Vectors: Force and Position

2.1E Using Vectors in EES

Vectors are useful in many areas of engineering. You will find yourself working with vectors throughout your mechanics classes as well as others. To help you work with vectors it is possible to define a special, vector variable in EES. A vector variable has three scalar variables corresponding to its *x*-, *y*-, and *z*-components. Many problems are two-dimensional (2D) and in this case you should define your vector as being a 2D vector so that the *z*-component is automatically ignored by EES.

Assigning and Using Vector Variables

To declare that an EES variable is a vector you must use the $Vector directive followed by a list of one or more vector variables. Each variable that is defined as a vector will have three scalar components that are designated with the subscripts _x, _y, and _z. Vector variables can be assigned using the **VectorAssign** command, which takes three arguments corresponding to the *x*-, *y*-, and *z*-components. It is also possible to assign each of the scalar vector components separately. Finally, vectors can be assigned using the **VectorAssignPolar** command which requires three arguments: the magnitude and the angles with respect to the *x*- and *y*- axes, θ_x and θ_y, respectively. These *directional angles* are discussed in Section 2.3 of the text and shown in Figure 2.1(a).

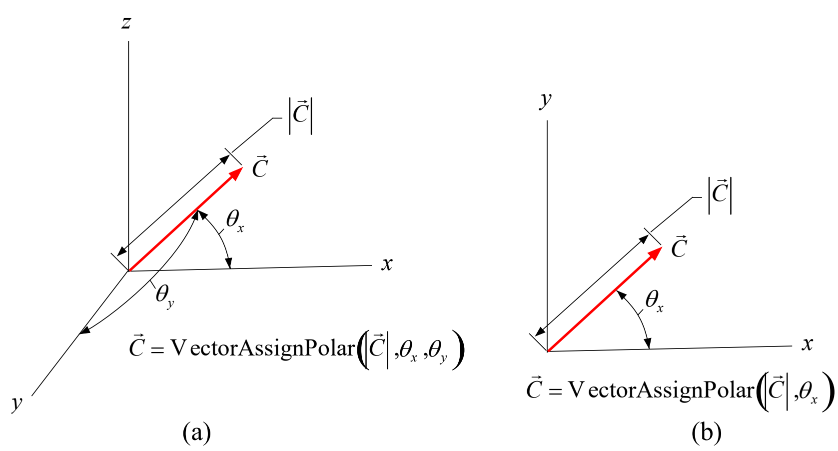
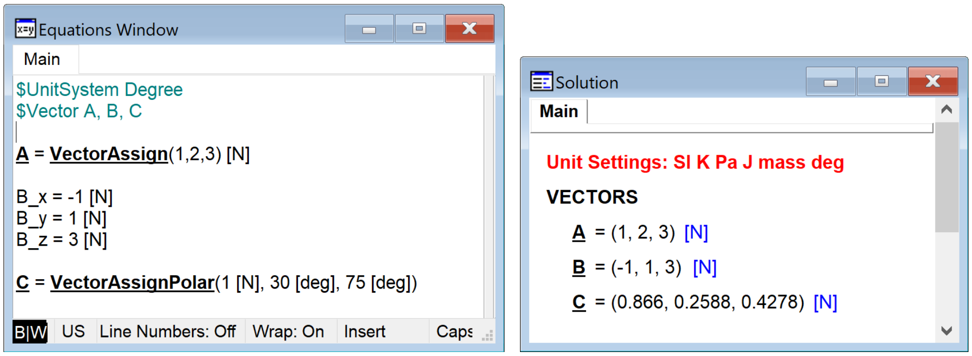


Figure 2.1. Arguments of the **VectorAssignPolar** command used to assign a (a) 3D vector from the magnitude and two directional angles and (b) 2D vector from the magnitude and one directional angle.

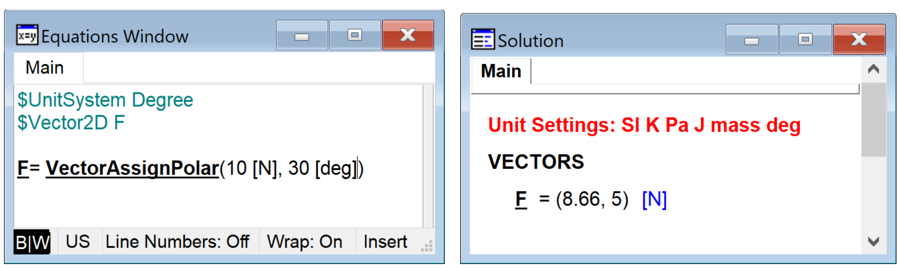
All three approaches for assigning a 3D vector are shown in Figure 2.2 (note that because **VectorAssignPolar** requires angles as arguments it is important that you have specified the unit system, as discussed in Section 1.3).



(a) (b)

Figure 2.2. (a) Equations window showing the vector variables A assigned using the **VectorAssign** command, B assigned component by component, and C assigned using the **VectorAssignPolar** command. (b) Solutions Window.

A 2D vector variable is a special case of the general, 3D vector variable in which the third component is ignored during any operations and computations. To declare that a variable is a 2D vector use the $Vector2D directive followed by a list of 2D vectors. 2D vectors can be assigned using the **VectorAssign** command but then only two arguments are permitted, corresponding to the *x*- and *y*-components. 2D vectors can also be assigned using the **VectorAssignPolar** command, but again only two arguments are permitted, corresponding to the magnitude and the angle relative to the *x*-axis as shown in Figure 2.1(b). Figure 2.3 defines the vector variable **F** to be a 2D force vector with magnitude 10 N and angle 30º.



(a) (b)

Figure 2.3. (a) Equations Window showing the use of the VectorAssignPolar function to assign a 2D vector with magnitude 10 N and angle 30º. (b) Solutions Window showing the *x*- and *y*-components of the resulting vector.

There are a few vector functions that provide useful pre-assigned vectors such as the zeros vector or unit vectors, as summarized in Table 2.1.

Table 2.1. Summary of functions that return pre-assigned vector variables.

Description	Function	Returns in 3D	Returns in 2D
Zeros vector	**VectorZeros**	$0\hat{i}+0\hat{j}+0\hat{k}$	$0\hat{i}+0\hat{j}$
Unit vector in *x*-direction	**VectorUnit_i**	$1\hat{i}+0\hat{j}+0\hat{k}$	$1\hat{i}+0\hat{j}$
Unit vector in *y*-direction	**VectorUnit_j**	$0\hat{i}+1\hat{j}+0\hat{k}$	$0\hat{i}+1\hat{j}$
Unit vector in *z*-direction	**VectorUnit_k**	$0\hat{i}+0\hat{j}+1\hat{k}$	N/A

You can access and manipulate each component of a vector in the same way as any other EES scalar variable. In this way it is possible to carry out any of the vector operations introduced in

Chapter 2 of the text manually. For example we can add two vectors $\vec{v}_1$ and $\vec{v}_2$ to get a resultant vector $\vec{R}$ by adding each of their components according to

$$\vec{R} = \vec{v}_1 + \vec{v}_2 = \left(v_{1,x} + v_{2,x}\right)\hat{i} + \left(v_{1,y} + v_{2,y}\right)\hat{j} + \left(v_{1,z} + v_{2,z}\right)\hat{k}. \tag{2.1}$$

EES allows you to accomplish vector addition automatically by simply adding vector variable **v_1** to vector variable **v_2**, as shown in Figure 2.4.

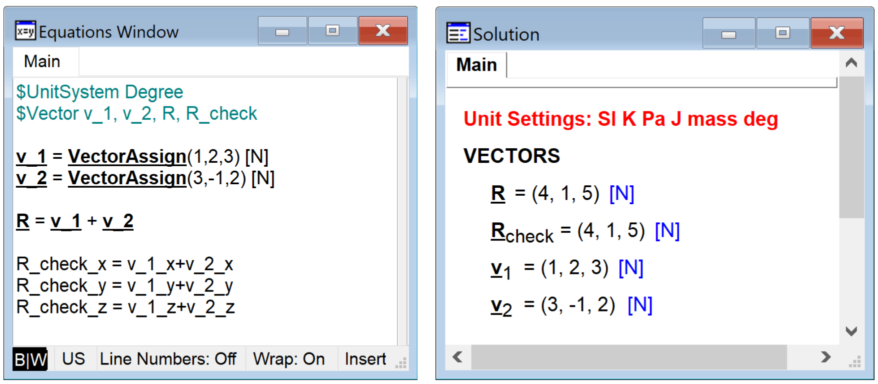


Figure 2.4. (a) Vector addition $\vec{R} = \vec{v}_1 + \vec{v}_2$ accomplished automatically using the vector addition operation and manually by adding each of the components according to Eq. (2.1). (b) Solutions Window showing that both methods provide the same answer.

Most of the vector operations introduced in Chapter 2 are available in EES for vector variables. For example, the magnitude of a vector is obtained according to

$$\left|\vec{v}\right| = \sqrt{v_x^2 + v_y^2 + v_z^2}, \tag{2.2}$$

and the angle between the vector and the various coordinate axes (the direction angles, shown in) can be computed according to:

$$\cos\left(\theta_x\right) = \frac{v_x}{\left|\vec{v}\right|},\ \cos\left(\theta_y\right) = \frac{v_y}{\left|\vec{v}\right|},\ \text{and}\ \cos\left(\theta_z\right) = \frac{v_z}{\left|\vec{v}\right|}. \tag{2.3}$$

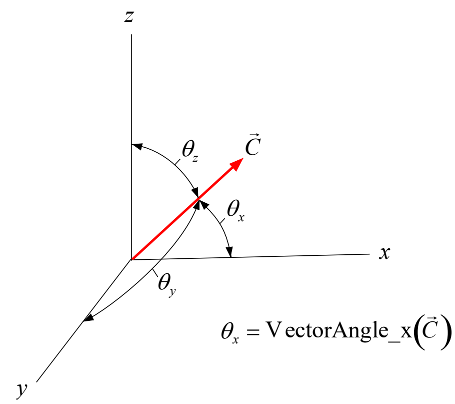
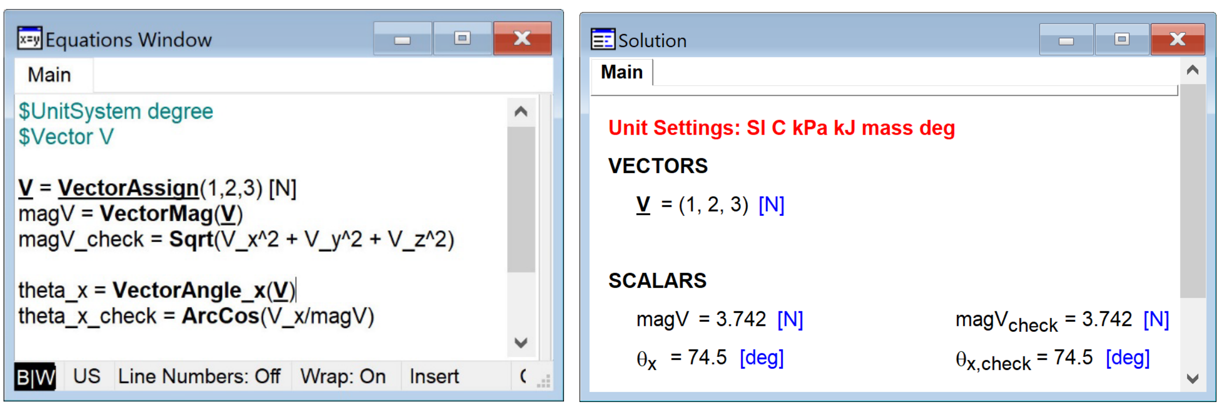


Figure 2.5. Direction angles for a 3D vector.

In EES, the magnitude of a vector variable can be obtained using the **VectorMag** function and the angles can be determined using the functions **VectorAngle_x**, **VectorAngle_y**, and **VectorAngle_z**, as shown in Figure 2.6.



(a) (b)

Figure 2.6. (a) Equations Window showing the use of the **VectorMag** and **VectorAngle_x** functions to determine the magnitude and angle relative to the *x*-coordinate as well as carrying out these calculations manually. (b) Solutions Window showing that the results are equivalent.

EXAMPLE E2.1 *Addition of Vectors and Working Loads*

We can revisit Example 2.6 from the textbook using EES to illustrate the application of vector variables and their operations. A short post *AB* has a commercially manufactured eyebolt screwed into its end. Three cables attached to the eyebolt apply the forces shown in Figure 1.

(a) Determine the resultant force applied to the eyebolt by the three cables, using a Cartesian vector representation.

(b) The manufacturer of the eyebolt specifies a maximum working load 2100 lb_f in the direction of the eyebolt's axis. When loads are not in the direction of the eyebolt's axis, the manufacturer specifies reduction of the maximum working load using the multipliers given in Figure 1. Determine if this size eyebolt is satisfactory.

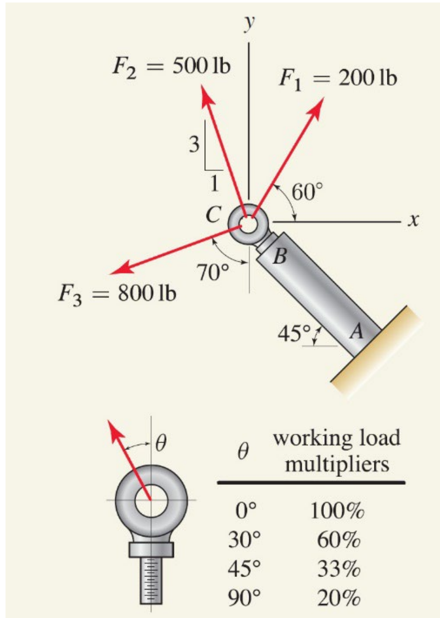


Figure 1

SOLUTION

Road Map The calculations will be the same as those carried out in Exercise 2.6 of the text, except that they will be accomplished using EES.

Part (a)

Governing Equations We will use the same equations and solution methodology as described in Example 2.6 of the textbook except that the calculations will be carried out using EES. This approach allows us to do some interesting additional analysis in the Discussion & Verification section.

Each of the applied forces are specified. The resultant force is the vector summation of the applied forces

$$\vec{R} = \vec{F}_1 + \vec{F}_2 + \vec{F}_3. \tag{1}$$

Computation We will enter the forces $\vec{F}_1$, $\vec{F}_2$, and $\vec{F}_3$ using the **VectorAssignPolar** and the **VectorAssign** commands in EES. Notice that each of these variables (F_1, F_2, and F_3) were declared to be 2D vector variables using a $Vector2D directive and also that the $UnitSystem Degree directive is used to ensure that angles are interpreted as being in degree.

```
$UnitSystem Degree

$Vector2D F_1, F_2, F_3
F_1 = VectorAssignPolar(200 [lbf], 60 [deg])          "force 1 on eyebolt"
F_2 = 500 [lbf]*VectorAssign(-1,3)/Sqrt(10)           "force 2"
F_3 = VectorAssignPolar(800 [lbf], -160 [deg])        "force 3"
```

The resultant vector $\vec{R}$ is obtained by vector addition of the three forces. The **VectorMag** function is used to determine its magnitude and the **VectorAngle_x** function provides its angle relative to the x-axis.

```
$Vector2D R
R = F_1 + F_2 + F_3                    "resultant of forces on the eyebolt"
magR=VectorMag(R)                      "magnitude of resultant"
angleR = VectorAngle_x(R)              "angle of resultant relative to x-axis"
```

The solution shows that vector $\vec{R} = \left(-809.9\hat{i} + 373.9\hat{j}\right)$ lb_f which has a magnitude of $\left|\vec{R}\right| = 892.0$ lb_f and an angle, relative to the x-axis, of $\measuredangle\vec{R} = 155.2°$. Notice that the angle calculated in Example 2.6 of the textbook was -24.8° but this angle was defined as being relative to the *negative* x-axis and therefore these results are consistent.

Part (b)

Governing Equations We need to determine the angle between the applied load and the eyebolt axis. The eyebolt axis is at an angle of 135° relative to the x-axis and so we can take the absolute value of the difference between the angle of vector $\vec{R}$ and the angle of the eyebolt axis

$$\theta = \left|\measuredangle\vec{R} - 135°\right|. \tag{2}$$

The multiplier on the 2100 lb_f maximum load must be selected based on the angle according to the table shown in Figure 1 to determine the maximum allowable working load that should be compared to the magnitude of the resultant force to assess whether the eyebolt is satisfactory.

Computation The angle is determined using Eq. (2).

```
theta = Abs(angleR - 135 [deg])        "angle relative to eyebolt axis"
```

which leads to $\theta = 20.22°$, also consistent with the answer calculated in Example 2.6. Because the angle is in the range $15° < \theta < 30°$ the maximum allowable load of 2100 lb_f must be multiplied by 60%.

```
multiplier = 60 [%]*Convert(%,-)       "multiplier on max. load due to angle"
MaxLoad = multiplier*2100 [lbf]        "maximum allowable load"
```

This leads to a maximum load of 1260 lb_f which is larger than the resultant force of 892.0 lb_f leading to the conclusion that the eyebolt is likely satisfactory.

Discussion & Verification The solution to the problem using EES removed a lot of the tedious work associated with determining magnitudes, angles, etc. These calculations are important, and you must know how they are done; however, once mastered they can be left to a computer. Also, we can visualize the solution more easily by plotting these vectors. Creating vector plots is the subject of the next section in this book.

Because we solved the problem using a computer it is relatively easy to exercise the solution to answer interesting questions. For example, we might want to know whether the eyebolt remains satisfactory under loading conditions where only a subset of the forces is applied. To "remove" force $\overrightarrow{F_2}$ we can remove the original **VectorAssign** statement that sets $\overrightarrow{F_2}$ and instead set $\overrightarrow{F_2}$ to the zeros vector (**VectorZeros**). To temporarily remove one or more lines of code simply highlight the line(s) in the Equations Window, right-click, and select Comment Out from the popup menu, as shown in Figure 3. The result will be that the highlighted code is surrounded by comments, removing it from the equation set. To bring the code back do the reverse process, highlight the equations, right-click and select Undo Comment.

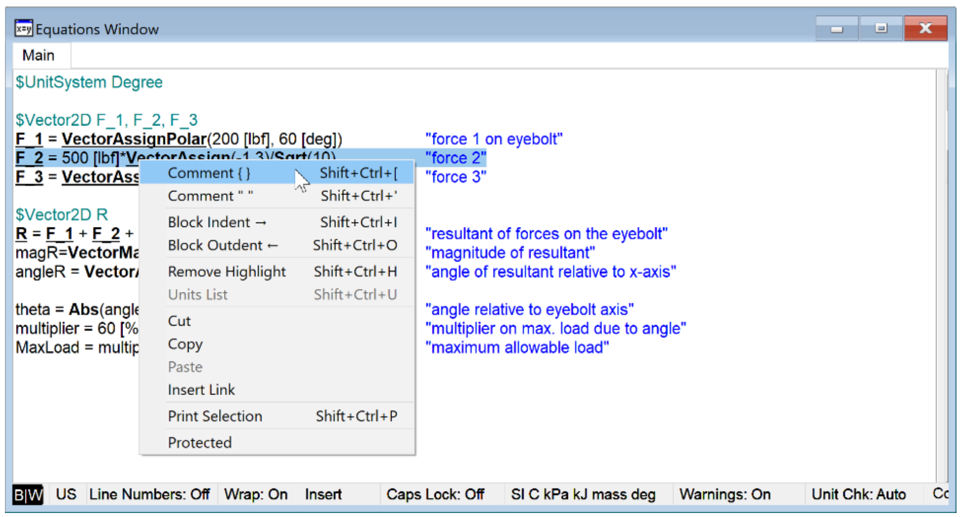
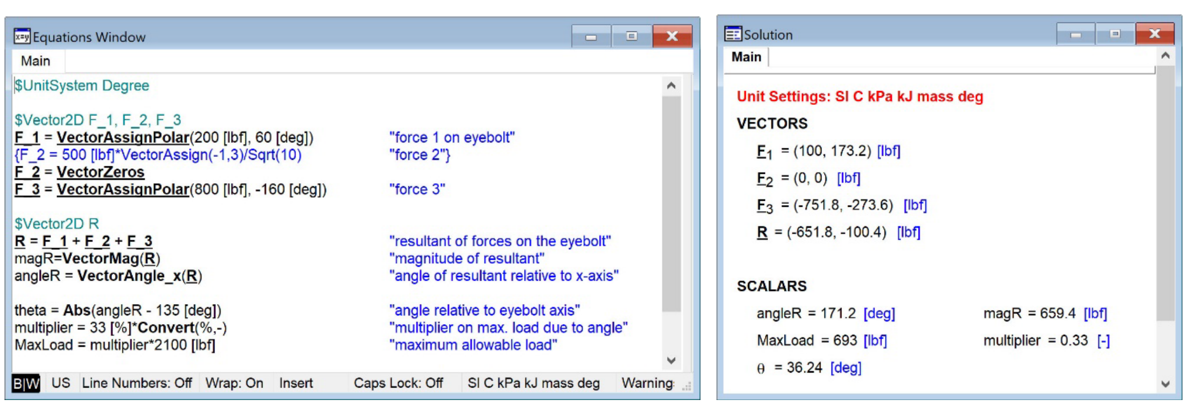


Figure 3. Remove code by highlighting, right-clicking, and selecting Comment.

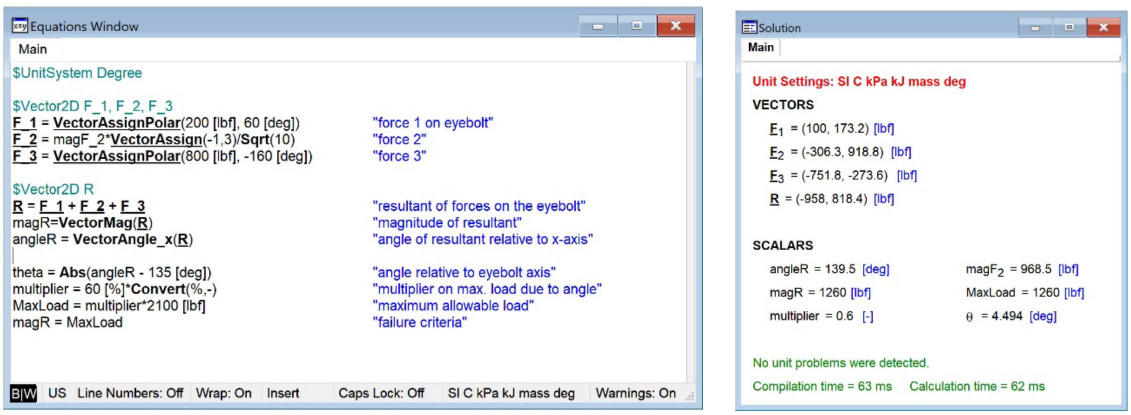
The resulting EES code is shown in Figure 4(a) and the solution is shown in Figure 4(b). The magnitude of the resultant has been reduced to $\left|\overrightarrow{R}\right|$ = 659.4 lb_f but the angle relative to the angle of the eyebolt has been increased to $\theta = 36.24^o$. Because the angle has increased, the multiplier on the maximum load allowed is now 33% according to Figure 1, leading to a maximum load of 693.0 lb_f. The eyebolt is still satisfactory, but it is actually closer to failure than it was with $\overrightarrow{F_2}$ applied.



(a) (b)

Figure 4. (a) Equations Window and (b) Solutions Window with force $\overrightarrow{F_2}$ removed from the eyebolt.

Finally, we can determine how large any one of the forces could be in order to cause the eyebolt to fail. Let's replace the line that assigns the force $\vec{F}_2$ and return the multiplier to 60%. Then let's see how large the force $\vec{F}_2$ would need to be in order to put the eyebolt at risk of failure. We will replace the 500 lb_f magnitude of $\vec{F}_2$ that was specified in the problem statement with a variable, magF_2. The resulting EES code will not solve, because we've added a variable but not an equation. The additional equation is that the magnitude of the resultant force must be equal to the maximum allowable load. The resulting Equations Window is shown in Figure 5(a) and the Solutions Window is shown in Figure 5(b). The solution shows that if $\left|\vec{F}_2\right|$ is increased from 500 lb_f to 968.5 lb_f then the magnitude of the resultant force will increase to the maximum allowable load. Note that the angle of the resultant force is 4.494° at this condition and so a multiplier of 60% remains valid.



(a) (b)

Figure 5. (a) Equations window with the magnitude of force $\vec{F}_2$ calculated in order that the magnitude of the resultant force is equal to the maximum allowable load. (b) Solutions Window showing that $\left|\vec{F}_2\right|$ = 968.5 lb_f results in the eyebolt reaching its maximum allowable load.

Dot Product and Cross Product

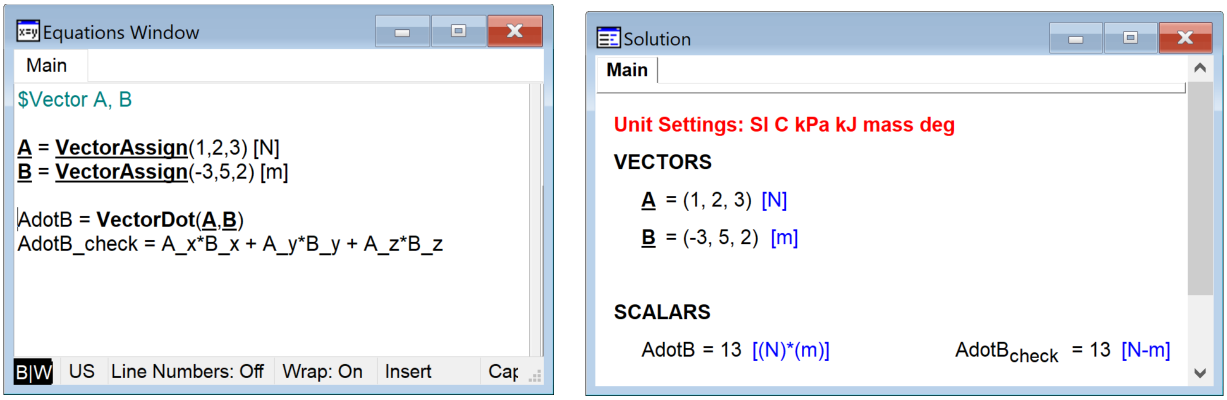
The dot product of two vectors is defined as

$$\vec{A}\cdot\vec{B} = \left|\vec{A}\right|\left|\vec{B}\right|\cos\left(\theta\right) \tag{2.4}$$

where θ is the angle between the two vectors. In Cartesian coordinates, the dot product is obtained according to

$$\vec{A}\cdot\vec{B} = A_x\,B_x + A_y\,B_y + A_z\,B_z \tag{2.5}$$

The dot product in EES can be obtained using the **VectorDot** function which takes as its arguments two vector variables, as shown in Figure 2.7.



(a) (b)

Figure 2.7. (a) Equations Window showing dot product of vector variables A and B computed using the VectorDot function as well as manually by adding the product of the components. (b) Solutions Window showing that these two calculations provide the same result.

The dot product is often used to decompose a force ($\vec{F}$) into components that are parallel to ($\vec{F}_{\parallel}$) and perpendicular to ($\vec{F}_{\perp}$) some direction defined by a position vector ($\vec{r}$), as shown in Figure 2.8. The magnitude of the component parallel to the position vector is given by

$$F_{\parallel} = \vec{F} \cdot \frac{\vec{r}}{|\vec{r}|}, \tag{2.6}$$

and the vector component is

$$\vec{F}_{\parallel} = F_{\parallel} \frac{\vec{r}}{|\vec{r}|}. \tag{2.7}$$

The magnitude of the component perpendicular to the position vector is

$$F_{\perp} = \sqrt{F^2 - F_{\parallel}^2}, \tag{2.8}$$

and the vector component is

$$\vec{F}_{\perp} = \vec{F} - \vec{F}_{\parallel}. \tag{2.9}$$

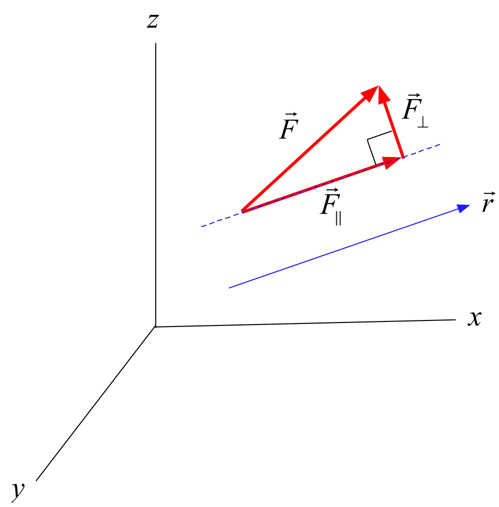


Figure 2.8. Decomposition of a force into its components parallel and perpendicular to a position vector.

EXAMPLE E2.2 *Component of a Force in a Particular Direction*

We can revisit Example 2.17 from the text using EES to illustrate the use of the **VectorDot** function. Rod *AB* is straight and has a bead at location *C*. An elastic cord having a 3 lb_f tensile force is attached between the bead and a support at *D*.

(a) Determine the components of the cord force in directions parallel and perpendicular to rod *AB*.

(b) Determine the vector components of the cord force in directions parallel and perpendicular to rod *AB*.

(c) If the bead at *C* is free to slide on rod *AB* and is released from rest, will the cord force tend to make the bead slide toward *A* or toward *B*?

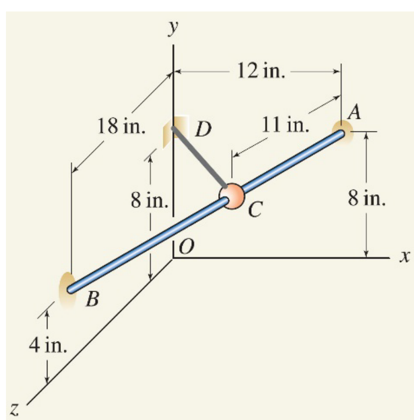


Figure 1.

SOLUTION

Road Map The calculations will be the same as those carried out in Exercise 2.17 of the text except that they will be accomplished using EES. The geometry will be specified by the coordinates that define points A, B, and D and these coordinates can be used to define the position vectors needed for the problem.

Part (a)

Governing Equations The position vector required to move from one end of the rod to the other (i.e., point A to point B) is given by

$$\vec{r}_{AB} = B - A. \tag{1}$$

The position vector from one end of the rod to the bead (i.e., point A to point C) has magnitude 11 inch and is in the same direction as $\vec{r}_{AB}$. Therefore, we can write $\vec{r}_{AC}$ as the product of its magnitude and the unit vector pointing in the direction of $\vec{r}_{AB}$,

$$\vec{r}_{AC} = \left|\vec{r}_{AC}\right| \frac{\vec{r}_{AB}}{\left|\vec{r}_{AB}\right|}. \tag{2}$$

The position vector from the end of the bar to the fixed end of the cord (i.e., from point A to point D) is

$$\vec{r}_{AD} = D - A. \tag{3}$$

The position vector from the bead to the fixed end of the cord (i.e., from point C to point D) is obtained from the vector equation

$$\vec{r}_{CD} = \vec{r}_{CA} + \vec{r}_{AD} = -\vec{r}_{AC} + \vec{r}_{AD}. \tag{4}$$

The cord force $\vec{F}_{CD}$ has a magnitude of 3 lb_f and is in the direction of $\vec{r}_{CD}$; therefore we can write $\vec{F}_{CD}$ as the product of its magnitude and a unit vector pointing in the direction of $\vec{r}_{CD}$

$$\vec{F}_{CD} = \left|\vec{F}_{CD}\right| \frac{\vec{r}_{CD}}{\left|\vec{r}_{CD}\right|}. \tag{5}$$

In order to obtain the component of the component of the cord force that is in the direction of the bar we will take the dot product of the cord force and a unit vector in the direction of the bar, as indicated by Eq. (2.7)

$$F_{\|} = \vec{F}_{CD} \cdot \frac{\vec{r}_{AB}}{\left|\vec{r}_{AB}\right|}. \tag{6}$$

The component of the cord force perpendicular to the bar is given by Eq. (2.8)

$$F_{\perp} = \sqrt{\left|\vec{F}_{CD}\right|^2 - F_{\parallel}^2}. \tag{7}$$

Equations (1) through (7) are sufficient to solve this problem as they can be entered and solved sequentially.

Computation The coordinates *A*, *B*, and *D* are entered as 3D vectors using the **VectorAssign** command.

```
$Vector A, B, D
A = VectorAssign(12, 8, 0) [inch]          "point A - one end of rod"
B = VectorAssign(0, 4, 18) [inch]          "point B - other end of rod"
D = VectorAssign(0,8,0) [inch]             "point D - fixed end of cord"
```

The position vector $\vec{r}_{AB}$ is entered in EES according to Eq. (1).

```
$Vector r_AB
r_AB = B - A                               "position vector associated with the rod"
```

The position vector $\vec{r}_{AC}$ is computed using Eq. (2)

```
$Vector r_AC
magr_AC = 11 [inch]                        "distance from rod end to bead"
r_AC = magr_AC*r_AB/VectorMag(r_AB)        "position vector from rod end to bead"
```

which leads to $\vec{r}_{AC} = (-6\hat{i} - 2\hat{j} + 9\hat{k})\,\text{inch}$. The position vector $\vec{r}_{AD}$ is entered in EES according to Eq. (3) and Eq. (4) is used to determine $\vec{r}_{CD}$

```
$Vector r_AD, r_CD
r_AD = D - A                               "vector from end of rod to fixed end of cord"
r_CD = -r_AC + r_AD                        "vector from bead to fixed end of cord"
```

The force acting on the bead is computed from Eq. (5)

```
$Vector F_CD
magF_CD = 3 [lbf]                          "magnitude of force from cord"
F_CD = magF_CD*r_CD/VectorMag(r_CD)        "force on bead"
```

which leads to $\vec{F}_{CD} = (-1.636\hat{i} + 0.5455\hat{j} - 2.455\hat{k})\,\text{lb}_\text{f}$. The magnitude of the force acting on the bead in the directions parallel and perpendicular to the bar are computed using Eqs. (6) and (7), respectively.

```
magF_par = VectorDot(F_CD,r_AB)/VectorMag(r_AB) "mag. of force on bead parallel to rod"
magF_perp = Sqrt(magF_CD^2 - magF_par^2)        "mag. of force on bead perpendicular to rod"
```

Solving provides $F_{\|}$ = -1.215 lb$_f$ and $F_{\perp}$ = 2.743 lb$_f$.

Part (b)

Governing Equations The vector corresponding to the force parallel to the bar is the component multiplied by the unit vector in the direction of the bar, as given by Eq. (2.7)

$$\vec{F}_{\|} = F_{\|} \cdot \frac{\vec{r}_{AB}}{\left|\vec{r}_{AB}\right|}. \tag{8}$$

The vector corresponding to the force perpendicular to the bar is obtained from Eq. (2.9)

$$\vec{F}_{\perp} = \vec{F}_{CD} - \vec{F}_{\|}. \tag{9}$$

Computation Equations (8) and (9) are computed directly using EES.

```
$Vector F_par, F_perp
F_par = magF_par*r_AB/VectorMag(r_AB)          "force on bead parallel to rod"
F_perp = F_CD - F_par                          "force on bead perpendicular to rod"
```

Solving provides $\vec{F}_{\|} = (0.6627\,\hat{i} + 0.2209\,\hat{j} - 0.9940\,\hat{k})$ lb$_f$ and $\vec{F}_{\|} = (0.6627\,\hat{i} + 0.2209\,\hat{j} - 0.9940\,\hat{k})$ lb$_f$.

Part (c)

The component of the force parallel to the rod in the direction defined by $\vec{r}_{AB}$ is negative. Therefore, the bead will tend to slide in the negative $\vec{r}_{AB}$ direction, which is towards point A.

Discussion & Verification The solution to the problem using EES again removed a lot of tedious work and probably saved some mistakes. The real power of solving a problem using a computer is that you can play with the solution and see how it behaves in order to gain a deeper understanding of the physical situation. For example, we could ask ourselves what position the bead would "want" to be in if it were allowed to slide unrestrained along the rod. Intuitively, the rod will slide as long as it experiences a non-zero force parallel to the rod direction (i.e., a non-zero $F_{\|}$).

We can modify our EES program by commenting out the magnitude of the position vector $\vec{r}_{AC}$ that defines the position of the bead relative to the end of the rod. This will lead to an undefined equation set as we've removed one equation but not a variable. Trying to solve will lead to a message such as the one shown in Figure 2.

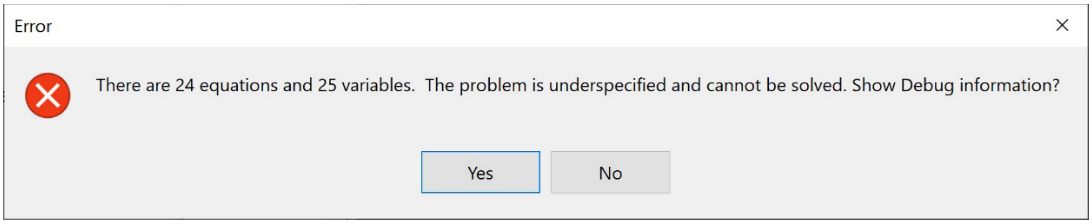
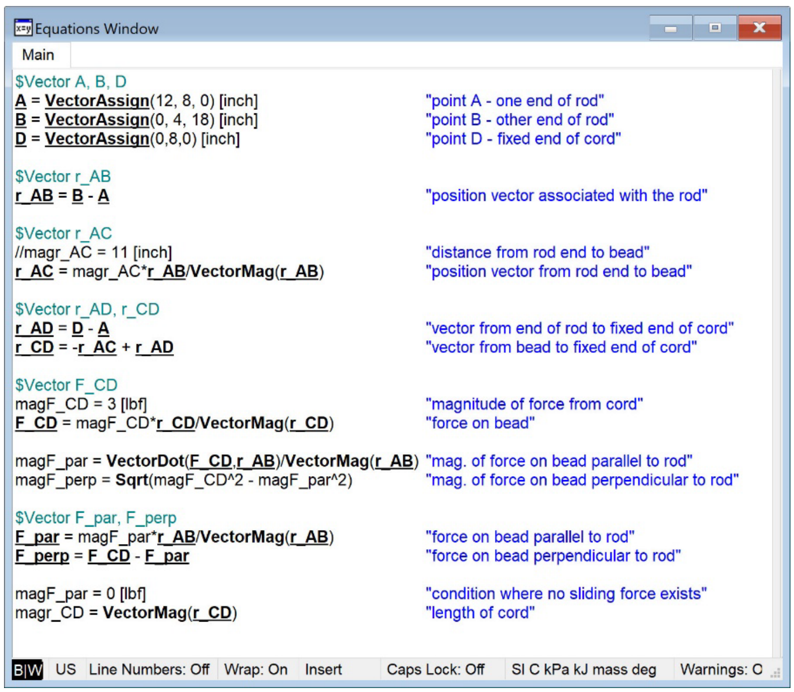
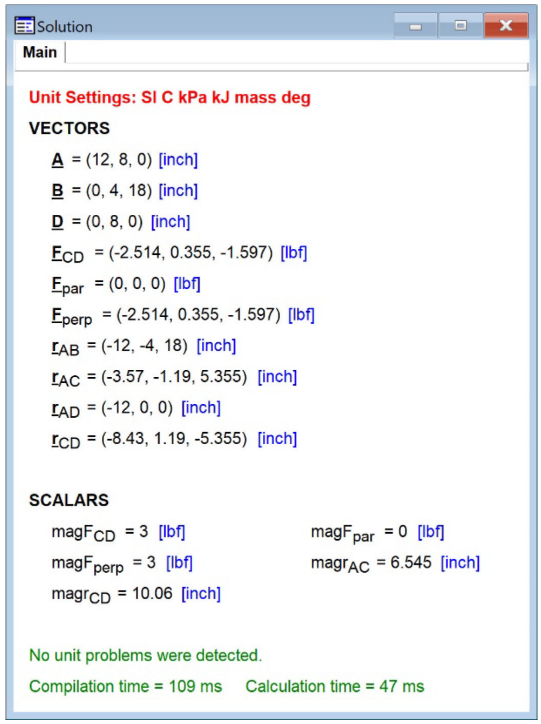


Figure 2: Error message that occurs when the value of the variable magr_AC corresponding to $|\vec{r}_{AC}|$ is commented out.

We need to add another equation, in this case that $F_{\|} = 0$ which corresponds to the condition that the bead is not experiencing a force tending to move it along the rod. The resulting Equations Window is shown in Figure 3(a) and the Solution Window in Figure 3(b).



(a)



(b)

Figure 3. (a) Equations Window showing that the equation specifying the value of the variable magr_AC corresponding to $\left|\vec{r}_{AC}\right|$ has been commented out and instead an equation requiring that the variable magF_par must be zero has been added. (b) Solutions Window showing that the bead will tend to come to rest at a position along the bar that is 6.545 inch from point *A*.

The magnitude of the position vector $\vec{r}_{CD}$ at this point is 10.06 inch which corresponds to the answer to Exercise 2.18 in the text, which asks for the shortest distance between the rod and the end of the cord at point *D*.

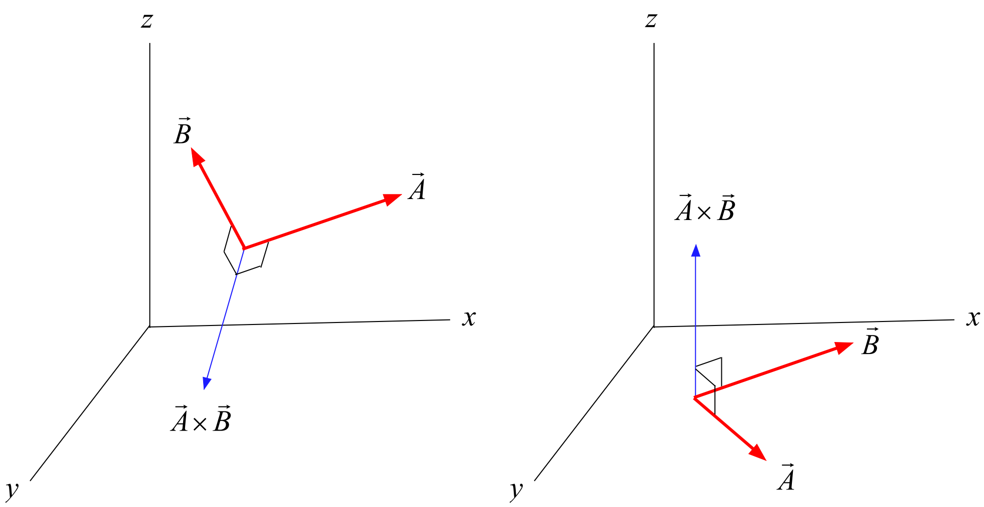
The cross product between two vectors is defined as

$$\vec{A}\times\vec{B}=\left[\left|\vec{A}\right|\left|\vec{B}\right|\sin\left(\theta\right)\right]\hat{u} \tag{2.10}$$

where θ is the angle between the two vectors and $\hat{u}$ is the unit vector that is normal to the plane containing the two vectors (in the direction defined by the right-hand rule), as shown in (a)

(b)

Figure 2.9(a).



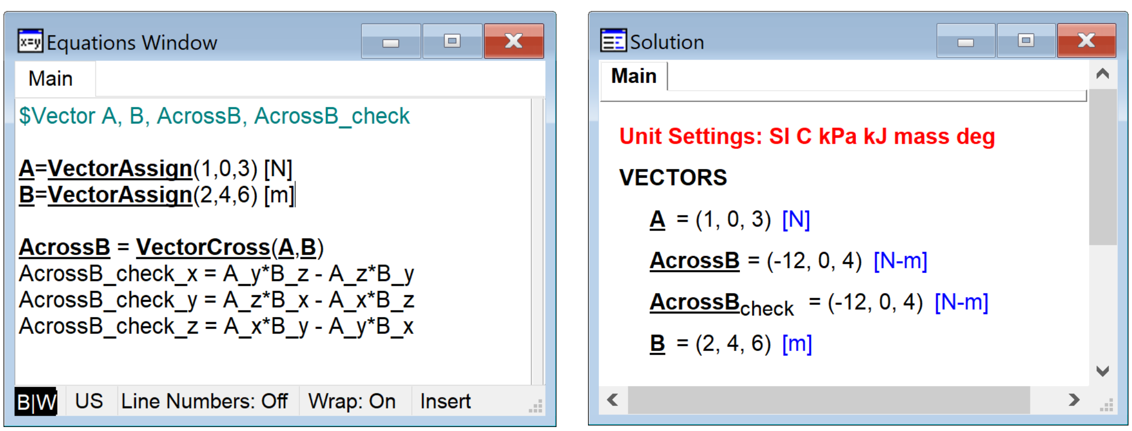
(a) (b)

Figure 2.9. (a) Cross product of two 3D vectors. (b) Cross product of two 2D vectors.

In Cartesian coordinates, the dot product is obtained according to

$$\vec{A}\times\vec{B}=\left(A_y\,B_z-A_z\,B_y\right)\hat{i}+\left(A_z\,B_x-A_x\,B_z\right)\hat{j}+\left(A_x\,B_y-A_y\,B_x\right)\hat{k} \tag{2.11}$$

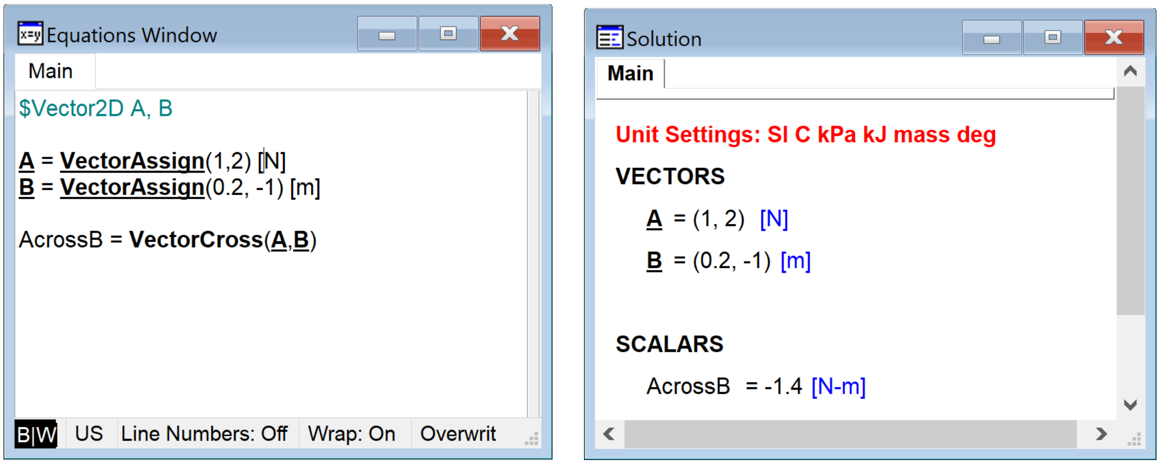
The cross product of two 3D vectors in EES can be obtained using the **VectorCross** function which takes as its arguments two vector variables, as shown in Figure 2.10.



(a) (b)

Figure 2.10. (a) Equations Window showing cross product of vector variables A and B computed using the **VectorCross** function as well as manually according to Eq. (2.11). (b) Solutions Window showing that these two calculations provide the same result.

Note that the product of two, 2D vectors in EES is always a vector in the *z*-direction, as shown in
(a) (b)
Figure 2.9(b). This can be seen from Eq. (2.11) by realizing that the *z*-components of the two, 2D vectors must be zero. Therefore, the function **VectorCross** applied to 2D vectors returns a scalar, which is the magnitude of this vector in the *z*-direction. This can be seen in Figure 2.11.



(a) (b)

Figure 2.11. (a) Equations Window showing the cross product of 2D vector variables A and B becomes (b) a scalar result in the Solutions Window.

Cross products are often used to find the direction normal to a plane and moments related to a force.

EXAMPLE E2.3 *Components of Force in Directions Normal and Tangent to a Plane*

We can revisit Example 2.20 from the text using EES to illustrate the use of the **VectorCross** function. A house with 95 Mg (95×10^6 g) mass is built on a steep slope defined by points *A*, *B*, and *C*. To help assess the possibility of slope failure (and mud slide), it is necessary to:

(a) Determine the components of the weight in directions normal and parallel (tangent) to the slope.

(b) Determine the component vectors of the weight in directions normal and parallel (tangent) to the slope.

(c) Determine the smallest distance from point *O* to the slope.

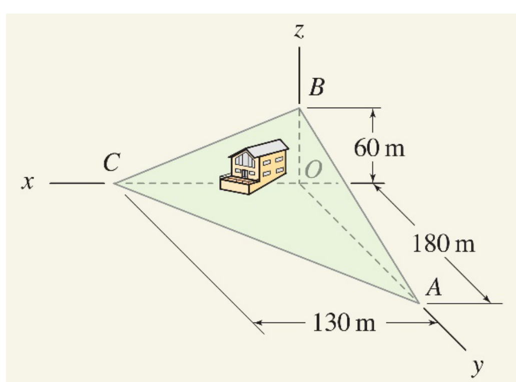


Figure 1

SOLUTION

Road Map The calculations will be the same as those carried out in Exercise 2.20 of the text except that they will be accomplished using EES. The weight can be determined from the mass of the house and the three points defining the plane can be used to define two vectors that both lie in the plane. The cross product of these two in-plane vectors provides a vector normal to the plane. The dot product of the weight with the normal vector can be used to establish the component of the weight perpendicular to the plane which can then be used to determine the component parallel to the plane. This process is similar to what was done in Example E2.1. Finally, we can define a vector from origin to any point on the plane and use the dot product of this point with the normal vector to determine the shortest distance from the origin to the plane.

Part (a)

Governing Equations The three points that define the plane (A, B, and C) are entered and used to determine two vectors that lie in the plane according to:

$$\vec{r}_{AB} = B - A, \text{ and} \tag{1}$$

$$\vec{r}_{AC} = C - A. \tag{2}$$

A vector $\vec{n}$ in the direction normal to the plane and facing toward the sky, is obtained from

$$\vec{n} = \vec{r}_{AB} \times \vec{r}_{AC}. \tag{3}$$

The vector corresponding to the weight of the house is the mass multiplied by the gravity vector

$$\vec{W} = m\,g\left(0\,\hat{i} + 0\,\hat{j} - 1\,\hat{k}\right). \tag{4}$$

The component of the weight vector in the direction of the normal vector is obtained according to

$$W_n = \vec{W} \cdot \frac{\vec{n}}{|\vec{n}|}. \tag{5}$$

The component of the weight vector that is parallel to the plane is obtained from

$$W_t = \sqrt{\left|\vec{W}\right|^2 - W_n^2}. \tag{6}$$

Equations (2) through (4) are sufficient to solve this problem.

Computation The inputs to the problem are entered in EES; these include the points that define the plane as well as the mass of the house. Note that the $UnitSystem SI directive is used to specify that the SI unit system is used so that the EES constant g# can be used to obtain the acceleration of gravity in the correct units.

```
$UnitSystem SI

m = 95e6 [g]*Convert(g,kg)                "mass of house"

$Vector A, B, C                           "three points on the plane"
A = VectorAssign(0, 180, 0) [m]
B = VectorAssign(0, 0, 60) [m]
C = VectorAssign(130, 0, 0) [m]
```

Equations (1) and (2) are used to define the two vectors in the plane:

```
$Vector r_AB, r_AC
r_AB = B - A                              "one position vector on plane"
r_AC = C - A                              "another position vector on plane"
```

Equations (3) through (6) are entered directly into EES.

```
$Vector n
n=VectorCross(r_AB,r_AC)                  "normal vector"

$Vector W
W = m*g#*VectorAssign(0,0,-1)             "weight vector"

magW_n = VectorDot(W,n)/VectorMag(n)      "mag of weight in direction of normal vector"
magW_t = Sqrt(VectorMag(W)^2 - magW_n^2)  "mag of weight in direction tangent to plane"
```

Solving provides W_n = -809.6 kN and W_t = 460.9 kN. The negative sign means the W_n is normal to the plane and directed towards the ground.

Part (b)

Governing Equations The weight normal to the plane is a vector defined by the magnitude W_n in the direction $\vec{n}$

$$\vec{W}_n = W_n \cdot \frac{\vec{n}}{|\vec{n}|}, \tag{5}$$

and the weight tangent to the plane is obtained according to

$$\vec{W} = \vec{W}_n + \vec{W}_t. \tag{6}$$

Computation Equations (5) and (6) are solved directly using EES.

```
$Vector W_n, W_t
W_n = magW_n*n/vectormag(n)               "weight vector normal to plane"
W = W_n + W_t                             "weight vector tangent to plane"
```

Solving provides $\vec{W}_n = (-324.7\,\hat{i} - 234.5\,\hat{j} - 703.6\,\hat{k})\,\text{kN}$ and $\vec{W}_t = (324.7\,\hat{i} + 234.5\,\hat{j} - 228.1\,\hat{k})\,\text{kN}$.

Part (c)

Governing Equations The shortest distance from point *O* to the plane is the dot product of a vector from point *O* to any point in the plane (e.g., $\vec{r}_{OA}$) with the unit normal vector from the plane.

$$r = \vec{r}_{OA} \cdot \frac{\vec{n}}{|\vec{n}|} \tag{7}$$

Computation The position vector $\vec{r}_{OA}$ is defined and Eq. (7) is entered in EES.

```
$Vector r_OA
r_OA = A - VectorZeros                    "position vector from origin to a point on plane"
r = VectorDot(r_OA,n)/VectorMag(n)        "distance from origin to plane"
```

Solving leads to $r = 52.14$ m.

Discussion & Verification As was pointed out in Example 2.20, the shortest distance calculated in part (c) is between point *O* and an infinite plane. We can determine whether the intersection point actually lies on the plane by computing the vector

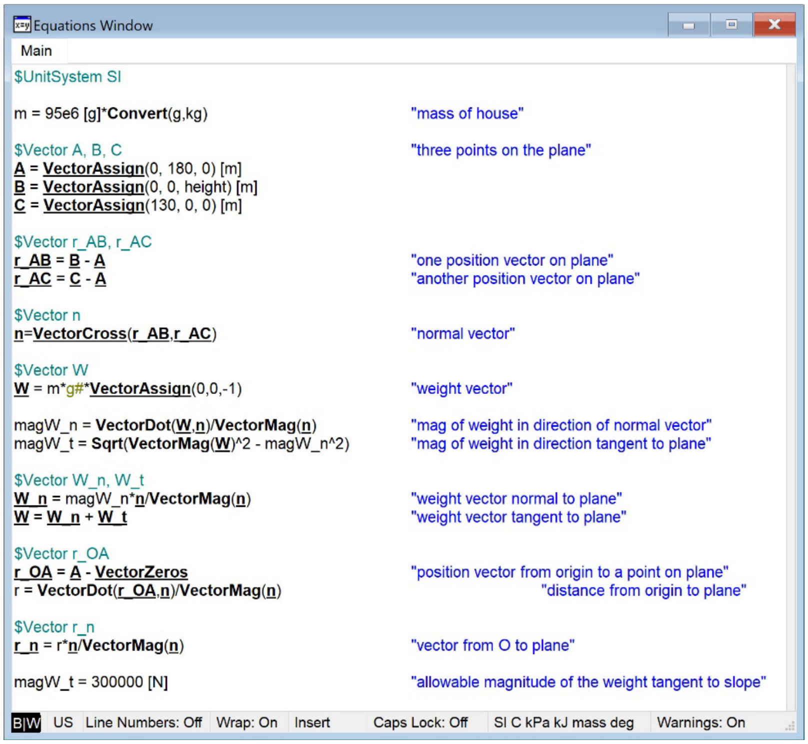
$$\vec{r}_n = r\frac{\vec{n}}{|\vec{n}|} \tag{8}$$

which is a vector of length *r* that starts at point *O* and has a direction normal to the plane.

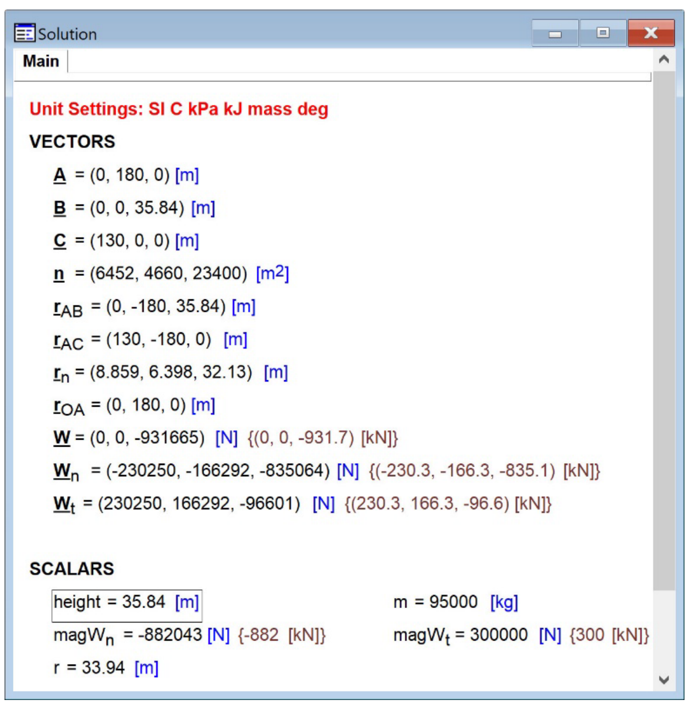
```
$Vector r_n
r_n = r*n/vectormag(n)                    "vector from O to plane"
```

Solving provides $\vec{r}_n = (20.81\hat{i} + 15.1\,\hat{j} + 45.31\hat{k})\,\text{m}$, which is inside the slope given in the problem.

Because we have gone to the trouble to develop a computer program to solve the problem, we can think about some interesting questions to ask. For example, how steep can our slope be before the house is in danger of due to slope failure? If we had some maximum allowable weight in the direction parallel to the slope, say $W_t = 300$ kN, then what elevation could we tolerate at the origin (i.e., what is the maximum allowable *z*-coordinate of point *B*)? As in our previous examples, it is not necessary to start the problem over; rather we can manipulate our computer program to quickly solve this design-type problem. Here we will make the *z*-coordinate of point *B* a variable (height) and specify the value of W_t. Note that by doing this we are asking EES to solve a coupled set of nonlinear algebraic equations which it does numerically using an iterative technique that starts from a guessed solution. Because we have a reasonable solution at this point, it is a good idea to update the guess values used by EES to the current values of all of the variables. This is done by selecting Update Guesses from the Calculate menu. The result is shown in Figure 2 and indicates that point *B* cannot be higher than 35.84 m.



(a)



(b)

Figure 2. (a) Equations Window with the z-coordinate of point B the variable height and W_t set to 300 kN. (b) Solutions Window showing that height is equal to 35.84 m.

2.2E Plotting Vectors in EES

Vectors in EES can be used to make vector plots which help visualize vectors in either 2D or 3D. Vector plots are more complicated than other types of plots due to the need to specify not only the vector but also its origin.

2D Vector Plots

In order to develop a vector plot, it is first necessary to have defined some vector variables in EES. For example, the code below generates two, 2D vectors ($\vec{A}$ and $\vec{B}$) and then adds them together to create the resultant vector ($\vec{R}$).

```
$UnitSystem Degree

$Vector2D A, B, R
A = VectorAssignPolar(1 [m],35 [degree])
B = VectorAssignPolar(1.5 [m],110 [degree])
R = A + B
```

Select New Plot Window from the Plots menu and then select Vector Plots to bring up the Vector Plot Dialog shown in Figure 2.12.

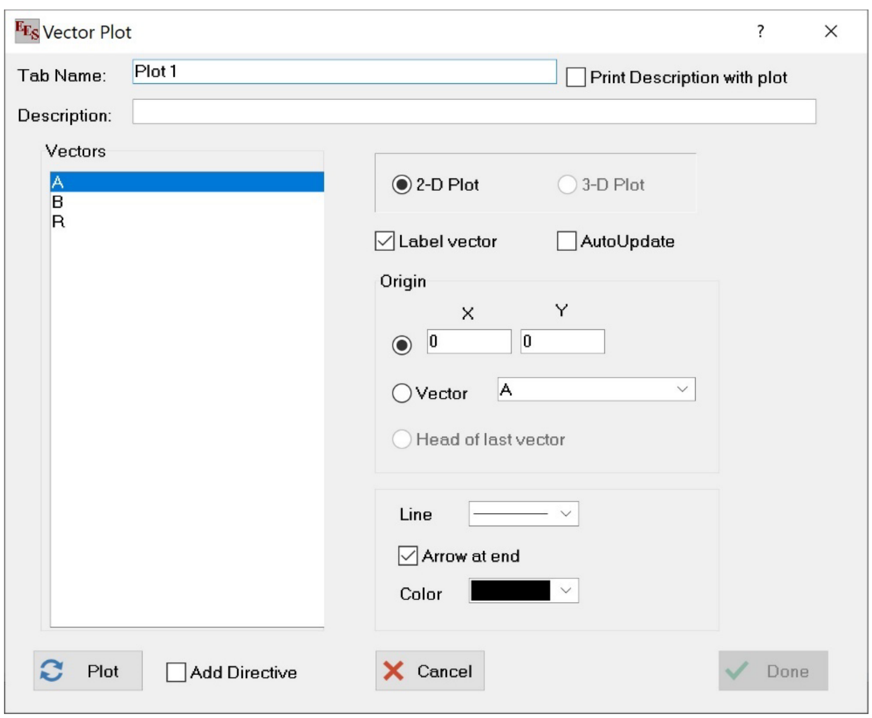


Figure 2.12. Vector Plot Dialog.

When the 2D plot radio button is selected, the box on the left side of the dialog will contain a list of all of the 2D vectors in the EES code. You can select them (one at a time) and add them to your vector plot by hitting the Plot button. When you are done adding vectors select the Done button.

For each vector you can specify its origin in one of three ways: by entering the coordinates, by specifying a vector, or by specifying that it be placed at the head of the last plotted vector (for all but the first vector). This last option is useful for developing force polygon type plots. The format of the vector can be adjusted by changing the controls located at the bottom of the Vector Plot Dialog. Figure 2.13 shows a 2D Vector Plot containing the vectors $\vec{A}$ and $\vec{B}$ where $\vec{A}$ is used as the origin of $\vec{B}$. The resultant vector $\vec{R}$ is also shown.

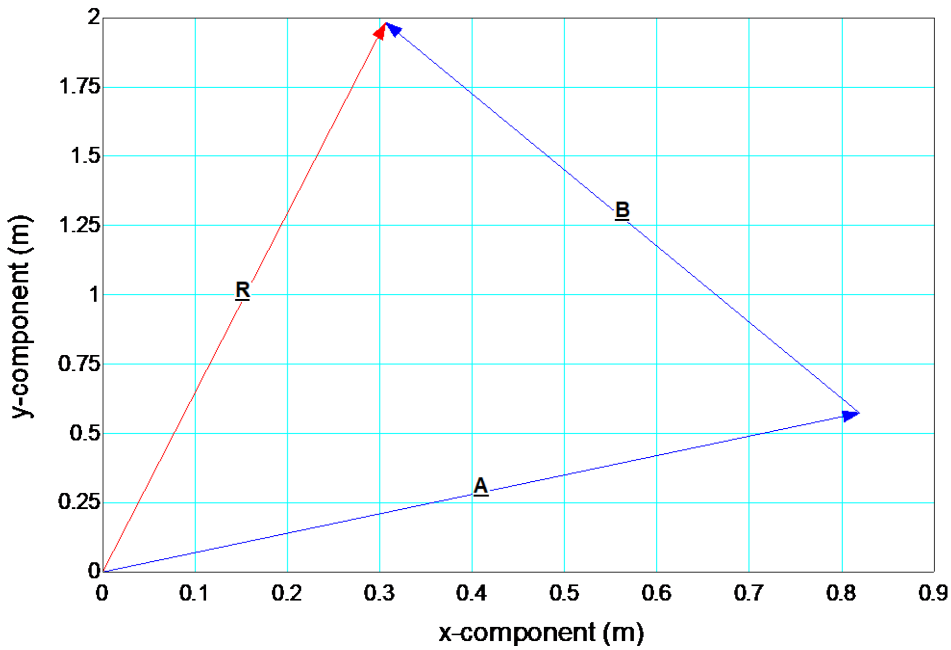


Figure 2.13: 2D Vector Plot showing that $\vec{R} = \vec{A} + \vec{B}$.

3D Vector Plots

The 3D Vector Plots work in a manner that is similar to the 2D Vector Plots except that a 3D plot is developed. The code below defines 3D vectors $\vec{A}$ and $\vec{B}$ and their resultant $\vec{R}$.

```
$Vector A, B, R

A = VectorAssign(1, 2, 2) [m]
B = VectorAssign(0.5, 0.5, -0.5) [m]
R = A + B
```

Select New Plot Window from the Plots menu and then Vector Plots to access the Vector Plot Dialog. Select the 3D Plot radio button to obtain a list of 3D vectors. Figure 2.14 shows a 3D Vector Plot containing the vectors $\vec{A}$ and $\vec{B}$ where $\vec{A}$ is used as the origin of $\vec{B}$. The resultant vector $\vec{R}$ is also shown. It is possible to spin the 3D plot by clicking on the plot and holding down the left mouse button. It is often easiest to visualize the vectors by setting the *x*-, *y*-, and *z*-grid positions to be 0 as was done in Figure 2.14.

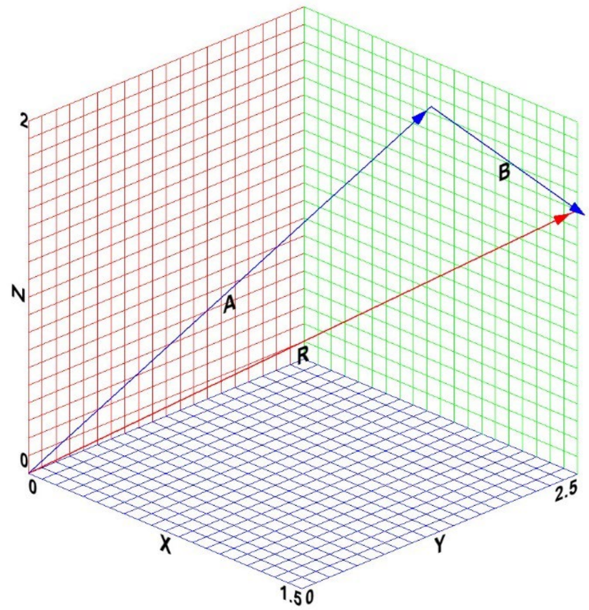


Figure 2.14: 3D Vector Plot showing that $\vec{R} = \vec{A} + \vec{B}$.

EXAMPLE E2.4 *Holding up a Pole with 3 Cables*

A 5 m tall pole is to be held up by three cables as shown in Figure 1. Two of these cables are already attached and their forces have been set, $F_1 = 2$ kN and $F_2 = 1.5$ kN. The ground connection points for cables 1 and 2 are $A = (-1, 3, 0)$ m and $B = (-2, -3, 0)$ m, respectively.

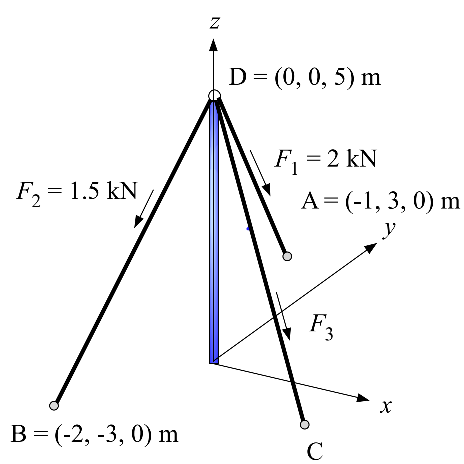


Figure 1

(a) Determine the coordinate of the ground attachment point for the cable, point C $(x, y, 0)$, and force (F_3) that is required for the third cable such that the resultant force of the three cables has a magnitude of 5 kN and is directed along the axis of the pole.

(b) Develop a 3D vector plot that shows the three forces ($\vec{F}_1$, $\vec{F}_2$, and $\vec{F}_3$) and the resultant force.

(c) Determine the component of the resultant force that is perpendicular to the pole in the event that cable 1 snaps. Assume the force exerted by cable 3 remains the same as what was calculated in (a).

SOLUTION

Road Map We will proceed by first determining the position vectors that define cables 1 and 2 and, from them, the associated force vectors. The resultant of the three cables is known and therefore we can determine the required force vector associated with cable 3. The force vector provides the required direction for cable 3 which will allow us to determine the coordinates of the necessary connection point, point C.

Part (a)

Governing Equations and Computation We will declare vector variables that correspond to the four points in Figure 1 and use these to define the coordinates of the three known points.

```
$Vector A, B, C, D
A = VectorAssign(-1,3,0) [m]          "coordinates of connection pt for cable 1"
B = VectorAssign(-2,-3,0) [m]         "coordinates of connection pt for cable 2"
D = VectorAssign(0,0,5) [m]           "coordinates of top of pole"
```

Next we will define position vectors corresponding to each of the three cables and use the known coordinates of A, B, and D to define position vectors for cables 1 and 2, $\vec{r}_{DA}$ and $\vec{r}_{DB}$, respectively.

```
$Vector r_DA, r_DB, r_DC
r_DA = A - D                          "position vector corresponding to cable 1"
r_DB = B - D                          "position vector corresponding to cable 2"
```

Force vectors are defined for each of the three cables and the forces for cables 1 and 2 are computed by multiplying the known magnitudes of these forces with unit vectors corresponding to their directions.

$$\vec{F}_1 = F_1 \frac{\vec{r}_{DA}}{\left|\vec{r}_{DA}\right|}, \text{ and} \tag{1}$$

$$\vec{F}_2 = F_2 \frac{\vec{r}_{DB}}{\left|\vec{r}_{DB}\right|}. \tag{2}$$

```
$Vector F_1, F_2, F_3
F_1 = 2 [kN]*r_DA/VectorMag(r_DA)             "force vector for cable 1"
F_2 = 1.5 [kN]*r_DB/VectorMag(r_DB)           "force vector for cable 2"
```

A vector is defined for the resultant and assigned based on the requirement that the resultant vector have a magnitude of 5 kN and be directed along the axis of the pole. The sum of the three vectors defines the resultant, allowing us to solve for $\vec{F}_3$

$$\vec{F}_1 + \vec{F}_2 + \vec{F}_3 = \vec{R}. \tag{3}$$

```
$Vector R
R = VectorAssign(0,0,-5) [kN]          "required resultant vector"
R = F_1 + F_2 + F_3                    "sum of forces"
magF_3 = VectorMag(F_3)                "magnitude of force required in cable 3"
```

Solving shows that the force on cable 3 is $\vec{F}_3$ = (0.8247, -0.2842, -2.093) kN and the magnitude of force 3 is F_3 = 2.268 kN.

The direction of cable 3 is defined by $\vec{F}_3$ and therefore we may write that the ground connection point for cable 3 (point C) is given by

$$C = D + L_3 \frac{\vec{F}_3}{\left|\vec{F}_3\right|} \tag{4}$$

where L_3 is the length of cable 3 which starts at point D and is in the direction defined by $\vec{F}_3$. Equation (4) represents three equations, one for each of the three components:

$$C_x = D_x + L_3 \frac{F_{3,x}}{\left|\vec{F}_3\right|}, \tag{5}$$

$$C_y = D_y + L_3 \frac{F_{3,y}}{\left|\vec{F}_3\right|}, \tag{6}$$

$$C_z = D_z + L_3 \frac{F_{3,z}}{\left|\vec{F}_3\right|}, \tag{7}$$

Equations (5) through (7), or equivalently Eq. (4), are three equations in the four unknowns C_x, C_y, C_z, and L_3. However, the z-coordinate of point C must be zero

$$C_z = 0. \tag{8}$$

Equations (4) and (8) are entered in EES.

```
D + L_3*F_3/VectorMag(F_3)=C           "ground connection point"
C_z = 0 [m]                            "z-coordinate of ground"
```

Solving leads to $C = (1.970, -0.6789, 0)$ m.

Part (b)

Governing Equations and Computation To solve part (b) we will make a 3D vector plot of the four force vectors involved in the problem. Select New Plot Window from the Plots menu and then select Vector Plot to bring up the Vector Plot Dialog shown in Figure 2. Select the 3D Plot radio button and then one by one plot the vectors **F_1**, **F_2**, **F_3**, and **R** indicating that each one should start at point defined by vector **D**. The resulting vector plot is shown in Figure 3.

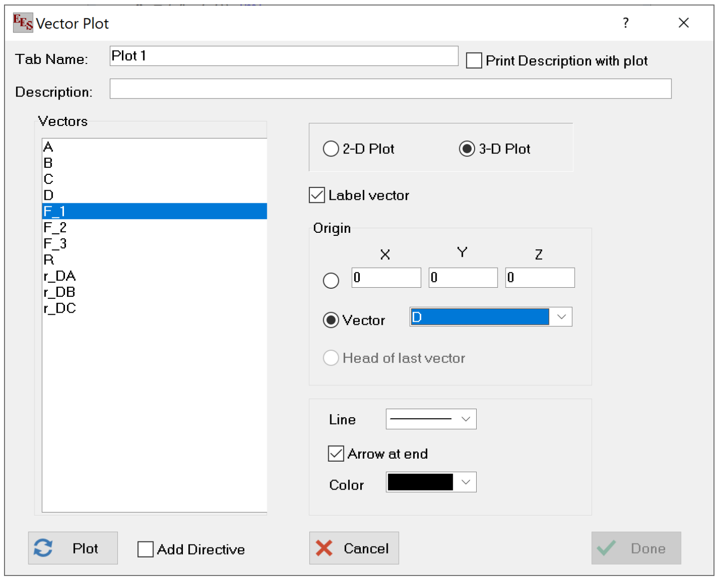


Figure 2. Vector Plot Dialog.

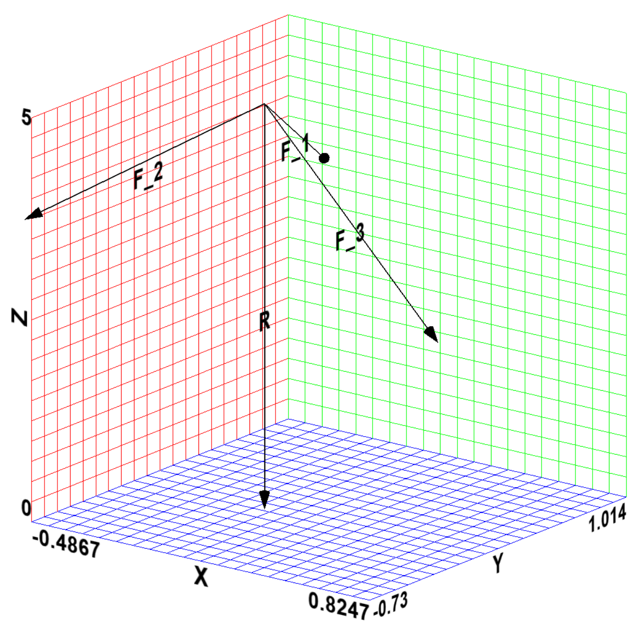
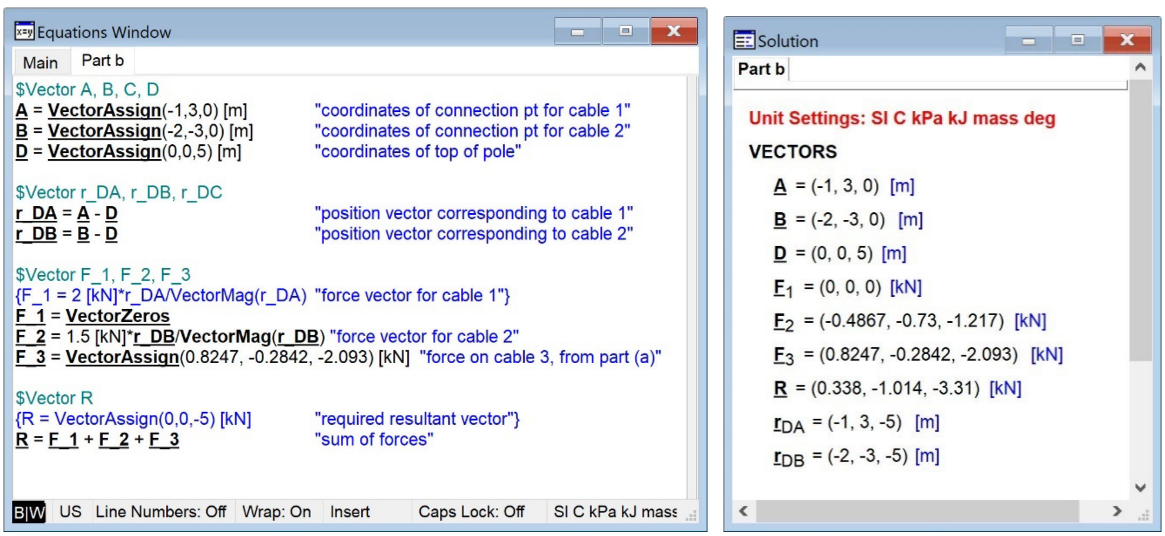


Figure 3. Vector plot showing the forces from the cables and the resultant force.

Part (c)

Governing Equations and Computation To solve part (c) we will modify our solution from part (a). We can either modify the existing EES code or start a new program. If you are using a Professional License of EES you can simply define a new Equations Tab (right click on the tab and select Add Tab).

In the solution for part (c) we need to make the following changes. First, set the magnitude of the force on cable 1 to zero, corresponding to it having snapped. Second, assign the force vector for cable 3 to the value calculated in part (a), $\vec{F}_3$ = (0.8247, -0.2842, -2.093) kN. Third, remove the assignment for the resultant force $\vec{R}$, as it will no longer be 3 kN directed along the axis of the pole after cable 1 snaps. The resulting EES code and solution is shown in Figure 4. Notice that the resultant force $\vec{R}$ now has significant components in the *x*- and *y*-directions which would tend to cause the pole to snap.



(a) (b)

Figure 4. (a) Equations Window with modifications made for the case where cable 1 has snapped. (b) Solution Window showing that the resultant force now has significant *x*- and *y*-components.

In order to obtain the magnitude of the component of $\vec{R}$ parallel to the pole we will take the dot product of $\vec{R}$ with a unit vector in the pole direction, $\vec{u}$:

$$\vec{u} = -\hat{k}, \tag{9}$$

$$R_{par} = \vec{R} \cdot \vec{u}. \tag{10}$$

```
$Vector u
u = -VectorUnit_k                  "unit vector in the pole direction"
magR_par=VectorDot(R,u)            "magnitude of component of R in z"
```

The component of $\vec{R}$ parallel to the pole is then

$$\vec{R}_{par} = R_{par}\,\vec{u}. \tag{10}$$

This is, not surprisingly, equivalent to the z-component of the resultant force. The component of $\vec{R}$ perpendicular to the pole is obtained from

$$\vec{R} = \vec{R}_{par} + \vec{R}_{perp}. \tag{11}$$

This is, again not surprisingly, simply the resultant of the x- and y-components of $\vec{R}$.

```
$Vector R_par, R_perp
R_par = magR_Par*u                      "component of R in pole direction"
R = R_par + R_perp                      "component of R perpendicular to z"
magR_perp = VectorMag(R_perp)           "magnitude of component of R perpendicular to z"
```

The result is that $\left|\vec{R}_{perp}\right| = 1.069$ kN.