column.

Activate the Matrix Writer again by using ← MTRW , and press NXT to check out the second soft key menu at the bottom of the display. It will show the keys:

+ROW -ROW +COL -COL →STK GOTO

The **+ROW** key will add a row full of zeros at the location of the selected cell of the spreadsheet.
The **-ROW** key will delete the row corresponding to the selected cell of the spreadsheet.
The **+COL** key will add a column full of zeros at the location of the selected cell of the spreadsheet.
The **-COL** key will delete the column corresponding to the selected cell of the spreadsheet.
The **→STK** key will place the contents of the selected cell on the stack.
The **GOTO** key, when pressed, will request that the user indicate the number of the row and column where he or she wants to position the cursor.

Pressing NXT once more produces the last menu, which contains only one function **DEL** (delete).

The function **DEL** will delete the contents of the selected cell and replace it with a zero.

To see these keys in action try the following exercise:

(1) Activate the Matrix Writer by using ← MTRW . Make sure the **VEC**■ and **GO**→■ keys are selected.

(2) Enter the following:

1 ENTER 2 ENTER 3 ENTER
NXT **GOTO** 2 **OK** 1 **OK** **OK**
2 ENTER 1 ENTER 5 ENTER
4 ENTER 5 ENTER 6 ENTER
7 ENTER 8 ENTER 9 ENTER

(3) Move the cursor up two positions by using ▲▲▲. Then press **-ROW**. The second row will disappear.

(4) Press **+ROW**. A row of three zeroes appears in the second row.

(5) Press **-COL**. The first column will disappear.

(6) Press **+COL**. A row of two zeroes appears in the first row.

(7) Press **GOTO** 3 **OK** 3 **OK** **OK** to move to position (3,3).

(8) Press **→STK**. This will place the contents of cell (3,3) on the stack, although you will not be able to see it yet.

(9) Press ENTER to return to normal display. Element (3,3) and the full matrix will be available in the screen.

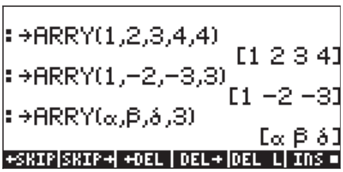
Summary of Matrix Writer use for entering vectors
In summary, to enter a vector using the Matrix Writer, simply activate the writer (← MTRW), and place the elements of the vector, pressing ENTER after each of them. Then, press ENTER ENTER . Make sure that the VEC■ and GO→■ keys are selected.

Example: ← MTRW ' ALPHA ← X Y^x 2 ENTER 2 ENTER 5 +/- ENTER ENTER

produces: ['x^2' 2 –5]

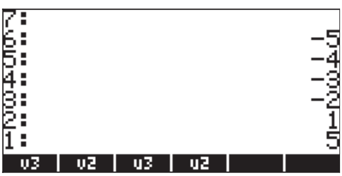
Building a vector with →ARRY

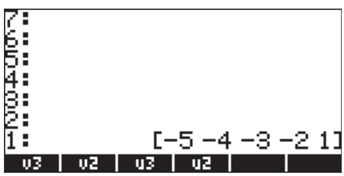
The function →ARRY, available in the function catalog (→ CAT → ⟶ , use ▲▼ to locate the function), can also be used to build a vector or array in the following way. In ALG mode, enter →ARRY(*vector elements, number of elements*), e.g.,



In RPN mode:
(1) Enter the n elements of the array in the order you want them to appear in the array (when read from left to right) into the RPN stack.
(2) Enter *n* as the last entry.
(3) Use function →ARRY.

The following screen shots show the RPN stack before and after applying function →ARRY:



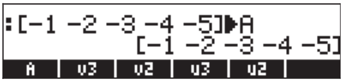


In RPN mode, the function [→ARRY] takes the objects from stack levels *n+1, n, n-1*, …, down to stack levels 3 and 2, and converts them into a vector of *n* elements. The object originally at stack level n+1 becomes the first element, the object originally at level n becomes the second element, and so on.

Note: Function →ARRY is also available in the PRG/TYPE menu (← PRG)

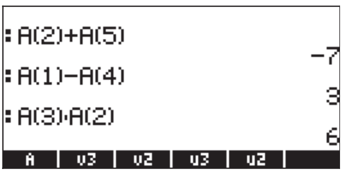
Identifying, extracting, and inserting vector elements

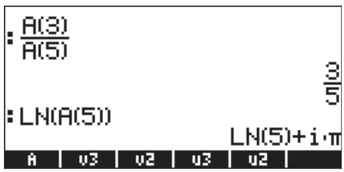
If you store a vector into a variable name, say A, you can identify elements of the vector by using A(i), where i is an integer number less than or equal to the vector size. For example, create the following array and store it in variable A: [-1, -2, -3, -4, -5]:

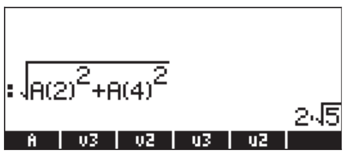


To <u>recall</u> the third element of A, for example, you could type in A(3) into the calculator. In ALG mode, simply type A(3). In RPN mode, type 'A(3)' ENTER EVAL.

You can <u>operate with elements of the array</u> by writing and evaluating algebraic expressions such as:





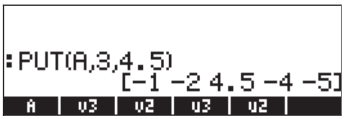


More complicated <u>*expressions involving elements*</u> of A can also be written. For example, using the Equation Writer (→ EQW), we can write the following summation of the elements of A:

Highlighting the entire expression and using the EVAL soft menu key, we get the result: -15.

Note: The vector A can also be referred to as an *indexed variable* because the name A represents not one, but many values identified by a sub-index.

To <u>replace an element in an array</u> use function PUT (you can find it in the function catalog → CAT , or in the PRG/LIST/ELEMENTS sub-menu – the later was introduced in Chapter 8). In ALG mode, you need to use function PUT with the following arguments: PUT(*array, location to be replaced, new value*). For example, to change the contents of A(3) to 4.5, use:



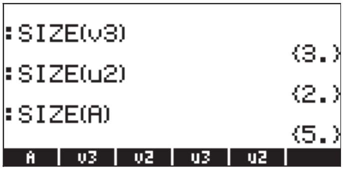
In RPN mode, you can *<u>change the value of an element</u>* of A, by storing a new value in that particular element. For example, if we want to change the contents of A(3) to read 4.5 instead of its current value of –3., use:

4 . 5 ENTER ' ALPHA A ← () 3 ENTER STO▶

To verify that the change took place use: → A . The result now shown is: [-1 -2 4.5 -4 -5].

Note: This approach for changing the value of an array element is not allowed in ALG mode, if you try to store 4.5 into A(3) in this mode you get the following error message: Invalid Syntax.

<u>To find the length of a vector</u> you can use the function SIZE, available through the command catalog (N) or through the PRG/LIST/ELEMENTS sub-menu. Some examples, based on the arrays or vectors stored previously, are shown below:

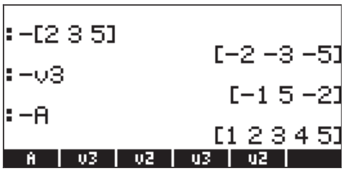


Simple operations with vectors

To illustrate operations with vectors we will use the vectors A, u2, u3, v2, and v3, stored in an earlier exercise.

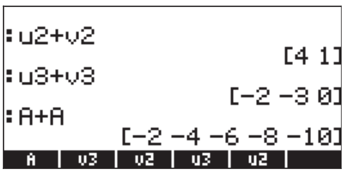
Changing sign

To change the sign of a vector use the key +/-, e.g.,



Addition, subtraction

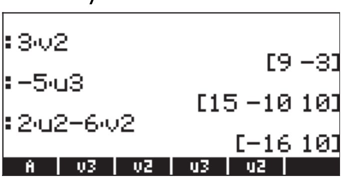
Addition and subtraction of vectors require that the two vector operands have the same length:

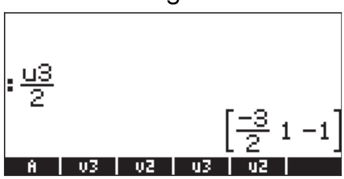


Attempting to add or subtract vectors of different length produces an error message (Invalid Dimension), e.g., v2+v3, u2+u3, A+v3, etc.

Multiplication by a scalar, and division by a scalar

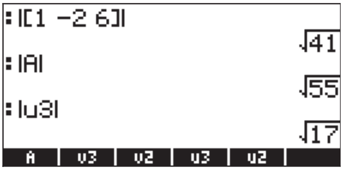
Multiplication by a scalar or division by a scalar is straightforward:





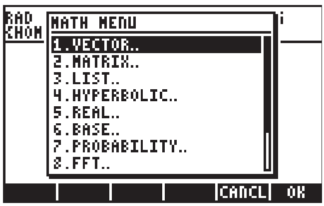
Absolute value function

The absolute value function (ABS), when applied to a vector, produces the magnitude of the vector. For a vector A = [A_1,A_2,…,A_n], the magnitude is defined as $|A| = \sqrt{A_x^2 + A_y^2 + \cdots + A_z^2}$. In the ALG mode, enter the function name followed by the vector argument. For example: `ABS([1,-2,6])`, `ABS(A)`, `ABS(u3)`, will show in the screen as follows:

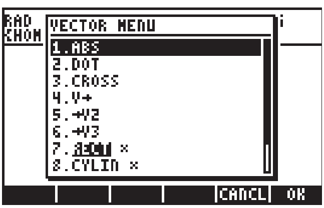


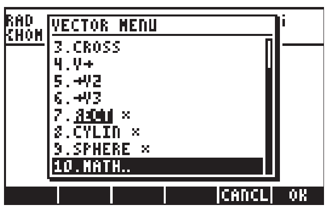
The MTH/VECTOR menu

The MTH menu (← MTH) contains a menu of functions that specifically to vector objects:



The VECTOR menu contains the following functions (system flag 117 set to CHOOSE boxes):



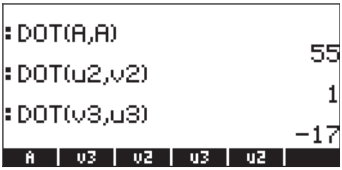


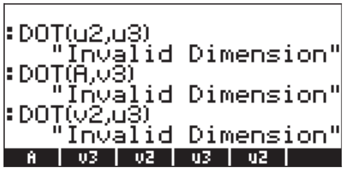
Magnitude

The magnitude of a vector, as discussed earlier, can be found with function ABS. This function is also available from the keyboard (← ABS). Examples of application of function ABS were shown above.

Dot product

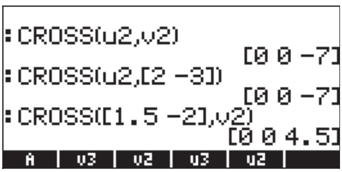
Function DOT is used to calculate the dot product of two vectors of the same length. Some examples of application of function DOT, using the vectors A, u2, u3, v2, and v3, stored earlier, are shown next in ALG mode. Attempts to calculate the dot product of two vectors of different length produce an error message:

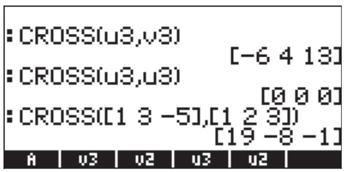




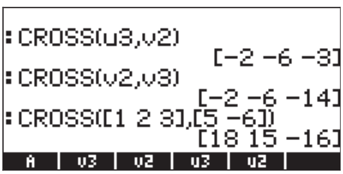
Cross product

Function CROSS is used to calculate the cross product of two 2-D vectors, of two 3-D vectors, or of one 2-D and one 3-D vector. For the purpose of calculating a cross product, a 2-D vector of the form [A_x, A_y], is treated as the 3-D vector [A_x, A_y,0]. Examples in ALG mode are shown next for two 2-D and two 3-D vectors. Notice that the cross product of two 2-D vectors will produce a vector in the z-direction only, i.e., a vector of the form [0, 0, C_z]:





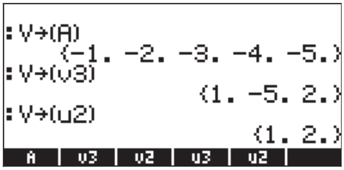
Examples of cross products of one 3-D vector with one 2-D vector, or vice versa, are presented next:



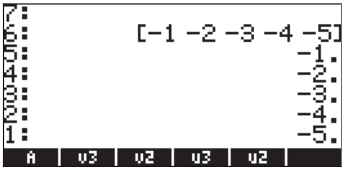
Attempts to calculate a cross product of vectors of length other than 2 or 3, produce an error message (Invalid Dimension), e.g., CROSS(v3,A), etc.

Decomposing a vector

Function V→ is used to decompose a vector into its elements or components. If used in the ALG mode, V→ will provide the elements of the vector in a list, e.g.,



In the RPN mode, application of function V→ will list the components of a vector in the stack, e.g., V→(A) will produce the following output in the RPN stack (vector A is listed in stack level 6:).

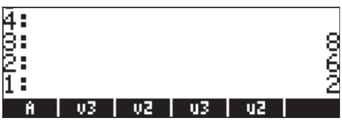


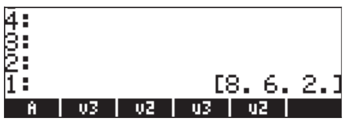
Building a two-dimensional vector

Function →V2 is used in the RPN mode to build a vector with the values in stack levels 1: and 2:. The following screen shots show the stack before and after applying function →V2:

Building a three-dimensional vector

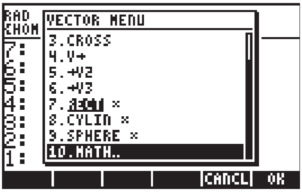
Function →V3 is used in the RPN mode to build a vector with the values in stack levels 1: , 2:, and 3:. The following screen shots show the stack before and after applying function →V2:





Changing coordinate system

Functions RECT, CYLIN, and SPHERE are used to change the current coordinate system to rectangular (Cartesian), cylindrical (polar), or spherical coordinates. The current system is shown highlighted in the corresponding CHOOSE box (system flag 117 unset), or selected in the corresponding SOFT menu label (system flag 117 set). In the following figure the RECTangular coordinate system is shown as selected in these two formats:



When the rectangular, or Cartesian, coordinate system is selected, the top line of the display will show an XYZ field, and any 2-D or 3-D vector entered in the calculator is reproduced as the (x,y,z) components of the vector. Thus, to enter the vector A = 3**i**+2**j**-5**k**, we use [3,2,-5], and the vector is shown as:

If instead of entering Cartesian components of a vector we enter cylindrical (polar) components, we need to provide the magnitude, r, of the projection of the vector on the x-y plane, an angle θ (in the current angular measure) representing the inclination of r with respect to the positive x-axis , and a z-component of the vector. The angle θ must be entered preceded by the angle character (∠), generated by using ALPHA ↱ 6. For example, suppose that we have a vector with r = 5, θ = 25° (DEG should be selected as the angular measure), and z = 2.3, we can enter this vector in the following way:

↰ [] 5 ↱ , ALPHA ↱ 6 2 5 ↱ , 2 . 3

Before pressing ENTER, the screen will look as in the left-hand side of the following figure. After pressing ENTER, the screen will look as in the right-hand side of the figure (For this example, the numerical format was changed to Fix, with three decimals).

Notice that the vector is displayed in Cartesian coordinates , with components x = r cos(θ), y = r sin(θ), z = z, even though we entered it in polar coordinates. This is because the vector display will default to the current coordinate system. For this case, we have x = 4.532, y = 2.112, and z = 2.300.

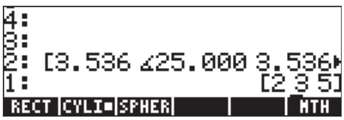
Suppose that we now enter a vector in spherical coordinates (i.e., in the form (ρ,θ,φ), where ρ is the length of the vector, θ is the angle that the xy projection of the vector forms with the positive side of the x-axis, and φ is the angle that ρ forms with the positive side of the z axis), with ρ = 5, θ = 25°, and φ = 45°. We will use: ↰ [] 5 ↱ , ALPHA ↱ 6 2 5 ,
ALPHA ↱ 6 4 5

The figure below shows the transformation of the vector from spherical to Cartesian coordinates, with x = ρ sin(ϕ) cos(θ), y = ρ sin (ϕ) cos (θ), z = ρ cos(ϕ). For this case, x = 3.204, y = 1.494, and z = 3.536.

If the CYLINdrical system is selected, the top line of the display will show an R∠ Z field, and a vector entered in cylindrical coordinates will be shown in its cylindrical (or polar) coordinate form (r,θ,z). To see this in action, change the coordinate system to CYLINdrical and watch how the vector displayed in the last screen changes to its cylindrical (polar) coordinate form. The second component is shown with the angle character in front to emphasize its angular nature.

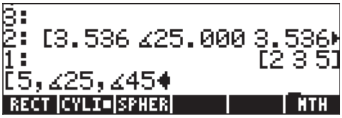
The conversion from Cartesian to cylindrical coordinates is such that r = $(x^2+y^2)^{1/2}$, θ = $\tan^{-1}(y/x)$, and z = z. For the case shown above the transformation was such that (x,y,z) = (3.204, 2.112, 2.300), produced (r,θ,z) = (3.536,25°,3.536).

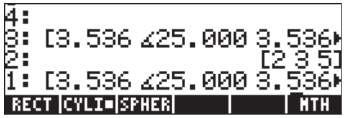
At this point, change the angular measure to Radians. If we now enter a vector of integers in Cartesian form, even if the CYLINdrical coordinate system is active, it will be shown in Cartesian coordinates, e.g.,



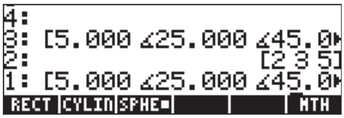
This is because the integer numbers are intended for use with the CAS and, therefore, the components of this vector are kept in Cartesian form. To force the conversion to polar coordinates enter the vector components as real numbers (i.e., add a decimal point), e.g., [2., 3., 5.].

With the cylindrical coordinate system selected, if we enter a vector in spherical coordinates it will be automatically transformed to its cylindrical (polar) equivalent (r,θ,z) with r = ρ sin φ, θ = θ, z = ρ cos φ. For example, the following figure shows the vector entered in spherical coordinates, and transformed to polar coordinates. For this case, ρ = 5, θ = 25°, and φ = 45°, while the transformation shows that r = 3.563, and z = 3.536. (Change to DEG):





Next, let's change the coordinate system to spherical coordinates by using function SPHERE from the VECTOR sub-menu in the MTH menu. When this coordinate system is selected, the display will show the R∠∠ format in the top line. The last screen will change to show the following:



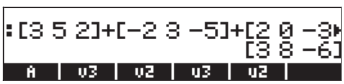
Notice that the vectors that were written in cylindrical polar coordinates have now been changed to the spherical coordinate system. The transformation is such that $\rho = (r^2+z^2)^{1/2}$, θ = θ, and $\phi = \tan^{-1}(r/z)$. However, the vector that originally was set to Cartesian coordinates remains in that form.

Application of vector operations

This section contains some examples of vector operations that you may encounter in Physics or Mechanics applications.

Resultant of forces

Suppose that a particle is subject to the following forces (in N): $\mathbf{F}_1$ = 3**i**+5**j**+2**k**, $\mathbf{F}_2$ = -2**i**+3**j**-5**k**, and F_3 = 2**i**-3**k**. To determine the resultant, i.e., the sum, of all these forces, you can use the following approach in ALG mode:



Thus, the resultant is R = $\mathbf{F}_1$+ $\mathbf{F}_2$ + $\mathbf{F}_3$ = (3**i**+8**j**-6**k**)N. RPN mode use:

[3,5,2] ENTER [-2,3,-5] ENTER [2,0,3] ENTER + +

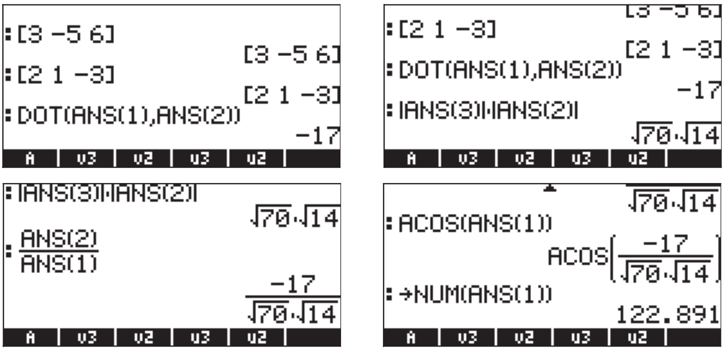
Angle between vectors

The angle between two vectors **A**, **B**, can be found as θ =cos^{-1}(**A•B**/|**A**||**B**|)

Suppose that you want to find the angle between vectors **A** = 3**i**-5**j**+6**k**, **B** = 2**i**+**j**-3**k**, you could try the following operation (angular measure set to degrees) in ALG mode:

1 - Enter vectors [3,-5,6], press ENTER, [2,1,-3], press ENTER.
2 - DOT(ANS(1),ANS(2)) calculates the dot product
3 - ABS(ANS(3))*ABS((ANS(2)) calculates product of magnitudes
4 - ANS(2)/ANS(1) calculates cos(θ)
5 - ACOS(ANS(1)), followed by ,→NUM(ANS(1)), calculates θ

The steps are shown in the following screens (ALG mode, of course):



Thus, the result is θ = 122.891°. In RPN mode use the following:

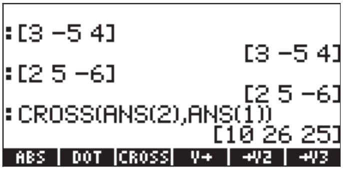
[3,-5,6] ENTER [2,1,-3] ENTER DOT

[3,-5,6] ENTER ABS [2,1,-3] ENTER ABS ×

÷ ACOS →NUM

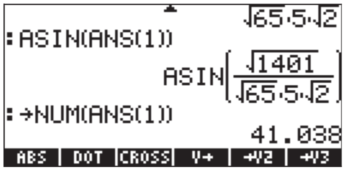
Moment of a force

The moment exerted by a force **F** about a point O is defined as the cross-product **M** = **r**×**F**, where **r**, also known as the arm of the force, is the position vector based at O and pointing towards the point of application of the force. Suppose that a force **F** = (2**i**+5**j**-6**k**) N has an arm **r** = (3**i**-5**j**+4**k**)m. To determine the moment exerted by the force with that arm, we use function CROSS as shown next:



Thus, **M** = (10**i**+26**j**+25**k**) m·N. We know that the magnitude of **M** is such that |**M**| = |**r**||**F**|sin(θ), where θ is the angle between **r** and **F**. We can find this angle as, θ = $\sin^{-1}$(|**M**| /|**r**||**F**|) by the following operations:
1 – ABS(ANS(1))/(ABS(ANS(2))*ABS(ANS(3)) calculates sin(θ)
2 – ASIN(ANS(1)), followed by →NUM(ANS(1)) calculates θ

These operations are shown, in ALG mode, in the following screens:

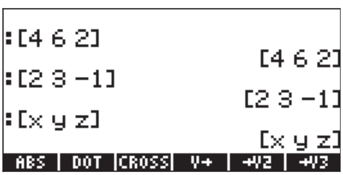


Thus the angle between vectors **r** and **F** is θ = 41.038°. RPN mode, we can use: [3,-5,4] ENTER [2,5,-6] ENTER CROSS ABS [3,-5,4] ENTER ABS [2,5,-6] ENTER ABS × ÷ ASIN →NUM

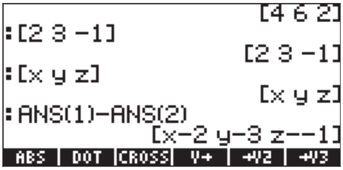
Equation of a plane in space

Given a point in space $P_0(x_0,y_0,z_0)$ and a vector **N** = N_x**i**+N_y**j**+N_z**k** normal to a plane containing point P_0, the problem is to find the equation of the plane. We can form a vector starting at point P_0 and ending at point P(x,y,z), a generic point in the plane. Thus, this vector **r** = P_0P = $(x-x_0)$**i**+ $(y-y_0)$**j** + $(z-z_0)$**k**, is perpendicular to the normal vector **N**, since **r** is contained entirely in the plane. We learned that for two normal vectors **N** and **r**, **N•r** =0. Thus, we can use this result to determine the equation of the plane.

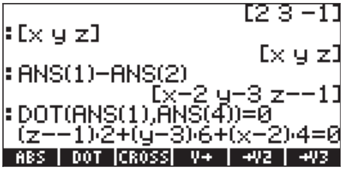
To illustrate the use of this approach, consider the point $P_0(2,3,-1)$ and the normal vector **N** = 4**i**+6**j**+2**k**, we can enter vector **N** and point P_0 as two vectors, as shown below. We also enter the vector [x,y,z] last:



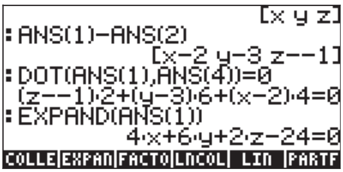
Next, we calculate vector P_0P = **r** as ANS(1) – ANS(2), i.e.,



Finally, we take the dot product of ANS(1) and ANS(4) and make it equal to zero to complete the operation **N•r** =0:



We can now use function EXPAND (in the ALG menu) to expand this expression:



Thus, the equation of the plane through point P_0(2,3,-1) and having normal vector **N** = 4**i**+6**j**+2**k**, is 4x + 6y + 2z – 24 = 0. In RPN mode, use:

`[2,3,-1]` ENTER `['x','y','z']` ENTER − `[4,6,2] DOT EXPAND`

Row vectors, column vectors, and lists

The vectors presented in this chapter are all row vectors. In some instances, it is necessary to create a column vector (e.g., to use the pre-defined statistical functions in the calculator). The simplest way to enter a column vector is by enclosing each vector element within brackets, all contained within an external set of brackets. For example, enter:

`[[1.2],[2.5],[3.2],[4.5],[6.2]]` ENTER

This is represented as the following column vector:

In this section we will showing you ways to transform: a column vector into a row vector, a row vector into a column vector, a list into a vector, and a vector (or matrix) into a list.

We first demonstrate these transformations using the RPN mode. In this mode, we will use functions OBJ→, →LIST, →ARRY and DROP to perform the transformation. To facilitate accessing these functions we will set system flag 117 to SOFT menus (see Chapter 1). With this flag set, functions OBJ→, →ARRY, and →LIST will be accessible by using ← PRG TYPE. Functions OBJ→, →ARRY, and →LIST will be available in soft menu keys F1, F2, and F3. Function DROP is available by using ← PRG STACK DROP.

Following we introduce the operation of functions OBJ→, →LIST, →ARRY, and DROP with some examples.

Function OBJ→

This function decomposes an object into its components. If the argument is a list, function OBJ→ will list the list elements in the stack, with the number of elements in stack level 1, for example: {1,2,3} ENTER ← PRG TYPE OBJ→ results in:

When function OBJ→ is applied to a vector, it will list the elements of the vector in the stack, with the number of elements in level 1: enclosed in braces (a list). The following example illustrates this application: [1,2,3] ENTER ← PRG TYPE OBJ→ results in:

If we now apply function OBJ→ once more, the list in stack level 1:, {3.}, will be decomposed as follows:

Function →LIST

This function is used to create a list given the elements of the list and the list length or size. In RPN mode, the list size, say, n, should be placed in stack level 1:. The elements of the list should be located in stack levels 2:, 3:, …, n+1:. For example, to create the list {1, 2, 3}, type: 1 ENTER 2 ENTER 3 ENTER 3 ENTER ← PRG TYPE →LIST.

Function →ARRY

This function is used to create a vector or a matrix. In this section, we will use it to build a vector or a column vector (i.e., a matrix of n rows and 1 column). To build a regular vector we enter the elements of the vector in the stack, and in stack level 1: we enter the vector size as a list, e.g., 1 ENTER 2 ENTER 3 ENTER ← {} 3 ENTER ← PRG TYPE →ARRY.

To build a column vector of n elements, enter the elements of the vector in the stack, and in stack level 1 enter the list {n 1}. For example, 1 ENTER 2 ENTER 3 ENTER ← {} 1 → , 3 ENTER ← PRG TYPE →ARRY.

Function DROP

This function has the same effect as the delete key (←).

Transforming a row vector into a column vector

We illustrate the transformation with vector `[1,2,3]`. Enter this vector into the RPN stack to follow the exercise. To transform a row vector into a column vector, we need to carry on the following operations in the RPN stack:

1 - Decompose the vector with function OBJ→

2 - Press 1 + to transform the list in stack level 1: from {3} to {3,1}

3 - Use function →ARRY to build the column vector

These three steps can be put together into a UserRPL program, entered as follows (in RPN mode, still): ↱ « » ↰ PRG TYPE OBJ→ 1 + →ARRY ENTER ' ALPHA ALPHA R X C ENTER STO▶

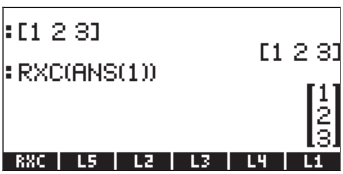
A new variable, RXC, will be available in the soft menu labels after pressing VAR :

Press ↱ RXC to see the program contained in the variable RXC:

<< OBJ→ 1 + →ARRY >>

This variable, RXC, can now be used to directly transform a row vector to a column vector. In RPN mode, enter the row vector, and then press RXC. Try, for example: [1,2,3] ENTER RXC.

After having defined this variable , we can use it in ALG mode to transform a row vector into a column vector. Thus, change your calculator's mode to ALG and try the following procedure: [1,2,3] ENTER VAR RXC ↰ ( ) ↰ ANS , resulting in:



Transforming a column vector into a row vector

To illustrate this transformation, we'll enter the column vector [[1],[2],[3]] in RPN mode. Then, follow the next exercise to transform a row vector into a column vector:

1 - Use function OBJ→ to decompose the column vector

2 - Use function OBJ→ to decompose the list in stack level 1:

3 - Press the delete key ◄ (also known as function DROP) to eliminate the number in stack level 1:

4 - Use function →LIST to create a list

5 - Use function →ARRY to create the row vector

These five steps can be put together into a UserRPL program, entered as follows (in RPN mode, still):

→ « » ← PRG TYPE OBJ→ OBJ→

← PRG STACK DROP ← PRG TYPE →LIST →ARRY ENTER

' ALPHA ALPHA C X R ENTER STO►

A new variable, CXR, will be available in the soft menu labels after pressing VAR :

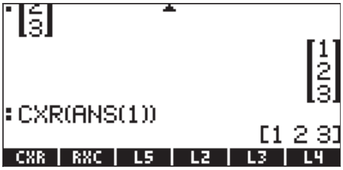
Press → CXR to see the program contained in the variable CXR:

<< OBJ→ OBJ→ DROP →ARRY >>

This variable, CXR, can now be used to directly transform a column vector to a row vector. In RPN mode, enter the column vector, and then press CXR. Try, for example: [[1],[2],[3]] ENTER CXR.
After having defined variable CXR, we can use it in ALG mode to transform a row vector into a column vector. Thus, change your calculator's mode to ALG and try the following procedure:

[[1],[2],[3]] ENTER VAR CXR ← () ← ANS

resulting in:



Transforming a list into a vector

To illustrate this transformation, we'll enter the list {1,2,3} in RPN mode. Then, follow the next exercise to transform a list into a vector:

1 - Use function OBJ→ to decompose the column vector

2 - Type a 1 and use function →LIST to create a list in stack level 1:

3 - Use function →ARRY to create the vector

These three steps can be put together into a UserRPL program, entered as follows (in RPN mode):

→ « » ← PRG TYPE OBJ→ 1 →LIST →ARRY ENTER

' ALPHA ALPHA L X V ENTER STO▶

A new variable, LXV, will be available in the soft menu labels after pressing VAR :

Press ↱ LXV to see the program contained in the variable LXV:

<< OBJ→ 1 →LIST →ARRY >>

This variable, LXV, can now be used to directly transform a list into a vector. In RPN mode, enter the list, and then press LXV. Try, for example: {1,2,3} ENTER LXV.

After having defined variable LXV, we can use it in ALG mode to transform a list into a vector. Thus, change your calculator's mode to ALG and try the following procedure: {1,2,3} ENTER VAR LXV ↰ () ↰ ANS , resulting in:

Transforming a vector (or matrix) into a list

To transform a vector into a list, the calculator provides function AXL. You can find this function through the command catalog, as follows:

↱ CAT ALPHA ALPHA A X L ALPHA OK

As an example, apply function AXL to the vector [1,2,3] in RPN mode by using:[1,2,3] ENTER AXL. The following screen shot shows the application of function AXL to the same vector in ALG mode.