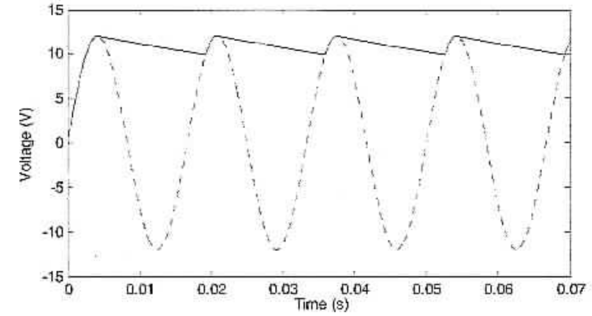
diode is off. The `switch` command is used for switching between the two parts. The calculations start with the diode on (the variable `state='on'`), and when $i_R - i_C \leq 0$ the value of `state` is changed to `'off'`, and the program switches to the commands that calculate v_R for this state. These calculations continue until $v_s \geq v_R$ when the program switches back to the equations that are valid when the diode is on.

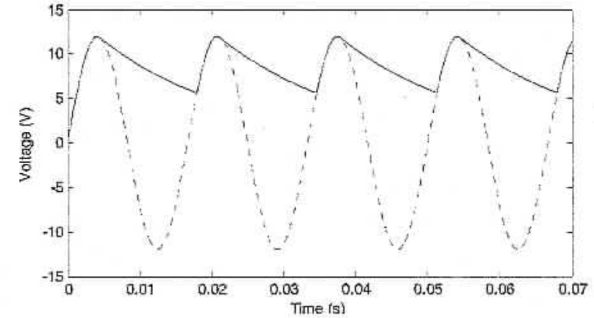
```
V0=12; C=45e-6; R=1800; f=60;
Tf=70e-3; w=2*pi*f;
clear t VR Vs
t=0:0.05e-3:Tf;
n=length(t);
state='on'                          Assign 'on' to the variable state.
for i=1:n
    Vs(i)=V0*sin(w*t(i));           Calculate the voltage of the source at time t.
    switch state
       case 'on'                    Diode is on.
          VR(i)=Vs(i);
          iR=Vs(i)/R;
          iC=w*C*V0*cos(w*t(i));
          sumI=iR+iC;
          if sumI <= 0              Check if iR - iC <= 0.
             state='off';           If true, assign 'off' to state.
             tA=t(i);               Assign value to tA.
          end
       case 'off'                   Diode is off.
          VR(i)=V0*sin(w*tA)*exp(-(t(i)-tA)/(R*C));
          if Vs(i) >= VR(i)         Check if vs >= vR.
             state='on';            If true, assign
          end                       'on' to the
    end                             variable state.
end
plot(t,Vs,':',t,VR,'k','linewidth',1)
xlabel('Time (s)'); ylabel('Voltage (V)')
```

The two plots generated by the program are shown below. One plot shows the result with C = 45 μF, and the other with C = 10 μF. It can be observed that with

larger capacitor the DC voltage is smoother (smaller ripple in the wave).



C = 45 μF



C = 10 μF

7.8 PROBLEMS

1. Evaluate the following expressions without using MATLAB. Check the answer with MATLAB.
   a) $14 > 15/3$
   b) $y = 8/2 < 5 \times 3 + 1 > 9$
   c) $y = 8/(2 < 5) \times 3 + (1 > 9)$
   d) $2 + 4 \times 3 \sim= 60/4 - 1$

   No spaces allowed! They must be connected,

2. Given: $a = 4,\ b = 7$. Evaluate the following expressions without using MATLAB. Check the answer with MATLAB.
   a) $y = a + b >= a \times b$
   b) $y = a + (b > = a) \times b$
   c) $y = b - a < a < a/b$

3. Given: $v = [4\ -2\ -1\ 5\ 0\ 1\ -3\ 8\ 2]$ and $w = [0\ 2\ 1\ -1\ 0\ -2\ 4\ 3\ 2]$. Evaluate the following expressions without using MATLAB. Check the answer with MATLAB.
   a) $v <= w$ *b)* $w = v$
   c) $v < w + v$ *d)* $(v < w) + v$

4. Use the vectors v and w from the previous problem. Use relational operators to create a vector y that is made up from the elements of w that are smaller than the elements of v.

5. Evaluate the following expressions without using MATLAB. Check the answer with MATLAB.
   a) -3&0 *b)* 4<-1&5>0
   c) 8-12|6+5&~-2 *d)* ~4&0+8*~(4|0)

6. The maximum daily temperature (in °F) for New York City and Anchorage, Alaska during the month of January, 2001 are given in the vectors below (data from the U.S. National Oceanic and Atmospheric Administration).
   TNY = [31 26 30 33 33 39 41 41 34 33 45 42 36 39 37 45 43 36 41 37 32 32 35 42 38 33 40 37 36 51 50]
   TANC = [37 24 28 25 21 28 46 37 36 20 24 31 34 40 43 36 34 41 42 35 38 36 35 33 42 42 37 26 20 25 31]
   Write a program in a script file to answer the following:
   a) Calculate the average temperature for the month in each city.
   b) How many days was the temperature below the average in each city?
   c) How many days, and which dates in the month, was the temperature in Anchorage higher then the temperature in New York?
   d) How many days, and which dates in the month, was the temperature the same in both cities?
   e) How many days, and which dates in the month, was the temperature in both cities above freezing (above 32 °F)?

7. Use MATLAB in the two different ways described below to plot the function:

$$f(x) = \begin{cases} 15 & \text{for} \quad x \le -1 \\ -5x+10 & \text{for} \quad -1 \le x \le 1 \\ -10x^2+35x-20 & \text{for} \quad 1 \le x \le 3 \\ -5x+10 & \text{for} \quad 3 \le x \le 4 \\ -10 & \text{for} \quad x \ge 4 \end{cases}$$

   in the domain $-2 \le x \le 5$.
   a) Write a program in a script file, using conditional statements and loops.
   b) Create a user-defined function for $f(x)$, and then use the function in a script file to make the plot.

8. Use loops to create a 3×5 matrix in which the value of each element is the difference between the indices divided by the sum of its indices (the row number and column number of the element). For example, the value of the element (2,5) is $(2-5)/(2+5) = -0.4286$.

9. Write a program in a script file that determines the real roots of a quadratic equation $ax^2+bx+c = 0$. Name the file `quadroots`. When the file runs it asks the user to enter the values of the constants a, b, and c. To calculate the roots of the equation the program calculates the discriminant D given by:

$$D = b^2 - 4ac$$

   If $D > 0$ the program displays a message: "The equation has two roots," and the roots are displayed in the next line.
   If $D = 0$ the program displays a message: "The equation has one root," and the root is displayed in the next line.
   If $D < 0$ the program displays a message: "The equation has no real roots."
   Run the script file in the Command Window three times to obtain solutions to the following three equations:
   a) $-2x^2+16x-32 = 0$
   b) $8x^2+9x+3 = 0$
   c) $3x^2+5x-6 = 0$

10. Write a program (using a loop) that determines the expression:

$$\left(12\sum_{n=1}^{m}\frac{(-1)^n}{n^2}\right)^{\frac{1}{2}} \qquad (n = 1, 2,..., m)$$

   Should be (n-1). This is a typo

   Run the program with $m = 10$, $m = 1000$, and $m = 10{,}000$. Compare the result with π.

11. A vector is given by: $x = [15\ -6\ 0\ 8\ -2\ 5\ 4\ -10\ 0.5\ 3]$. Using conditional statements and loops, write a program that determines and displays the sum of the positive elements of the vector, and the sum of the negative elements of the vector.

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

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

   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 average mean is useful for calculating the average return of a stock. The following table gives the return of an IBM stock in the last ten years (a return of 16% means 1.16 etc.). 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

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 calculated, and the output argument `y` is the value $x!$. The function displays an error message if a negative or a 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. The Taylor's series expansion for $\cos(x)$ is:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{(2n)}$$

where x is in radians. Write a user-defined function that determines $\cos(x)$ using Taylor's series expansion. For function name and arguments use `y=cosTaylor(x)`, where the input argument `x` is the angle in degrees and the output argument `y` is the value for $\cos(x)$. In the program of the user-defined function use a loop for adding the terms of the Taylor's series. If a_n is the nth term in the series, than the sum S_n of the n terms is $S_n = S_{n-1} + a_n$. In each pass calculate the estimated error E given by: $E = \left|\frac{S_n - S_{n-1}}{S_{n-1}}\right|$. Stop adding terms when $E \le 0.000001$. If you have a solution for Problem 13, insert it as a subfunction inside `cosTaylor`, and use it for calculating the factorial term in the equation (otherwise use MATLAB's built-in function `factorial`).
Use `sinTaylor` for calculating: *a)* $\cos(55°)$ *b)* $\cos(190°)$. Compare the values with the values obtained by using a calculator.

You mean to say cosine Taylor

15. Write a program in a script file that finds the smallest even integer that is also divisible by 7 and whose cube is greater than 40,000. Use a loop in the program. The loop should start from 1 and stop when the number is found. The program prints a message: "The required number is:" and then prints the number.

16. Write a user-defined function that finds all the prime numbers between 1 and n. Name the function `pr=prime(n)`, where the input argument `n` is a positive integer, and the output argument `pr` is a vector with the prime numbers. If a negative number or a number that is not an integer is entered when the function is called, an error message "The input argument must be a positive integer." is displayed. Use the function with:
a) `prime(30)`. *b)* `prime(52.5)`.
c) `prime(79)`. *d)* `prime(-20)`.

17. Write a user-defined function that sorts the elements of a vector (of any length) 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 functions `sort`, `max`, or `min`. Test your function on a vector with 14 numbers (integers) randomly distributed between –30 and 30. Use the MATLAB `rand` function to generate the initial vector.

18. 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 where the (1,1) element is the largest, and the (m,n) element is the smallest. Use the user-defined function `downsort` from Problem 17 as a subfunction within `matrixsort`.
Test your function on a 4×7 matrix with elements (integers) randomly distributed between –30 and 30. Use the MATLAB `rand` function to generate the initial matrix.

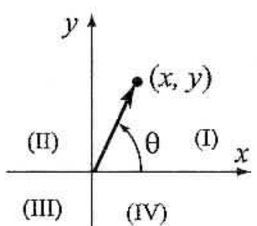
19. The Karvonen formula is one method for calculating the training heart rate (*THR*).

$$THR = [(220 - AGE) - RHR] \times INTEN + RHR$$

where *AGE* is the age, *RHR* the rest heart rate, and *INTEN* is the fitness level (0.55 for low, 0.65 for medium, and 0.8 for high fitness level). Write a program in a script file that determines the *THR*. The program asks the user to enter his/her age (number), rest heart rate (number), and fitness level (low, medium, or high). The program then displays the training heart rate.

20. 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 `[theta radius]=CartesianToPolar(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, II, and III, and a negative number in the IV quadrant. Use the function to determine the polar coordinates of points (15, 3), (–7, 12), (–17, –9), and (10, –6.5).

neg in



21. Write a program in a script file that calculates the cost of renting a car according to the following price schedule:

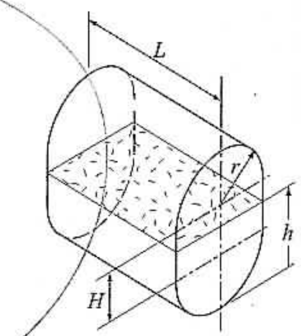
Type of car	Rental period		
	1-6 days	**7-27 days**	**28-60 days**
Class B	\$27 per day	\$162 for 7 days, + \$25 for each additional day.	\$662 for 28 days, + \$23 for each additional day.
Class C	\$34 per day	\$204 for 7 days, + \$31 for each additional day.	\$284 for 28 days, + \$28 for each additional day.
Class D	Class D cannot be rented for less than 6 days.	\$276 for 7 days, + \$43 for each additional day.	\$1,136 for 28 days, + \$38 for each additional day.

The program asks the user to enter the rental period and the type of car. The program then displays the cost. If a period longer than 60 days is entered, a message "Rental is not available for more than 60 days" is displayed. If a rental period of less than 6 days is entered for class D, a message "Class D cars cannot be rented for less than 6 days" is displayed.

Run the program nine times for the following cases:
Class B, for 3, 14, and 50 days. Class C, for 20, 28 and 61 days. Class D for 6, 18, and 60 days.

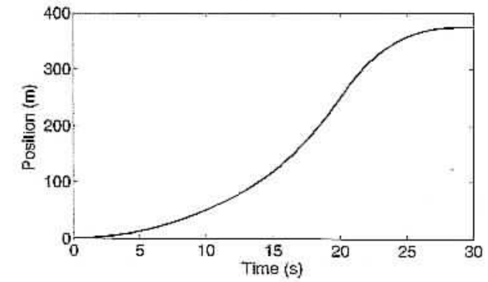
22. A fuel tank is made of a rectangular prism center and half circular cylinders at the top and bottom as shown, where $r = 20$ cm, $H = 15$ cm, and $L = 60$ cm.

Write a user-defined function (for the function name and arguments use `V = Volfuel(h)`) that gives the volume of the fuel in the tank (in liters) 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 $0 \le h \le 55$ cm.



this was hard

23. The position x as a function of time, of a particle that moves along a straight line is shown in the figure, and is given by the following equations:



$$x(t) = \begin{cases} 0.5t^2 \text{ m} & \text{for } 0 \le t \le 10 \text{ s} \\ 0.05t^3 - t^2 + 15t - 50 \text{ m} & \text{for } 10 \le t \le 20 \text{ s} \\ 0.0025t^4 - 0.15t^3 + 135t - 1650 \text{ m} & \text{for } 20 \le t \le 30 \text{ s} \end{cases}$$

Write three user-defined functions: one calculates the position of the particle at time t (for the function name and arguments use `x = position(t)`). The second function calculates the velocity of the particle at time t (for the function name and arguments use `v = velocity(t)`), and the third function calculates the acceleration of the particle at time t (for the function name and arguments use `a = acceleration(t)`). In a script file, write a program that creates plots of the position, velocity, and acceleration as function of time (three plots on the same page). In the program, first create a vector t, $0 \le t \le 30$ s, and then use the functions `position`, `velocity`, and `acceleration` to create vectors of velocity and acceleration that are used for the plots.

24. Write a program that determines the change given back to a customer in a self-service checkout machine of a supermarket for purchases of up to \$10. The program generates a random number between 0.01 and 10.00 and displays the number as the amount to be paid. The program then asks the user to enter his payment, which can be one \$1 bill, one \$5 bill, or one \$10. If the payment is less than the amount to be paid, an error message is displayed. If the payment is sufficient, the program calculates the change and lists the bills and/or the coins that make up the change, which has to be composed of the least number of bills and coins. For example, if the amount to be paid is \$2.33, and a \$10 bill is entered as payment, then the change is: 1 \$5 bill, 2 \$1 bills, 2 quarters, 1 dime, 1 nickel, and 2 cents.

Write the main program in a script file, and write two user-defined functions that are used in the script file. One user-defined function generates a random number between 0.01 and 10.00. The other user-defined function calculates the composition of the change.

25. The concentration of a drug in the body C_P can be modeled by the equation:

$$Cp = \frac{D_G}{V_d}\frac{k_a}{(k_a - k_e)}(e^{-k_e t} - e^{-k_a t})$$

where D_G is the dosage administrated (mg), V_d is the volume of distribution (L), k_a is the absorption rate constant (h^{-1}), k_e is the elimination rate constant (h^{-1}), and t is the time (h) since the drug was administered. For a certain drug, the following quantities are given: $D_G = 150$ mg, $V_d = 50$ L, $k_a = 1.6\ h^{-1}$, and $k_e = 0.4\ h^{-1}$.

a) A single dose is administered at $t = 0$. Calculate and plot C_P versus t for 10 hours.

a) A first dose is administered at $t = 0$, and subsequently four more doses are administered at intervals of 4 hours (i.e. at $t = 4, 8, 12, 16$). Calculate and plot C_P versus t for 24 hours.

26. The solution of the nonlinear equation $x^3 - P = 0$ gives the cubic 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 $x_2 = x_1 - \frac{x_1^3 - P}{3x_1^2}$, which is then used for calculating a third, 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: $x_{i+1} = x_i - \frac{x_i^3 - P}{3x_i^2}$. Write a user-defined function that calculates the cubic root of a number. For function name and arguments use `y=cubic(P)`, where the input argument `P` is the number whose cubic root is to be determined, and the output argument `y` is the value $\sqrt[3]{P}$. 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) $\sqrt[3]{100}$ *b)* $\sqrt[3]{9261}$ *c)* $\sqrt[3]{-70}$

Chapter 8

Polynomials, Curve Fitting, and Interpolation

Polynomials are mathematical expressions that are frequently used for problem solving and modeling in science and engineering. In many cases an equation that is written in the process of solving a problem is a polynomial, and the solution of the problem is the zero of the polynomial. MATLAB has a wide selection of functions that are specifically designed for handling polynomials. How to use polynomials in MATLAB is described in Section 8.1.

Curve fitting is a process of finding a function that can be used to model data. The function does not necessarily pass through any of the points, but models the data with the smallest possible error. There are no limitations to the type of the equations that can be used for curve fitting. Often, however, polynomial, exponential, and power functions are used. In MATLAB curve fitting can be done by writing a program, or by interactively analyzing data that is displayed in the Figure Window. Section 8.2 describes how to use MATLAB programming for curve fitting with polynomials and other functions. Section 8.4 describes the basic fitting interface which is used for interactive curve fitting and interpolation.

Interpolation is the process of estimating values between data points. The simplest interpolation is done by drawing a straight line between the points. In a more sophisticated interpolation, data from additional points is used. How to interpolate with MATLAB is discussed in Sections 8.3 and 8.4.

8.1 POLYNOMIALS

Polynomials are functions that have the form:

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$$

The coefficients $a_n, a_{n-1}, \ldots, a_1, a_0$ are real numbers, and *n*, which is a nonnega-