

Features Used

[\angle], [MODE], \blacktriangleright Rect, Cylind, \blacktriangleright Sphere, unitV(), dotP(), crossP(), norm(), Func, \uparrow , \ominus , [COPY], [PASTE], NewProb, \approx

Setup

\square 1, NewFold emag

Vectors This chapter describes how to use the TI-89 for coordinate system transformations, vector algebra, and vector component transformations. The functions **rec2cyl()**, **cyl2rec()**, **rec2sph()**, and **sph2rec()** are created in this chapter for vector component transformations and are used in Chapters 10 and 11.

Topic 37: Coordinate Systems and Coordinate Transformations

Points in 3-D space can be entered in rectangular, cylindrical, and spherical coordinate systems. The TI-89 does this with position vectors, which are vectors that point from the origin to the coordinates of the point in space. On the TI-89, each position vector is represented by the coordinates of its endpoint—(x,y,z) in rectangular, (r, θ ,z) in cylindrical, or (ρ , ϕ , θ) in spherical coordinates.

The TI-89 notation differs in two ways from the standard form of (ρ , ϕ ,z) for cylindrical and (r, θ , ϕ) for spherical coordinates used in most electrical engineering texts. The coordinates are related as $\rho_{STD}=r_{TI}$, $r_{STD}=\rho_{TI}$, $\phi_{STD}=\theta_{TI}$, and $\theta_{STD}=\phi_{TI}$. This difference in angle notation appears to interchange the angles. The two vector forms are related as $[\rho_{STD}, \angle\phi_{STD}, z]_{STD}=[r_{TI}, \angle\theta_{TI}, z]_{TI}$ for cylindrical coordinates and $[r_{STD}, \angle\theta_{STD}, \angle\phi_{STD}]_{STD}=[\rho_{TI}, \angle\phi_{TI}, \angle\theta_{TI}]_{TI}$ for spherical coordinates. Check the Guidebook to be sure of the definition of these variables. The coordinate transformations of this section use the TI-89 form.

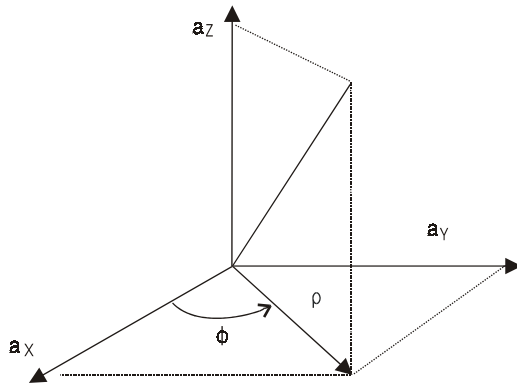


Figure 1a. Cylindrical coordinates

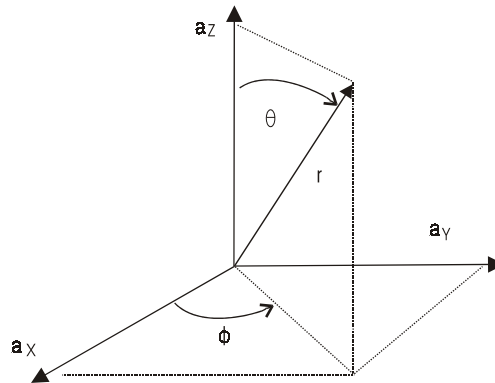


Figure 1b. Spherical coordinates

The TI-89 defines vectors as either a 1x3 row matrix or a 3x1 column matrix. The *vector entry mode* is determined solely by the format of the vector elements and establishes the coordinate system in which the vector is interpreted, but not necessarily displayed.

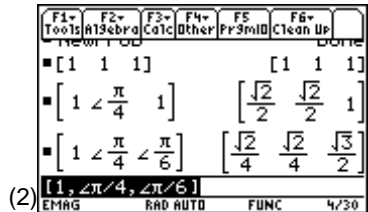
- For rectangular coordinates, the vector is entered as three scalar expressions, for example, $[1, \cos(x+\sqrt{y}), -2.9]$,
- For cylindrical coordinates, the second entry is entered as an angle with a preceding \angle (2^{nd} $[\angle]$), for example, $[z^{\wedge}3, \angle 25, \sin(y*z)]$,
- For spherical coordinates, the second and third entries are entered as angles, for example, $[1, \angle \pi/4, \angle (\theta-3)]$.

Regardless of the entry mode, vectors are displayed according to the **Vector Format** mode—rectangular, cylindrical, or spherical.

1. Clear the TI-89 by pressing 2^{nd} $[F6]$ **2:NewProb** $[ENTER]$. Press $[MODE]$ to verify that **Vector Format** is set to **RECTANGULAR** (screen 1).



2. On the Home screen, enter the rectangular position vector $[1,1,1]$ as shown in the top of screen 2. Enter the cylindrical position vector $[1, \angle \pi/4, 1]$. Enter the spherical position vector $[1, \angle \pi/4, \angle \pi/6]$.



Note that the forms are all displayed in the **RECTANGULAR** format regardless of the entry mode.

3. The TI-89 has three commands that change the display format of these coordinates from one coordinate system to another for the current entry only. The rectangular-to-standard cylindrical coordinate transformation is based upon

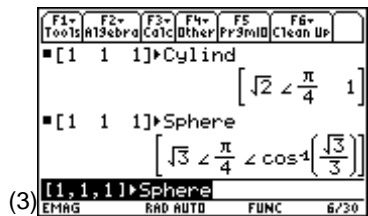
$$\rho = \sqrt{x^2 + y^2} \quad x = \rho \cos \phi$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) \quad y = \rho \sin \phi$$

$$z = z$$

Use **Cylind** to display $[1,1,1]$ in cylindrical components as shown in the top of screen 3.

2^{nd} $[[]]$ 1 $[,]$ 1 $[,]$ 1 2^{nd} $[[]]$ **CATALOG** $[>]$ **Cylind**



4. The rectangular-to-standard spherical coordinate transformation is based upon

$$r = \sqrt{x^2 + y^2 + z^2} \quad x = r \sin \theta \cos \phi$$

$$\theta = \cos^{-1}\left(\frac{z}{r}\right) \quad y = r \sin \theta \sin \phi$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) \quad z = r \cos \theta$$

Use ► **Sphere** to display [1,1,1] in spherical coordinates as shown in the bottom of screen 3.

2nd [] 1 [] 1 [] 1 **2nd** [] **CATALOG** [►] **Sphere**

5. To convert back to original rectangular form, use **ans(1)** to get the previous answer, and then enter ► **Rect** as shown in screen 4.

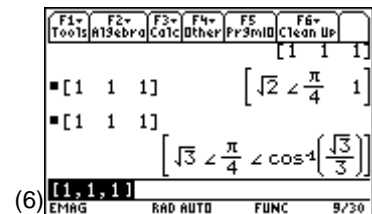
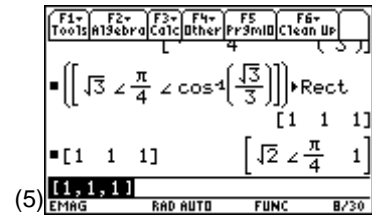
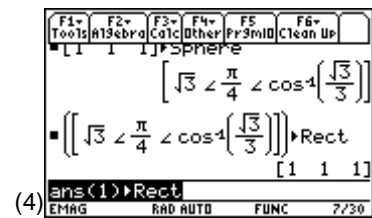
2nd [ANS] **CATALOG** [►] **Rect**

6. Set the **Vector Format** mode to **CYLINDRICAL**.

MODE [▼] [▼] [▼] [▼] [▼] [▼] **2:CYLINDRICAL** **ENTER**

7. Enter [1, 1, 1] which is now displayed in cylindrical format in screen 5.

8. Try this for **SPHERICAL Vector Format** mode also as shown in screen 6.



Topic 38: Vector Components

In the common notation, vectors are given as $\mathbf{v} = A_1 \mathbf{a}_1 + A_2 \mathbf{a}_2 + A_3 \mathbf{a}_3$ where all vectors are noted by **bold** symbols, such as \mathbf{v} , and unit vectors in the i th direction as \mathbf{a}_i .

On the TI-89, the magnitudes of the components, A_1 , A_2 , and A_3 , for \mathbf{v} are entered as a vector. A vector, \mathbf{v} , can be represented in any orthogonal coordinate system as long as the unit vectors, \mathbf{a}_1 , \mathbf{a}_2 , and \mathbf{a}_3 form a right-hand coordinate system.

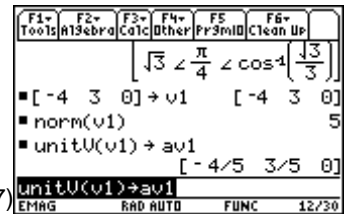
A right-hand coordinate system is one for which the first vector crossed into the second gives the third, the second crossed into the third gives the first, and the third crossed into the first gives the second. Mathematically this is written as $\mathbf{a}_1 \times \mathbf{a}_2 = \mathbf{a}_3$, $\mathbf{a}_2 \times \mathbf{a}_3 = \mathbf{a}_1$, and $\mathbf{a}_3 \times \mathbf{a}_1 = \mathbf{a}_2$. The proper right-hand orders for the standard coordinate systems are $(\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z)$, $(\mathbf{a}_\rho, \mathbf{a}_\phi, \mathbf{a}_z)$, and $(\mathbf{a}_r, \mathbf{a}_\theta, \mathbf{a}_\phi)$.

For operations on vector components, it is best to use the rectangular vector entry and display modes.

1. Set the **Vector Format** mode to **RECTANGULAR**.
2. Enter $\mathbf{v1} = -4\mathbf{a}_x + 3\mathbf{a}_y + 0\mathbf{a}_z$ as shown in screen 7.

$\boxed{2\text{nd}} \boxed{[]} \boxed{(-)} \boxed{4} \boxed{,} \boxed{3} \boxed{,} \boxed{0} \boxed{2\text{nd}} \boxed{[]} \boxed{\text{STO}} \boxed{\triangleright} \boxed{\mathbf{v1}}$

Although $\mathbf{v1}$ was given in rectangular coordinates, this could represent the cylindrical vector $\mathbf{v1} = -4\mathbf{a}_\rho + 3\mathbf{a}_\phi + 0\mathbf{a}_z$ or the spherical vector $\mathbf{v1} = -4\mathbf{a}_r + 3\mathbf{a}_\theta + 0\mathbf{a}_\phi$ equally well.



Note: The convention in this book is to save the unit vector of \mathbf{vn} as \mathbf{avn} .

3. Use $\text{norm}()$ to find the magnitude of $\mathbf{v1}$.
 $\boxed{\text{CATALOG}} \boxed{\text{norm}(\mathbf{v1})} \boxed{[]}$
4. Use $\text{unitV}()$ to find the unit vector in the direction of $\mathbf{v1}$.
 $\boxed{\text{CATALOG}} \boxed{\text{unitV}(\mathbf{v1})} \boxed{[]} \boxed{\text{STO}} \boxed{\triangleright} \boxed{\mathbf{av1}}$

Topic 39: Angle between Vectors

Calculate the angle between $\mathbf{v1}$ from Topic 38 and a second vector $\mathbf{v2}$, which extends from (2,-5,4) to (1,1,3).

1. Enter the two endpoints in rectangular coordinates using position vectors of $[2,-5,4] \rightarrow \mathbf{v2a}$ and $[1,1,3] \rightarrow \mathbf{v2b}$, respectively, as shown in screen 8.
2. $\mathbf{v2}$ is the difference between the two position vectors, see Figure 2. Enter $\mathbf{v2b} - \mathbf{v2a} \rightarrow \mathbf{v2}$ as shown in screen 9.

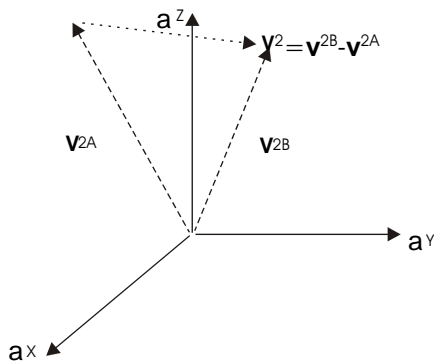
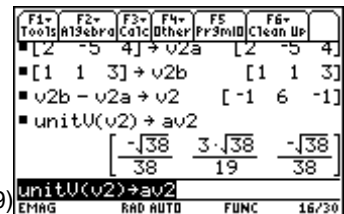
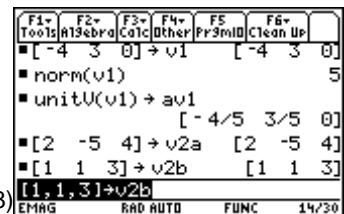


Figure 2. Vector addition

- Use **unitV()** to find the unit vector in the direction of **v2**.

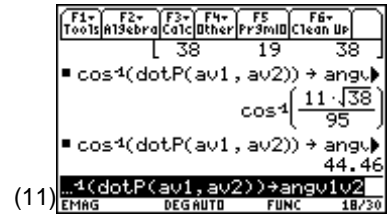
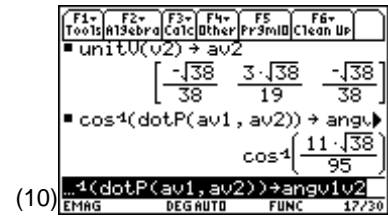
CATALOG **unitV**(**v2**) **STO** **av2**

- Set **Angle** mode to **DEGREE** using **MODE**.

- The angle between two vectors is given by the arccosine of the dot product of their unit vectors, that is, $\cos^{-1}(\text{dotP}(\text{av1}, \text{av2})) \rightarrow \text{angv1v2}$. This is entered as shown in screen 10.

COS⁻¹ **CATALOG** **dotP**(**av1** , **av2**) **STO** **angv1v2**

- Use **◀** **[≈]** to get a floating point value of about 45° (screen 11).



Topic 40: Parallel and Perpendicular Vectors

Other common vector operations are to find the vector components of **v1** that are parallel and perpendicular to **v2**.

Every vector can be decomposed into two orthogonal vector components so that **v1** is the vector sum of two components, one parallel and one perpendicular, **v1par** and **v1perp** as in Figure 3.

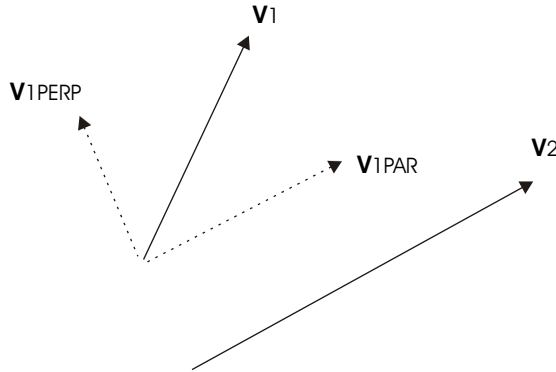


Figure 3. Parallel and perpendicular components

- The magnitude of the component of **v1** that is parallel to **v2** is the dot product of **v1** and **av2** as shown in screen 12.

CATALOG **dotP**(**v1** , **av2**) **STO** **v1parmag**

- The parallel component is the magnitude multiplied by the unit vector **av2** (screen 12).

v1parmag **×** **av2** **STO** **v1par**



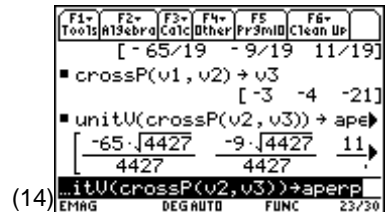
(12) **v1parmag** **×** **av2** **STO** **v1par**

3. Find $\mathbf{v1perp}$ as $\mathbf{v1-v1par} \rightarrow \mathbf{v1perp}$ as shown in screen 13.



4. An alternate method of finding $\mathbf{v1perp}$ begins by calculating the vector perpendicular to the plane containing $\mathbf{v1}$ and $\mathbf{v2}$ using the $\mathbf{crossP()}$ command as shown in screen 14.

```
CATALOG crossP(v1, v2) STO v3
```



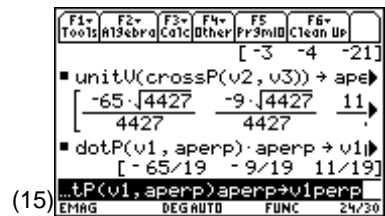
5. The cross product $\mathbf{crossP(v3,v2)} \rightarrow \mathbf{v4}$ is perpendicular to $\mathbf{v2}$ and so is in the direction of $\mathbf{v1perp}$. Therefore, the dot product $\mathbf{dotP(v1,av4)}$ gives the magnitude of the perpendicular component of $\mathbf{v1}$. The vector form of the component of $\mathbf{v1}$ perpendicular to $\mathbf{v2}$ is calculated by $\mathbf{unitV(crossP(v2,v3))}$ as shown in screen 14.

```
CATALOG unitV(CATALOG crossP(v2, v3)) STO aperp
```

6. Calculate the dot product of $\mathbf{dotP(v1,aperp)}$ \mathbf{aperp} .

```
CATALOG dotP(v1, aperp) * aperp STO v1perp
```

Screen 15 shows this result agrees with the earlier calculation of $\mathbf{v1perp}$ in screen 13.



Topic 41: Rectangular to Cylindrical Vector Transformation

Vector components in one system are transformed to another according to the geometrical relationships between the two systems.

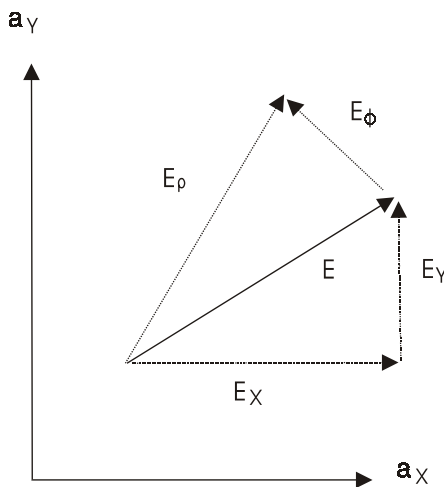


Figure 4. Rectangular and cylindrical components of a vector

The general procedure for transforming any vector is to find its components in the other system using the dot product as $E_i = \mathbf{E} \cdot \mathbf{a}_i$. The rectangular to standard cylindrical transformation of vector components is based on

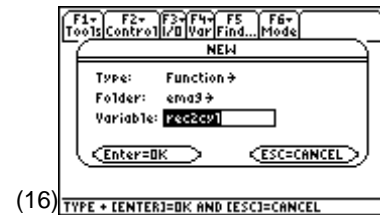
$$\begin{aligned} E_\rho &= E_x \cos \phi + E_y \sin \phi \\ E_\phi &= -E_x \sin \phi + E_y \cos \phi \\ E_z &= E_z \end{aligned}$$

These equations are given in matrix form as

$$\begin{bmatrix} E_\rho \\ E_\phi \\ E_z \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

However, it is time consuming to enter the matrix repeatedly for each transformation, particularly since it depends upon the vector components and the coordinates of the point. Functions are an ideal feature of the TI-89 for this use since they can be used in expressions and return results for different input values.

1. Press **[APPS]** **7:Program Editor 3:New**.
2. Select **2:Function** for **Type** and select **emag** for **Folder**.
3. Type **rec2cyl** for **Variable** and press **[ENTER]** **[ENTER]** to display a template for the new function, as shown in screens 16 and 17.



(16) TYPE + [ENTER]=BK AND [ESC]=CANCEL



(17) EMAG DEG AUTO FUNC

4. Input data is provided to functions through the arguments of the function. For the **rec2cyl** function, the arguments are the rectangular components of the vector (**vec**) and the coordinates of the point at which the transformation is to be evaluated (**pt**). Enter the arguments in the parentheses as **vec** and **pt** (screen 18).



(18) EMAG DEG AUTO FUNC

5. Enter the instructions for the function.

Local rho,x,y

Defines **rho**, **x**, and **y** as local variables. The function arguments as well as all variables listed in the **Local** instruction exist only within the function and are erased upon the completion of the function.

pt[1,1]->x
pt[1,2]->y

Gets local variables **x** and **y** from the input vector **pt**.

Note: To enter \uparrow , press **[2nd]** **[CHAR]** **2:Math 9:** \uparrow .

$$\sqrt{(x^2+y^2)} \rightarrow \rho$$

Computes local variable **rho** based on **x** and **y**.

$$[[x/\rho, y/\rho, 0][-y/\rho, x/\rho, 0][0, 0, 1]] * \text{vec}^T$$

Computes $\cos\phi = x/\rho$ and $\sin\phi = y/\rho$, forms the transformation matrix, and performs matrix multiplication. **vec** is transposed to a 3x1 column matrix in order to be multiplied by the transformation matrix. The results are transposed to be put back into the row matrix format used earlier.

- Return to the Home screen, and use the **rec2cyl** function to transform the rectangular vector components [1,1,1] to cylindrical components at the point [-1,2,3].

$$\text{rec2cyl}([1, 1, 1], [-1, 2, 3])$$

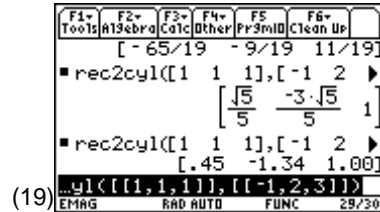
- Press **ENTER** to observe the exact result (top of screen 19).

$$\left[\frac{\sqrt{5}}{5} \quad \frac{-3\sqrt{5}}{5} \quad 1 \right]$$

- Press **2nd** **[=]** to observe the floating point result (bottom of screen 19).

$$[.45 \quad -1.34 \quad 1.00]$$

This result shows the cylindrical components $[E_\rho, E_\phi, E_z]$ of the original rectangular vector.



Topic 42: Cylindrical to Rectangular Vector Transformation

Standard cylindrical components are transformed to rectangular components by

$$\begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_\rho \\ E_\phi \\ E_z \end{bmatrix}$$

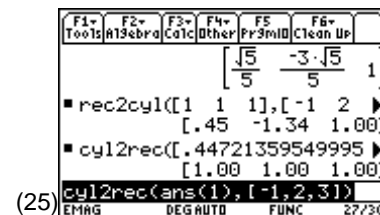
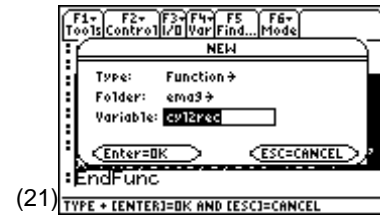
Rather than enter this transformation matrix, the **rec2cyl** function in Topic 41 can be copied to a new function and then edited by making use of the matrix relationship

$$\begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} = \begin{bmatrix} \text{Cyl} \\ \text{to} \\ \text{Rec} \end{bmatrix} \begin{bmatrix} E_\rho \\ E_\phi \\ E_z \end{bmatrix} = \begin{bmatrix} \text{Cyl} \\ \text{to} \\ \text{Rec} \end{bmatrix} \begin{bmatrix} \text{Rec} \\ \text{to} \\ \text{Cyl} \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

which leads to

$$\begin{bmatrix} \text{Cyl} \\ \text{to} \\ \text{Rec} \end{bmatrix} = \begin{bmatrix} \text{Rec} \\ \text{to} \\ \text{Cyl} \end{bmatrix}^{-1}$$

1. Press **[APPS]** **7:Program Editor 1:Current** to display **rec2cyl**.
2. Use **⏪** to move to the beginning of the third line. Hold down **[F]** and press **⏪⏪⏪⏪⏪⏪** to highlight the instructions to be copied as shown in screen 20.
3. To copy the highlighted instructions, press **[♦]** **[COPY]**.
4. Press **[F1]** **3:New** to open a new function. Name the function **cyl2rec** (screen 21).
5. Enter the arguments of **vec** and **pt** as before (screen 22).
6. Move the cursor to the third line and press **[♦]** **[PASTE]** to paste the copied instructions to **cyl2rec** (screen 23).
7. The transformation from cylindrical to rectangular components uses the inverse of the original matrix. Edit the last instruction by inserting **^** **[]** **(-)** **1** **[]** as highlighted in screen 24.
8. To verify that it works, return to the Home screen. Use **ans(1)** as the first argument of **cyl2rec** to transform the cylindrical components from Topic 41 back to rectangular components (screen 25).



Topic 43: Rectangular to Spherical Vector Transformation

Transformation from rectangular to standard spherical coordinates is based on

$$\begin{bmatrix} E_r \\ E_\theta \\ E_\phi \end{bmatrix} = \begin{bmatrix} \sin \theta \cos \phi & \sin \theta \sin \phi & \cos \theta \\ \cos \theta \sin \phi & \cos \theta \cos \phi & -\sin \theta \\ -\sin \phi & \cos \phi & 0 \end{bmatrix} \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

A function **rec2sph** is constructed in a manner similar to **rec2cyl** (Topic 41).

1. Press **[APPS]** **7:Program Editor 3:New** and name the function **rec2sph**.
2. Enter the function lines shown below.

```
:rec2sph(vec,pt)
:Func
:Local r,ra,x,y,z
:pt[1,1]->x:pt[1,2]->y:pt[1,3]->z
:√(x^2+y^2)->ra:√(ra^2+z^2)->r
:([[x/r,y/r,z/r][x*z/r/ra,y*z/r/ra,-ra/r][ -y/ra,x/ra,0]]*vec^T)
:EndFunc
```

3. Return to the Home screen and use **rec2sph** to transform the rectangular components [1,1,1] to spherical components at the point [-1,2,3].

```
rec2sph ([ 2nd [ ] 1 [ ] 1 [ ] 1 2nd [ ] [ ] 2nd [ ] (-) 1 [ ] 2 [ ] 3 2nd [ ] [ ] )
```

The exact result shown in screen 26 is

$$\left[\frac{2\sqrt{14}}{7} \quad \frac{-\sqrt{70}}{35} \quad \frac{-3\sqrt{5}}{5} \right]$$



(26)

Note: The function **rec2sph** gives undefined results for $x=y=0$. A more robust function could be written using the **when()** command.

Topic 44: Spherical to Rectangular Vector Transformation

The standard spherical to rectangular transformation **sph2rec** uses the matrix inverse as in Topic 42 and transforms the spherical components back to the original rectangular components.

Screen 27 shows **sph2rec** with the instructions copied from **rec2sph**. Follow the procedure from Topic 42.

Screen 28 shows how to edit to function for the matrix inverse.

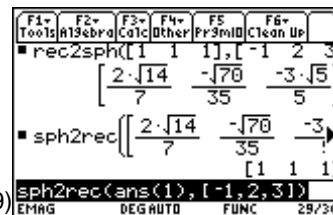
Screen 29 shows the spherical components transformed back to the rectangular components in Topic 43.



(27)



(28)



(29)

Tips and Generalizations

Mathematicians and engineers have chosen different “standard” ways of representing vectors. The TI-89 follows the mathematicians’ standards. The functions developed here always follow the engineers’ standards.

All of the transformations in this chapter work for symbolic entries as well, although they often lead to quite complicated results.

The real excitement begins when Chapter 10 builds on this chapter and introduces vector calculus using the TI-89.

