Moment of a Force and Equivalent Force Systems

4.1E One Dimensional Optimization in EES

To this point we have talked about using EES to solve problems that involve fixed inputs and result in a single solution. In Section 3.1E, the use of Parametric Tables was presented as a method for carrying out a parametric study where we could vary the value of an independent variable to see its effect on one or more dependent variables. Optimization is the process where one or more variables are adjusted in order to minimize or maximize an objective function. In engineering, the objective function can be the cost, weight, efficiency, or some other important characteristic of a system or component. Optimization is often the reason why engineers develop a model of a physical system as it allows the identification of an optimal design or operating condition. This section discusses the powerful algorithms that are available in EES for accomplishing single-variable optimization.

The first step in an optimization process is to develop a model of the component, system, or process to be optimized. The model will include some inputs and predict some outputs, one of which is the objective function. To illustrate the process, consider the objective function

$$f = \sin(x)\left[2 + \cos^2(x - 10°)\right], \tag{4.1}$$

which can easily be entered in EES.

```
$UnitSystem degree
f = Sin(x)*(2+Cos(x-10 [degree])^2)
```

We are interested in finding the maximum value of f within the range $0° < x < 180°$. This can be done manually by generating a parametric table that includes the variables `x` and `f` and making a plot, as described in Section 3.1E and shown in Figure 4.1. Examination of Figure 4.1 shows that f is maximized at approximately $x = 70°$.

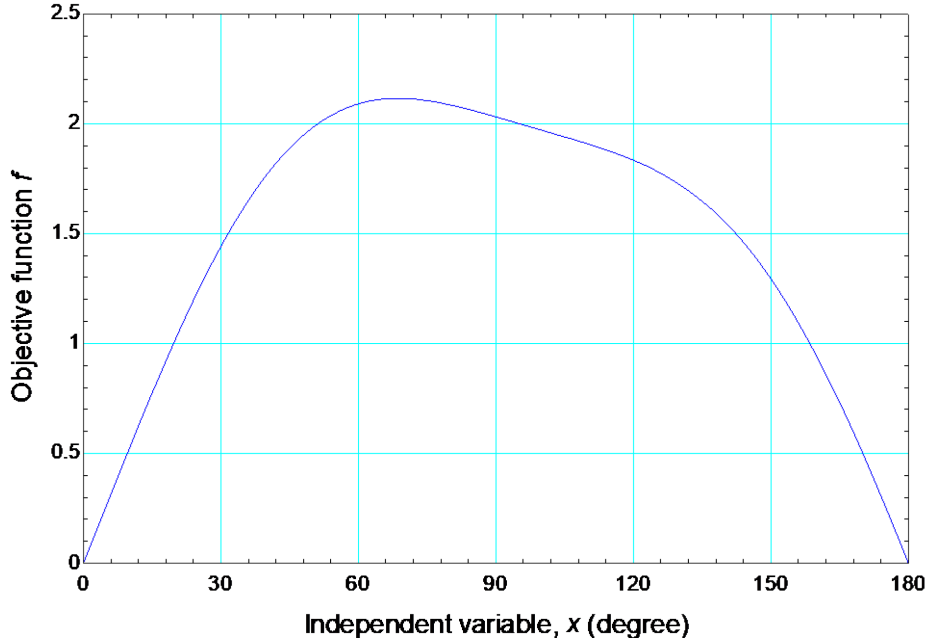


Figure 4.1. The objective function f as a function of x from $0° < x < 180°$.

Degrees of Freedom

The optimal value of x can be determined using the built-in optimization algorithms in EES. In order to carry out optimization, EES requires at least one free parameter that can be varied. If the value of x were set in the Equations Window

```
$UnitSystem degree
x=20 [degree]
f = Sin(x)*(2+Cos(x-10 [degree])^2)
```

then the problem is completely specified. Therefore, if you select Min/Max from the Calculate menu to initiate the optimization, the message shown in Figure 4.2 will be displayed.

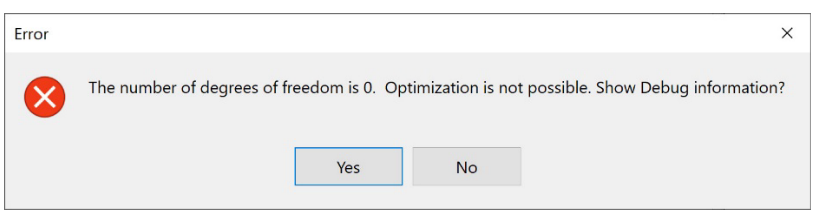


Figure 4.2. Message indicating that the number of degrees of freedom is zero.

It is necessary to comment out the equations that specify value(s) of the independent variables that will be adjusted to proceed with the optimization. In this case, the equation that specifies x is commented out.

```
$UnitSystem degree
{x=20 [degree]}
f = Sin(x)*(2+Cos(x-10 [degree])^2)
```

This leads to a problem that has one degree of freedom (i.e., there is one more unknown variable than there are equations) and therefore one-dimensional optimization is possible.

Find Minimum or Maximum Dialog

In order to use EES' optimization algorithms, select Min/Max from the Calculate menu to access the Find Minimum or Maximum dialog shown in Figure 4.3.

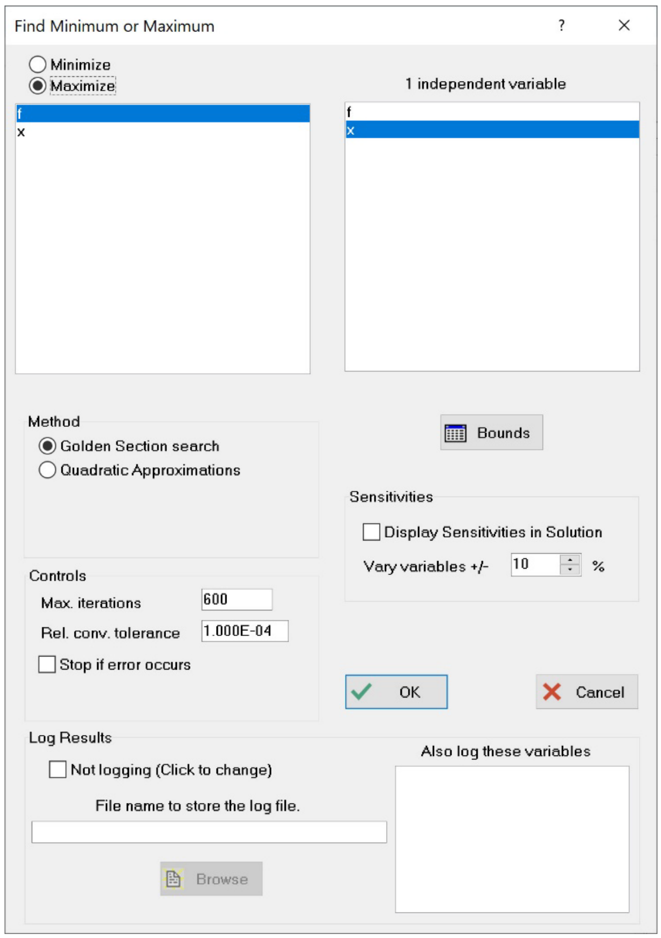


Figure 4.3. Find Minimum or Maximum Dialog.

The top left portion of the dialog requires that you select the optimization target (i.e., the objective function) from the list of variables and specify whether the optimization target is to be minimized or maximized. In this case we will maximize the value of the variable f, as shown. The top right window allows you to select the independent (i.e., the optimization) variable(s). EES determines the number of free parameters associated with the equations and requires that this number of independent parameters be selected. For this problem (with the variable x commented out) there is only one free parameter.

Stopping Criteria

The Controls box in the Find Minimum or Maximum dialog shown in Figure 4.3 allows the termination criteria for the optimization process to be specified. The process will terminate when either the maximum number of calculations is reached or the relative convergence tolerance is achieved. The relative convergence tolerance refers to the change in the optimization target that occurs between successive iterations normalized by its value. If the Stop if error occurs box is unchecked then EES will not terminate the optimization process if it attempts to solve the equations for a value of the optimization variable that results in an error. Rather, it will consider this value of the independent variable to be non-viable (and therefore not optimal) and move on to other values.

Guess Value and Bounds

EES requires that you set both upper and lower bounds and the guess value for each of the independent variables. To set these bounds, select Bounds from the Find Minimum or Maximum dialog which will bring up the Variable Information window showing the independent variable(s). For this problem, there is only one independent variable, x, and so the dialog appears as shown in Figure 4.4. For this problem we are interested in identifying the maximum value of *f* that occurs in the range 0° < *x* < 180° which sets the bounds. The guess value is set to an arbitrary number within this range.

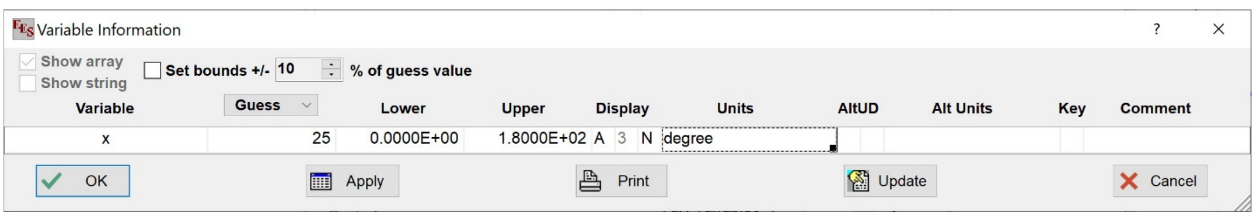


Figure 4.4. Variable Information Window for independent variable where guess values and bounds are entered for the optimization.

Select OK in order to return to the Find Minimum or Maximum dialog. All that remains is to select the optimization method. In this problem there is only one free parameter, therefore a one-dimensional optimization will be accomplished using either of the two methods shown: Golden Section search or Quadratic approximations. The details of the implementation of the optimization techniques are programmed in EES. However, it is useful to have a high-level understanding of how the techniques work so that you can select the appropriate one for your problem.

The Golden Section Search

The Golden Section search operates by successively narrowing the range in which the extremum (minimum or maximum value) is known to exist. The process starts by evaluating the objective function (in this case *f*) at the lower and upper limits of the independent variable (i.e., at $x = 0°$ and $x = 180°$). The objective function is found to be near zero at these two points, which are labeled points 1 and 2 in Figure 4.5.

Next, two test points are located within the range. The first test point, identified as point 3 in Figure 4.5, is located a distance of $g\ (x_2 - x_1)$ from the left edge of the range, where g is the Golden Ratio equal to 0.6182. The range of x is initially from 0° to 180° and therefore point 3 is located at x_3 = 111.3°. The second test point is located at a value of x that is $g\ (x_2 - x_1)$ from the right edge of the range, which corresponds to x_4 = 68.7°. The objective function (f) is evaluated at points 3 and 4 and it can be seen that point 4 exhibits a larger value of f than point 3. The result is that the range of x containing the maximum value of f can be narrowed by eliminating all points to the right of point 3. The process is repeated by locating two points within the reduced range between point 1 and point 3. However, point 4 is already located a distance of $g\ (x_3 - x_1)$ from the left edge of the new range. This, in fact, is how the value of g was determined. Therefore, it is not necessary to reevaluate f at this point. Point 5 is located a distance of $g\ (x_3 - x_1)$ from the right edge of the new range (x_5 = 42.5°) and the value of f is evaluated at this point. Since f is higher at point 4 then at point 5, all values to the left of point 5 are eliminated and the reduced range is now between points 3 and 5. This process continues until the stopping criteria selected for the optimization is achieved.

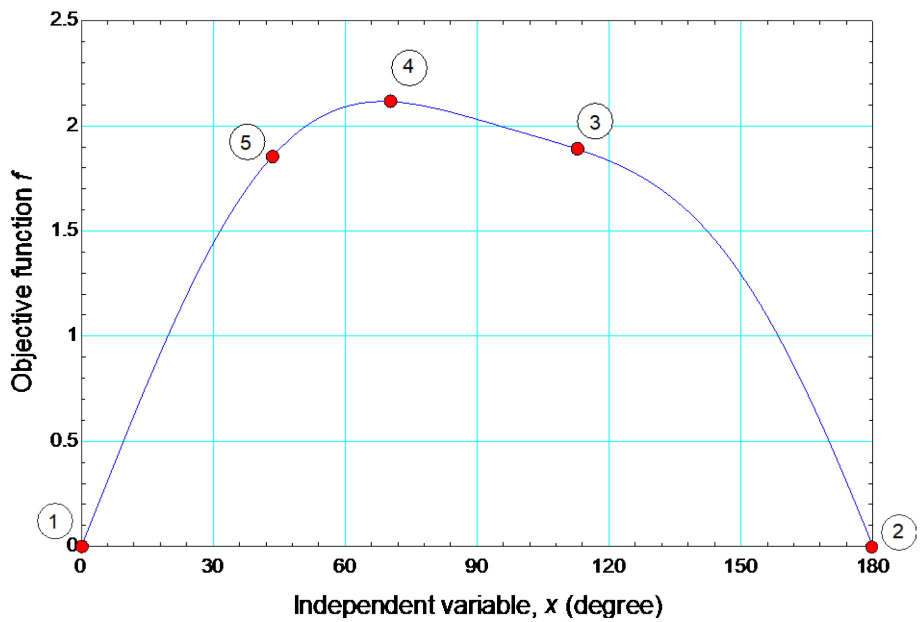


Figure 4.5. Progression of the Golden Section search method for the example optimization problem.

Select the Golden Section search and then hit OK to initiate the optimization. The result is shown in Figure 4.6. The optimal value of x has been identified to be 68.6° resulting in an objective function value of f = 2.115. The optimization required 25 iterations to find this result.

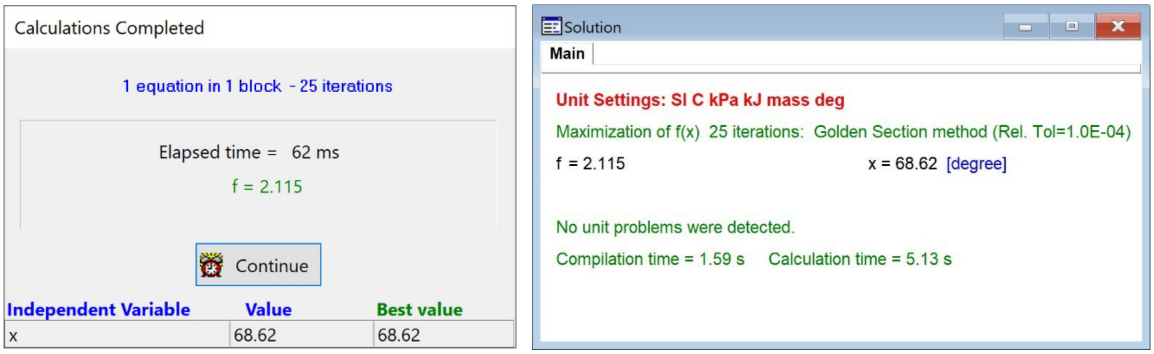


Figure 4.6. Result of the Golden Section search indicates the optimized value of the independent variable and the associated value(s) of the dependent variable as well as the number of iterations required.

The Quadratic Approximations Optimization Method

An alternative method for one-dimensional optimization is Quadratic Approximations. Quadratic Approximations proceeds three points at a time. For the first iteration, these three points correspond to the bounds of the problem (points 1 and 2 in Figure 4.7) and one point within the range (which is the guess value that was set, point 3).

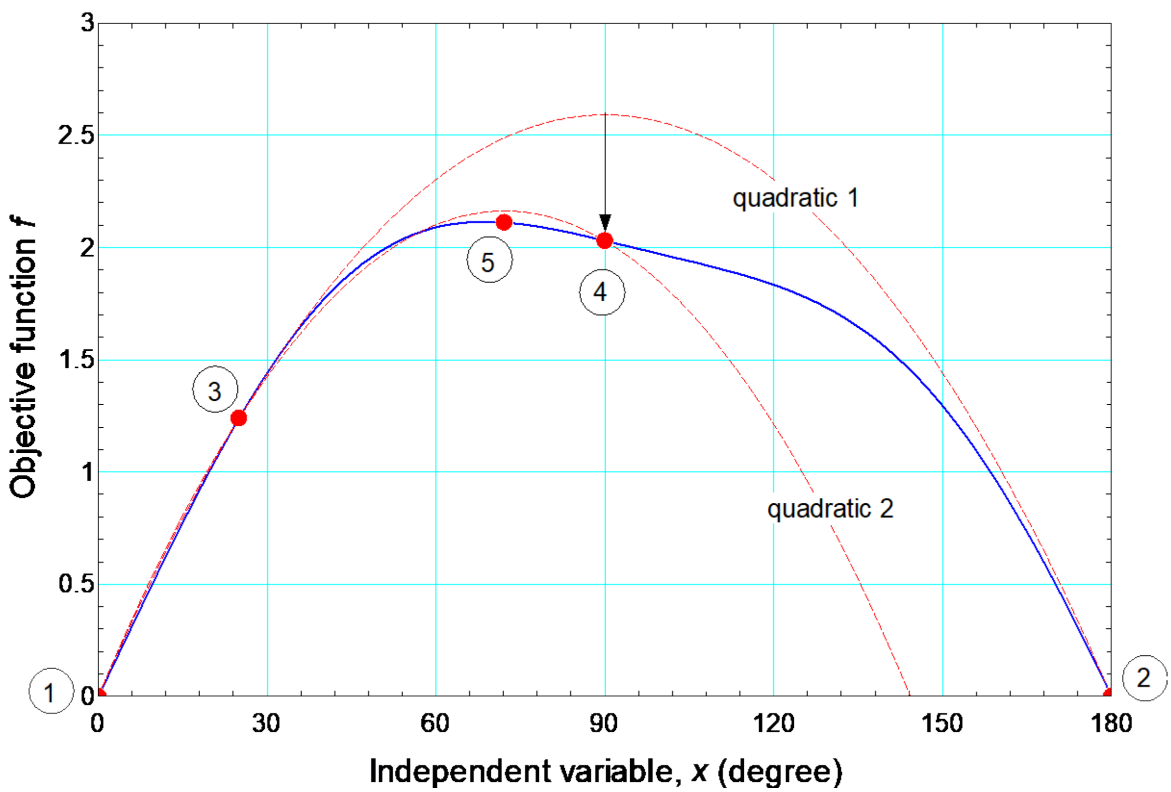


Figure 4.7. Progression of the Quadratic Approximations optimization method.

The quadratic approximation technique assumes that the objective function (f) depends on the optimization variable (x) in a quadratic manner

$$f = a\,x^2 + b\,x + c. \tag{4.2}$$

The coefficients a, b, and c in Eq. (4.2) are selected in order to fit the three points. The resulting quadratic function that passes through points 1, 2, and 3 is shown as quadratic 1 in Figure 4.7. The optimal value of the optimization variable is predicted using the quadratic equation by setting the derivative to zero. Taking the derivative of Eq. (4.2) provides

$$2\,a\,x_{opt} + b = 0 \tag{4.3}$$

which can be solved for x_{opt}:

$$x_{opt} = -\frac{b}{2\,a}. \tag{4.4}$$

The value of the objective function at x_{opt} is computed, leading to point 4 in Figure 4.7. The point with the smallest value of the objective function (for maximization) is discarded and the process is carried out again using the remaining three points (in this case points 1, 3, and 4). The new quadratic equation is labeled quadratic 2 in Figure 4.7 and leads to the identification of a new optimal point 5. This process continues until the stopping criteria set for the optimization is achieved.

For most functions, the Quadratic Approximations method will converge to the optimal solution more quickly than the Golden Section method. If Quadratic Approximations is selected from the Find Minimum or Maximum Dialog then the result is shown in Figure 4.8. Notice that the same result as the Golden Section search (shown in Figure 4.6) was achieved, but the process required 18 iterations and about 0.5 s of compilation/calculation time, compared to 25 iterations and almost 7 s for the Golden Section method.

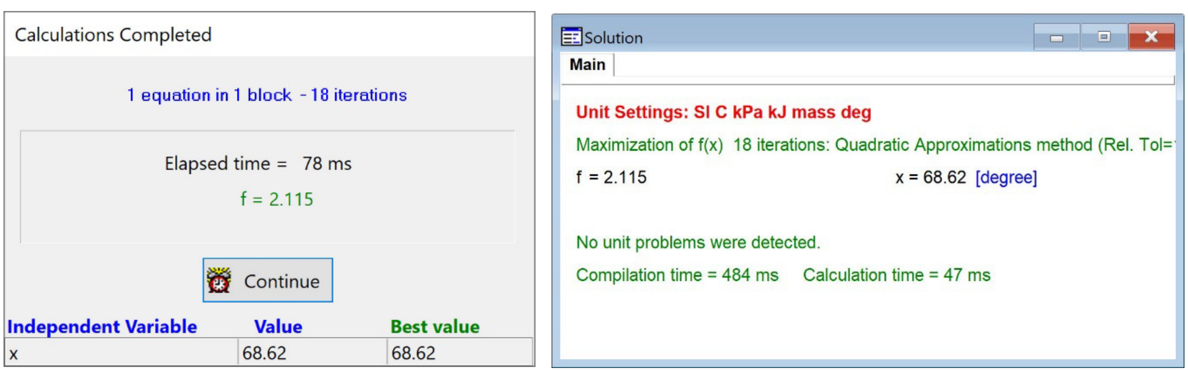


Figure 4.8. Result of the Quadratic Approximations search indicates the optimized value of the independent variable and the associated value(s) of the dependent variable and shows that fewer iterations were required compared to the Golden Section Search.

EXAMPLE E4.1 *Maximizing the Moment of a Force*

We will revisit Example 4.4 from the text. The belt tensioner *ABC* is attached to an engine using a bearing at *A* having a torsional spring.

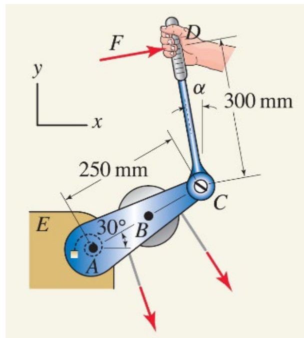


Figure 1. Problem from Example 4.4 in Gray et al. (2023).

To release the tension, a ratchet wrench *CD* is applied to the tensioner at point *C*. If a moment about point *A* of 50 N-m is required to release the belt tension, determine the smallest force *F* required and the angle α at which the wrench should be positioned. Consider the following two cases:

(a) The force *F* is always perpendicular to the handle of the wrench.

(b) The force *F* is always horizontal (parallel to the *x* axis).

SOLUTION

Road Map The textbook solves Example 4.4 using a scalar approach. Here we will use a vector approach. Also, the textbook solution graphically determines the wrench angle that would maximize the moment whereas we will set the problem up as an optimization problem. That is, we will setup the model using a reasonable value of α and then employ the 1-D optimization algorithms in EES to determine the value of α that maximizes the moment.

Part (a)

Governing Equations The position vector $\vec{r}_{AC}$ is a two-dimensional vector corresponding to the tensioner that is defined by its angle, θ = 30° (defined relative to the *x*-axis) and length, L_t = 250 mm. The position vector $\vec{r}_{CD}$ corresponds to the wrench and is defined by its angle, α (defined relative to the *y*-axis), and length L_w = 300 mm. The position vector $\vec{r}_{AD}$ goes from the bearing (at *A*) to the point where the force is applied to the wrench (at *D*)

$$\vec{r}_{AD} = \vec{r}_{AC} + \vec{r}_{CD}. \tag{1}$$

The direction of the force vector, $\vec{F}$, is perpendicular to the wrench and therefore is defined by the angle α. The moment that the force vector applies about the bearing is in the *z*-direction and can be obtained from:

$$\vec{M} = \vec{r}_{AD} \times \vec{F}. \tag{2}$$

The magnitude of the force vector must be such that the moment is the required 50 N-m in the clockwise direction (i.e., -50 N-m).

Computation Because we will be using trigonometric functions, the $UnitSystem directive is used to specify that the units of angles must be degree. The inputs include the tensioner angle and length as well as the wrench angle and length.

```
$UnitSystem degree

theta = 30 [deg]                          "angle of the tensioner (from horizontal)"
L_t = 250 [mm]*Convert(mm,m)              "length of the tensioner"
L_w = 300 [mm]*Convert(mm,m)              "length of the wrench"
alpha = 20 [deg]                          "angle of the wrench (from vertical)"
```

The position vectors $\vec{r}_{AC}$, $\vec{r}_{CD}$, and $\vec{r}_{AD}$ are declared, The position vectors $\vec{r}_{AC}$ and $\vec{r}_{CD}$ are assigned using the **VectorAssignPolar** command with the appropriate length and angle while the position vector $\vec{r}_{AD}$ is defined using Eq. (1).

```
$Vector2D r_AC, r_CD, r_AD
r_AC = VectorAssignPolar(L_t,theta)            "position vector for tensioner"
r_CD = VectorAssignPolar(L_w,90 [deg] + alpha) "position vector for wrench"
r_AD = r_AC + r_CD                             "position vector from bearing to end of wrench"
```

The force vector $\vec{F}$ is defined using the **VectorAssignPolar** command and the moment is determined from Eq. (2) with the **VectorCross** command. Because both $\vec{r}_{AD}$ and $\vec{F}$ are 2-D vectors the result of the **VectorCross** command is a scalar that is the magnitude of the moment. The moment must be set to the required value of -50 N-m, which completes the equation set.

```
$Vector2D F
F = VectorAssignPolar(magF, alpha)        "force"
M = VectorCross(r_AD,F)                   "moment"
M = -50 [N-m]                             "required moment"
```

The equation set can be checked by solving the system of equations which leads to F = 145.6 N for $\alpha = 20°$.

The question asks for the value of α that minimizes F which can be determined using EES' 1-D optimization algorithm. First, we will comment out the specified value of α.

```
{alpha = 20 [deg]          "angle of the wrench (from vertical)"}
```

Select Min/Max from the Calculate menu to access the Find Minimum or Maximum Dialog, shown in Figure 2. The objective function is the magnitude of the force, magF, and the independent parameter is the angle, alpha. The bounds used for the optimization were set to be from -180° < α < 180°.

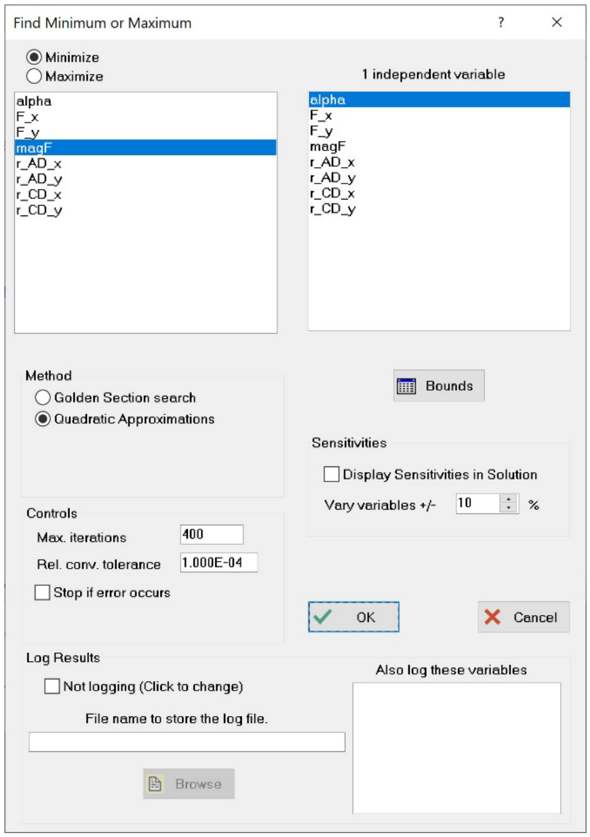


Figure 2. Find Minimum or Maximum Dialog setup to minimize the force by varying the wrench angle.

The results of the optimization are shown in Figure 3 and show that the optimal value of α is -60° which leads to a minimum force of F = 90.91 N; these results match those reported in the textbook for Example 4.4.

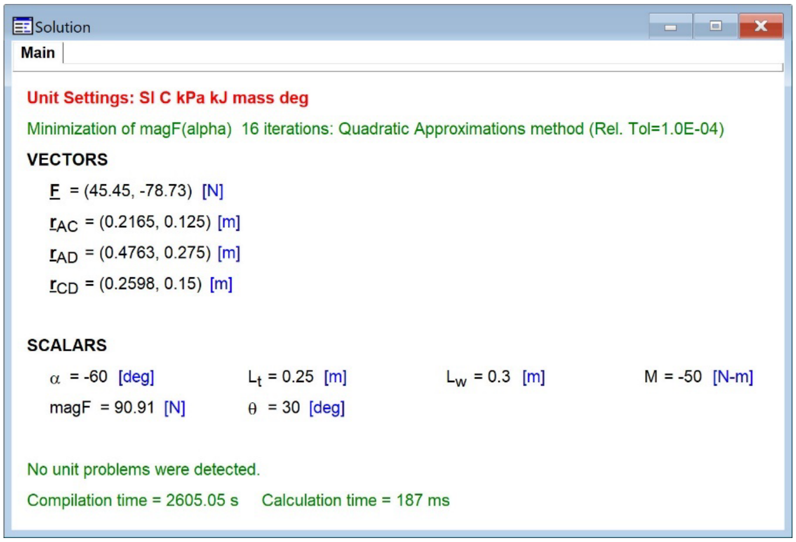


Figure 3. Results of the optimization.

Part (b)

Governing Equations The governing equations remain the same except that the direction of the force $\vec{F}$ is now aligned with the x-axis.

Computation The equation that defines the force vector is commented out. Instead, the force vector is defined using the **VectorAssign** command with the y-component set to zero.

```
{F = VectorAssignPolar(magF, alpha)          "force"}
F = VectorAssign(magF, 0)                    "force"
```

The optimization is carried out as before and the result is that the minimum value of F is 117.6 N which occurs when $\alpha = 0°$. These results are also consistent with the answer from the textbook.

Discussion & Verification We can double check our answers by creating a plot showing the value of F as a function of α for the two cases examined in parts (a) and (b). The result is shown in Figure 4 and is consistent with the answers we obtained through optimization.

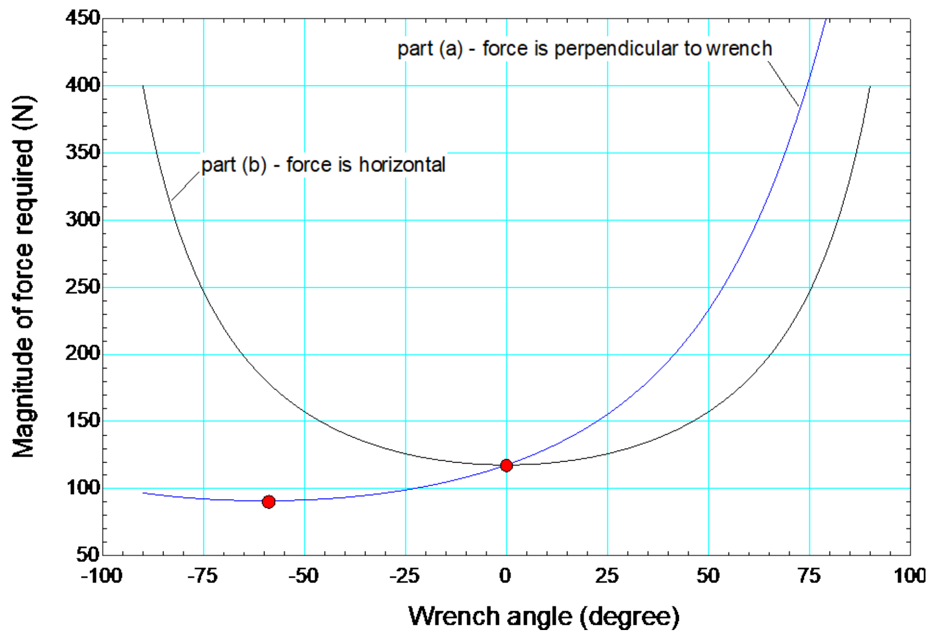


Figure 4. Required force as a function of wrench angle for the case where force is perpendicular to the wrench and horizontal.

EXAMPLE E4.2 *Calculating the Moment about a Point*

We will revisit Problem 4.24 from the text. Structure *OAB* is attached to the ground at point *O* and supports forces from two cables.

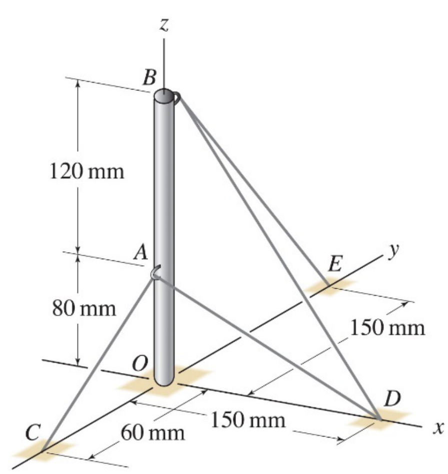


Figure 1. Problem 4.24 in Gray et al. (2023).

Cable *CAD* passes through a frictionless ring at point *A*, and cable *DBE* passes through a frictionless ring at point *B*. If the force in cable *CAD* is 250 N and the force in cable *DBE* is 100 N, use a vector approach to determine:

(a) The moment of all cable forces about point *A*.

(b) The moment of all cable forces about point *O*.

SOLUTION

Road Map The problem is a relatively simple application of Varignon's theorem. The results of the calculations are visualized using a 3-D vector plot created using the $VectorPlot directive discussed in Section 3.3E.

Part (a)

Governing Equations The geometry is specified by the points *O*, *A*, *B*, *C*, *D*, and *E* which can be used to determine the position vectors that define both the direction of the various forces as well as the location of the force application. The vector forces are determined based on the specified magnitudes and their directions. Finally, the moment of the force about point *A* can be calculated using the cross product of the position vector defining the location of the application of the force relative to point *A* and the force.

Computation The inputs include the coordinates of the points as well as the magnitude of the forces in the cables.

```
"Coordinates of points"
$Vector A, B, C, D, E, O
O=VectorAssign(0,0,0) [mm]
A = VectorAssign(0,0,80) [mm]
B = VectorAssign(0,0,200) [mm]
C = VectorAssign(0,-60,0) [mm]
D = VectorAssign(150,0,0) [mm]
E = VectorAssign(0,150,0) [mm]

"magnitudes of forces"
magF_CAD = 250 [N]
magF_DEB = 100 [N]
```

The position vectors that dictate the directions of the forces are $\vec{r}_{BE}$, $\vec{r}_{BD}$, $\vec{r}_{AD}$ and $\vec{r}_{AC}$.

```
"position vectors defining force directions"
$Vector r_BE, r_BD, r_AD, r_AC
r_BE = E - B
r_BD = D - B
r_AD = D - A
r_AC = C - A
```

The force vectors are the product of the magnitude of the force and the unit vector defined by the appropriate position vector. For example, $\vec{F}_{BE}$ is given by

$$\vec{F}_{BE} = F_{DEB} \frac{\vec{r}_{BE}}{\left|\vec{r}_{BE}\right|}. \quad (1)$$

```
"forces"
$Vector F_BE, F_BD, F_AD, F_AC
F_BE = magF_DEB*r_BE/VectorMag(r_BE)
F_BD = magF_DEB*r_BD/VectorMag(r_BD)
F_AD = magF_CAD*r_AD/VectorMag(r_AD)
F_AC = magF_CAD*r_AC/VectorMag(r_AC)
```

The moment about A associated with each force is the cross product of the position vector from A to the point of application and the force. The forces that are applied at point A ($\vec{F}_{AC}$ and $\vec{F}_{AD}$) apply no moment because the position vector is zero. Therefore, the moment about A is the sum of the moments produced by the two forces applied at point B ($\vec{F}_{BE}$ and $\vec{F}_{BD}$)

$$\vec{M}_A = \vec{r}_{AB} \times \vec{F}_{BE} + \vec{r}_{AB} \times \vec{F}_{BD}, \quad (2)$$

where

$$\vec{r}_{AB} = B - A. \quad (3)$$

```
"moment about A"
$Vector r_AB, M_A
r_AB = B - A
M_A = VectorCross(r_AB,F_BE)+VectorCross(r_AB,F_BD)
```

Solving provides $\vec{M}_A$ = (-7200, 7200, 0) N-mm.

It is helpful to use a 3-D vector plot to visualize the forces and resulting moment. We will use the $VectorPlot directive discussed in Section 3.3E for this purpose. The scale of the moment is much larger than the scale of the position vectors or forces and therefore we will reduce it by a factor of 100 so that the forces, position vectors, and moment can all be placed on the same plot.

```
$Vector M_A_scaled
M_A_scaled = M_A/100

$VectorPlot Name='Part a' r_AB:A/black F_BE:B/red F_BD:B/red  M_A_scaled:A/purple
```

The resulting 3-D vector plot is shown in Figure 2. Notice that the twisting action produced by the two forces involved corresponds to the direction of the resulting moment according to the right-hand rule.

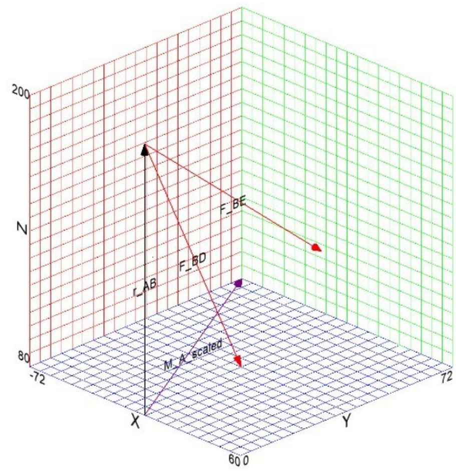


Figure 2. 3-D vector plot showing the position vector corresponding to the portion of the pole *AB* as well as the two forces $\vec{F}_{BE}$ and $\vec{F}_{BD}$ and the moment that the produce about point *A*, $\vec{M}_A$.

Part (b)

Governing Equations The approach remains the same except that the moment is determined about point *O*, therefore all four forces applied contribute to the moment.

Computation The equations from part (a) that define the coordinates, position vectors, and forces are copied into a new tab, labeled part b.

```
"Coordinates of points"
$Vector A, B, C, D, E, O
O=VectorAssign(0,0,0) [mm]
A = VectorAssign(0,0,80) [mm]
B = VectorAssign(0,0,200) [mm]
C = VectorAssign(0,-60,0) [mm]
D = VectorAssign(150,0,0) [mm]
E = VectorAssign(0,150,0) [mm]

"magnitudes of forces"
magF_CAD = 250 [N]
magF_DEB = 100 [N]

"position vectors defining force directions"
$Vector r_BE, r_BD, r_AD, r_AC
r_BE = E - B
r_BD = D - B
r_AD = D - A
r_AC = C - A

"forces"
$Vector F_BE, F_BD, F_AD, F_AC
F_BE = magF_DEB*r_BE/VectorMag(r_BE)
F_BD = magF_DEB*r_BD/VectorMag(r_BD)
F_AD = magF_CAD*r_AD/VectorMag(r_AD)
F_AC = magF_CAD*r_AC/VectorMag(r_AC)
```

The moment about point *O* is given by

$$\vec{M}_O = \vec{r}_{OB} \times \vec{F}_{BE} + \vec{r}_{OB} \times \vec{F}_{BD} + \vec{r}_{OA} \times \vec{F}_{AD} + \vec{r}_{OA} \times \vec{F}_{AC}, \tag{4}$$

where

$$\vec{r}_{OB} = B - O \tag{5}$$

and

$$\vec{r}_{OA} = A - O. \tag{6}$$

```
$Vector r_OB, r_OA, M_O
r_OB = B - O
r_OA = A - O
M_O = VectorCross(r_OB, F_BE)+VectorCross(r_OB,F_BD)+VectorCross(r_OA,F_AC)+&
VectorCross(r_OA,F_AD)
```

The calculations lead to $\vec{M}_O$= (0, 29647, 0) N-mm. A 3-D vector plot containing the pole, the four forces, and the moment (scaled to appear on the plot) is generated using the $VectorPlot directive below and shown in Figure 3.

```
$Vector M_O_scaled
M_O_scaled = M_O/100

$VectorPlot Name='Part b' r_OB:O/black F_BE:B/red F_BD:B/red  F_AD:A/red F_AC:A/red &
M_O_scaled:O/purple
```

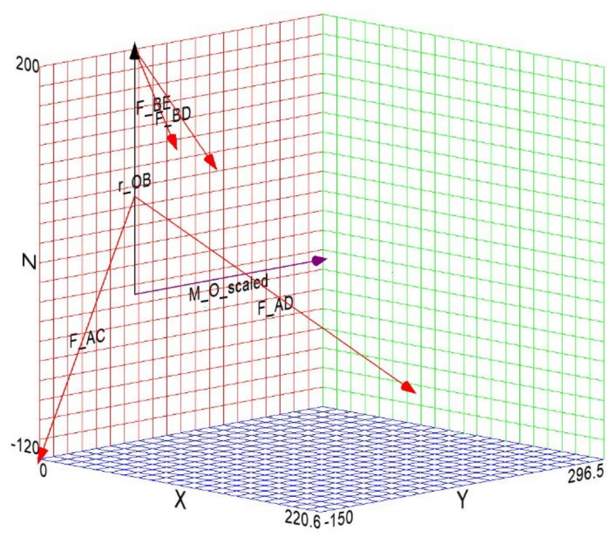


Figure 4. Required force as a function of wrench angle for the case where force is perpendicular to the wrench and horizontal.

Notice that the net result of the four forces is to create a moment about point *O* that causes the pole to twist about the *y*-axis.

EXAMPLE E4.3 *Moment of a Force About a Line – Vector Solution*

We will revisit Example 4.7 from the text. The steering linkage for the front wheel of an off-road vehicle is shown.

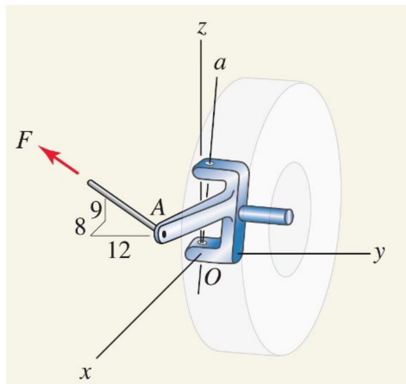


Figure 1. Example 4.7 in Gray et al. (2023).

Force F causes the assembly to rotate about line a so that the vehicle can be steered. Point A is located at (160, -20, 100) mm. Determine the force F needed to produce a moment about line a of 10 N-m if

(a) line a lies in the yz plane and has direction angle $\theta_z = 10^{\circ}$.

(b) line a coincides with the z axis.

SOLUTION

Road Map We can determine the vector moment produced about any point on a (e.g., point O) as we did in Example 4.2. The moment about line a is the projection of the resulting vector moment onto the line.

Part (a)

Governing Equations The coordinates of point A are given in the problem statement allowing the position vector $\vec{r}_{OA}$ to be determined

$$\vec{r}_{OA} = A - O. \tag{1}$$

The direction of the force vector $\vec{F}$ is given by the unit vector

$$\hat{u}_F = \frac{\left(-8\hat{\imath} - 12\hat{\jmath} + 9\hat{k}\right)}{\sqrt{(-8)^2 + (-12)^2 + (9)^2}}. \tag{2}$$

The force vector is then

$$\vec{F} = F\,\hat{u}_F, \tag{3}$$

where F is the unknown magnitude of the force. The vector moment produced by force $\vec{F}$ about point O is

$$\vec{M}_O = \vec{r}_{OA} \times \vec{F}. \tag{4}$$

A unit vector in the direction of line a is given by

$$\hat{u}_a = 0\,\hat{i} + \sin(\theta_z)\,\hat{j} + \cos(\theta_z)\,\hat{k}. \tag{5}$$

The magnitude of the projection of $\vec{M}_O$ onto line a is

$$M_a = \vec{M}_O \cdot \hat{u}_a. \tag{6}$$

The value of F must be set such that M_a = 10 N-m.

Computation The inputs include the coordinates of points O and A as well as the unit vectors that define the direction of the force ($\hat{u}_F$) and the line a ($\hat{u}_a$).

```
$UnitSystem degree

$Vector O, A, u_F, u_a
O = VectorAssign(0,0,0) [mm]
A = VectorAssign(160, -20, 100) [mm]                  "point of force application"
theta_z = 10 [degree]                                 "angle between line a and the z axis in the y-z plane"
u_a = VectorAssign(0, Sin(theta_z), Cos(theta_z))     "unit vector defining the direction of line a"
u_F = VectorAssign(-8,-12,9)/Sqrt(8^2+12^2+9^2)       "unit vector defining the direction of the force"
```

The position vector $\vec{r}_{OA}$ is computed using Eq. (1).

```
$Vector r_OA
r_OA = A - O                                          "position vector"
```

The magnitude of the force vector is not known. One method of moving forward is to enter Eqs. (3), (4), and (6) along with the specification that M_a must be 10 N-m before solving the equations. A better approach is to enter a guess for the magnitude of the force and then enter each equation sequentially, solving as we go to ensure that we have not introduced any errors. Finally, we can update our guess values, comment out the guessed value of F and require that M_a must be 10 N-m.

A guess value of F is entered.

```
magF = 10 [N]                                   "guess value for F"
```

Equation (3) is used to determine the force vector.

```
$Vector F
F = magF*u_F                                    "force vector"
```

Equation (4) is used to determine the moment about point *O*.

```
$Vector M_O
M_O = VectorCross(r_OA, F)                      "moment vector about O"
```

Equation (5) is used to determine the magnitude of moment $\vec{M}_O$ projected onto line *a*.

```
magM_A = VectorDot(M_O, u_a)                    "mag. of M_O on line a"
```

Solving shows that M_a is equal to -1434 N-mm (-1.434 N-m) for the assumed value of F = 10 N; clearly a larger force is required to reach a moment of 10 N-m. Also notice that the value of the moment is negative since a positive force $\vec{F}$ will tend to rotate the mechanism clockwise (i.e., in the negative direction) about line *a*. Next we will update our guess values (select Update Guesses from the Calculate menu), comment out the assumed value of F

```
{magF = 10 [N]                                  "guess value for F"}
```

and specify that M_a must be -10 N-m.

```
magM_A = -10 [N-m]*Convert(N-m,N-mm)      "required value of magM_A"
```

Solving provides F = 69.75 N, which agrees with the answer in the text.

Discussion & Verification It is instructive to visualize these results using a 3-D vector plot. The vector moment along line *a* is

$$\vec{M}_a = M_a \hat{u}_a. \tag{7}$$

The $VectorPlot directive below will plot the position vector $\vec{r}_{OA}$, the force $\vec{F}$, and the two moment vectors $\vec{M}_O$ and $\vec{M}_a$ (scaled so that they show up in the plot), as shown in Figure 2.

```
$Vector M_A
M_A = magM_A*u_a                                "moment projected onto line a"

$Vector M_A_scaled, M_O_scaled
M_A_scaled = M_A/100
M_O_scaled = M_O/100

$VectorPlot Name = 'Vectors' r_OA:O/black F:A/red M_A_scaled:O/purple M_O_scaled:O/green
```

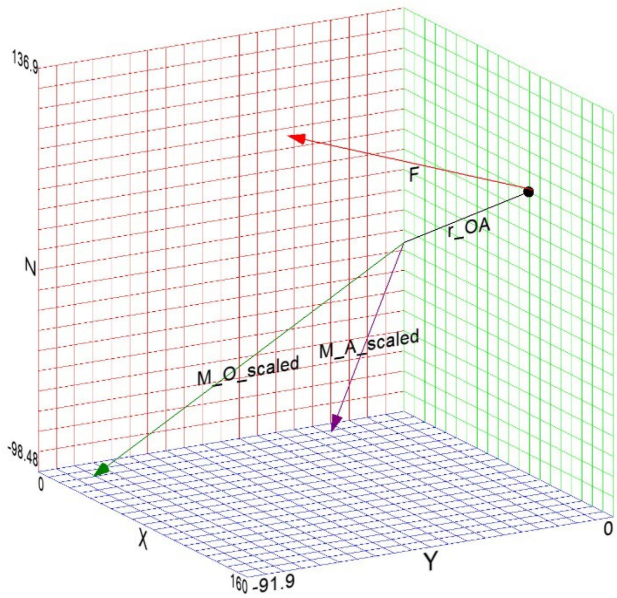


Figure 2. Vector plot showing the relevant position vector, force, and moments.

Part (b)

Governing Equations The approach remains the same except that the unit vector defining line *a* is now directed along the *z*-axis.

Computation The equations from part (a) are all the same – the only change is that the angle θ_z must be changed to 0°.

```
{theta_z = 10 [degree]                          "angle between line a and the z axis in the y-z plane"}
theta_z = 0 [degree]                            "angle between line a and the z axis in the y-z plane"
```

Solving leads to F = 81.73 N, which agrees with the text.

Discussion & Verification The vector plot is automatically refreshed when the equations are solved due to the $VectorPlot directive; the new vector plot is shown in Figure 3.

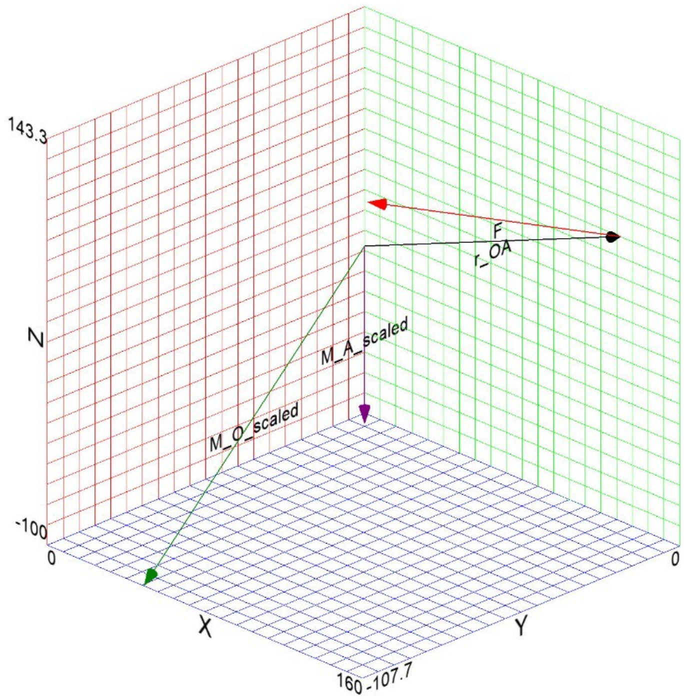
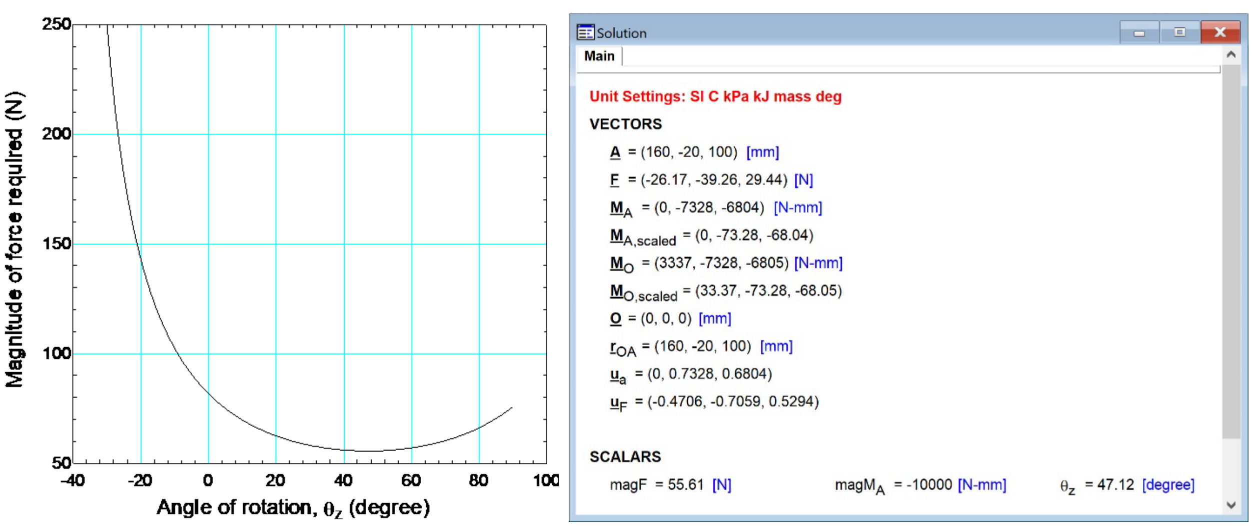


Figure 3. Vector plot showing the relevant position vector, force, and moments with $\theta_z = 0°$.

We could examine the relationship between the force required and the angle of line *a*. A Parametric Table containing the variables theta_z and magF is generated and used to develop the plot shown in Figure 4(a). We can use EES' internal optimization algorithm to minimize the value of *F* by varying θ_z; the result is shown in Figure 4(b).



(a) (b)

Figure 4. (a) Magnitude of force required as a function of the rotation angle, θ_z and (b) results of using EES' 1-D optimization to vary the rotation angle in order to minimize the required value of *F*.

4.2E Multi-dimensional Optimization in EES

Section 4.1E discussed one-dimensional optimization using EES. In this case an objective function was optimized (minimized or maximized) by varying a single independent variable. Most engineering designs will involve more than one independent variable and therefore require multi-dimensional optimization. In this section we will introduce the multi-dimensional optimization algorithms that are available in EES in the context of the two-dimensional objective function defined by

$$f = 0.7\exp\left(-\frac{r_1^2}{0.2^2}\right) + \left[1 - 0.7\exp\left(-\frac{r_1^2}{0.2^2}\right)\right]\exp\left(-\frac{r_2^2}{0.05^2}\right), \quad (4.5)$$

where

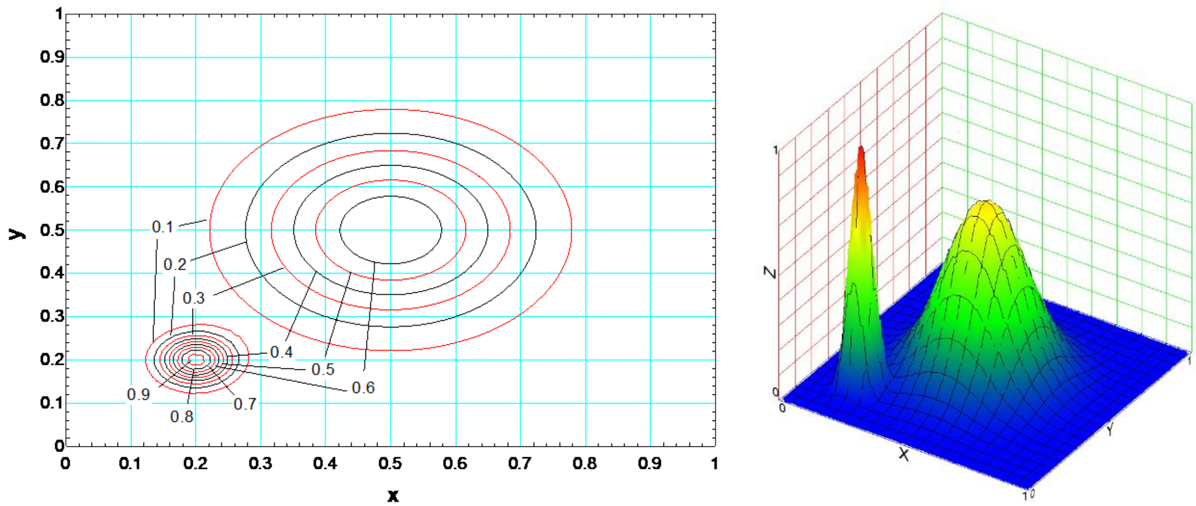
$$r_1 = \sqrt{(x-0.5)^2 + (y-0.5)^2}, \text{ and} \quad (4.6)$$

$$r_2 = \sqrt{(x-0.2)^2 + (y-0.2)^2}. \quad (4.7)$$

Note that the multi-dimensional optimization algorithms discussed in this section can be easily extended to more independent variables, but the techniques are most easily visualized in two-dimensions. The objective function can be entered in EES, as shown below.

```
"Objective Function"
r1=Sqrt((x-0.5)^2+(y-0.5)^2)
r2=Sqrt((x-0.2)^2+(y-0.2)^2)
f=0.7*Exp(-r1^2/0.2^2)+(1-0.7*exp(-r1^2/0.2^2))*exp(-r2^2/0.05^2)
```

In Chapter 5 we will discuss 3-D plots in EES including both contour and surface plots. A contour plot of the objective function given by Eqs. (4.5) through (4.7) is shown in Figure 4.9(a) and a surface plot is shown in Figure 4.9(b).



(a) (b)

Figure 4.9: (a) Contour plot and (b) surface plot of the objective function (*f*) as a function of the two independent variables *x* and *y*.

Because we have not specified either *x* or *y* in the EES code there are two degrees of freedom. Therefore, if we select Min/Max from the Calculate menu we can access the Find Minimum or Maximum shown in Figure 4.10.

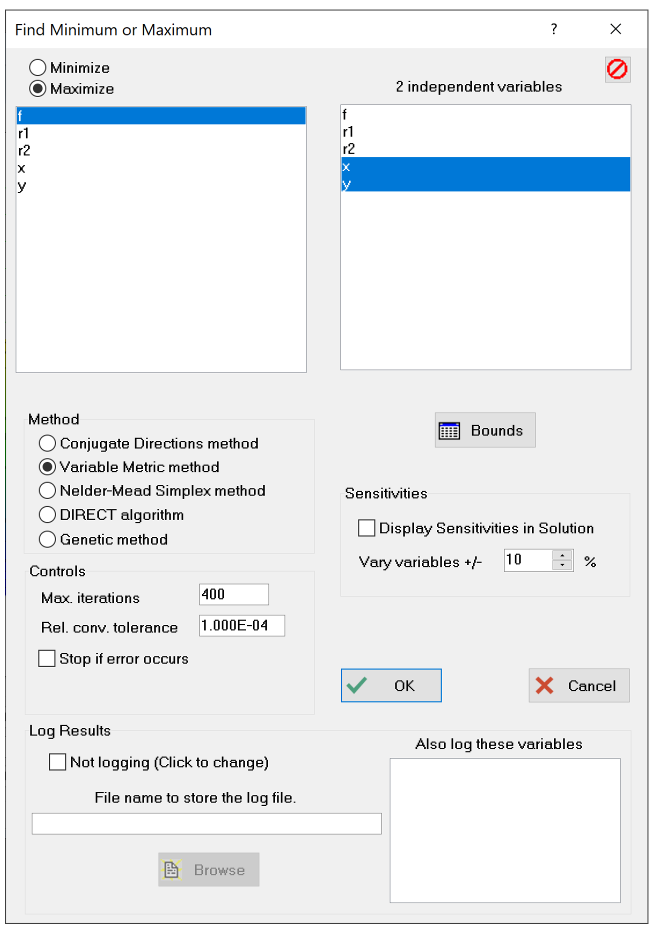


Figure 4.10. Find Minimum or Maximum Dialog.

The optimization problem is set up to maximize the objective function *f* by adjusting the two independent variables (*x* and *y*). Select the Bounds button to set the bounds and guess values for the two independent variables, *x* and *y*. In this case the parameter space to explore will be from 0 to 1 for both *x* and *y*.

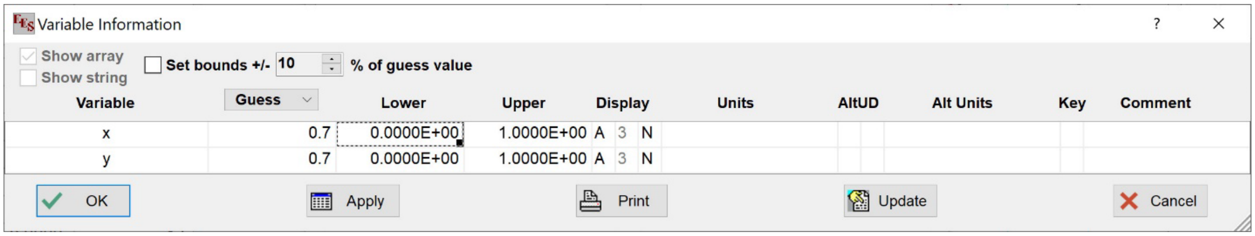


Figure 4.11. Bounds and guess values for the independent variables.

There are several methods available for multi-dimensional optimization. Because they have been programmed in EES it is not necessary to understand the details of their implementation in order to use them. However, it is useful to have a general idea of how each method works so that you can select the appropriate method for your problem.

The Conjugate Directions Method

The Conjugate Directions method uses a series of one-dimensional searches to locate the optimum. In its simplest form, EES will hold all but one of the optimization variables constant and then vary the single remaining parameter in order to locate the value at which the objective function is maximized along a one-dimensional path. This process can be accomplished using one of the one-dimensional optimization techniques discussed in Section E4.1. This process is repeated for each independent variable multiple times until the stopping criteria are achieved.

As an example, consider Figure 4.12. The process begins at the specified guess values for the independent variables (x = 0.7 and y = 0.7, labeled point 1) and then progresses by holding x constant while varying y (i.e., moving along path 1). The optimal value of y is identified along this path (labeled point 2). Next the value of y is held constant and x is varied (i.e., we move along path 2), this leads us to point 3. Next the process repeats by holding x constant and varying y (i.e., moving along path 3) and continues in this manner. EES employs a method referred to as conjugate directions in order to improve the efficiency of the optimization. The conjugate directions method makes the one-dimensional searches along directions that are oriented more favorably than those defined by any of the independent variables in the problem.

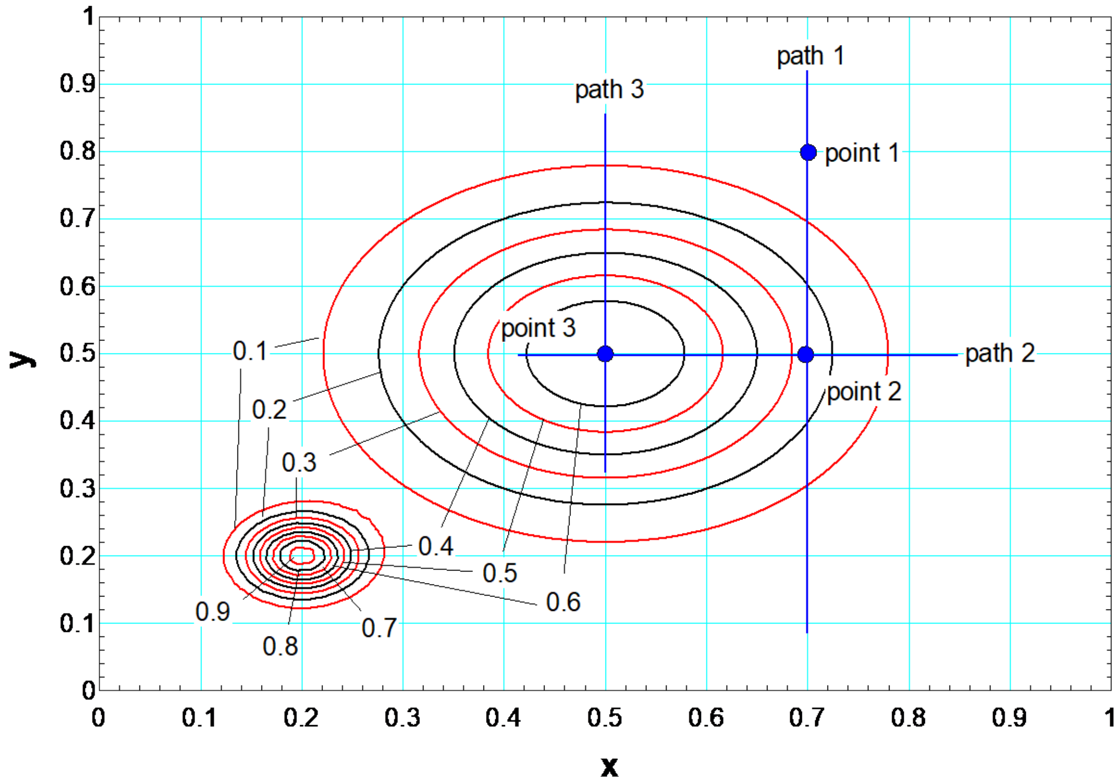


Figure 4.12. Progression of the Conjugate Directions method.

In order to implement the technique, select Conjugate Directions method and then OK. The result is shown in Figure 4.13. The optimization required 45 iterations (i.e., function evaluations) to arrive at the optimal value $x = 0.5$ and $y = 0.5$ where the objective function is $f = 0.7$. Notice from Figure 4.12 that the technique did not identify the global maximum but rather a local maximum; the true optimal solution corresponds to the smaller but taller peak at $x = 0.2$ and $y = 0.2$. The Conjugate Directions method is sensitive to the guess values for the optimization variables (which correspond to the starting point of the optimization process). If a guess value that is closer to the peak is specified (e.g., $x = 0.15$ and $y = 0.15$) then the Conjugate Directions method is likely to converge to the global optimal value of $x = 0.2$ and $y = 0.2$ where the objective function is $f = 1$.

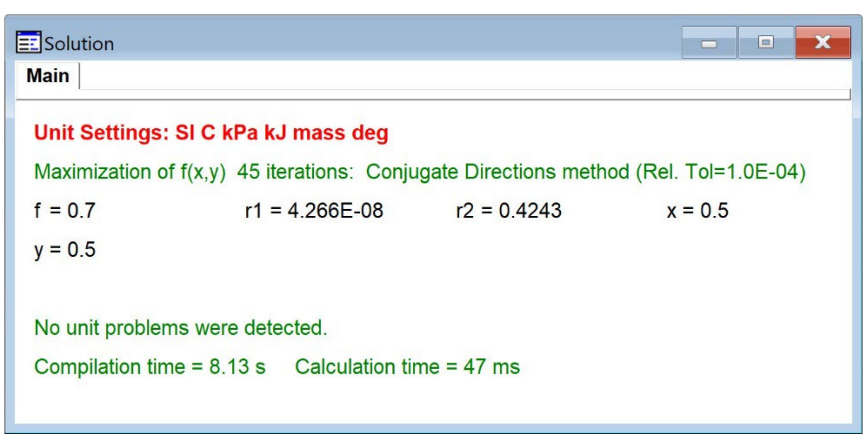


Figure 4.13. Results of using the Conjugate Directions method.

The Variable Metric Method

The Variable Metric method is a multi-dimensional version of the quadratic approximations method that was discussed for one-dimensional optimization in Section 4.1. The objective function is fit to a quadratic function of all of the independent variables. The function is used to locate an estimate of the optimal value, which leads to a new trial point and the process is continued. The Variable Metric method, like the Conjugate Directions method, is sensitive to the guess values that are used to start the optimization and may not find a global optimum.

In order to use the Variable Metric Method it is only necessary to select the Variable Metric method radio button in the Find Minimum or Maximum dialog. The method is slightly more efficient than the Conjugate Directions Method for this application, requiring only 39 iterations as shown in Figure 4.14. Note that the Variable Metric Method also failed to identify the global optimum for this problem.

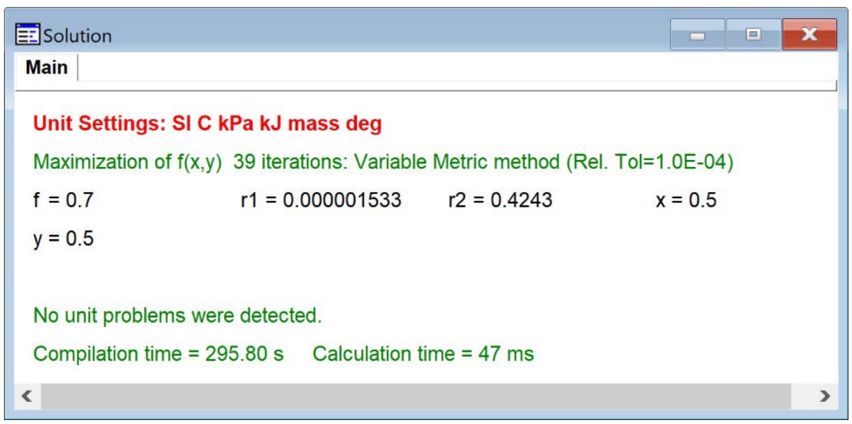


Figure 4.14. Results of using the Variable Metric method.

The Nelder-Mead Simplex Method

The Professional license of EES provides the Nelder-Mead simplex method of multi-dimensional optimization. The algorithm uses N+1 test points at a time, where N is the dimension of the problem (e.g., for our two-dimensional optimization problem the method would use three points). The test points define a simplex; for this 2-D problem, the simplex is a triangle. The method proceeds by modifying the simplex during each iteration by adjusting one of its points. The process continues until the stopping criterion is achieved. The method is illustrated in Figure 4.15 where the simplex associated with iterations 12 through 16 are shown.

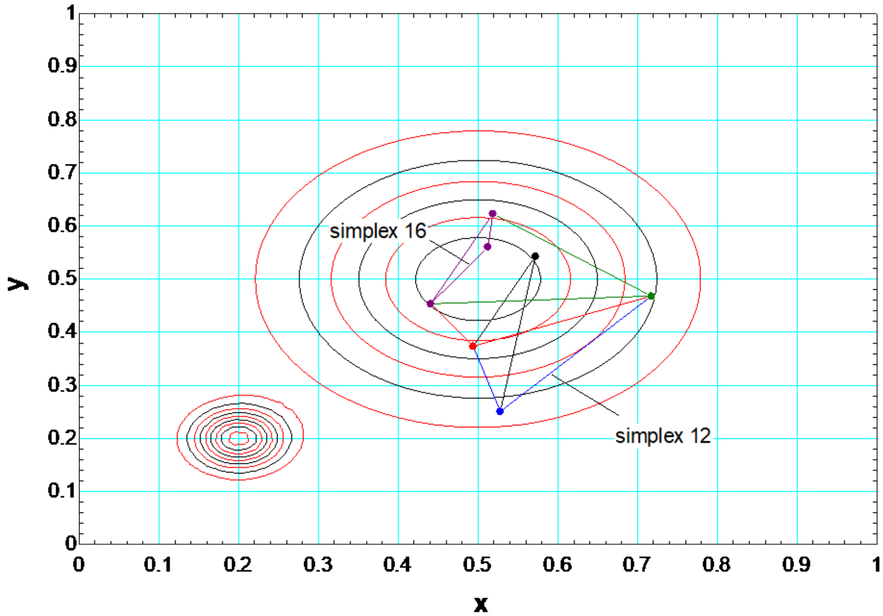


Figure 4.15. Progression of the Nelder-Mead Simplex method.

Select the Nelder-Mead Simplex Method in the Find Minimum or Maximum dialog. The result is shown in Figure 4.16. Notice the method is not as efficient as the others discussed since it requires 62 iterations and also that it failed to find the global optimal solution.

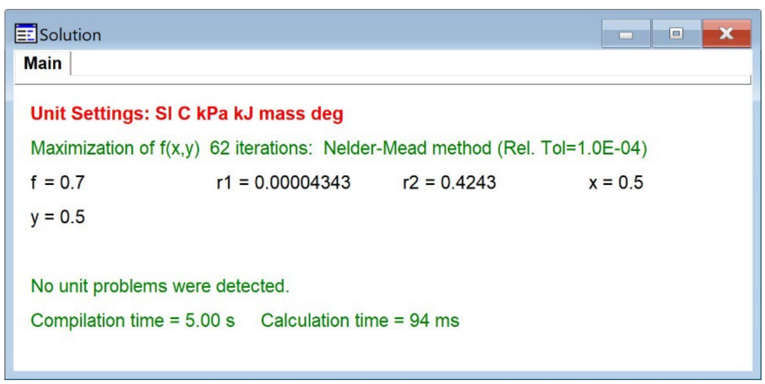


Figure 4.16. Results of using the Nelder-Mead Simplex Method.

The DIRECT Algorithm

The DIRECT Optimization Algorithm implements the original DIRECT algorithm which subdivides the parameter space into successively smaller regions using a regular grid. Each region is then sampled and subdivided as necessary. The progression of the algorithm for this problem is shown in Figure 4.17 where you can see the initial, large region sampled by the algorithm followed by the successive smaller regions.

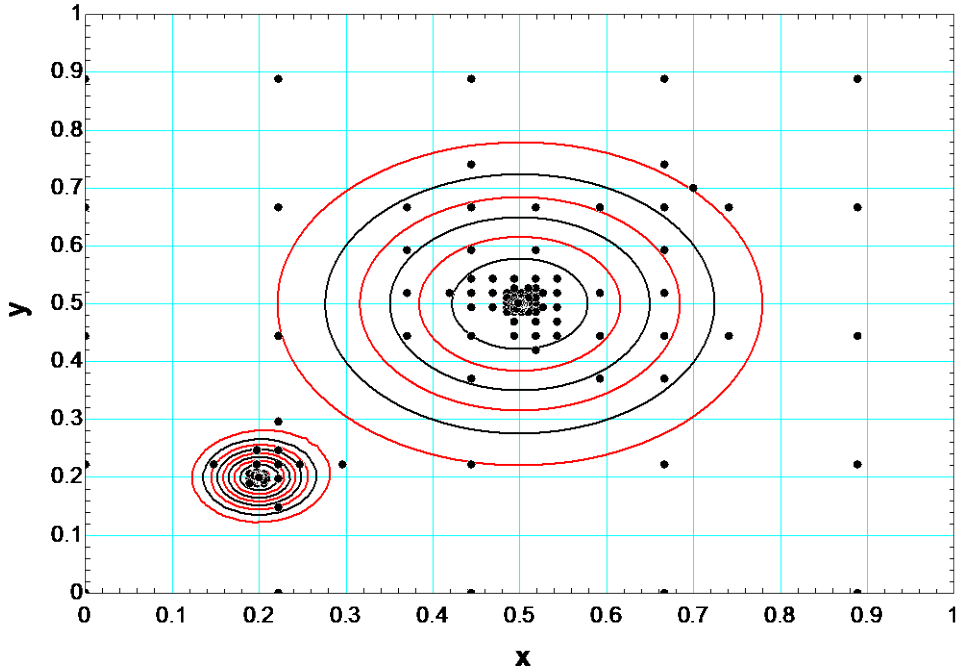


Figure 4.17. Progression of the Nelder-Mead Simplex method.

The DIRECT algorithm is sometimes better at finding a global optimal value than the techniques discussed thus far since it samples the entire parameter space, regardless of the guess value. Figure 4.18 illustrates the result of using the DIRECT Method for this problem. Notice that it was able to find the global optimal solution at $x = 0.2$ and $y = 0.2$ but that it required 310 iterations.

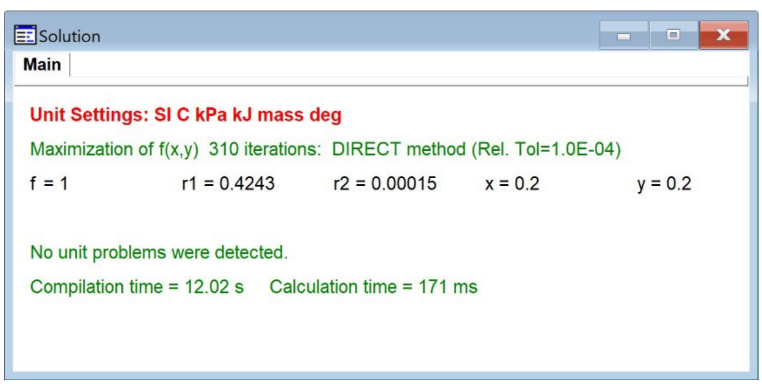


Figure 4.18. Results of using the DIRECT Method.

The Genetic Method

The Professional license of EES includes the Genetic Method which is designed to reliably locate global optimal values even if the surface has many local peaks. The genetic method mimics the processes occurring in biological evolution. A population of individuals (i.e., sample points) is initially chosen at random from the range specified by the bounds of the independent variables. The individuals in this population are surveyed to determine their fitness (i.e., the values of the objective function). Then a new generation of individuals is generated in a stochastic manner by 'breeding' selected members of the current population according to their fitness. The characteristics of an individual that are passed on to the next generation are represented by encoded values of its independent variables. The probability that an individual in the current population will be selected for breeding the next generation is an increasing function of its fitness. Additional random variations are introduced by the possibility of 'mutations' that cause the offspring to have characteristics that differ markedly from those of the parents. In the implementation programmed in EES, the number of individuals in the population remains constant for each generation. The number of individuals, number of generations, and mutation rate can be adjusted by slider bars in the Find Minimum or Maximum Dialog once the Genetic Method is selected, as shown in Figure 4.19.

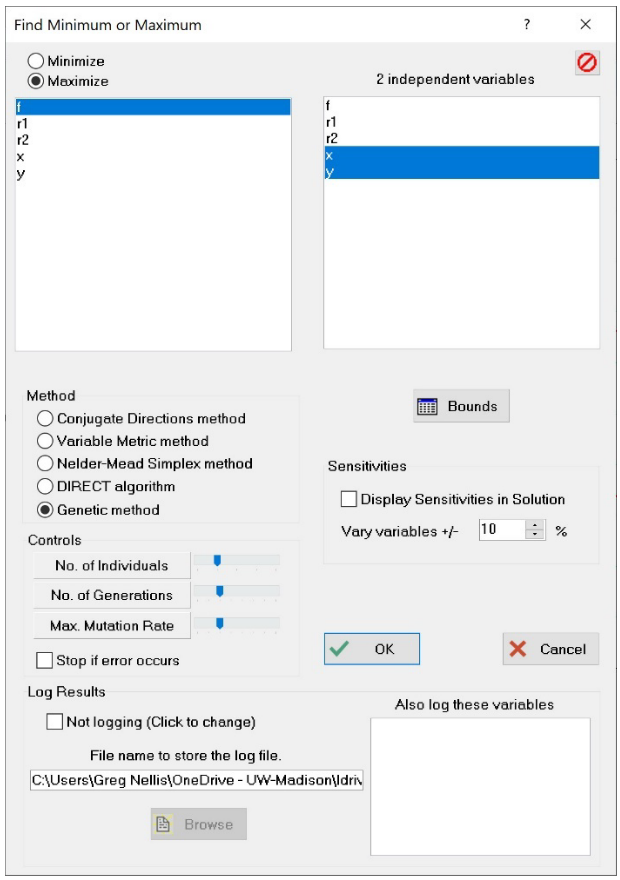


Figure 4.19. Find Minimum or Maximum Dialog for the Genetic Method.

Figure 4.20 illustrates the progression of the genetic optimization method; the populations for a few selected generations are shown. In the first generation, Figure 4.20(a), the members of the population are uniformly distributed throughout the parameter space (although one member is located exactly at the coordinates associated with the guess value, x = 0.7 and y = 0.7). The population slowly converges towards the broad peak, as shown in Figure 4.20(b) for generation 5. Mutations cause individual members to move away from this broad peak, as shown in Figure 4.20(c) for generation 23. If even one member happens to come near the second, smaller but steeper peak then the majority of the population will eventually be attracted there. This is shown in Figure 4.20(d) for generation 77.

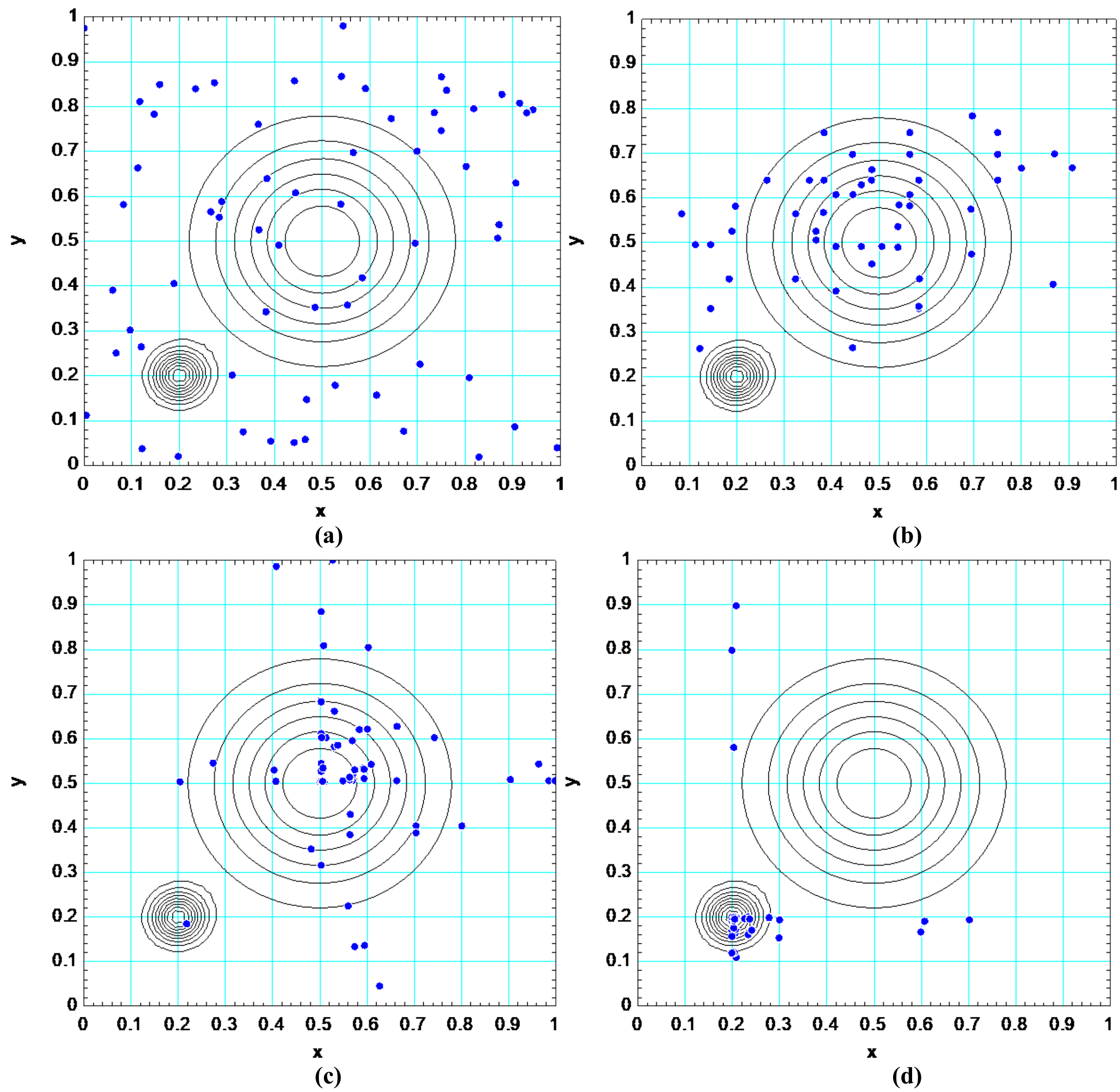


Figure 4.20. Progression of the genetic optimization method: (a) generation 1, (b) generation 5, (c) generation 23, and (d) generation 77.

The primary advantage of genetic optimization is its ability to reliably find a global optimum even if there are many local optimal values in the problem. Genetic optimization will inevitably find the sharp peak given a sufficiently large population and number of generations. The disadvantage of genetic optimization is that it takes a long time since many of the members of each population are not useful. Figure 4.21 shows the results of using the Genetic Method. Note that the global optimal was identified, but that it required 2151 iterations to locate it.

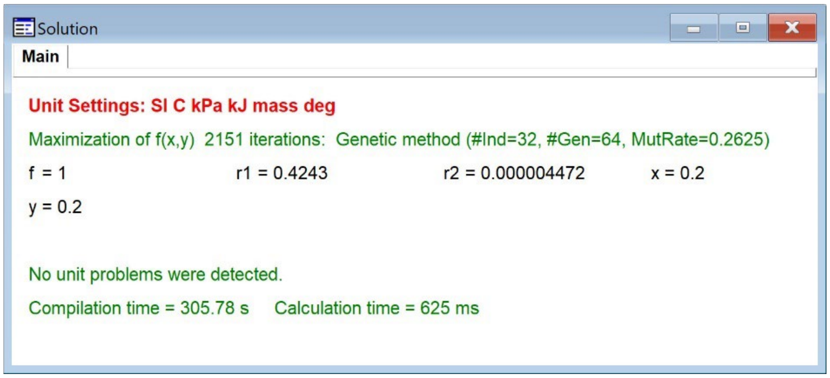


Figure 4.21. Results of using the DIRECT Method.

EXAMPLE E4.4 *Determination of an Equivalent Force System*

We will revisit Example 4.13 from the text. A table supports the vertical forces shown.

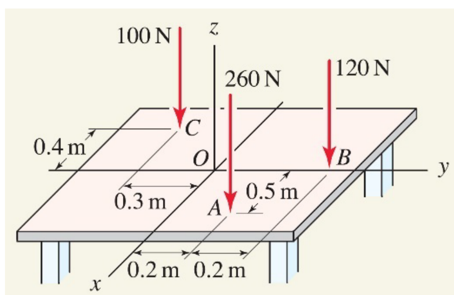


Figure 1. Example 4.13 in Gray et al. (2023).

(a) Determine an equivalent force system at the center of the table, point *O*.

(b) Determine an equivalent force system consisting of a single force, and specify the *x* and *y* coordinates of the point where the force's line of action intersects the table.

SOLUTION

Road Map We will use a vector approach for this problem. The equivalent force system for part (a) is the resultant of the applied forces together with the net moment applied by the forces about point *O*. In part (b) the equivalent force system is again the resultant of the applied forces this resultant force must be applied at a point that leads to the same net moment calculated in part (a).

Part (a)

Governing Equations The coordinates of the center of the table (*O*) and the points where each force is applied (*A*, *B*, and *C*) are all specified. Therefore, the position vectors from the center to each point of application can be determined

$$\vec{r}_{OA} = A - O, \tag{1}$$

$$\vec{r}_{OB} = B - O, \text{ and} \tag{2}$$

$$\vec{r}_{OC} = C - O. \tag{3}$$

The magnitude of each force and their direction (vertically downwards, or in the $-\hat{j}$ direction) are also specified, allowing each force vector ($\vec{F}_A$, $\vec{F}_B$, and $\vec{F}_C$) to be determined. The resultant of these forces is

$$\vec{F}_R = \vec{F}_A + \vec{F}_B + \vec{F}_C, \tag{4}$$

and the net moment that these forces produce about point O is

$$\vec{M}_{RO} = \vec{r}_{OA} \times \vec{F}_A + \vec{r}_{OB} \times \vec{F}_B + \vec{r}_{OC} \times \vec{F}_C. \tag{5}$$

Computation The inputs include the coordinates of the points O, A, B, and C as well as magnitudes of the applied forces.

```
$Vector O, A, B, C
O=VectorAssign(0,0,0) [m]                 "center of table"
A=VectorAssign(0.5, 0.2, 0) [m]           "point of application for force A"
B=VectorAssign(0,0.4,0) [m]               "point of application of force B"
C=VectorAssign(-0.4,-0.3,0) [m]           "point of application of force C"
magF_A = 260 [N]                          "magnitude of force A"
magF_B = 120 [N]                          "magnitude of force B"
magF_C = 100 [N]                          "magnitude of force"
```

The forces are specified based on their magnitude and direction.

```
$Vector F_A, F_B, F_C
F_A = -magF_A*VectorUnit_k                "force A"
F_B = -magF_B*VectorUnit_k                "force B"
F_C = -magF_C*VectorUnit_k                "force C"
```

The position vectors from the center of the table to each force's point of application are obtained using Eqs. (1) through (3).

```
$Vector r_OA, r_OB, r_OC
r_OA = A - O                              "position vector from center to A"
r_OB = B - O                              "position vector from center to B"
r_OC = C - O                              "position vector from center to C"
```

The force and moment associated with the equivalent force system applied at point O are obtained using Eqs. (4) and (5).

```
$Vector F_R, M_RO
F_R = F_A + F_B + F_C                     "resultant force"
M_RO =VectorCross(r_OA,F_A) + VectorCross(r_OB,F_B)+VectorCross(r_OC,F_C)
```

```
"resultant moment about point O"
```

Solving leads to $\vec{F}_R$ = (0, 0, -480) N and $\vec{M}_{RO}$ = (-70, 90, 0) N-m.

Discussion & Verification It is instructive to visualize these results using a 3-D vector plot. The EES code below converts the coordinates and position vectors from m to mm so that they have approximately the same scale as the forces involved in the problem and generates a 3-D vector plot. The result is shown in Figure 2. Notice that the net moment lies entirely in the *x*-*y* plane since all of the applied forces are in the *z*-direction.

```
$Vector r_OA_mm, r_OB_mm, r_OC_mm, A_mm, B_mm, C_mm
r_OA_mm = r_OA*Convert(m,mm)
r_OB_mm = r_OB*Convert(m,mm)
r_OC_mm = r_OC*Convert(m,mm)
A_mm = A*Convert(m,mm)
B_mm = B*Convert(m,mm)
C_mm = C*Convert(m,mm)

$VectorPlot Name = 'Part a' r_OA_mm: O/black r_OB_mm: O/black r_OC_mm:O/black F_A:A_mm/red&
F_B:B_mm/red F_C:C_mm/red F_R:O/red M_RO:O/purple
```

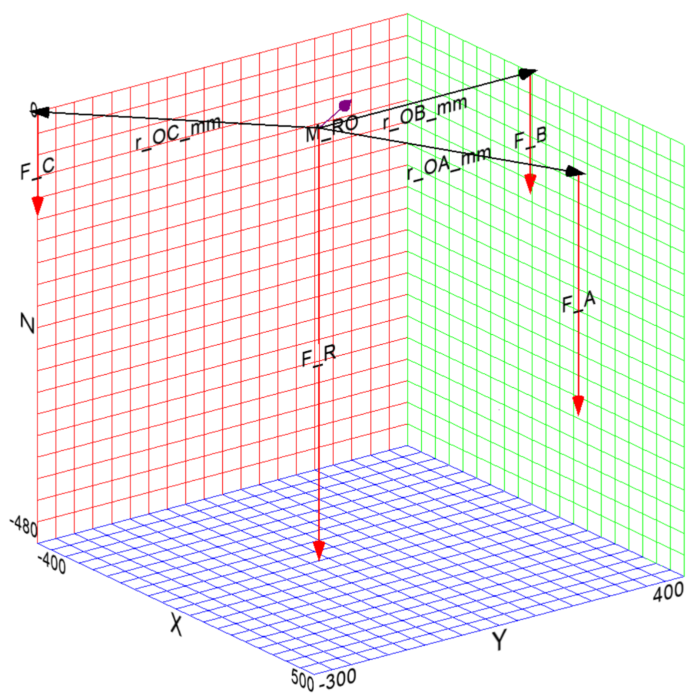


Figure 2. 3-D Vector plot showing applied forces and the equivalent force and moment.

Part (b)

Governing Equations and Computation We need to find the coordinates of a point *D* such that the resultant force, $\vec{F}_R$, applied at that point produces the same total moment $\vec{M}_{RO}$. The portion of the code from part (a) that computes these quantities is copied into a new tab.

```
$Vector O, A, B, C
O=VectorAssign(0,0,0) [m]                 "center of table"
A=VectorAssign(0.5, 0.2, 0) [m]           "point of application for force A"
B=VectorAssign(0,0.4,0) [m]               "point of application of force B"
C=VectorAssign(-0.4,-0.3,0) [m]           "point of application of force C"
magF_A = 260 [N]                          "magnitude of force A"
magF_B = 120 [N]                          "magnitude of force B"
magF_C = 100 [N]                          "magnitude of force"

$Vector F_A, F_B, F_C
F_A = -magF_A*VectorUnit_k                "force A"
F_B = -magF_B*VectorUnit_k                "force B"
F_C = -magF_C*VectorUnit_k                "force C"

$Vector r_OA, r_OB, r_OC
r_OA = A - O                              "position vector from center to A"
r_OB = B - O                              "position vector from center to B"
r_OC = C - O                              "position vector from center to C"

$Vector F_R, M_RO
F_R = F_A + F_B + F_C                     "resultant force"
M_RO =VectorCross(r_OA,F_A) + VectorCross(r_OB,F_B)+VectorCross(r_OC,F_C)
                                          "resultant moment about point O"
```

The point D must lie on the table so it's z-coordinate must be zero ($D_z = 0$). The x- and y-components of point D are unknown. Initially we will set them to reasonable values to proceed with the solution.

```
$Vector r_OD, D
D_x = 0.3 [m]                             "assumed - will become an optimization variable"
D_y = 0.2 [m]                             "assumed - will become an optimization variable"
D_z = 0 [m]                               "point D lies on the table"
r_OD = D - O                              "position vector associated with force application"
```

The moment associated with applying force $\vec{F}_R$ at point D is given by

$$\vec{M}_{RO,calc} = \vec{r}_{OD} \times \vec{F}_R. \qquad (6)$$

```
$Vector M_RO_calc
M_RO_calc = VectorCross(r_OD, F_R)        "moment from resultant force"
```

We need to vary the coordinates of D so that the calculated value of the moment, $\vec{M}_{RO,calc}$, is equal to the resultant moment, $\vec{M}_{RO}$. The z-component of both $\vec{M}_{RO,calc}$ and $\vec{M}_{RO}$ will necessarily be zero, as shown in Figure 2. Therefore, we need only to vary D_x and D_y until

$$\vec{M}_{RO,calc,x} = \vec{M}_{RO,x}, \text{ and} \qquad (7)$$

$$\vec{M}_{RO,calc,y} = \vec{M}_{RO,y}. \qquad (8)$$

We can do this in several ways, one possibility is to set this up as a multi-dimensional optimization problem where the independent variables are D_x and D_y and the objective function is an error related to the degree to which Eqs. (7) and (8) are satisfied. For example, the total rms error is given by

$$error = \sqrt{\left(\vec{M}_{RO,calc,x} - \vec{M}_{RO,x}\right)^2 + \left(\vec{M}_{RO,calc,y} - \vec{M}_{RO,y}\right)^2}. \tag{9}$$

```
error = Sqrt((M_RO_calc_x-M_RO_x)^2 + (M_RO_calc_y-M_RO_y)^2)
```

Comment out the two independent variables, D_x and D_y, so that there are two degrees of freedom.

```
{D_x = 0.3 [m]                    "assumed - will become an optimization variable"
D_y = 0.2 [m]                     "assumed - will become an optimization variable"}
```

Select Min/Max from the Calculate menu and setup the optimization problem to minimize the value of the variable error by changing the values of the variables D_x and D_y, as shown in Figure 3. Set the bounds and guess values and select OK to run the optimization. The coordinates of point D found by the optimization are (0.1875, 0.1458, 0) m.

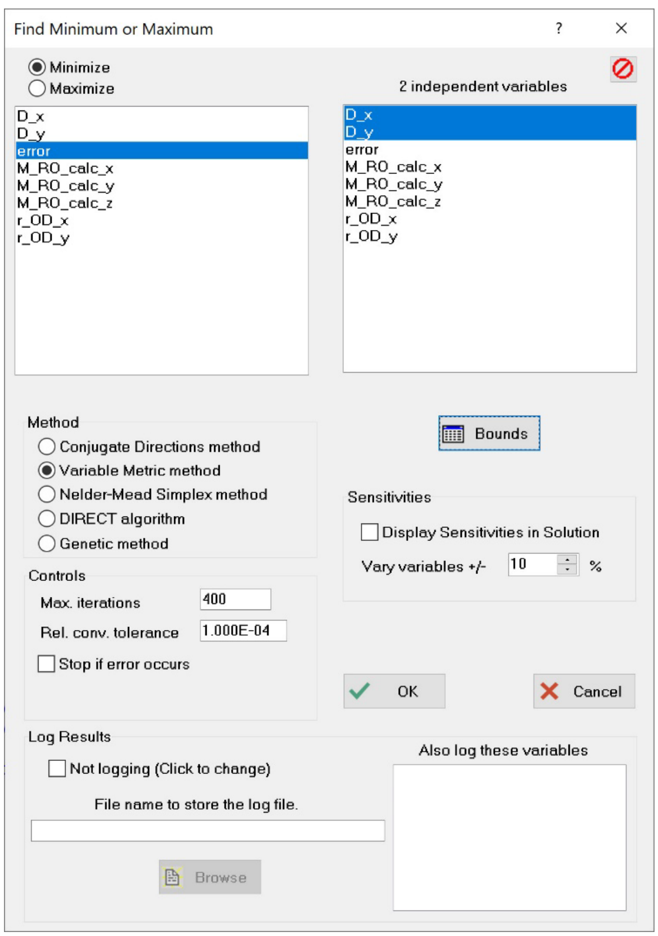


Figure 3. Find Minimum or Maximum Dialog.

The directive below creates the 3-D vector plot shown in Figure 4 which illustrates the three applied forces as well as their resultant applied at point *D* so that creates the same moment.

```
$Vector r_OA_mm, r_OB_mm, r_OC_mm, r_OD_mm, A_mm, B_mm, C_mm, D_mm
r_OA_mm = r_OA*Convert(m,mm)
r_OB_mm = r_OB*Convert(m,mm)
r_OC_mm = r_OC*Convert(m,mm)
r_OD_mm = r_OD*Convert(m,mm)
A_mm = A*Convert(m,mm)
B_mm = B*Convert(m,mm)
C_mm = C*Convert(m,mm)
D_mm = D*Convert(m,mm)

$VectorPlot Name = 'Part b' r_OA_mm: O/black r_OB_mm: O/black r_OC_mm:O/black&
r_OD_mm:O/blue F_A:A_mm/red F_B:B_mm/red F_C:C_mm/red F_R:D_mm/purple
```

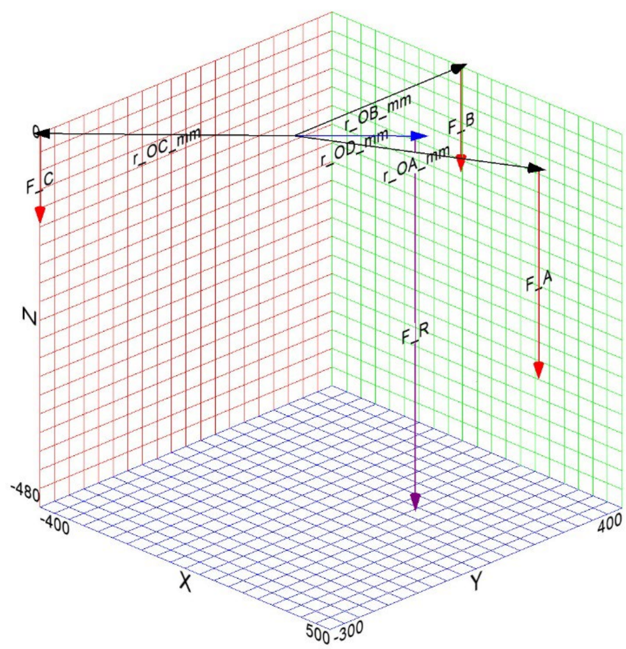


Figure 4. 3-D Vector plot showing applied forces and the resultant placed at point *D* which creates the same moment.