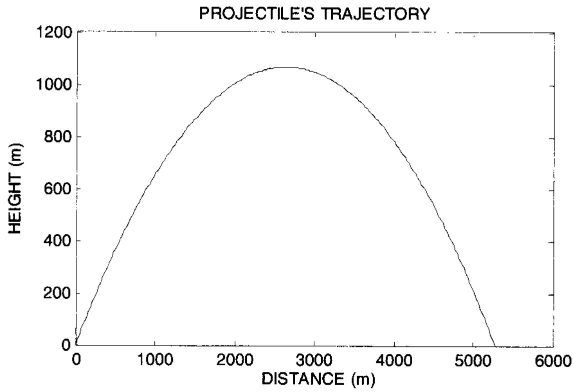
In addition, the following figure is created in the Figure Window:




## 7.13 *Problems*

1. The fuel efficiency of an automobile is measured in mi/gal (miles per U.S. gallon) or in km/L (kilometers per liter). Write a MATLAB user-defined function that converts fuel efficiency values from km/L to mi/gal. For the function name and arguments use `mpg=kmlTOmpg(kml)`. The input argument `kml` is the efficiency in km/L, and the output argument `mpg` is the efficiency in mi/gal. Use the function in the Command Window to:
   (*a*) Determine the fuel efficiency in mi/gal of a car that consumes 9 km/L.
   (b) Determine the fuel efficiency in mi/gal of a car that consumes 14 km/L.

2. Write a user-defined MATLAB function for the following math function:
$$y(x) = -0.2x^4 + e^{-0.5x}x^3 + 7x^2$$
   The input to the function is $x$ and the output is $y$. Write the function such that $x$ can be a vector (use element-by-element operations).
   (*a*) Use the function to calculate $y(-2.5)$, and $y(3)$.
   (*b*) Use the function to make a plot of the function $y(x)$ for $-3 \le x \le 4$.

3. Write a user-defined MATLAB function, with two input and two output arguments, that determines the height in centimeters and mass in kilograms of a person from his height in inches and weight in pounds. For the function name and arguments use `[cm, kg] = STtoSI(in, lb)`. The input arguments are the height in inches and weight in pounds, and the output arguments are the height in centimeters and mass in kilograms. Use the function in the Command Window to:

   (*a*) Determine in SI units the height and mass of a 5 ft 8 in. person who weighs 175 lb.
   (b) Determine your own height and weight in SI units.

4. Write a user-defined MATLAB function that converts speed given in units of miles per hour to speed in units of meters per second. For the function name and arguments use `mps = mphTOmets(mph)`. The input argument is the speed in mi/h, and the output argument is the speed in m/s. Use the function to convert 55 mi/h to units of m/s.

5. Write a user-defined MATLAB function for the following math function:

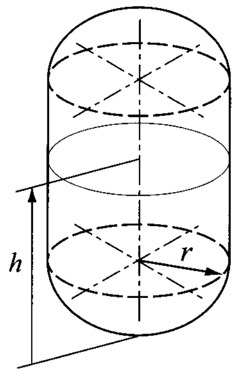
$$r(\theta) = 2\cos\theta\sin\theta\sin(\theta/4)$$

The input to the function is $\theta$ (in radians) and the output is $r$. Write the function such that $\theta$ can be a vector.
(*a*) Use the function to calculate $r(3\pi/4)$ and $r(7\pi/4)$.
(*b*) Use the function to plot (polar plot) $r(\theta)$ for $0 \le \theta \le 2\pi$.

6. Write a user-defined MATLAB function that determines the area of a triangle when the lengths of the sides are given. For the function name and arguments use `[Area] = triangle(a,b,c)`. Use the function to determine the areas of triangles with the following sides:
(*a*) $a = 3, b = 8, c = 10$. (*b*) $a = 7, b = 7, c = 5$.

7. A cylindrical vertical fuel tank has hemispheric end caps as shown. The radius of the cylinder and the caps is $r$ = 15 in., and the height of the cylindrical middle section is 40 in.

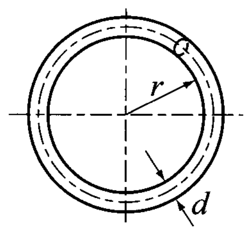
Write a user-defined function (for the function name and arguments use `V = Volfuel(h)`) that gives the volume of fuel in the tank (in gallons) as a function of the height $h$ (measured from the bottom). Use the function to make a plot of the volume as a function of $h$ for $0 \le h \le 70$ in.




8. The surface area $S$ of a ring in shape of a torus with an inner radius $r$ and a diameter $d$ is given by:

$$S = \pi^2(2r + d)d$$

The ring is to be plated with a thin layer of coating. The weight of the coating $W$ can be calculated approximately as $W = \gamma S t$, where $\gamma$ is the specific weight of the coating material and $t$ is its thickness. Write an anonymous function that calculates the weight of the coating. The function should have four input arguments, $r$, $d$, $t$, and $\gamma$. Use the anonymous function to calculate the weight of a gold coating ($\gamma$ = 0.696 lb/in.$^3$) of a ring with $r$ = 0.35 in., $d$ = 0.12 in., and $t$ = 0.002 in.



9. The monthly deposit into a savings account $S$ needed to reach an investment goal $B$ can be calculated by the formula

$$M = S\frac{\frac{r}{1200}}{(1+\frac{r}{1200})^{12N}-1}$$

where $M$ is the monthly deposit, $S$ is the saving goal, $N$ is the number of years, and $r$ is the annual interest rate (%). Write a MATLAB user-defined function that calculates the monthly deposit into a savings account. For the function name and arguments use `M = invest(S,r,N)`. The input arguments are `S` (the investment goal), `r` (the annual interest rate, %), and `N` (duration of the savings in years). The output `M` is the amount of the monthly deposit. Use the function to calculate the monthly deposit for a 10-year investment if the investment goal is $25,000 and the annual interest rate is 4.25%.

10. The heat index, $HI$ (in degrees F), is an apparent temperature. For temperatures higher than 80°F and humidity higher than 40% it is calculated by:

$$HI = C_1 + C_2T + C_3R + C_4TR + C_5T^2 + C_6R^2 + C_7T^2R + C_8TR^2 + C_9R^2T^2$$

where $T$ is temperature in degrees F, $R$ is the relative humidity in percent, $C_1 = -42.379$, $C_2 = 2.04901523$, $C_3 = 10.14333127$, $C_4 = -0.22475541$, $C_5 = -6.83783 \times 10^{-3}$, $C_6 = -5.481717 \times 10^{-2}$, $C_7 = 1.22874 \times 10^{-3}$, $C_8 = 8.5282 \times 10^{-4}$, and $C_9 = -1.99 \times 10^{-6}$. Write a user-defined function for calculating $HI$ for given $T$ and $R$. For the function name and arguments use `HI=HeatIn(T,R)`. The input arguments are `T` in °F and, `R` in %, and the output argument is `HI` in °F (rounded to the nearest integer). Use the function to determine the heat index for the following conditions:

(*a*) $T = 95$ °F, $R = 80$ %.

(*b*) $T = 100$ °F, $R = 100$ % (condition in a sauna).

11. The body fat percentage ($BFP$) of a person can be estimated by the formula

$$BFP = 1.2 \times BMI + 0.23 \times Age - 10.8 \times Gender - 0.54$$

where $BMI$ is the body mass index, given by $BMI = 703\frac{W}{H^2}$, in which $W$ is the weight in pounds and $H$ is the height in inches, $Age$ is the person's age, and $Gender = 1$ for a male and $Gender = 0$ for a female.

Write a MATLAB user-defined function that calculates the body fat percentage. For the function name and arguments use `BFP = BodyFat(w,h,age,gen)`. The input arguments are the weight, height, age, and gender (1 for male, 0 for female), respectively. The output argument is the $BEF$ value. Use the function to calculate the body fat percentage of:

*a*) A 35-years-old, 6 ft 2 in. tall, 220 lb male.

*b*) A 22-years-old, 5 ft 7 in. tall, 135 lb female.

12. Write a user-defined function that calculates grade point average (GPA) on a scale of 0 to 4, where $A = 4$, $B = 3$, $C = 3$, $D = 1$, and $E = 0$. For the function name and arguments use `av = GPA(g,h)`. The input argument `g` is a vector whose elements are letter grades *A*, *B*, *C*, *D*, or *E* entered as strings. The input argument `h` is a vector with the corresponding credit hours. The output argument `av` is the calculated GPA. Use the function to calculate the GPA for a student with the following record:

| Grade | B | A | C | E | A | B | D | B |
|---|---|---|---|---|---|---|---|---|
| Credit Hours | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 2 |

For this case the input arguments are:
`g=['BACEABDB']` and `h=[3 4 3 4 3 4 3 2]`.

13. The factorial $n!$ of a positive number (integer) is defined by $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$, where $0! = 1$. Write a user-defined function that calculates the factorial $n!$ of a number. For function name and arguments use `y=fact(x)`, where the input argument `x` is the number whose factorial is to be calculated, and the output argument `y` is the value $x!$. The function displays an error message if a negative or non-integer number is entered when the function is called. Use `fact` with the following numbers:
(*a*) 12! (*b*) 0! (*c*) –7! (*d*) 6.7!

14. Write a user-defined MATLAB function that determines the vector connecting two points (*A* and *B*). For the function name and arguments use `V=vector(A,B)`. The input arguments to the function are vectors `A` and `B`, each with the Cartesian coordinates of points *A* and *B*. The output `V` is the vector from point *A* to point *B*. If points *A* and *B* have two coordinates each (they are in the *x y* plane), then `V` is a two-element vector. If points *A* and *B* have three coordinates each (general points in space), then `V` is a three-element vector. Use the function `vector` for determining the following vectors.
(*a*) The vector from point (0.5, 1.8) to point (–3, 16).
(*b*) The vector from point (–8.4, 3.5, –2.2) to point (5, –4.6, 15).

15. Write a user-defined MATLAB function that determines the dot product of two vectors. For the function name and arguments use `D=dotpro(u,v)`. The input arguments to the function are two vectors, which can be two- or three-dimensional. The output `D` is the result (a scalar). Use the function `dotpro` for determining the dot product of:
(*a*) Vectors $a = 3i + 11j$ and $b = 14i - 7.3j$.
(*b*) Vectors $c = -6i + 14.2j + 3k$ and $d = 6.3i - 8j - 5.6k$.

16. Write a user-defined MATLAB function that determines the unit vector in the direction of the line that connects two points ($A$ and $B$) in space. For the function name and arguments use `n = unitvec(A,B)`. The input to the function are two vectors `A` and `B`, each with the Cartesian coordinates of the corresponding point. The output is a vector with the components of the unit vector in the direction from $A$ to $B$. If points $A$ and $B$ have two coordinates each (they are in the $x\,y$ plane), then `n` is a two-element vector. If points $A$ and $B$ have three coordinate each (general points in space), then `n` is a three-element vector. Use the function to determine the following unit vectors:
(*a*) In the direction from point (1.2, 3.5) to point (12, 15).
(*b*) In the direction from point (–10, –4, 2.5) to point (–13, 6, –5).

17. Write a user-defined MATLAB function that determines the cross product of two vectors. For the function name and arguments use `w=crosspro(u,v)`. The input arguments to the function are the two vectors, which can be two- or three-dimensional. The output `w` is the result (a vector). Use the function `crisper` for determining the cross product of:
(*a*) Vectors $a = 3i + 11j$ and $b = 14i - 7.3j$.
(*b*) Vectors $c = -6i + 14.2j + 3k$ and $d = 6.3i - 8j - 5.6k$.

18. The area of a triangle $ABC$ can be calculated by:

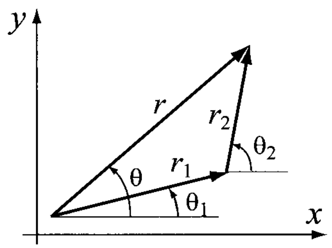
$$A = \frac{1}{2}|\boldsymbol{AB} \times \boldsymbol{AC}|$$

where $\boldsymbol{AB}$ is the vector from point $A$ to point $B$ and $\boldsymbol{AC}$ is the vector from point $A$ to point $C$. Write a user-defined MATLAB function that determines the area of a triangle given its vertices' coordinates. For the function name and arguments use `[Area] = TriArea(A,B,C)`. The input arguments `A`, `B`, and `C`, are vectors, each with the coordinates of the corresponding vertex. Write the code of `TriArea` such that it has two subfunctions—one that determines the vectors $\boldsymbol{AB}$ and $\boldsymbol{AC}$ and an other that executes the cross product. (If available, use the user-defined functions from Problems 15 and 17. The function should work for a triangle in the $x\,y$ plane (each vertex is defined by two coordinates) or for a triangle in space (each vertex is defined by three coordinates). Use the function to determine the areas of triangles with the following vertices:
(*a*) $A = (1, 2)$, $B = (10, 3)$, $C = (6, 11)$.
(*b*) $A = (-1.5, -4.2, -3)$, $B = (-5.1, 6.3, 2)$, $C = (12.1, 0, -0.5)$.

19. Write a user-defined function that plots a circle given the coordinates of the center and the radius. For the function name and arguments use `circleplot(x,y,R)`. The input arguments are the $x$ and $y$ coordinates of the center and the radius. This function has no output arguments. Use the function to plot the following circles:
(*a*) $x = 3.5$, $y = 2.0$, $R = 8.5$. (*b*) $x = -4.0$, $y = -1.5$, $R = 10$.

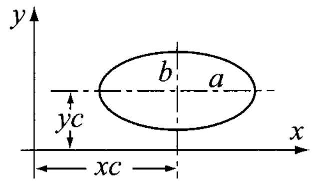
20. Write a user-defined function that plots a circle that passes through three given points. For the function name and arguments use `cirpnts(P)`. The input arguments is a $3 \times 2$ matrix in which the two elements of a row are the $x$ and $y$ coordinates of one point. This function has no output arguments. The figure that is created by the function displays the circle and the three points marked with asterisks. Use the function to plot a circle that passes through the points (6, 1.5), (2, 4), (–3, –1.8).

21. In polar coordinates a two-dimensional vector is given by its radius and angle $(r, \theta)$. Write a user-defined MATLAB function that adds two vectors that are given in polar coordinates. For the function name and arguments use `[r th]= AddVecPol(r1,th1,r2,th2)`, where the input arguments are $(r_1, \theta_1)$ and $(r_2, \theta_2)$, and the output arguments are the radius and angle of the result. Use the function to carry out the following additions:




(*a*) $r_1 = (5, 23°)$, $r_2 = (12, 40°)$. (*b*) $r_1 = (6, 80°)$, $r_2 = (15, 125°)$.

22. Write a user-defined function that plots an ellipse with axes that are parallel to the $x$ and $y$ axes, given the coordinates of its center and the length of the axes. For the function name and arguments use `ellipseplot(xc,yc,a,b)`. The input arguments `xc` and `yc` are the coordinates of the center, and `a` and `b` are half the lengths of the horizontal and vertical axes (see figure), respectively. This function has no output arguments. Use the function to plot the following ellipses:




(*a*) $xc = 3.5$, $yc = 2.0$, $a = 8.5$, $b = 3$.
(*b*) $xc = -5$, $yc = 1.5$, $a = 4$, $b = 8$.

23. Write a user-defined function that finds all the prime numbers between two numbers $m$ and $n$. Name the function `pr=prime(m,n)`, where the input arguments `m` and `n` are positive integers, and the output argument `pr` is a vector with the prime numbers. If $m > n$ is entered when the function is called, the error message "The value of $n$ must be larger than the value of $m$." is displayed. If a negative number or a number that is not an integer is entered when the function is called, the error message "The input argument must be a positive integer." is displayed. Use the function with:
(*a*) `prime(12,80)` (*b*) `prime(21,63.5)`
(*c*) `prime(100,200)` (*d*) `prime(90,50)`

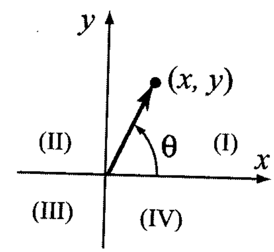
24. The geometric mean *GM* of a set of *n* positive numbers $x_1, x_2, \ldots, x_n$ is defined by:

$$GM = (x_1 \cdot x_2 \cdot \ldots \cdot x_n)^{1/n}$$

Write a user-defined function that calculates the geometric mean of a set of numbers. For function name and arguments use `GM=Geomean(x)`, where the input argument `x` is a vector of numbers (any length) and the output argument `GM` is their geometric mean. The geometric mean is useful for calculating the average return of a stock. The following table gives the returns for IBM stock over the last ten years (a return of 16% means 1.16). Use the user-defined function `Geomean` to calculate the average return of the stock.

| Year | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 |
|---|---|---|---|---|---|---|---|---|---|---|
| Return | 1.38 | 1.76 | 1.17 | 0.79 | 1.42 | 0.64 | 1.2 | 1.06 | 0.83 | 1.18 |

25. Write a user-defined function that determines the polar coordinates of a point from the Cartesian coordinates in a two-dimensional plane. For the function name and arguments use `[th rad]=CartToPolar(x,y)`. The input arguments are the *x* and *y* coordinates of the point, and the output arguments are the angle θ and the radial distance to the point. The angle θ is in degrees and is measured relative to the positive *x* axis, such that it is a positive number in quadrants I and II, and a negative number in quadrant III and IV. Use the function to determine the polar coordinates of points (14, 9), (–11, –20), (–15, 4), and (13.5, –23.5).




26. Write a user-defined function that sorts the elements of a vector from the largest to the smallest. For the function name and arguments use `y=downsort(x)`. The input to the function is a vector `x` of any length, and the output `y` is a vector in which the elements of `x` are arranged in a descending order. Do not use the MATLAB built-in function `sort`, `max`, or `min`. Test your function on a vector with 14 numbers (integers) randomly distributed between –30 and 30. Use the MATLAB `randi` function to generate the initial vector.

27. Write a user-defined function that sorts the elements of a matrix. For the function name and arguments use `B = matrixsort(A)`, where `A` is any size matrix and `B` is a matrix of the same size with the elements of `A` rearranged in descending order row after row with the (1,1) element the largest and the (*m*,*n*) element the smallest. If available, use the user-defined function `downsort` from Problem 26 as a subfunction within `matrixsort`.

Test your function on a $4 \times 7$ matrix with elements (integers) randomly distributed between –30 and 30. Use MATLAB's `randi` function to generate the initial matrix.

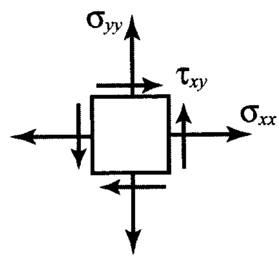
28. Write a user-defined MATLAB function that calculates the determinant of a $3 \times 3$ matrix by using the formula:

$$det = A_{11}\begin{vmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{vmatrix} - A_{12}\begin{vmatrix} A_{21} & A_{23} \\ A_{31} & A_{33} \end{vmatrix} + A_{13}\begin{vmatrix} A_{21} & A_{22} \\ A_{31} & A_{32} \end{vmatrix}$$

For the function name and arguments use `d3 = det3by3(A)`, where the input argument `A` is the matrix and the output argument `d3` is the value of the determinant. Write the code of `det3by3` such that it has a subfunction that calculates the $2 \times 2$ determinant. Use `det3by3` for calculating the determinants of:

(a) $\begin{bmatrix} 1 & 3 & 2 \\ 6 & 5 & 4 \\ 7 & 8 & 9 \end{bmatrix}$ (b) $\begin{bmatrix} -2.5 & 7 & 1 \\ 5 & -3 & -2.6 \\ 4 & 2 & -1 \end{bmatrix}$

29. A two-dimensional state of stress at a point in a loaded material is defined by three components of stress $\sigma_{xx}$, $\sigma_{yy}$, and $\tau_{xy}$. The maximum and minimum normal stresses (principal stresses) at the point, $\sigma_{max}$ and $\sigma_{min}$, are calculated from the stress components by:




$$\sigma_{\substack{max\\min}} = \frac{\sigma_{xx} + \sigma_{yy}}{2} \pm \sqrt{\left(\frac{\sigma_{xx} - \sigma_{yy}}{2}\right)^2 + \tau_{xy}^2}$$

Write a user-defined MATLAB function that determines the principal stresses from the stress components. For the function name and arguments use `[Smax,Smin] = princstress(Sxx,Syy,Sxy)`. The input arguments are the three stress components, and the output arguments are the maximum and minimum stresses.

Use the function to determine the principal stresses for the following states of stress:

(a) $\sigma_{xx} = -190\,\text{MPa}$, $\sigma_{yy} = 145\,\text{MPa}$, and $\tau_{xy} = 110\,\text{MPa}$.

(b) $\sigma_{xx} = 14\,\text{ksi}$, $\sigma_{yy} = -15\,\text{ksi}$, and $\tau_{xy} = 8\,\text{ksi}$.

30. The dew point temperature $T_d$ and the relative humidity *RH* can be calculated (approximately) from the dry-bulb $T$ and wet-bulb $T_w$ temperatures by (http://www.wikipedia.org)

$$e_s = 6.112\exp\left(\frac{17.67T}{T+243.5}\right) \qquad e_w = 6.112\exp\left(\frac{17.67T_w}{T_w+243.5}\right)$$

$$e = e_w - p_{sta}(T - T_w)0.00066(1 + 0.00115T_w)$$

$$RH = 100\frac{e}{e_s} \quad T_d = \frac{243.5\ln(e/6.112)}{17.67 - \ln(e/6.112)}$$

where the temperatures are in degrees Celsius, $RH$ is in %, and $p_{sta}$ is the barometric pressure in units of millibars.

Write a user-defined MATLAB function that calculates the dew point temperature and relative humidity for given dry-bulb and wet-bulb temperatures and barometric pressure. For the function name and arguments use `[Td,RH] = DewptRhum(T,Tw,BP)`, where the input arguments are $T$ , $T_w$ and $p_{sta}$, and the output arguments are $T_d$ and $RH$. The values of the output arguments should be rounded to the nearest tenth. Use the user-defined function `dewpoint` for calculating the dew point temperature and relative humidity for the following cases:

(*a*) $T = 25$ °C, $T_w = 19$ °C, $p_{sta} = 985$ mbar.

(*b*) $T = 36$ °C, $T_w = 31$ °C, $p_{sta} = 1020$ mbar.

31. Write a user-defined MATLAB function that calculates a student's final grade in a course using the scores from three midterm exams, a final exam, and six homework assignments. The midterms are graded on a scale from 0 to 100 and each accounts for 15% of the course grade. The final exam is graded on a scale from 0 to 100 and accounts for 40% of the course grade. The six homework assignments are each graded on a scale from 0 to 10. The homework assignment with the lowest grade is dropped, and the average of the remaining assignments accounts for 15% of the course grade. In addition, the following adjustment is made when the grade is calculated. If the average grade for the three midterms is higher than the grade for the final exam, then the grade of the final exam is not used and the average grade of the three midterms accounts for 85% of the course grade. The program calculates a course grade that is a number between 0 and 100.

    For the function name and arguments use `g = fgrade(R)`. The input argument `R` is a matrix in which the elements in each row are the grades of one student. The first six columns are the homework grades (numbers between 0 and 10), the next three columns are the midterm grades (numbers between 0 and 100), and the last column is the final exam grade (a number between 0 and 100). The output from the function, `g`, is a column vector with the student grades for the course. Each row has the course grade of the student with the grades in the corresponding row of the matrix `R`.

    The function can be used to calculate the grades of any number of students. For one student the matrix `R` has one row. Use the function for the following cases:

    (*a*) Use the Command Window to calculate the course grade of one student with the following grades: 8, 9, 6, 10, 9, 7, 76, 86, 91, 80.

(b) Write a program in a script file. The program asks the user to enter the students' grades in an array (one student per row). The program then calculates the course grades by using the function `fgrade`. Run the script file in the Command Window to calculate the grades of the following four students:

Student A: 7, 10, 6, 9, 10, 9, 91, 71, 81, 88.
Student B: 5, 5, 6, 1, 8, 6, 59, 72, 66, 59.
Student C: 6, 8, 10, 4, 5, 9, 72, 78, 84 78.
Student D: 7, 7, 8, 8, 9, 8, 83, 82, 81 84.

32. In a lottery the player has to select several numbers out of a list. Write a MATLAB program that generates a list of $n$ integers that are uniformly distributed between the numbers $a$ and $b$. All the selected numbers on the list must be different.

(*a*) Use the function to generate a list of seven numbers from the numbers 1 through 59.

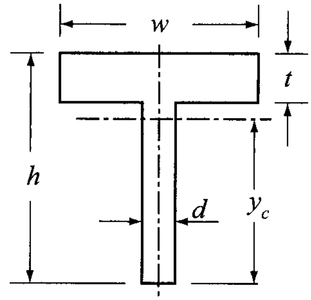
(*b*) Use the function to generate a list of eight numbers from the numbers 50 through 65.

(*c*) Use the function to generate a list of nine numbers from the numbers –25 through –2.

33. The solution of the nonlinear equation $x^5 - P = 0$ gives the fifth root of the number $P$. A numerical solution of the equation can be calculated with Newton's method. The solution process starts by choosing a value $x_1$ as a first estimate of the solution. Using this value, a second, more accurate solution $x_2$ can be calculated with $x_2 = x_1 - \frac{x_1^5 - P}{5x_1^4}$, which is then used for calculating a third, still more accurate solution $x_3$, and so on. The general equation for calculating the value of the solution $x_{i+1}$ from the solution $x_i$ is $x_{i+1} = x_i - \frac{x_i^5 - P}{5x_i^4}$. Write a user-defined function that calculates the fifth root of a number. For function name and arguments use `y=fifthroot(P)`, where the input argument `P` is the number whose fifth root is to be determined, and the output argument `y` is the value $\sqrt[5]{P}$. In the program use $x = P$ for the first estimate of the solution. Then, by using the general equation in a loop, calculate new, more accurate solutions. Stop the looping when the estimated relative error $E$ defined by $E = \left|\frac{x_{i+1} - x_i}{x_i}\right|$ is smaller than 0.00001. Use the function `cubic` to calculate:

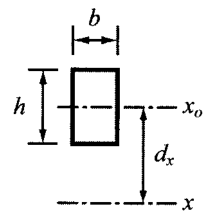
(*a*) $\sqrt[5]{120}$ (*b*) $\sqrt[5]{16807}$ (*c*) $\sqrt[5]{-15}$

34. Write a user-defined function that determines the coordinate $y_c$ of the centroid of the T-shaped cross-sectional area shown in the figure. For the function name and arguments use `yc = centroidT(w,h,t,d)`, where the input arguments `w`, `h`, `t` and `d`, are the dimensions shown in the figure, and the output argument `yc` is the coordinate $y_c$.

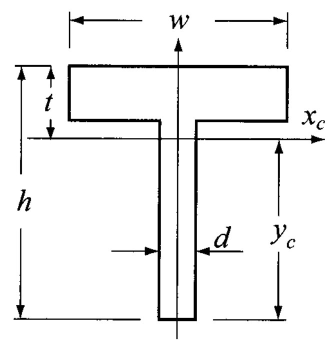



Use the function to determine $y_c$ for an area with $w = 240$ mm, $h = 380$ mm, $d = 42$ mm, and $t = 60$ mm.

35. The area moment of inertia $I_{x_o}$ of a rectangle about the axis $x_o$ passing through its centroid is $I_{x_o} = \frac{1}{12}bh^3$. The moment of inertia about an axis $x$ that is parallel to $x_o$ is given by $I_x = I_{x_o} + Ad_x^2$, where $A$ is the area of the rectangle, and $d_x$ is the distance between the two axes.




Write a MATLAB user-defined function that determines the area moment of inertia $I_{x_c}$ of a "T" beam about the axis that passes through its centroid (see drawing). For the function name and arguments use `Ixc = IxcTBeam(w,h,t,d)`, where the input arguments `w`, `h`, `t`, and `d` are the dimensions shown in the figure, and the output argument `Ixc` is $I_{x_c}$. For finding the coordinate $y_c$ of the of the centroid use the user-defined function `centroidT` from Problem 34 as a subfunction inside `IxcTBeam`.

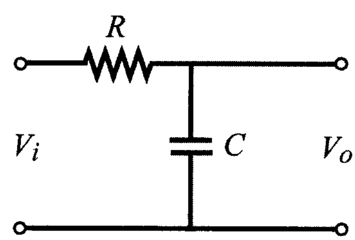



(The moment of inertia of a composite area is obtained by dividing the area into parts and adding the moments of inertia of the parts.)

Use the function to determine the moment of inertia of a "T" beam with $w = 240$ mm, $h = 380$ mm, $d = 42$ mm, and $t = 60$ mm.

36. In a low-pass $RC$ filter (a filter that passes signals with low frequencies), the ratio of the magnitudes of the voltages is given by:

$$RV = \left|\frac{V_o}{V_i}\right| = \frac{1}{\sqrt{1 + (\omega RC)^2}}$$



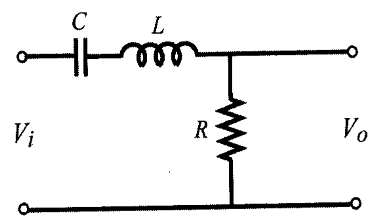

where $\omega$ is the frequency of the input signal.

Write a user-defined MATLAB function that calculates the magnitude ratio. For the function name and arguments use `RV = lowpass(R,C,w)`. The input arguments are `R`, the size of the resistor in Ω (ohms); `C`, the size of the capacitor in F (farads); and `w`, the frequency of the input signal in rad/s. Write the function such that `w` can be a vector.

Write a program in a script file that uses the `lowpass` function to generate a plot of $RV$ as a function of $\omega$ for $10^{-2} \leq \omega \leq 10^{6}$ rad/s. The plot has a logarithmic scale on the horizontal axis ($\omega$). When the script file is executed, it asks the user to enter the values of $R$ and $C$. Label the axes of the plot.

Run the script file with $R = 1200\,\Omega$, and $C = 8\,\mu\text{F}$.

37. A bandpass filter passes signals with frequencies that are within a certain range. In this filter the ratio of the magnitudes of the voltages is given by




$$RV = \left|\frac{V_o}{V_i}\right| = \frac{\omega RC}{\sqrt{(1-\omega^2 LC)^2 + (\omega RC)^2}}$$

where $\omega$ is the frequency of the input signal.

Write a user-defined MATLAB function that calculates the magnitude ratio. For the function name and arguments use `RV = bandpass(R,C,L,w)`. The input arguments are `R` the size of the resistor in Ω (ohms); `C`, the size of the capacitor in F (farads); `L`, the inductance of the coil in H (henrys); and `w`, the frequency of the input signal in rad/s. Write the function such that `w` can be a vector.

Write a program in a script file that uses the `bandpass` function to generate a plot of $RV$ as a function of $\omega$ for $10^{-2} \leq \omega \leq 10^{7}$ rad/s. The plot has a logarithmic scale on the horizontal axis ($\omega$). When the script file is executed, it asks the user to enter the values of $R$, $L$, and $C$. Label the axes of the plot.

Run the script file for the following two cases:

(*a*) $R = 1100\,\Omega$, $C = 9\ \mu\text{F}$, $L = 7$ mH.
(*b*) $R = 500\,\Omega$, $C = 300\ \mu\text{F}$, $L = 400$ mH.

38. The first derivative $\frac{df(x)}{dx}$ of a function $f(x)$ at $x = x_0$ can be approximated with the four-point central difference formula

$$\frac{df(x)}{dx} = \frac{f(x_0 - 2h) - f(x_0 - h) + f(x_0 + h) - f(x_0 + 2h)}{12h}$$

where $h$ is a small number relative to $x_0$. Write a user-defined function function (see Section 7.9) that calculates the derivative of a math function $f(x)$ by using the four-point central difference formula. For the user-defined function name use `dfdx=Funder(Fun,x0)`, where `Fun` is a name for the function that is passed into `Funder`, and `x0` is the point where the derivative is calculated. Use $h = x_0/10$ in the four-point central difference formula. Use the user-defined function `Funder` to calculate the following:

(*a*) The derivative of $f(x) = x^2e^x$ at $x_0 = 0.25$.

(*b*) The derivative of $f(x) = \frac{2^x}{x}$ at $x_0 = 2$.

In both cases compare the answer obtained from `Funder` with the analytical solution (use format long).

39. The new coordinates $(X_r, Y_r)$ of a point in the $x\,y$ plane that is rotated about the $z$ axis at an angle $\theta$ (positive is clockwise) are given by

$$X_r = X_0\cos\theta - Y_0\sin\theta$$
$$Y_r = X_0\sin\theta + Y_0\cos\theta$$

where $(X_0, Y_0)$ are the coordinates of the point before the rotation. Write a user-defined function that calculates $(X_r, Y_r)$ given $(X_0, Y_0)$ and $\theta$. For function name and arguments use `[xr,yr]=rotation(x,y,q)`, where the input arguments are the initial coordinates and the rotation angle in degrees, and the output arguments are the new coordinates.

(*a*) Use `rotation` to determine the new coordinates of a point originally at $(6.5, 2.1)$ that is rotated about the $z$-axis by $25°$.

(*b*) Consider the function $y = (x-7)^2 + 1.5$ for $5 \le x \le 9$. Write a program in a script file that makes a plot of the function. Then use `rotation` to rotate all the points that make up the first plot and make a plot of the rotated function. Make both plots in the same figure and set the range of both axes at 0 to 10.