Scilab Textbook Companion for Trigonometry by M. Corral[1]

Created by
S Sai Ashrith Reddy
B.Tech
Electrical Engineering
NITK
College Teacher
None
Cross-Checked by
Chaitanya Potti

July 31, 2019

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Trigonometry

Author: M. Corral

Publisher: Createspace Independent Publishing North Charlestn

Edition: 2

Year: 2010

ISBN: 9781475074574

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

List of Figures

Chapter 1

Right Triangle Trigonometry

Scilab code Exa 1.1 To determine unknown angle in 3 given triangles

```
1 clc,clear
2 //example 1.1
3 //To determine unknown angle in 3 given triangles
4
5 //Triangle ABC
6 A = 35//angle at vertex A in degrees
7 C = 20//angle at vertex C in degrees
8 B=180- (A+C) //unknown angle
9 printf('Triangle ABC: B = %.0f degree\n',B)
10
11 //Triangle DEF
```

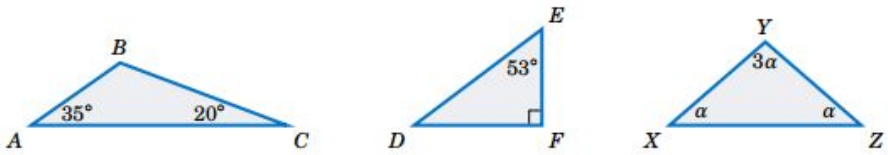

Figure 1.1: To determine unknown angle in 3 given triangles

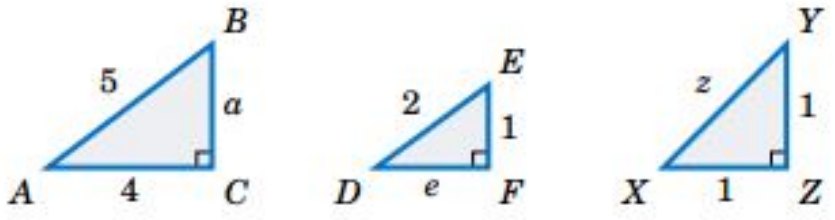

Figure 1.2: To determine length of unknown side in 3 given right triangles

```
12 E = 53//angle at vertex E in degree
13 //F = 90, DEF is right triangle
14 //So angles E and D are complimentary
15 D = 90- E //unknown angle
16 printf(' Triangle DEF: D = %.0f degree\n',D)
17
18 //Triangle XYZ
19 sum_multiple= 1+3+1 //for solvong for alpha
20 alpha = 180/ sum_multiple
21 X= alpha //unknown angle
22 Y= 3* alpha //unknown angle
23 Z= alpha //unknown angle
24 printf(' Triangle XYZ: X=%.0f degree Y=%.0f degree Z
     =%.0f degree',X,Y,Z)
```

Scilab code Exa 1.3 To determine length of unknown side in 3 given right triangles

```
1 clc,clear
2 //example 1.3
3 //To determine length of unknown side in 3 given
     right triangles
4
5 //Triangle ABC
6 AB=5 //given
```

```
7 AC=4 //given
8 a=sqrt(AB^2- AC^2) //by pythagoras theorem
9 printf('Triangle ABC: a=%f units \n',a)
10
11 //Triangle DEF
12 DE=2 //given
13 EF=1 //given
14 e=sqrt(DE^2- EF^2) //by pythagoras theorem
15 printf(' Triangle DEF: e=%f units = sqrt(%f) units\n
    ',e,e^2)
16
17 //Triangle XYZ
18 XZ=1 //given
19 YZ=1 //given
20 z=sqrt(XZ^2+YZ^2)//by pythagoras theorem
21 printf(' Triangle XYZ: z=%f units = sqrt(%f) units\n
    ',z,z^2)
```

Scilab code Exa 1.4 To determine height of the top of ladder touching the wall

```
1 clc,clear
2 //example 1.4
3 //To determine height of the top of ladder touching
    the wall
4
5 ladder = 17 //length of ladder or hypotenuse in feet
6 base = 8 //distance between lower tip of ladder and
    wall in feet
7 //Using pythagoras theorem
8 h=sqrt(ladder^2 - base^2) //required height
9 printf('Required height of top of ladder in contact
    with wall  = %.0f ft',h)
```

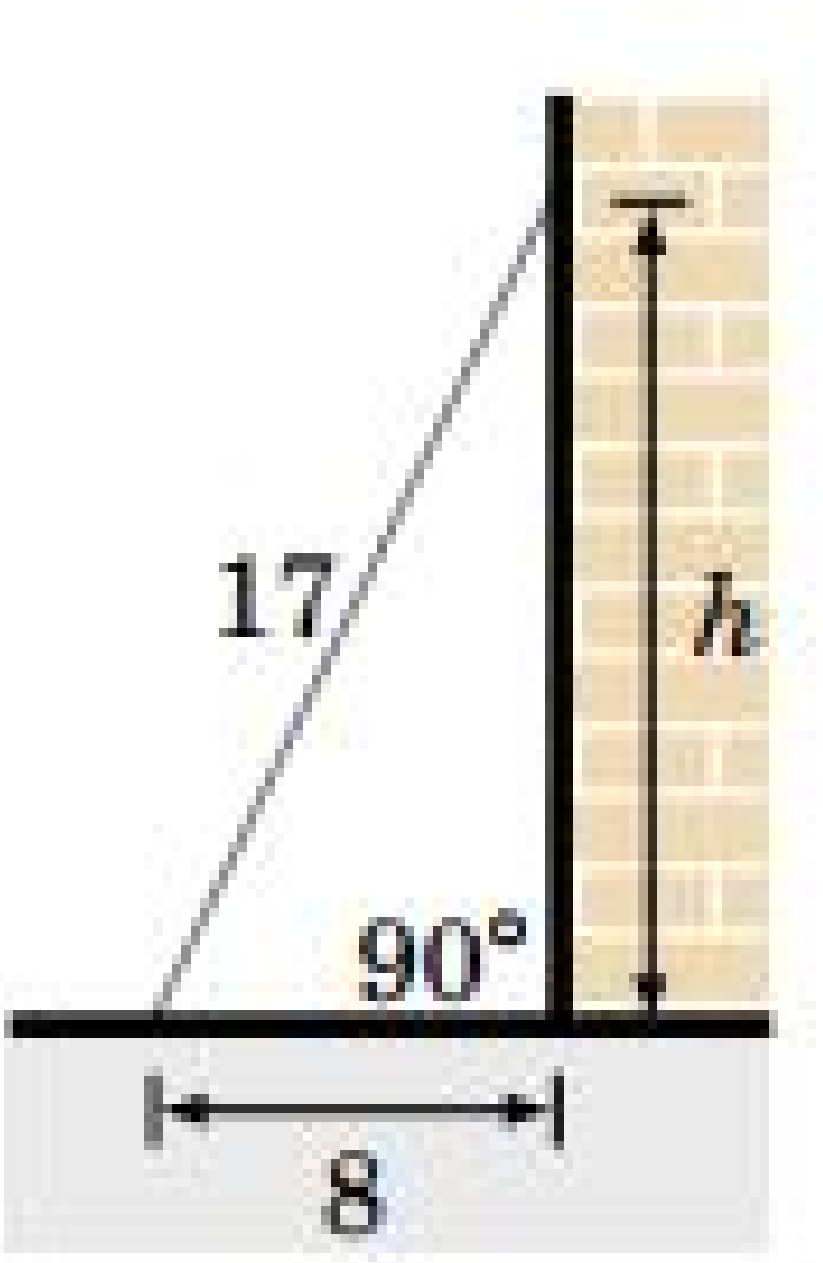

Figure 1.3: To determine height of the top of ladder touching the wall

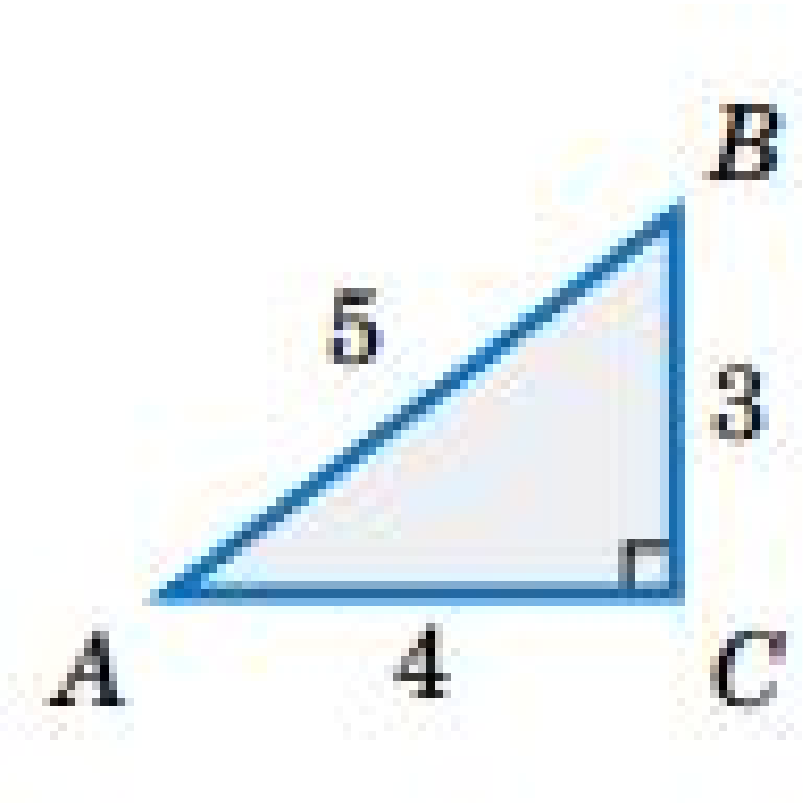

Figure 1.4: To find values of all trigonometric functions for angles A and B

Scilab code Exa 1.5 To find values of all trigonometric functions for angles A and

```
1 clc,clear
2 //example 1.5
3 //To find values of all trigonometric functions for
     angles A and B
4
5 //Angle at vertex A
```

```
6  opposite = 3;
7  adjacent = 4;
8  hypotenuse=5;
9
10 sin_A = opposite / hypotenuse;
11 cos_A = adjacent / hypotenuse;
12 tan_A = opposite / adjacent;
13 csc_A = hypotenuse/opposite;
14 sec_A = hypotenuse/adjacent;
15 cot_A = adjacent / opposite;
16 printf('ANGLE A')
17 printf('\nsin(A)= %.1f ; cos(A)= %.2f; tan(A)=
     %.2f;\n',sin_A,cos_A,tan_A)
18 printf('csc(A)= %.3f ; sec(A)= %.2f; cot(A)= %.2f
     ;',csc_A,sec_A,cot_A)
19
20 //Angle at vertex B
21 opposite = 4;
22 adjacent = 3;
23 hypotenuse=5;
24
25 sin_B = opposite / hypotenuse;
26 cos_B = adjacent / hypotenuse;
27 tan_B = opposite / adjacent;
28 csc_B = hypotenuse/opposite;
29 sec_B = hypotenuse/adjacent;
30 cot_B = adjacent / opposite;
31 printf('\n\nANGLE B')
32 printf('\nsin(B)= %.1f ; cos(B)= %.2f; tan(B)=
     %.2f;\n',sin_B,cos_B,tan_B)
33 printf('csc(B)= %.2f ; sec(B)= %.2f; cot(B)= %.2
     f;',csc_B,sec_B,cot_B)
```

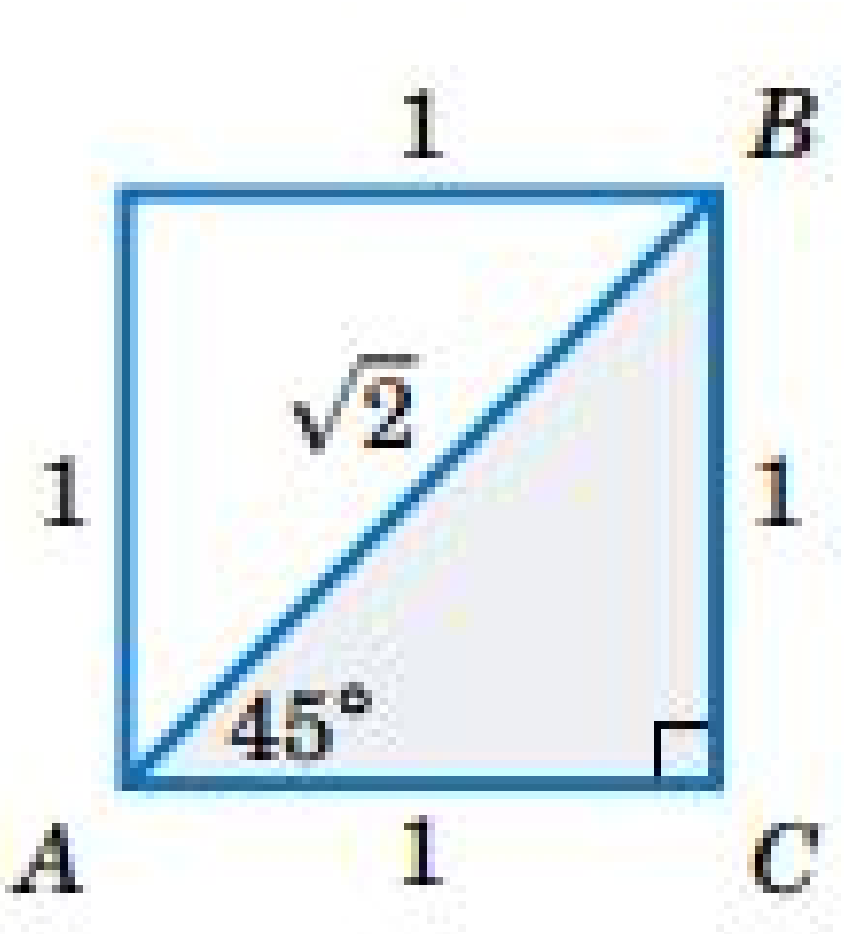

Figure 1.5: To find values of all trigonometric functions for 45 degree

Scilab code Exa 1.6 To find values of all trigonometric functions for 45 degree

```
clc,clear
//example 1.6
//To find values of all trigonometric functions for
   45 degree

//Consider a square of side 1 and divide it half
   diagonally
//ABC is now an isosceles triangle
//angle A and B are now equal and = 45 degree

AC=1;
BC=1;
AB=sqrt(AC^2+BC^2) //by pythagoras theorem
c=AB //we denote AB by c as its opposite to C

//conside angle BAC=45 degree
opposite = BC;
adjacent = AC;
hypotenuse = c;
sin_45 = opposite / hypotenuse;
cos_45 = adjacent / hypotenuse;
tan_45 = opposite / adjacent;
csc_45 = hypotenuse/opposite;
sec_45 = hypotenuse/adjacent;
cot_45 = adjacent / opposite;

printf('ANGLE = 45 degree')
printf('\nsin(45)= %.4f ; cos(45)= %.4f;  tan(45)=
   %.2f;\n',sin_45,cos_45,tan_45)
printf('csc(45)= %.4f ; sec(45)= %.4f;  cot(45)= %
   .2f;',csc_45,sec_45,cot_45)
```

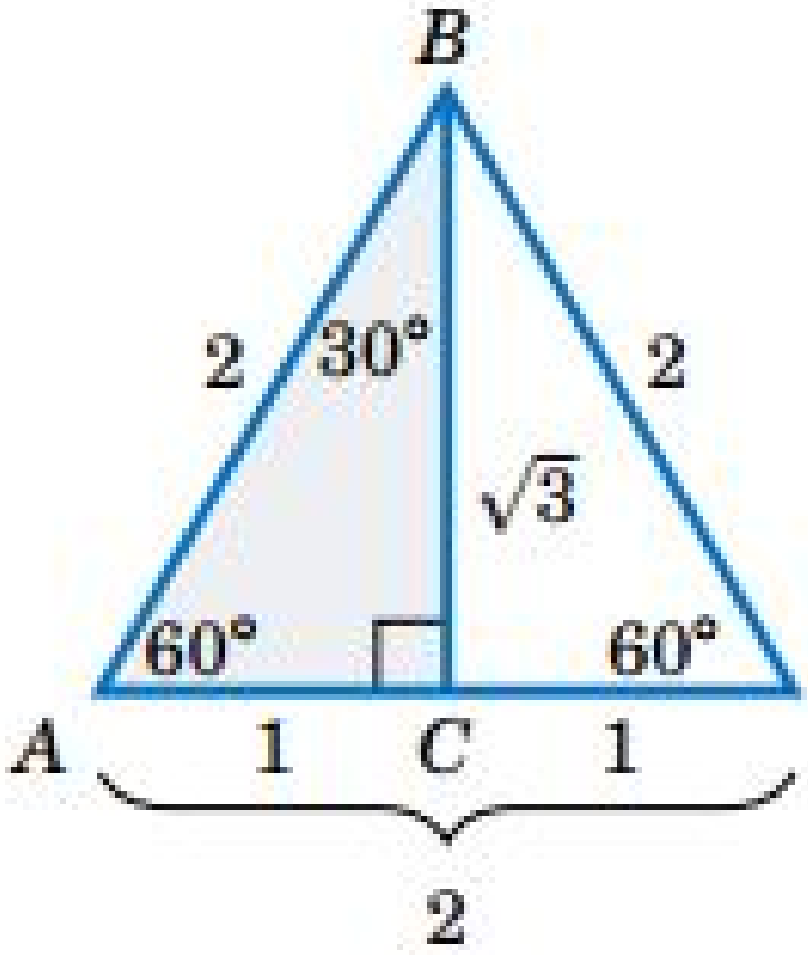

Figure 1.6: To find values of all trigonometric functions for 60 degree

Scilab code Exa 1.7 To find values of all trigonometric functions for 60 degree

```
clc,clear
//example 1.7
//To find values of all trigonometric functions for
   60 degree

//take an equilateral triangle of side 2 and divide
   it by half
//all 3 angles of equilateral triangle are same as
   60 degree
//the bisector of angle is also the perepndicual
   bisector of oppsoite side
// Thus, A=60 B=30 C=90 in new triangle as shown in
   figure

AB = 2; c=AB;
AC = AB/2; b=AC;
a=sqrt(c^2-b^2)//pythagoras theorem

//For angle A=60 degree
opposite = a;
adjacent = b;
hypotenuse = c;
sin_60 = opposite / hypotenuse;
cos_60 = adjacent / hypotenuse;
tan_60 = opposite / adjacent;
csc_60 = hypotenuse/opposite;
sec_60 = hypotenuse/adjacent;
cot_60 = adjacent / opposite;
printf('ANGLE = 60 degree')
printf('\nsin(60)= %.4f ; cos(60)= %.4f;    tan(60)=
    %.4f;\n',sin_60,cos_60,tan_60)
printf('csc(60)= %.4f ; sec(60)= %.4f;    cot(60)= %
   .4f;',csc_60,sec_60,cot_60)

//For angle ABC=30 degree
opposite = b;
```

```
30 adjacent = a;
31 hypotenuse = c;
32 sin_30 = opposite / hypotenuse;
33 cos_30 = adjacent / hypotenuse;
34 tan_30 = opposite / adjacent;
35 csc_30 = hypotenuse/opposite;
36 sec_30 = hypotenuse/adjacent;
37 cot_30 = adjacent / opposite;
38 printf('\n\nANGLE = 30 degree')
39 printf('\nsin(30)= %.4f ; cos(30)= %.4f;    tan(30)=
       %.4f;\n',sin_30,cos_30,tan_30)
40 printf('csc(30)= %.4f ; sec(30)= %.4f;    cot(30)= %
       .4f;',csc_30,sec_30,cot_30)
```

Scilab code Exa 1.8 To find all trigonometric functions when sine functions is giv

```
1 clc,clear
2 //example 1.8
3 //To find all trigonometric functions when sine
     functions is given
4
5 sin_A=2/3 //given
6 //since sine function  is opposite/hypotenuse and
7 //T-ratios are defined interms of ratio of sided of
     right triangle
8 opposite=2;
9 hypotenuse=3;
10 BC = opposite;
11 AB = hypotenuse;
12 b  = sqrt(hypotenuse^2- opposite^2) //by pythagoras
     theorem
13 adjacent = b;
14
```

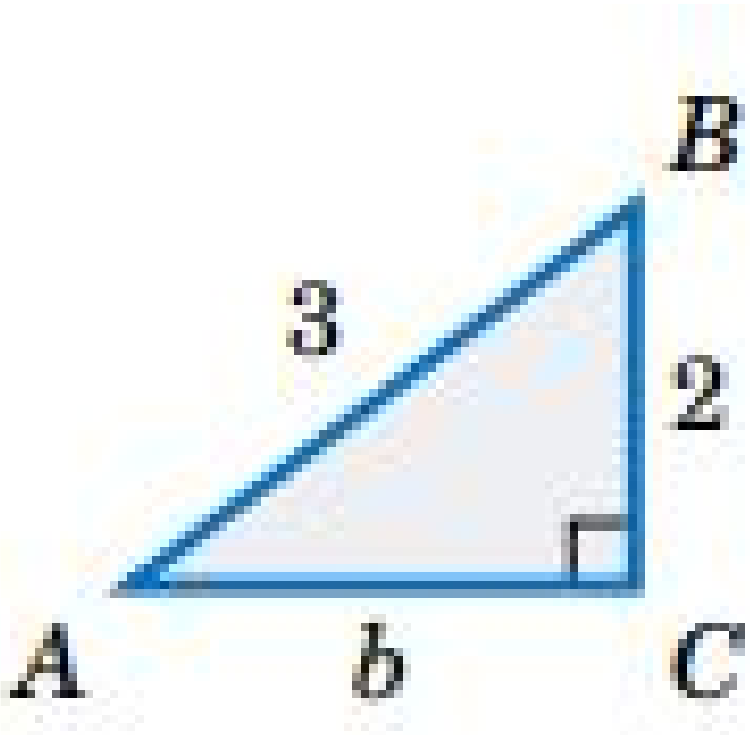

Figure 1.7: To find all trigonometric functions when sine functions is given

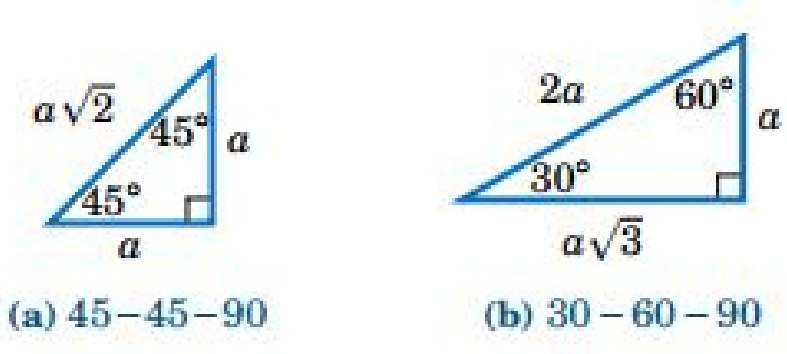

Figure 1.8: To convert given function into function of angle less than 45

```
15 cos_A = adjacent / hypotenuse;
16 tan_A = opposite / adjacent;
17 csc_A = hypotenuse/opposite;
18 sec_A = hypotenuse/adjacent;
19 cot_A = adjacent / opposite;
20
21 printf('for ANGLE A')
22 printf('\nsin(A)= %.4f ; cos(A)= %.4f;  tan(A)= %
   .4f;\n',sin_A,cos_A,tan_A)
23 printf('csc(A)= %.4f ; sec(A)= %.4f;  cot(A)= %.4f
   ;',csc_A,sec_A,cot_A)
```

Scilab code Exa 1.9 To convert given function into function of angle less than 45

```
1 clc,clear
2 //example 1.9
3 //To convert given function into function of angle
   less than 45
4
5 //(a)sin 65
```

```
angle = 65 ;
complement_angle = 90- 65 ;
//cofuction of sine is cosine
printf('(a)sin(%f)= cos (%f)\n',angle,
   complement_angle)

//(b)cos 78
angle = 78;
complement_angle = 90- 78 ;
//cofuction of cosine is sine
printf('(b)cos(%f)= sin (%f)\n',angle,
   complement_angle)

//(c)tan 59
angle = 59 ;
complement_angle = 90- 59 ;
//cofuction of tan is cot
printf('(c)tan(%f)= cot (%f)\n',angle,
   complement_angle)
```

Scilab code Exa 1.10 To find sine cosine and tangent functions for 75 degree

```
clc,clear
//example 1.10
//To find sine,cosine and tangent functions for 75
   degree

//triangle_ADB, angle_BAD = 30
AB=sqrt(3);BD=1;
AD=sqrt(AB^2+BD^2); //pythagoras theorem

//angle_DAB + angle_CAB = 75
//triangle_ABC, angle_BAC = 45
```

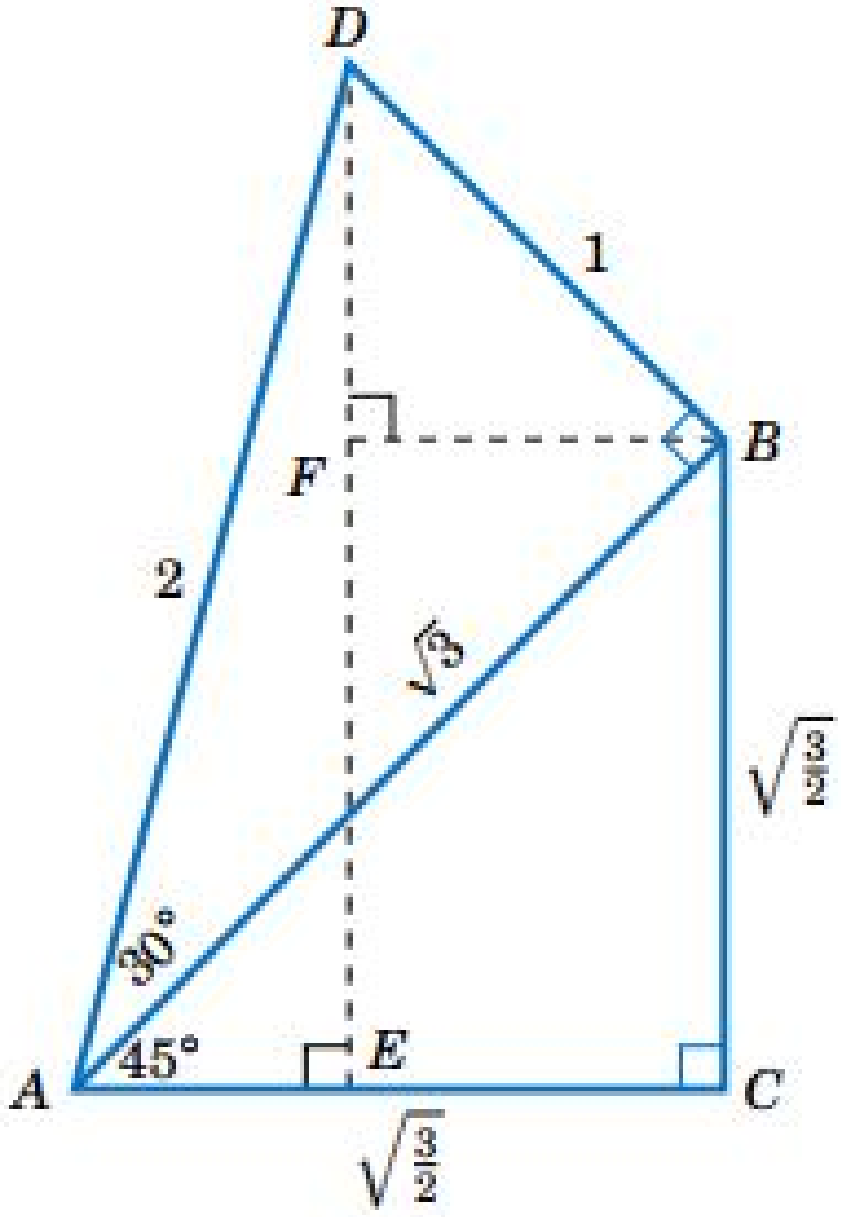

Figure 1.9: To find sine cosine and tangent functions for 75 degree

```
11 //pythagoras theorem and 45 degrees
12 AC=AB/sqrt(2);BC=AC;
13
14 angle_BAC = 45 ; angle_DAB = 30 ;
15 angle_DAE = angle_BAC + angle_DAB ;//required angle
16 angle_ADE = 90 - angle_DAE ;//complement of DAE
17 angle_ADB = 90 - angle_DAB ;//complement of DAB
18
19 //Draw BF perpendicular to DE
20 angle_BDF = angle_ADB - angle_ADE;
21 angle_DBF = 90 - angle_BDF; //complement of BDF
22 //By pythagoras theorem and 45 degree
23 DF=sqrt(BD/2);FB=DF;
24
25 EC=FB;//parallel sides of rectangle
26 FE= BC; //parallel sides of rectangle
27 DE=DF+FE;//from the figure
28 AE=AC-EC;//from the figure
29
30 sin_DAE = DE/AD;
31 cos_DAE = AE/AD;
32 tan_DAE = DE/AE;
33 csc_DAE = AD/DE;
34 sec_DAE = AD/AE;
35 cot_DAE = AE/DE;
36
37 printf('sin(%d)=%f\n',angle_DAE,sin_DAE);
38 printf('cos(%d)=%f\n',angle_DAE,cos_DAE);
39 printf('tan(%d)=%f\n',angle_DAE,tan_DAE);
40 printf('csc(%d)=%f\n',angle_DAE,csc_DAE);
41 printf('sec(%d)=%f\n',angle_DAE,sec_DAE);
42 printf('cot(%d)=%f\n',angle_DAE,cot_DAE);
```

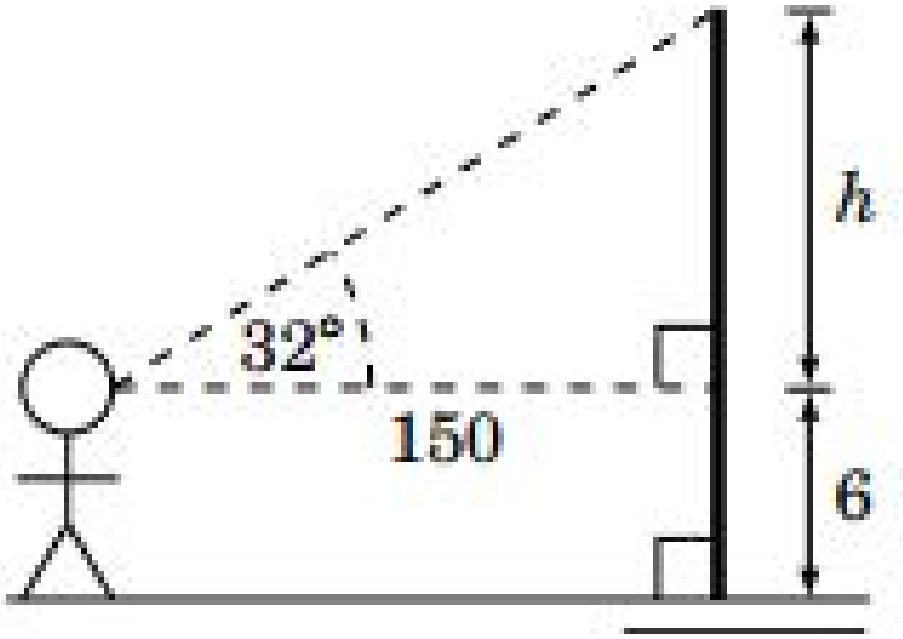

Figure 1.10: To find the height of the flagpole

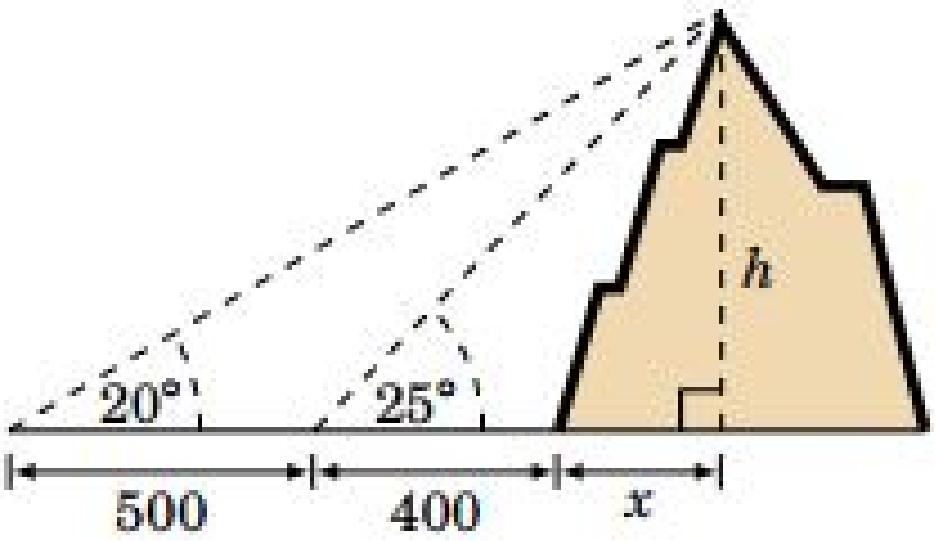

Figure 1.11: To find the height of mountain

Scilab code Exa 1.11 To find the height of the flagpole

```
clc,clear
//example 1.11
//To find the height of the flagpole

//conside the attached figure
d=150 //distance of person from flagpole in feet
angle_ele = 32 //angle of elevation in degree
height_eyes =6 //height of man's eyes
h= d*tand(angle_ele)
height_flagpole = height_eyes + h
printf('Required height of flagpole = %.0f ft',
   height_flagpole)
```

Scilab code Exa 1.12 To find the height of mountain

```
clc,clear
//Example 1.12
//To find the height of mountain

//from the figure
//h is height of mountain in degree
//x is distance from base of mountain to the point
   under top of mountain

d1=400 //initial ditance from base of mountain in
   feet
d2=500 //final ditance from base of mountain in feet
theta1=25 //initial angle of elevation in degrees
theta2=20 //final angle of elevation in degrees

//from the figure
//h= (x+d1) * tand(theta1)
//h= (x+d2) * tand(theta2)
//eliminating h and solving for x
x=((d1+d2)*tand(theta2) - d1*tand(theta1))/(tand(
   theta1)-tand(theta2))
//substituting x in expression for h
h= (x+d1) *tand(theta1)
printf('Height of mountain = %.0f feet',h)
```

Scilab code Exa 1.13 To find the horizontal distance from blimp to house

```
clc,clear
```

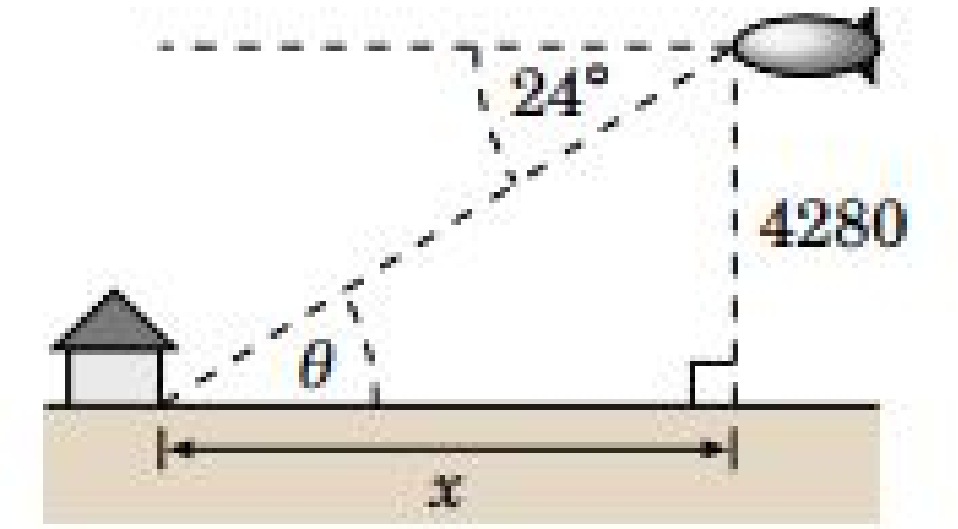

Figure 1.12: To find the horizontal distance from blimp to house

```
2 //Example 1.13
3 //To find the horizontal distance from blimp to
     house
4
5 //consider the figure attached
6 angle_dep = 24 //angle of depression in degrees
7 theta = angle_dep  //angle of elevation
8 height_blimp = 4280 //height of blimp from ground in
     feet
9 x = height_blimp  / tand(theta) //required distance
     in feet
10
11 printf('The house is %.0f ft far from blimp along
     the ground',x)
```

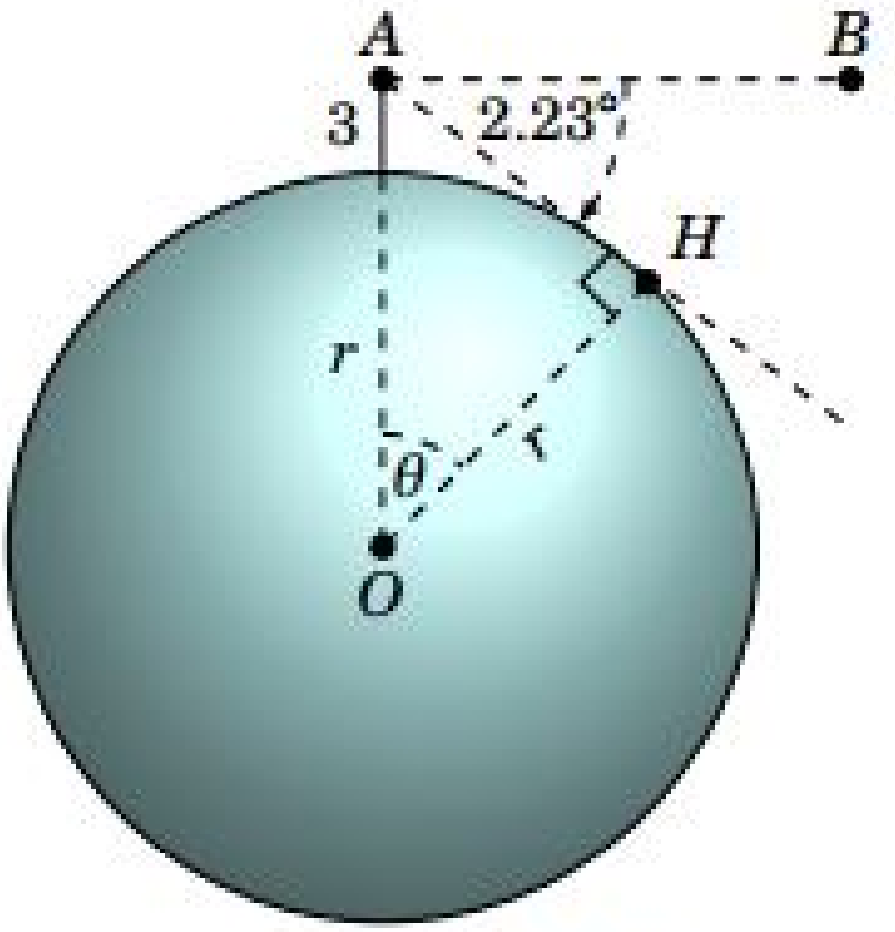

Figure 1.3.1

Figure 1.13: To estimate radius of earth when angle of depression is known

Scilab code Exa 1.14 To estimate radius of earth when angle of depression is known

```
clc,clear
//Example 1.14
//To estimate radius of earth when angle of
   depression is known

angle_dep = 2.23 //angle of depression in degrees
//In the figure,
//r is the radius of earth
//A represent the top of the mountain
//H be the ocean horizon in the line of sight from A
//O be the center of the earth
//B is a point on the horizontal line of sight from
   A

angle_OAH = 90 - angle_dep ;
theta = 180 - 90 - angle_OAH ;
height=3 //height of mountain
//r is radius of earth to be determined

//distance from top of mountain from centre = r +
   height
// cosd(theta)= r/r+height...solving further
r = height*cosd(theta)/(1-cosd(theta)) ;
printf('Radius of earth as calculated =%.1f miles\n
   ',r)
```

Scilab code Exa 1.15 To find the distance from centre of earth to sun

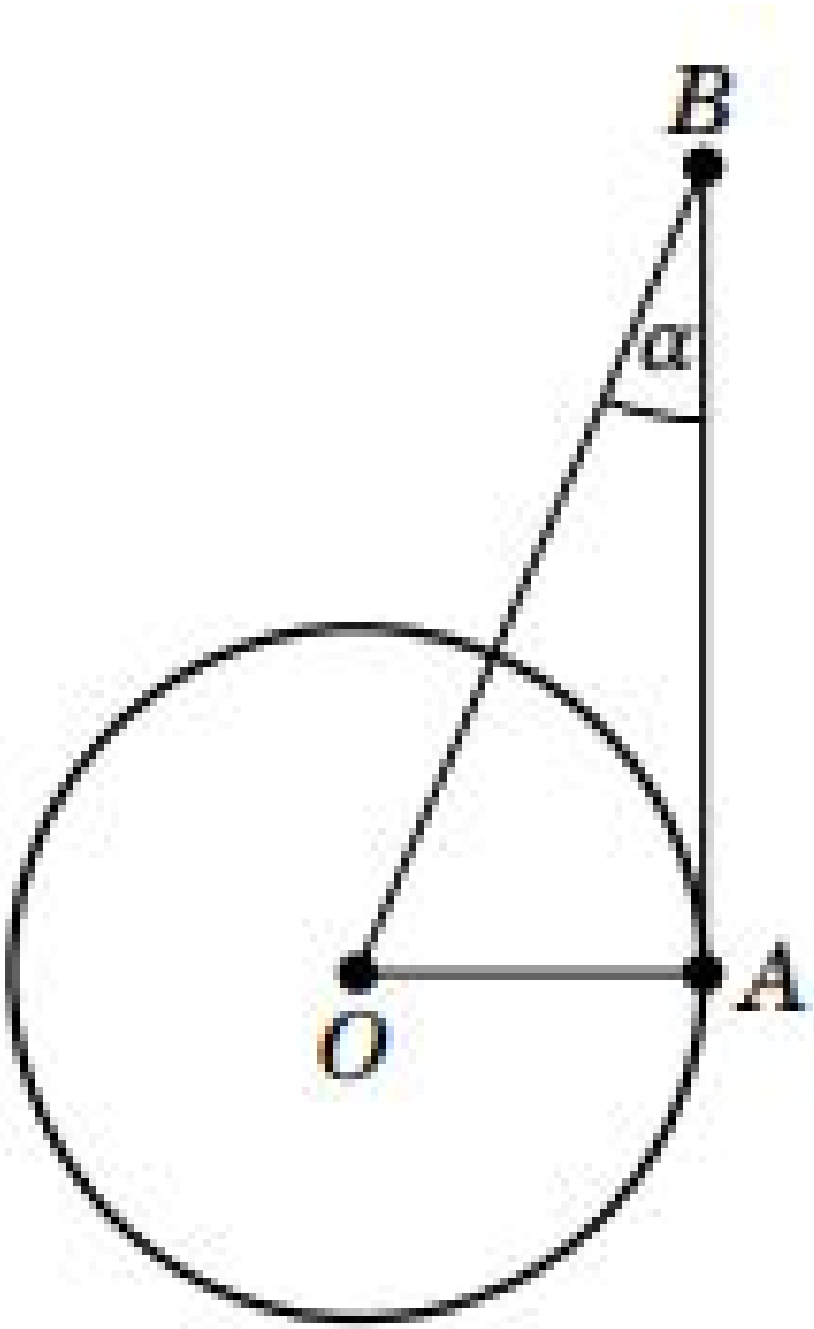

Figure 1.14: To find the distance from centre of earth to sun

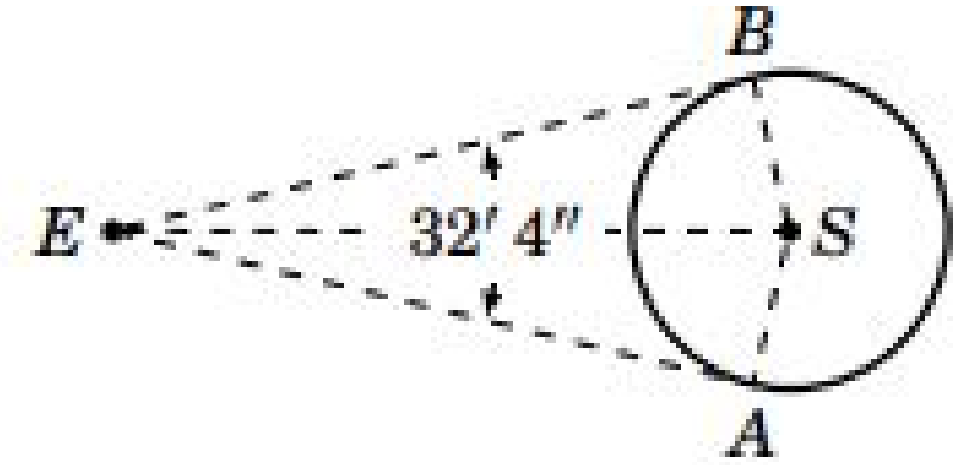

Figure 1.15: To determine the radius of sun

```
1  clc,clear
2  //Example 1.15
3  // To find the distance from centre of earth to sun
4
5  alpha = 0.00244; // equitorial paralalx in degree
6  OA = 3956.6; //radius of earth
7  angle_OAB = 90;
8
9  OB = OA / sind(alpha) ;
10 printf('Distance is obtained as %.0f miles = %.0f
     million miles',OB,OB/10^6)
```

Scilab code Exa 1.16 To determine the radius of sun

```
1  clc,clear
2  //Example 1.16
3  // To determine the radius of sun
4
```

```
5  angle_AEB =0 +(32/60)+ (4/60)/60//converting to
       degrees
6
7  //Triangle BES and AES are similar
8  //BS=AS as they are radius
9  //ES is common to both triangles
10 //angle_EBS=angle_ABS =90 as tangents are
       perpendicualar to radius
11 // angle_AES = angle_BES
12 angle_AES= angle_AEB /2;
13 angle_BES= angle_AEB /2;
14
15 //to find ditance from sun to centre of earth
16 //obtained from previous example
17 alpha = 0.00244; // equitorial paralalx in degree
18 OA = 3956.6 ;//radius of earth
19 angle_OAB = 90 ;//radius perpendicular to tangent
20 OB = OA / sind(alpha) ;
21
22 //ES is from earth surface to sun centre
23 //centre of earth to sun is  OB
24 //we initially treated sun as point
25 //that ditance is distance between their centres
26 radius_earth=3956.6 ;//in miles
27 ES = OB - radius_earth ;//in miles
28 AS=ES * sind(angle_AES) ;//in miles
29 printf('Required radius of sun = %.0f miles \n',AS)
30 printf('Answer might vary due to approximations in
       book and scilab precision')
```

Scilab code Exa 1.17 To determine the diameter of larger roller

```
1 clc,clear
```

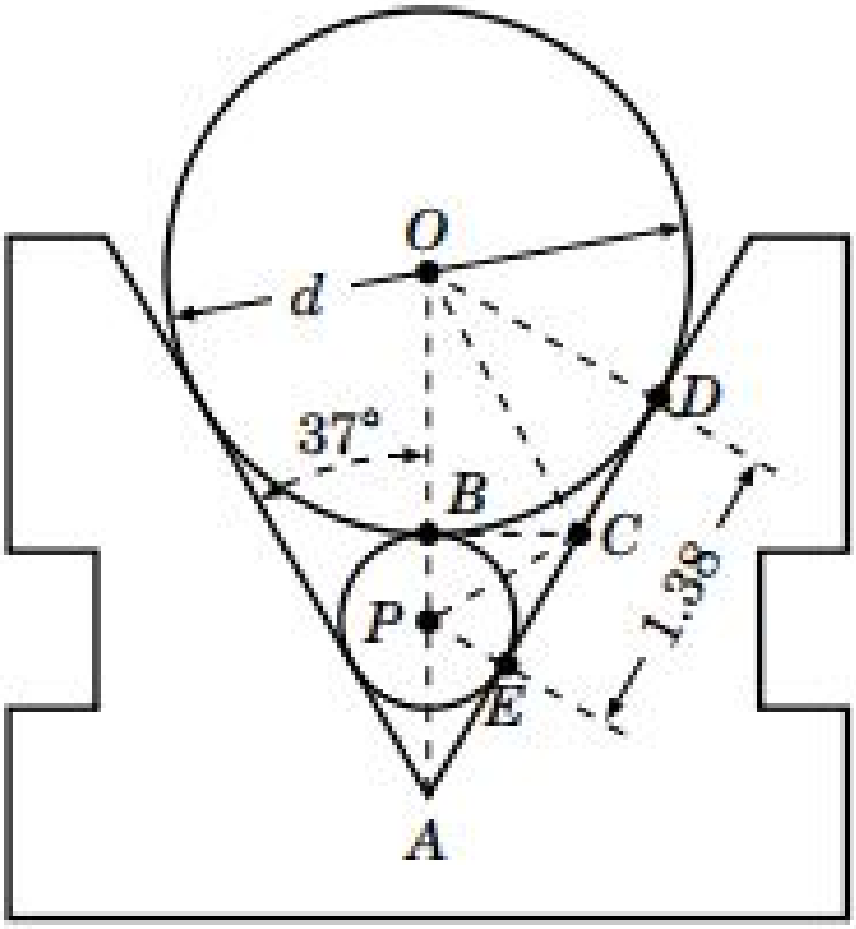

Figure 1.16: To determine the diameter of larger roller

```
2 //Example 1.17
3 //To find the diameter of larger roller
4
5 //since radius perpendicular to tangent
6 angle_ODA=90;
7 angle_PEC=90
8 angle_OAD=37 ;//by symmetry
9 ED=1.38 ;//given
10 //since DOA is right triangle, DOA and OAD are
     complementary angles
11 angle_DOA=90 - angle_OAD ;
12
13 //since radius perpendicular to tangent
14 angle_OBC=90;
15 angle_PBC=90;
16
17 //since ODA and ODC are right triangle,
18 //OD = OB as radius and BC=DC by pythagoras
19 //OBC and ODC are now congruent
20 //angle_BOC = angle_DOC
21 //angle_BOC + angle_DOC =(90-angle_OAD)
22 angle_BOC= (90-angle_OAD)/2;
23 angle_DOC= (90-angle_OAD)/2;
24
25 //BP=EP as radius
26 //since radius perpendicular to tangent
27 angle_PBC=90;
28 angle_PEC=90;
29 //Thus,BPC and EPC are congruent triangles
30 //Therefore,BC=DC and BC+DC = ED
31 BC = ED /2;
32 DC = ED /2;
33 OB = BC / tand(angle_BOC); //radius of large roller
34 diameter= 2* OB ;
35 printf('Diameter of larger roller = %.3f units',
     diameter)
```

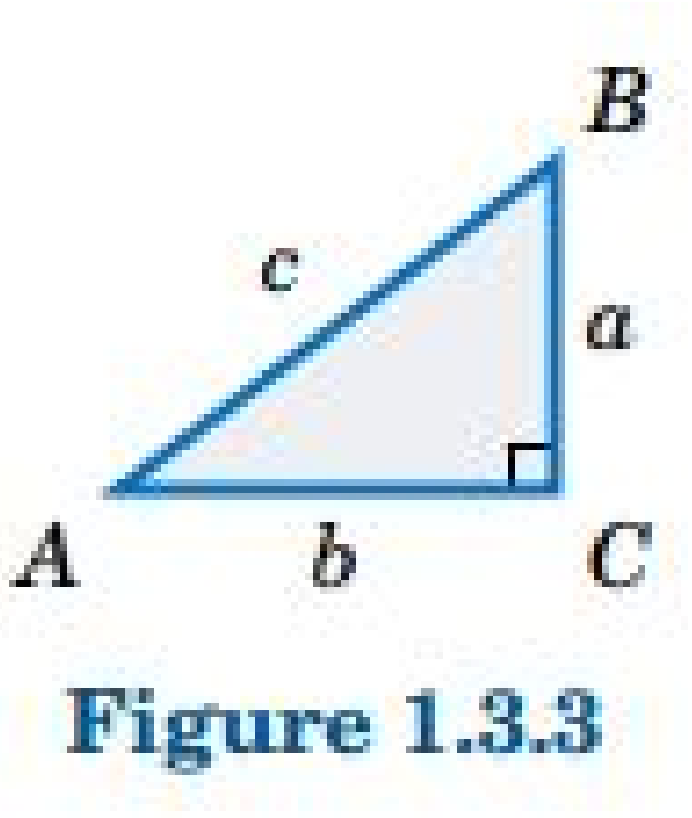

Figure 1.17: To solve the right triangle with given information

Scilab code Exa 1.19 To solve the right triangle with given information

```
1 clc,clear
2 //Example 1.19
3 //To solve the right triangle with given information
4
```

```
5 //part(a)
6 c=10 ;//side opposite to vertex C
7 A=22 ;//Angle at vertex A
8 a=c*sind(A);
9 b=c*cosd(A) ;
10 B=90 - A ;//since C is 90, A and B are complimentary
11 printf('(a)a= %.2f units ; b= %.2f  units; B = %.0f
     degree\n',a,b,B)
12
13 //part(b)
14 b=8 ;//side opposite to vertex B
15 A=40 ;//Angle at vertex A
16 a=b*tand(A);
17 c=b/cosd(A) ;
18 B=90 - A ;//since C is 90, A and B are complimentary
19 printf(' (b)a= %.2f units ; c= %.2f units; B = %.0f
     degree\n',a,c,B)
20
21 //part(c)
22 a=3 ;//side opposite to vertex A
23 b=4 ;//side opposite to vertex B
24 c=sqrt(a^2+b^2) ;//by pythagoras theorem
25 A = atand(a/b) ;//angle at vertex A
26 B=90 - A ;//since C is 90, A and B are complimentary
27 printf(' (c)c=%.0f units      ; A= %f degree;  B = %f
     degree',c,A,B)
```

Scilab code Exa 1.20 To find values of all trigonometric values of given angle of

```
1 clc,clear
2 //Example 1.20
3 //To find values of all trigonometric values of
     given angle of 120
```

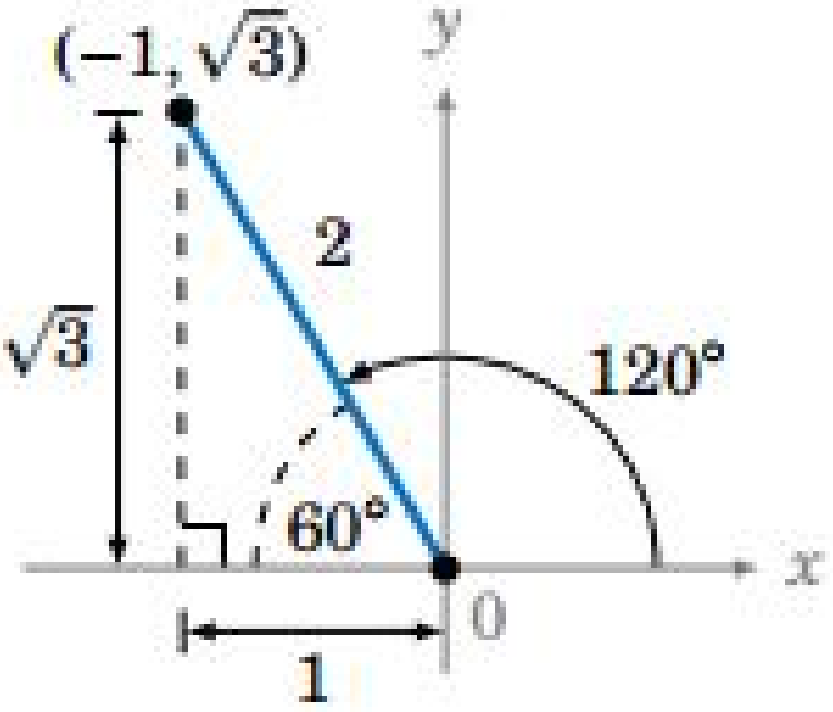

Figure 1.18: To find values of all trigonometric values of given angle of 120

```
4
5 theta=120; //given angle in degree
6 //Consider a point (-1,sqrt(3)) in 2nd quadrant
7 //This point can be used on terminal side of 120
8 //Thus for a basic right angled triangle formed in
     second quadrant
9 adjacent = 1;
10 opposite = sqrt(3);
11 //by pythagoras theorem
12 hypotenuse = sqrt(adjacent^2 + opposite^2);
13
14 //since its third quadrant
15 x=-adjacent;
16 y= opposite;
17 r=hypotenuse;
18
19 sin_120 =y/r ;
20 cos_120 =x/r ;
21 tan_120 =y/x ;
22 csc_120 =r/y ;
23 sec_120 =r/x ;
24 cot_120 =x/y ;
25
26 printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;'
     ,theta,sin_120,theta,cos_120,theta,tan_120)
27 printf('\ncsc(%d)= %f ; sec(%d)= %f ; cot(%d)= %f ;'
     ,theta,csc_120,theta,sec_120,theta,cot_120)
```

Scilab code Exa 1.21 To find values of all trigonometric values of given angle of

```
1 clc,clear
2 //Example 1.21
3 //To find values of all trigonometric values of
```

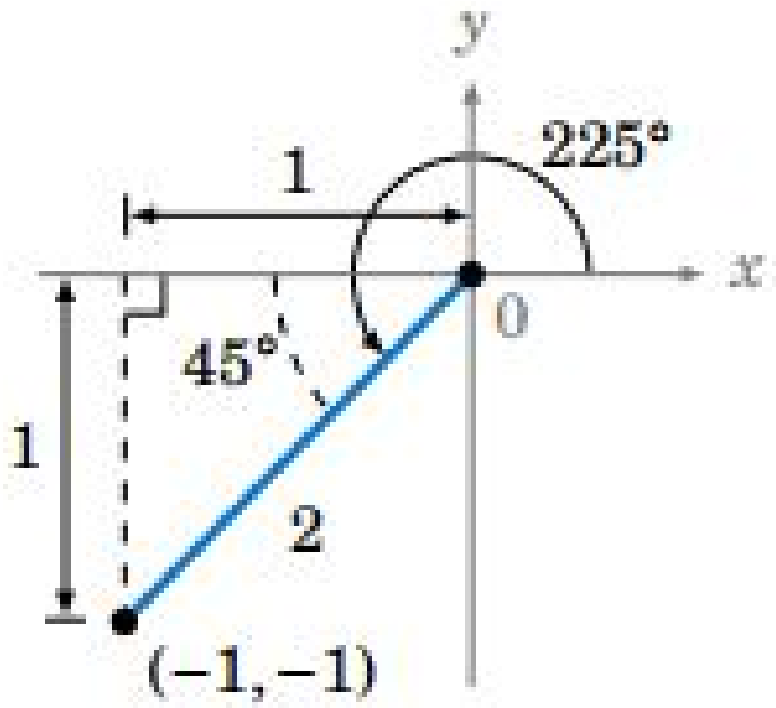

Figure 1.19: To find values of all trigonometric values of given angle of 225

```
    given angle of 225
4
5 theta=225 ;//given angle in degree
6 //Consider a point (-1,-1) in 3rd quadrant
7 //This point can be used on terminal side of 225
8 //Thus for a basic right angled triangle formed in 3
    rd quadrant
9 adjacent = 1 ;
10 opposite = 1 ;
11 //by pythagoras theorem
12 hypotenuse = sqrt(adjacent^2 + opposite^2) ;
13
14 //since its third quadrant
15 x=-adjacent ;
16 y=-opposite ;
17 r=hypotenuse ;
18
19 sin_225 =y/r ;
20 cos_225 =x/r ;
21 tan_225 =y/x ;
22 csc_225 =r/y ;
23 sec_225 =r/x ;
24 cot_225 =x/y ;
25
26 printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;'
    ,theta,sin_225,theta,cos_225,theta,tan_225)
27 printf('\ncsc(%d)= %f ; sec(%d)= %f ; cot(%d)= %f ;'
    ,theta,csc_225,theta,sec_225,theta,cot_225)
```

Scilab code Exa 1.22 To find values of all trigonometric values of given angle of

```
1 clc,clear
2 //Example 1.22
```

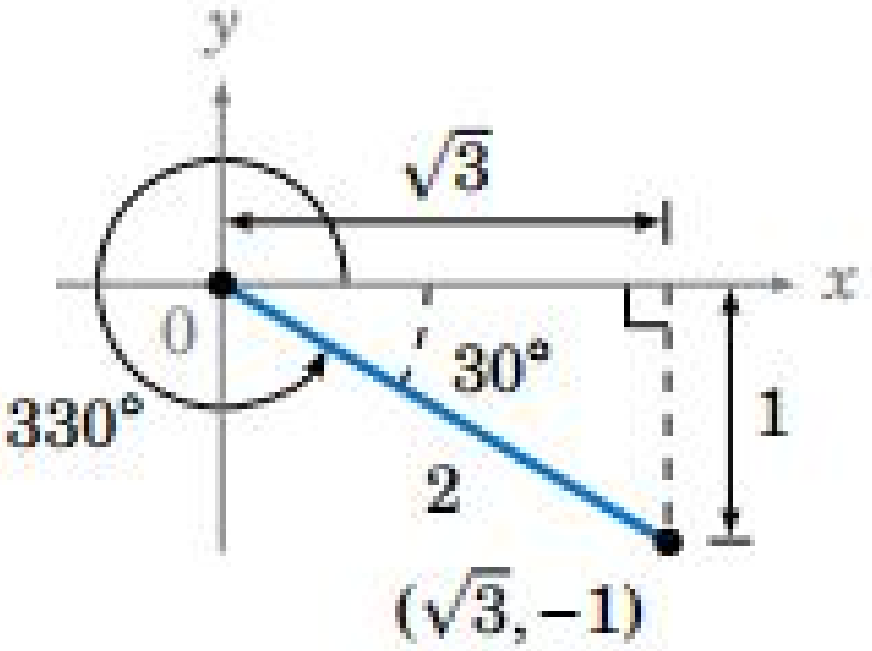

Figure 1.20: To find values of all trigonometric values of given angle of 330

```
3 //To find values of all trigonometric values of
     given angle of 330
4
5 theta=330 ; //given angle in degree
6 //Consider a point (sqrt(3),-1) in 4th quadrant
7 //This point can be used on terminal side of 330
8 //Thus for a basic right angled triangle formed in 4
     th quadrant
9 adjacent = sqrt(3);
10 opposite = 1 ;
11 hypotenuse = sqrt(adjacent^2 + opposite^2);
12 //by pythagoras theorem
13
14 //since its 4th quadrant
15 x=adjacent ;
16 y=-opposite ;
17 r=hypotenuse ;
18
19 sin_330 =y/r ;
20 cos_330 =x/r ;
21 tan_330 =y/x ;
22 csc_330 =r/y ;
23 sec_330 =r/x ;
24 cot_330 =x/y ;
25
26 printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;'
     ,theta,sin_330,theta,cos_330,theta,tan_330)
27 printf('\ncsc(%d)= %f ; sec(%d)= %f ; cot(%d)= %f ;'
     ,theta,csc_330,theta,sec_330,theta,cot_330)
```

Scilab code Exa 1.23 To find trigonometric ratios of 0 90 180 and 270 degrees

```
1 clc,clear
```

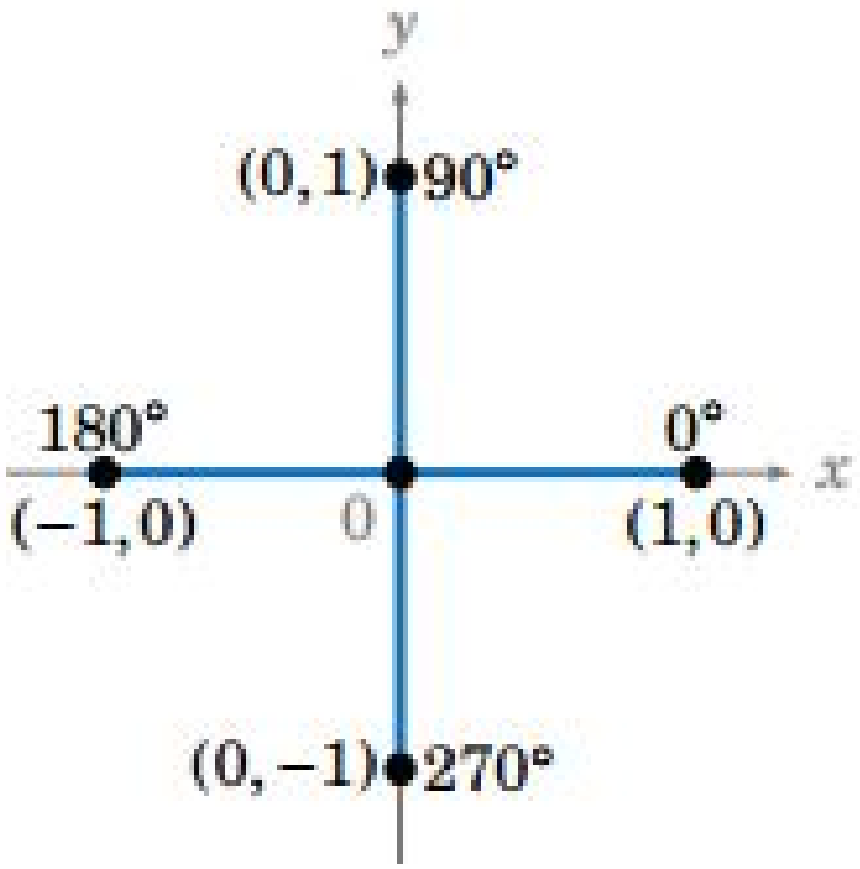

Figure 1.21: To find trigonometric ratios of 0 90 180 and 270 degrees

```
2 //Example 1.23
3 //To find trigonometric functions of various angles
4 //Note: Undefined ratios are commented to avoid
     divide by zero error
5
6 //zero degrees
7 //consider a point (1,0)
8 //Line segment joining (0,0) and (1,0) can be
     treated as triangle
9
10 x=1//base
11 y=0//height
12 r=1 //hypotenuse
13 sin_0 = y/r;
14 cos_0 = x/r;
15 tan_0 = y/x;
16 //csc_0 =r/y;
17 sec_0 = r/x;
18 //cot_0=x/y;
19 printf('\nZERO DEGREES:\n')
20 printf('sin(0)= %f;\n',sin_0)
21 printf('cos(0)= %f;\n',cos_0)
22 printf('tan(0)= %f;\n',tan_0)
23 printf('csc(0)= undefined = (1/0);\n')
24 printf('sec(0)= %f;\n',sec_0)
25 printf('cot(0)= undefined = (1/0);\n')
26
27 //90 degrees
28 //consider a point (0,1)
29 //Line segment joining (0,0) and (0,1) can be
     treated as triangle
30
31 x=0//base
32 y=1//height
33 r=1 //hypotenuse
34 sin_90 = y/r;
35 cos_90 = x/r;
36 //tan_90 = y/x;
```

```
37 csc_90 =r/y;
38 //sec_90 = r/x;
39 cot_90=x/y;
40 printf('\n90 DEGREES:\n')
41 printf('sin(90)= %f;\n',sin_90)
42 printf('cos(90)= %f;\n',cos_90)
43 printf('tan(90)= undefined = (1/0);\n')
44 printf('csc(90)= %f;\n',csc_90)
45 printf('sec(90)= undefined = (1/0);\n')
46 printf('cot(90)= %f;\n',cot_90)
47
48 //180 degrees
49 //consider a point (-1,0)
50 //Line segment joining (0,0) and (-1,0) can be
      treated as triangle
51
52 x=-1//base
53 y=0//height
54 r=1 //hypotenuse
55 sin_180 = y/r;
56 cos_180 = x/r;
57 tan_180 = y/x;
58 //csc_180 =r/y;
59 sec_180 = r/x;
60 //cot_180=x/y;
61 printf('\n180 DEGREES:\n')
62 printf('sin(180)= %f;\n',sin_180)
63 printf('cos(180)= %f;\n',cos_180)
64 printf('tan(180)= %f;\n',tan_180)
65 printf('csc(180)= undefined = (1/0);\n')
66 printf('sec(180)= %f;\n',sec_180)
67 printf('cot(180)= undefined = (-1/0);\n')
68
69
70 //270 degrees
71 //consider a point (0,-1)
72 //Line segment joining (0,0) and (0,-1) can be
      treated as triangle
```

```
73
74 x=0//base
75 y=-1//height
76 r=1 //hypotenuse
77 sin_270 = y/r;
78 cos_270 = x/r
79 //tan_90 = y/x;
80 csc_270 =r/y;
81 //sec_90 = r/x;
82 cot_270=x/y;
83 printf('\n270 DEGREES:\n')
84 printf('sin(270)= %f;\n',sin_270)
85 printf('cos(270)= %f;\n',cos_270)
86 printf('tan(270)= undefined = (-1/0);\n')
87 printf('csc(270)= %f;\n',csc_270)
88 printf('sec(270)= undefined = (1/0);\n')
89 printf('cot(270)= %f;\n',cot_270)
```

Scilab code Exa 1.24 To determine reference angle and angle between 0 to 360 with

```
1 clc,clear
2 //Example 1.24
3 //To determine reference angle and angle ( 0 to 360
    )with same terminal side as given angle
4
5 theta = 928 ;//given angle in degrees
6
7 //The while loop works for ALL VALUES OF theta
8 //It keeps subtracting 360 till a value in (0 to
    360) is obtained
9  result = theta ;
10 while 1==1,
11     if result<360 then
```

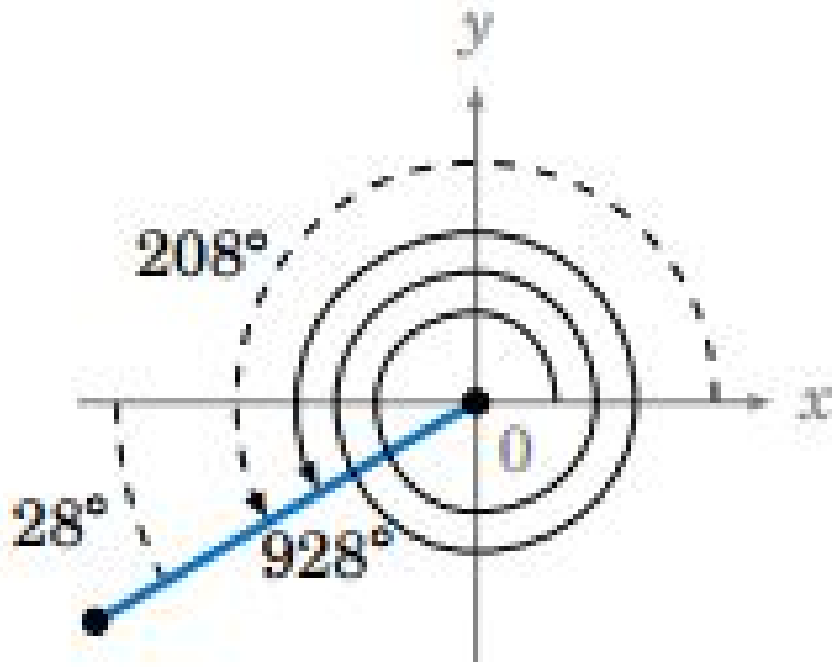

Figure 1.22: To determine reference angle and angle between 0 to 360 with same terminal side as given angle

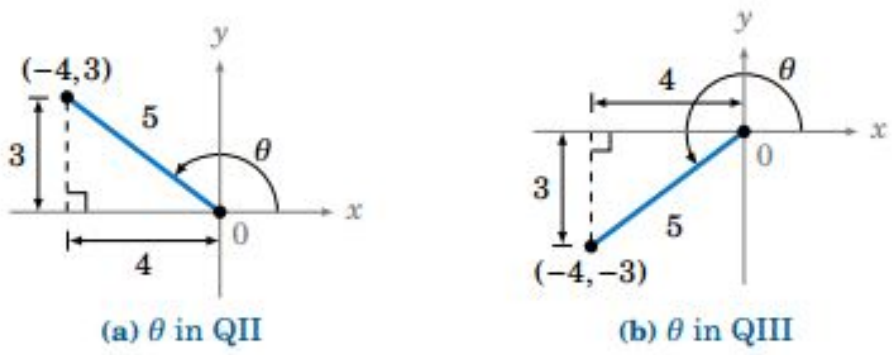

Figure 1.23: To find sin theta and tan theta when cos theta is given

```
12          printf('(a)Required angel between 0 and 360
                is %.0f degree',result);
13          break
14      end
15      result = result - 360 ;
16  end
17
18  //928 and 208 has same terminal side in 3rd quadrant
19  //so their reference angle is same
20  ref_angle_928 = result - 180 ; //required reference
        angle
21  printf('\n(b)Reference angel for %.0f is %.0f degree
        ',theta,ref_angle_928)
```

Scilab code Exa 1.25 To find sin theta and tan theta when cos theta is given

```
1  clc,clear
2  //Example 1.25
3  //To find sin_theta and tan_theta when cos_theta is
       given
```

```
4
5 cos_theta = -4/5;
6 adjacent =4 ; hypotenuse =5 ;
7 opposite = sqrt(hypotenuse ^2 - adjacent ^2) //by
    pythagoras theorem
8
9 //minus sign of cos_theta implies 2nd or 3rd
    quadrant
10 //Possibility 1 : 2nd quadrant
11 x= -adjacent ;
12 y=  opposite ;
13 r=  hypotenuse ;
14 sin_theta = y/r ;
15 tan_theta = y/x ;
16 printf('POSSIBILITY 1:Theta in 2nd quadrant\n')
17 printf('sin(theta)= %.2f ; tan(theta) = %.2f;  \n\n'
    ,sin_theta,tan_theta)
18
19 //Possibility 2 : 3rd quadrant
20 x=-adjacent ;
21 y=-opposite ;
22 r=hypotenuse ;
23 sin_theta = y/r ;
24 tan_theta = y/x ;
25 printf('POSSIBILITY 2:Theta in 3rd quadrant\n')
26 printf('sin(theta)= %.2f ; tan(theta) = %.2f;  ',
    sin_theta,tan_theta)
```

Scilab code Exa 1.27 To find all the angles with a given sine function value

```
1 clc,clear
2 //Example 1.27
3 //To find all the angles with a given sine function
```

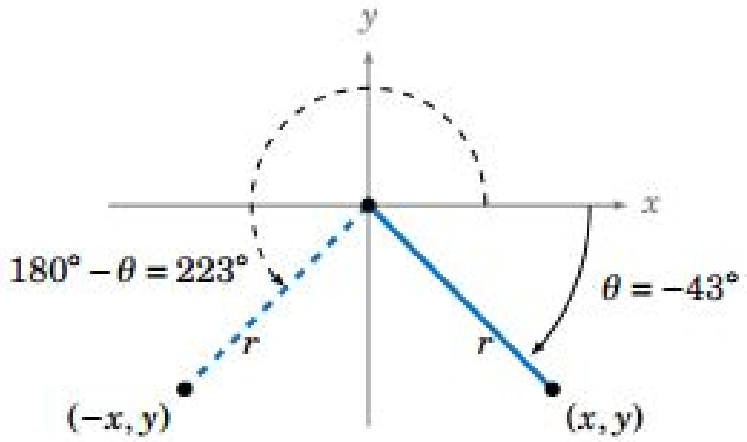

Figure 1.24: To find all the angles with a given sine function value

```
    value
4
5 sin_theta = -0.682;
6 theta=asind(sin_theta);
7 //This results -43 degree which isnt in 0 to 360
    range
8 //And theta exists in 4th quadrant
9 //Angles in 1st and 2nd quadrant have +ve sine
    values
10
11 theta1 = 180 - theta ;//reflection of theta in 3rd
    quadrant
12 theta2 = 360 + theta ;//both theta n theta 2 have
    same trigonometric values
13 printf('Required angles are %.0f and %.0f degrees',
    theta1,theta2)
```

Chapter 2

General Triangles

Scilab code Exa 2.1 To solve the triangle when one side and 2 angles are given

```
1 //Example 2.1
2 //To solve the triangle when one side and 2 angles
     are given
3 clc,clear
4
5 a=10 //side opposite to vertex A
6 A=41 //angle at vertex A
7 C=75 //angle at vertex C
8
9 B=180- (A+C)
10 b=a*sind(B)/sind(A) //law of sines
11 c=a*sind(C)/sind(A) //law of sines
12 printf('Angle B is %.0f degrees\n length of side b
     is %.1f units\n length of side c is %.1f units',B
     ,b,c)
```

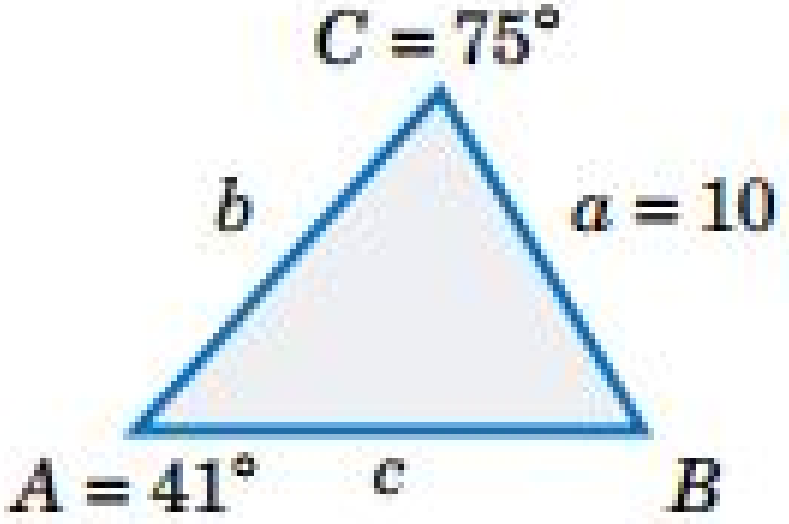

Figure 2.1: To solve the triangle when one side and 2 angles are given

Figure 2.2: To solve the triangle when 2 sides and one opposite angle is given

Scilab code Exa 2.2 To solve the triangle when 2 sides and one opposite angle is g

```
1 //Example 2.2
2 //To solve the triangle when 2 sides and one
     opposite angle is given
3 clc,clear
4
5 a=18 //side oposite to vertex A
6 A=25 //angle at vertex A
7 b=30 //side opposite to vertex B
8
9 sin_B=(b/a)*sind(A) //law of sines
10
11 //case 1
12 B=asind(sin_B)   //law of sines
13 C=180-(A+B)
14 c=a*sind(C)/sind(A) //law of sines
15 printf('1st possible solution set\nAngle at B =%.1f
     degree\nAngle at C=%.1f degree\nlength of side c=
     %.0f units\n',B,C,c)
16
17 //case 2
18 B=180 - asind(sin_B)  //law of sines
19 C=180-(A+B)
20 c=a*sind(C)/sind(A)  //law of sines
21 printf('\n\n2nd possible solution set\nAngle at B =%
     .1f degree\nAngle at C=%.1f degree\nlength of
     side c=%.1f units\n',B,C,c)
```

Scilab code Exa 2.3 To solve the triangle when 2 sides and opposite angle is given

```
1 //Example 2.3
```

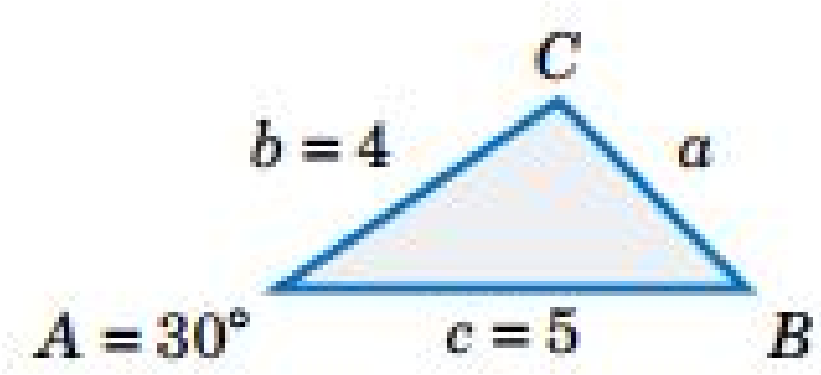

Figure 2.3: To solve the triangle when 2 sides and angle between them is given

```
2 //To solve the triangle when 2 sides and opposite
     angle is given
3 clc,clear
4
5 a=5 //side oposite to vertex A
6 A=30 //angle at vertex A
7 b=12 //side opposite to vertex B
8
9 sin_B=(b/a)*sind(A) //law of sines
10 printf("sin(B)=%f. But magnitude of sin(B) should
     be less than 1\nHence,there is no solution",sin_B
     )
```

Scilab code Exa 2.4 To solve the triangle when 2 sides and angle between them is g

```
1 //Example 2.4
```

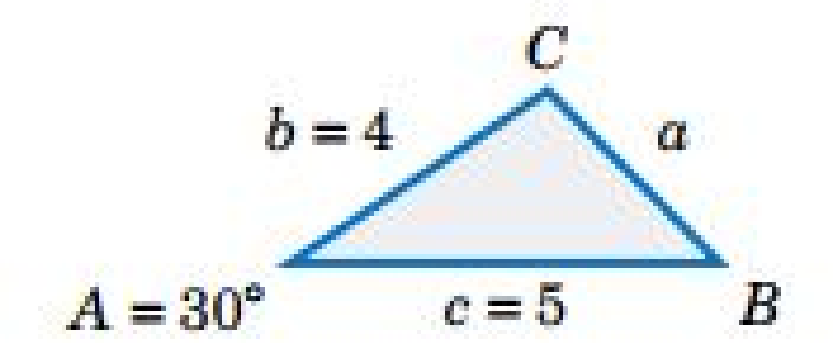

Figure 2.4: To solve the triangle when 2 sides and angle between them is given

```
2  //To solve the triangle when 2 sides and angle
       between them is given
3  clc,clear
4
5  c=5 //side oposite to vertex C
6  A=30 //angle at vertex A
7  b=4 //side opposite to vertex B
8
9  printf('By law of sines:\n')
10 printf('a/sin(30) = 4/sin(B) = 5 / sin(C)\n')
11 printf('\nEach of the equations has 2 unknowns.')
12 printf('\nFor eg: To obtain a , we can use 4/sin(B)
       =5/sin(C). Next we obtain B interms of C and put
       back.\n')
13 printf('Now we have a in terms of C which is unknown
       \n')
14 printf('Hence it is IMPOSSIBLE to solve this by law
       of sines')
```

Scilab code Exa 2.5 To solve the triangle when 2 sides and angle between them is g

```
1 //Example 2.5
2 //To solve the triangle when 2 sides and angle
     between them is given
3 clc,clear
4
5 c=5 //side oposite to vertex C
6 A=30 //angle at vertex A
7 b=4 //side opposite to vertex B
8
9 a = sqrt( b^2 + c^2 -2*b*c*cosd(A) ) //from law of
     cosines
10 printf('Length of a= %.2f units\n',a)
11 cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
12 B=acosd(cos_B)
13 printf('Angle B=%.1f degrees\n',B)
14 C=180-(A+B)
15 printf('Angle C=%.1f degrees\n',C)
```

Scilab code Exa 2.6 To solve the triangle when 3 sides are given

```
1 //Example 2.6
2 //To solve the triangle when 3 sides are given
3 clc,clear
4
5 c=4 //side oposite to vertex C
6 a=2 //side opposite to vertex A
7 b=3 //side opposite to vertex B
8
```

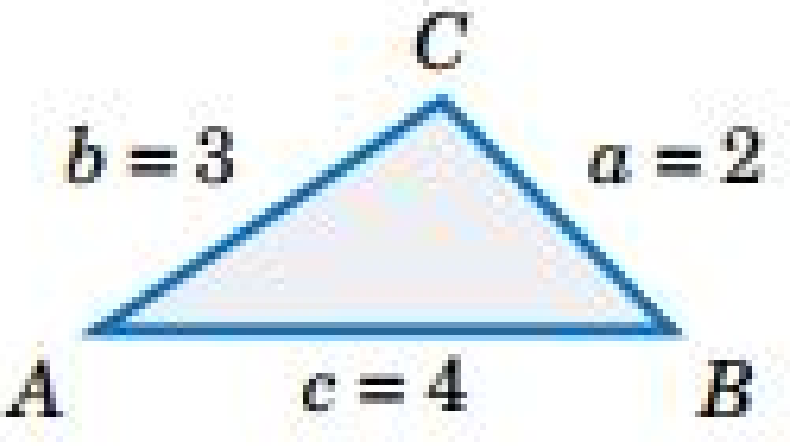

Figure 2.5: To solve the triangle when 3 sides are given

```
9  cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
10 B=acosd(cos_B)
11 printf('Angle B=%.1f degrees\n',B)
12 cos_C = (b^2+a^2-c^2)/(2*b*a) //from law of cosines
13 C=acosd(cos_C)
14 printf('Angle C=%.1f degrees\n',C)
15
16 A=180-(C+B)
17 printf('Angle A=%f degrees',A)
```

Scilab code Exa 2.7 To determine solution of a triangle when 3 sides are given

```
1 //Example 2.7
2 //To determine solution of a triangle when 3 sides
```

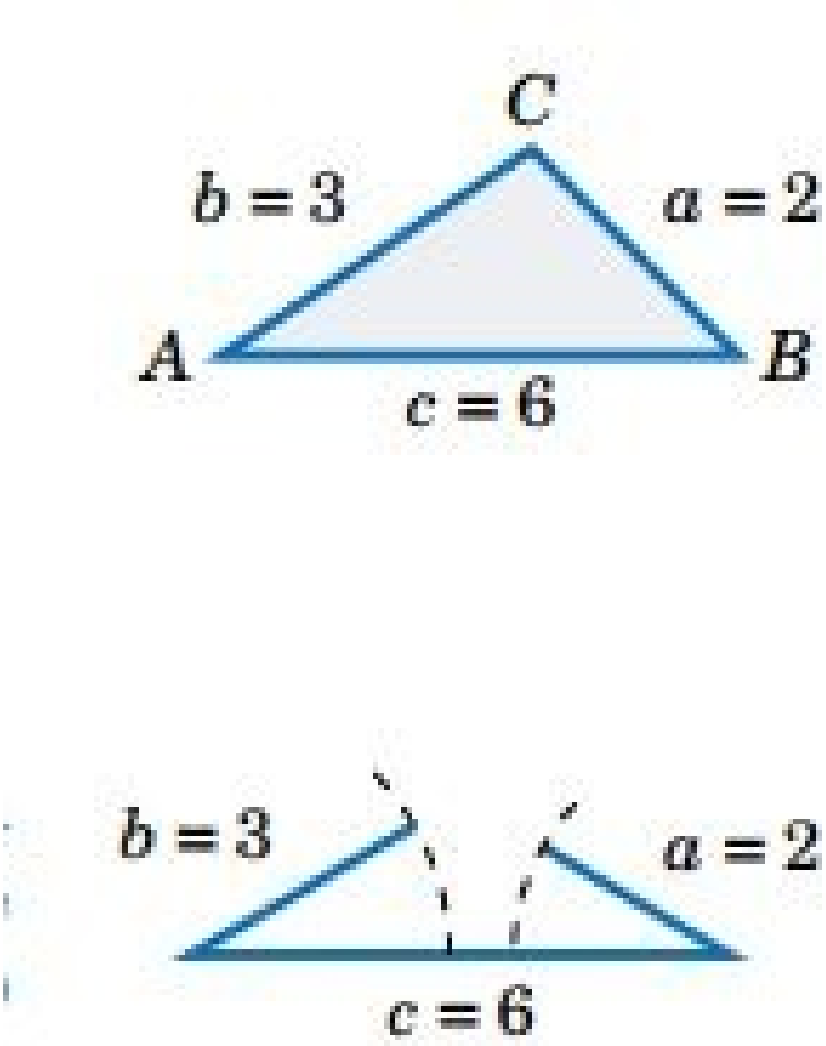

Figure 2.6: To determine solution of a triangle when 3 sides are given

```
    are given
clc,clear

c=6 //side oposite to vertex C
a=2 //side opposite to vertex A
b=3 //side opposite to vertex B

cos_A = (b^2+c^2-a^2)/(2*c*b) //from law of cosines
printf('cos(A)=%.3f as calculated\n',cos_A)
printf('But magnitude of cos(A) should always be
    less than 1. Hence NO SOLUTION exists\n\n')

printf('Note: We observe that a+b < c. But sum of
    any 2 sides should always exceed third side.')
printf('\nHence this triangle is impossible.')
```

Scilab code Exa 2.8 To solve the triangle when 2 sides and opposite angle is given

```
//Example 2.8
//To solve the triangle when 2 sides and opposite
    angle is given
clc,clear

a=18 //side oposite to vertex A
A=25 //angle at vertex A
b=30 //side opposite to vertex B

//using law of cosines solving for c
c_polynomial=[1 -54.38 576]
root_c=roots(c_polynomial)

//case 1
c=root_c(1)
cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
B= (180/%pi)*acos (cos_B)
```

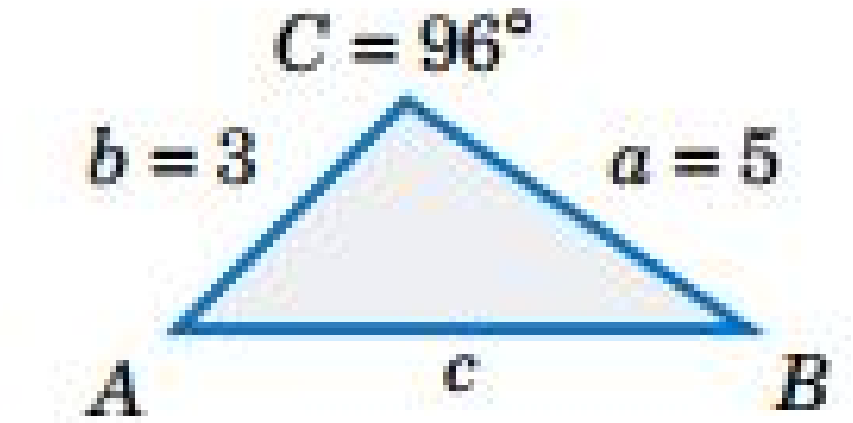

Figure 2.7: To solve the triangle when 2 sides and included angle is given

```
17 C=180-(A+B)
18 printf('1st possible answer set\nAngle B=%.1f degree
     \nAngle C=%.1f degree\nlength of c=%.0f units\n\n
     ',B,C,c)
19
20 //case 2
21 c=root_c(2)
22 cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
23 B=(180/%pi)*acos(cos_B)
24 C=180-(A+B)
25 printf('2nd possible answer set(which is not solved
     in book)\nAngle B=%.1f degree\nAngle C=%.1f
     degree\nlength of c=%.0f units\n',B,C,c)
```

Scilab code Exa 2.10 To solve the triangle when 2 sides and included angle is give

```
1 //Example 2.10
2 //To solve the triangle when 2 sides and included
     angle is given
3 clc,clear
4
5 a=5 //side oposite to vertex a
6 b=3 //side opposite to vertex b
7 C=96 //angle at vertex C
8
9 ApB=180-C //A + B
10 //using law of tangents
11  AmB =2* atand(  tand(ApB/2)*(a-b)/(a+b) )  //A-B
12
13 //solving for A and B using AmB and ApB
14 A= (AmB + ApB)/2
15 B=  ApB - A
16
17 c=a*sind(C)/sind(A) //law of sines
18 printf('Angle A= %.1f degree\nAngle B=%.1f degree\
     nlength of c=%.2f units',A,B,c)
```

Scilab code Exa 2.11 To check the solution of triangle using Mollweide equation

```
1 //Example 2.11
2 //To check the solution of triangle using Mollweide
     equation
3 clc,clear
4
5 c=6.09 //side oposite to vertex C
6 a=5 //side opposite to vertex A
7 b=3 //side opposite to vertex B
8
9 A=54.7 //angle at vertex A
```

```
10 B=29.3 //angle at vertex B
11 C=96 //angle at vertex C
12
13 LHS = (a-b)/c
14 RHS = sind((A-B)/2)/cosd(C/2)
15 printf(' LHS =  (a-b)/c                    = %.4f\n',LHS)
16 printf('  RHS =  sin((A-B)/2)/cos(C/2) = %.4f\n\n',
   RHS)
17
18 printf('Small difference in LHS and RHS is due to
   rounding off.\ni.e.Mollweides equation is holding
    true.\n')
19 printf('THE SOLUTION OF TRIANGLE IS CORRECT')
```

Scilab code Exa 2.12 To determine if a triangle can be formed with given dimension

```
1 //Example 2.12
2 //To determine if a triangle can be formed with
   given dimension
3 clc,clear
4
5 c=9 //side oposite to vertex C
6 a=6 //side opposite to vertex A
7 b=7 //side opposite to vertex B
8
9 A=55 //angle at vertex A
10 B=60 //angle at vertex B
11 C=65 //angle at vertex C
12 printf('Sum of angles=180\n')
13 printf('Smallest and largest sides are opposite to
   smallest and largest angle respectively\n\n')
14
15 LHS = (a+b)/c
16 RHS = cosd((A-B)/2)/sind(C/2)
17 printf(' LHS =  (a+b)/c                    = %.2f\n',LHS)
```

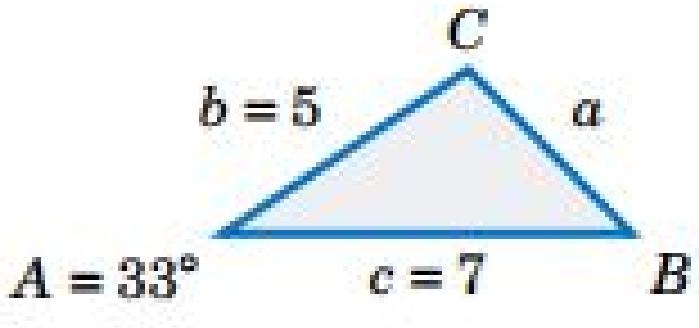

Figure 2.8: To determine area of triangle when 2 sides and an angle is given

```
18 printf(' RHS =  cos((A-B)/2)/sin(C/2) = %.2f\n\n',
     RHS)
19
20 printf('As we can see, LHS is not equal to RHS.\ni.e
     .Mollweides equation is not holding true.\n')
21 printf('THE TRIANGLE IS NOT POSSIBLE WITH GIVEN
     DIMENSIONS')
```

Scilab code Exa 2.13 To determine area of triangle when 2 sides and an angle is gi

```
1 //Example 2.13
2 //To determine area of triangle when 2 sides and an
     angle is given
3 clc,clear
4
5 c=7 //side oposite to vertex C
6 A=33 //angle at vertex A
7 b=5 //side opposite to vertex B
```

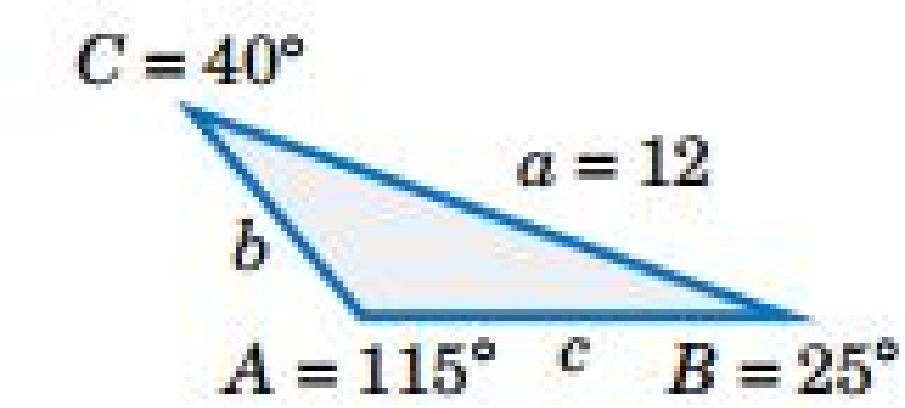

Figure 2.9: To determine area of triangle when 3 angles and a side is given

```
8
9  area_K = b*c*sind(A)/2
10 printf('Area of triangle ABC = %.2f square units',
       area_K)
```

Scilab code Exa 2.14 To determine area of triangle when 3 angles and a side is giv

```
1  //Example 2.14
2  //To determine area of triangle when 3 angles and a
       side is given
3  clc,clear
4
5  A=115 //angle at vertex A
6  a=12 //side opposite to vertex A
7  B=25 //angle at vertex B
```

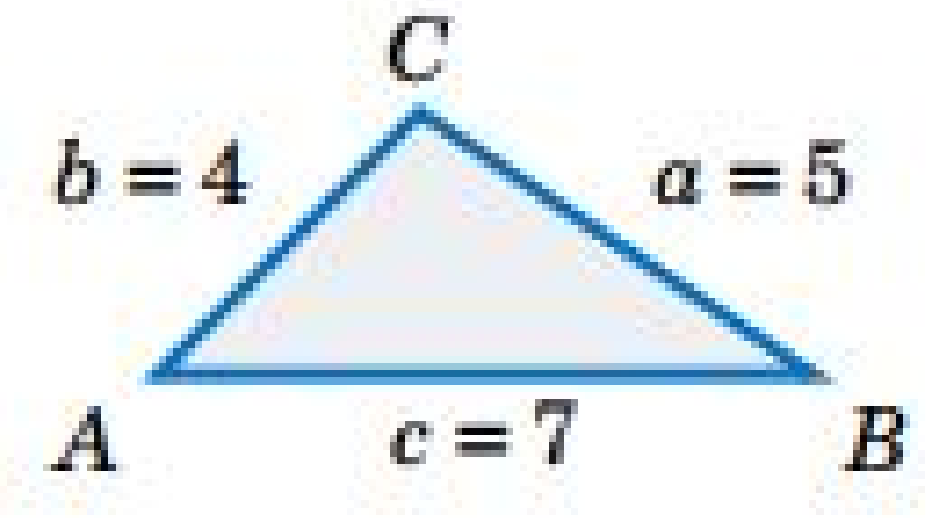

Figure 2.10: To determine area of triangle when 3 sides are given

```
8 C=40 //angle at vertex C
9
10 area_K = a^2*sind(B)*sind(C)/(2*sind(A))
11 printf('Area of triangle ABC = %.2f square units',
     area_K)
```

Scilab code Exa 2.15 To determine area of triangle when 3 sides are given

```
1 //Example 2.15
2 //To determine area of triangle when 3 sides are
     given
3 clc,clear
4
5 c=7 //side oposite to vertex C
```

```
6 a=5 //side opposite to vertex A
7 b=4 //side opposite to vertex B
8
9 s= (a+b+c)/2  //semi perimeter
10 area_K = sqrt(s*(s-a)*(s-b)*(s-c)) //using herons
    formula
11 printf('Area of triangle ABC = %.2f square units',
    area_K)
```

Scilab code Exa 2.16 To determine area of triangle when 3 sides are given

```
1 //Example 2.16
2 //To determine area of triangle when 3 sides are
    given
3 clc,clear
4
5 c=0.0000029 //side oposite to vertex C
6 a=1000000 //side opposite to vertex A
7 b=999999.9999979 //side opposite to vertex B
8
9 s= (a+b+c)/2  //semi perimeter
10 area_K = sqrt(s*(s-a)*(s-b)*(s-c)) //using herons
    formula
11 printf('Area of triangle ABC = %.3f square units\n\n
    ',area_K)
12
13 printf('Note:\n')
14 printf('In calculators like TI-83 plus, due to
    rounding off etc s will be 1000000\n')
15 printf('Therefore (s-a) is zero. And area will be
    zero according to herons formula\n')
16 printf('Due to large number of digits in scilab,(s-a
    ) is not zero. Thus, area is non-zero above.')
```

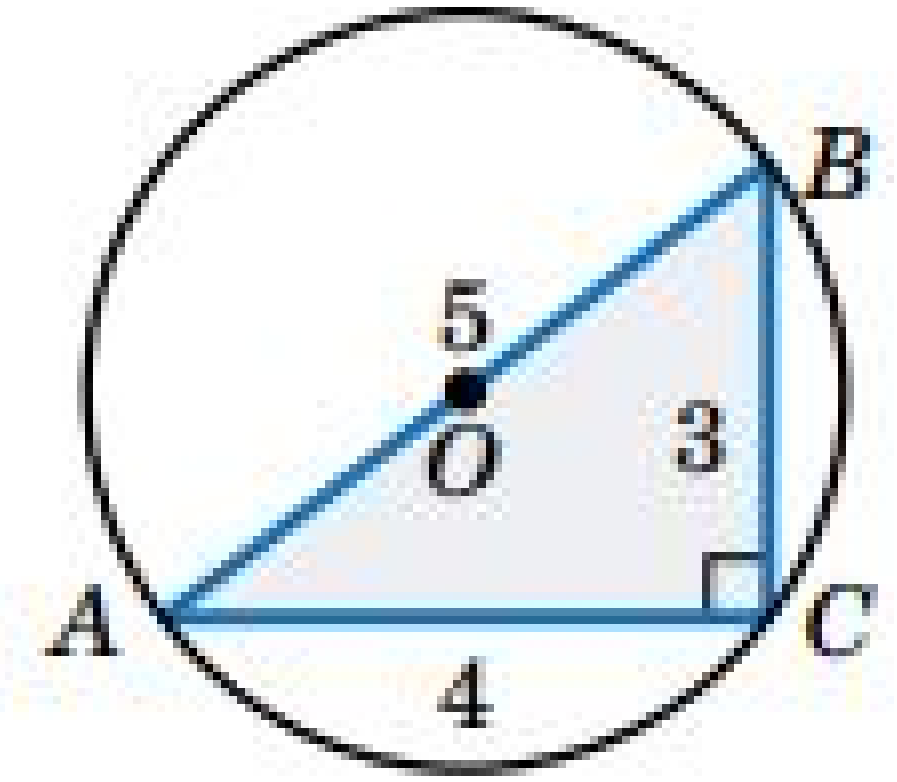

Figure 2.11: To find radius of circumscribed circle for triangle ABC

Scilab code Exa 2.17 To find radius of circumscribed circle for triangle ABC

```
1 //Example 2.17
2 //To find radius of circumscribed circle for
     triangle ABC
3 clc,clear
```

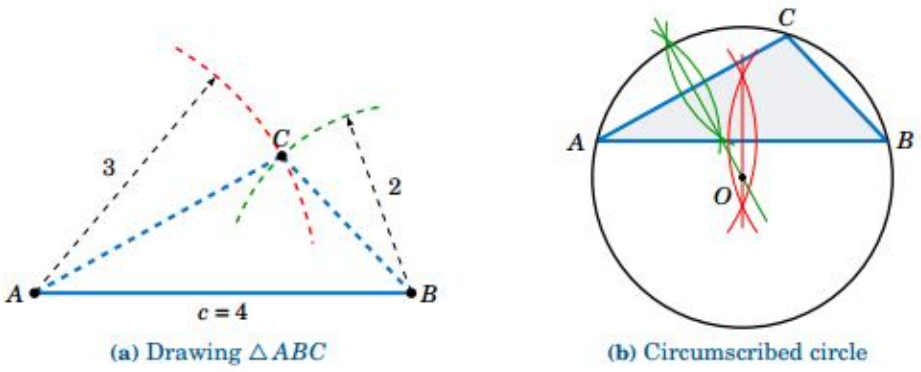

Figure 2.12: To find the radius of circumscribed circle for triangle ABC

```
4
5  c=5//side oposite to vertex C
6  a=3//side opposite to vertex A
7  b=4//side opposite to vertex B
8
9  cos_A = (c^2+b^2-a^2)/(2*c*b) //from law of cosines
10 A= acosd(cos_A)
11 diameter=(a/sind(A))
12 radius = diameter/2
13 printf('Radius of circumscribed circle = %.1f units
      \n',radius)
14 printf('\nNote :\n Diameter is same as AB i.e. c...
      So centre of circle is mipoint of AB')
```

Scilab code Exa 2.18 To find the radius of circumscribed circle for triangle ABC

```
1  //Example 2.18
2  //To find the radius of circumscribed circle for
      triangle ABC
3  clc,clear
```

```
4
5 c=4 //side oposite to vertex C
6 a=2 //side opposite to vertex A
7 b=3 //side opposite to vertex B
8
9 cos_A = (c^2+b^2-a^2)/(2*c*b) //from law of cosines
10 A= acosd(cos_A)
11 diameter=(a/sind(A))
12 radius = diameter/2
13 printf('Radius of circumscribed circle = %.2f units
   \n\n',radius)
14
15 //To draw the triangle
16 printf('NOTE:\nPROCEDURE TO DRAW THE TRIANGLE ABC\n'
   )
17 printf('Use a ruler to draw the longest side AB of
   length c = 4.\n')
18 printf('Use a compass to draw arcs of radius 3 and 2
    centered at A and B respectively.\n')
19 printf('The intersection of the arcs is the vertex C
   .\n\n')
20
21 //To draw the circumscribed circle
22 printf('PROCEDURE TO DRAW CIRCUMSCRIBED CIRCLE\n')
23 printf('Draw the perpendicular bisectors of AB and
   AC.\n')
24 printf('Their intersection is the center O of the
   circle.\n')
25 printf('Use a compass to draw the circle centered at
    O which passes through A.')
```

Scilab code Exa 2.19 To determine radius of inscribed circle ABC

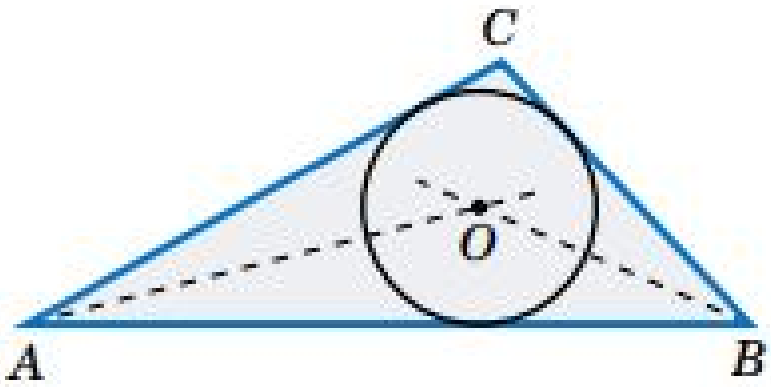

Figure 2.13: To determine radius of inscribed circle ABC

```
1  //Example 2.19
2  //To determine radius of inscribed circle ABC
3  clc,clear
4
5  c=4 //side oposite to vertex C
6  a=2 //side opposite to vertex A
7  b=3 //side opposite to vertex B
8
9  s= (a+b+c)/2  //semi perimeter
10 radius_r = sqrt((s-a)*(s-b)*(s-c)/s)
11 printf('Radius of inscribed circle is %.3f units=
      sqrt(5/12)units ',radius_r)
12 printf('\n\nNote:To obtain inscribed circle:\n(1)
      Intersect perpendicular bisectors of A and B\n')
13 printf('(2)Point of intersection is the centre of
      circle\n')
14 printf('(3)The radius is 0.645 as caculated above\n'
      )
15 printf('(4)Using a compass draw a circle with this
      centre\n')
```

Chapter 3

Identities

Scilab code Exa 3.8 To determine values of functions of sum of 2 angles when funct

```
clc,clear
//Example 3.8
//To determine values of functions of sum of 2
   angles when functions of 2 angles are given

sin_A = 4/5 ;
cos_A = 3/5 ;

sin_B = 12/13 ;
cos_B =  5/13 ;

//Apb refers to A plus B
sin_ApB = sin_A*cos_B +  cos_A*sin_B ;
cos_ApB = cos_A*cos_B -  sin_A*sin_B ;
tan_ApB = sin_ApB / cos_ApB ;

printf('sin(A+B) = %f\n',sin_ApB) ;
printf('cos(A+B) = %f\n',cos_ApB) ;
printf('tan(A+B) = %f\n',tan_ApB) ;
```

Chapter 4

Radian Measure

Scilab code Exa 4.1 To convert a degree measure to radians

```
1 clc,clear
2 //Example 4.1
3 //To convert a degree measure to radians
4
5 deg=18  //degree measure
6 radian=deg*(%pi/180) //radian measure
7 printf('Radian measure is %f rad\n(or)\n',radian)
8 printf('Radian measure is (pi/%.0f)rad',1/(radian/
    %pi))
```

Scilab code Exa 4.2 To convert a radian measure to degree

```
1 clc,clear
2 //Example 4.2
3 //To convert a radian meeasure to degree
4
5 radian=%pi/9  //radian measure
6 deg=radian/(%pi/180) //degree measure
7 printf('Degree measure is %.0f degree',deg)
```

Scilab code Exa 4.3 To determine length of the intercepted arc

```
clc,clear
//Example 4.3
//To determine length of the intercepted arc

r=2 //radius of circle
theta=1.2 //central angle in radian

s=r*theta //length of arc
printf('Length of arc intercepted = %.1f cm',s)
```

Scilab code Exa 4.4 To determine length of the arc intercepted

```
clc,clear
//Example 4.4
//To determine length of the arc intercepted

r=10 //radius of circle
theta=41*(%pi/180) //central angle in radian

s=r*theta //length of arc
printf('Length of arc intercepted = %.2f ft',s)
```

Scilab code Exa 4.5 To determine angle in radians and degrees

```
clc,clear
//Example 4.5
//To determine angle in radians and degrees
```

```
4
5 r=5 //radius of circle
6 s=2 //length of arc
7 theta = s/r //central angle in radian
8 printf('Measure of central angle = %.2f rad\n',theta
    )
9 printf('Measure of central angle = %.2f degree',
    theta*(180/%pi))
```

Scilab code Exa 4.6 To determine the length of the rope

```
1 clc,clear
2 //Example 4.6
3 //To determine the length of the rope
4
5 d=8 //distance between places in feet
6 r=2 //radius of cylinder in feet
7 //from the figure
8 DA=d/2,BE=r
9 DE=3 //distance from centre of container to wall
10
11 AE=sqrt(DE^2 + DA^2) //pythagoras theorem
12 AB=sqrt(AE^2 - BE^2) //pythagoras theorem
13
14 //all angles below are in radians
15 angle_AED = atan((d/2)/DE)
16 angle_AEB = acos(BE/AE)
17 angle_BEC = %pi - (angle_AED + angle_AEB)
18 arc_BC = BE*angle_BEC //length of arc BC
19 L = 2*(AB + arc_BC) //length of rope
20 printf('Length of the rope = %.1f ft',L)
```

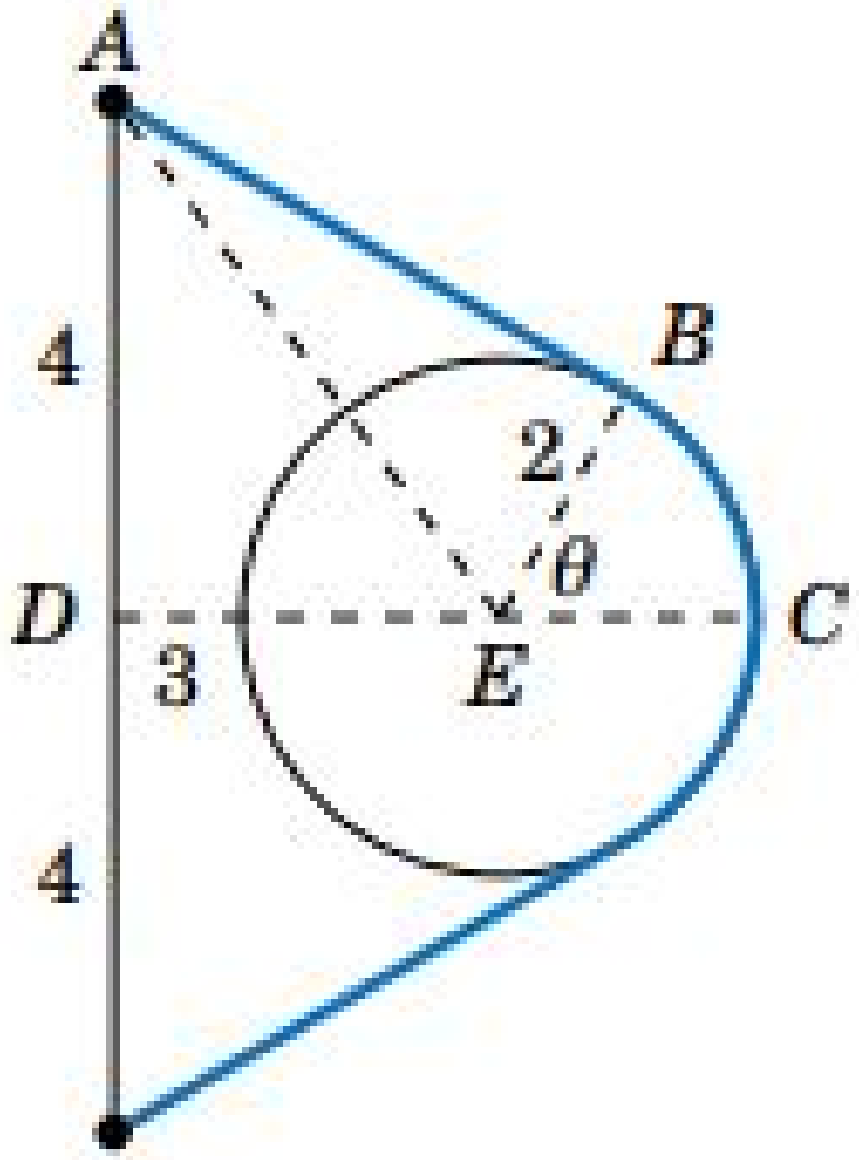

Figure 4.1: To determine the length of the rope

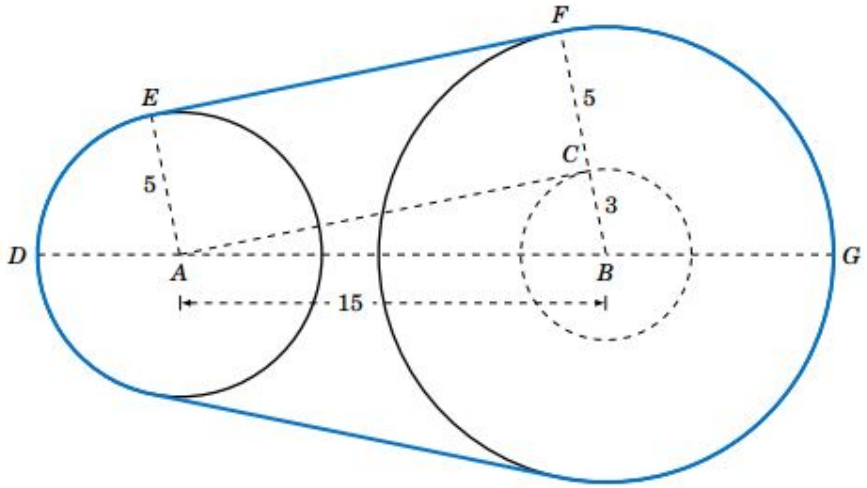

Figure 4.2: To determine the length of the belt around the pulleys

Scilab code Exa 4.7 To determine the length of the belt around the pulleys

```
clc,clear
//Example 4.7
//To determine the length of the belt around the
   pulleys

AE= 5 //radius of first pulley in cm
BF= 8 //radius of second pulley in cm
AB=15 //distance between centre of pulleys in cm

//from the figure
CF=AE //parallel side of rectangle ACFE
BC= BF- CF
AC = sqrt(AB^2 - BC^2) //by pythagoras theorem
EF=AC//parallel side of rectangle ACFE
```

```
14
15 angle_EAC = %pi/2
16 angle_BAC = asin(BC/AB)
17 angle_DAE = %pi -  angle_EAC - angle_BAC
18 angle_ABC = angle_DAE //AE and BF are parallel
19 angle_GBF= %pi - angle_ABC
20
21 arc_DE=AE*angle_ABC //length of arc DE
22 arc_FG=BF*angle_GBF //length of arc FG
23 L=2*(arc_DE + EF + arc_FG) //length of belt
24 printf('Length of belt around pulley = %f cm',L)
```

Scilab code Exa 4.8 To find the area of sector of circle

```
1 clc,clear
2 //Example 4.8
3 //To find the area of sector of circle
4
5 theta= %pi/5 //angle in radian
6 r=4 //radius in cm
7 A=r*r*theta/2 //Area of sector
8 printf('Area of sector = %.1f*pi cm^2\n(or)\n',A/%pi
   )
9 printf('Area of sector = %f cm^2',A)
```

Scilab code Exa 4.9 To determine area of sector of a circle

```
1 clc,clear
2 //Example 4.9
3 //To determine area of sector of a circle
4
5 theta= 117*(%pi/180) //angle in radian
6 r=3.5 //radius in m
```

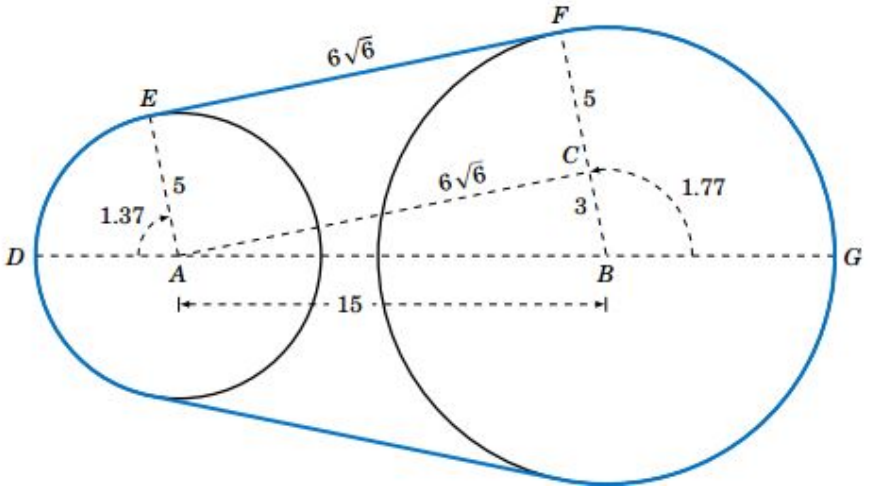

Figure 4.3: To determine area insude belt pulley system

```
7 A=r*r*theta/2 //Area of sector
8 printf('Area of sector = %.2f m^2',A)
```

Scilab code Exa 4.10 To determine area of sector of circle

```
1 clc,clear
2 //Example 4.10
3 //To determine area of sector of circle
4
5 s=6 //arc length in cm
6 r=9 //radius in cm
7 A=r*s/2 //Area of sector
8 printf('Area of sector = %.0f cm^2\n\n',A)
9 printf('Note: Angle subtended by arc = %f rad',s/r)
```

Scilab code Exa 4.11 To determine area insude belt pulley system

```
clc,clear
//Example 4.11
//To determine area insude belt pulley system

AE= 5 //radius of first pulley
BF= 8 //radius of second pulley
AB=15 //distance between centre of pulleys

//from the figure
CF=AE
BC= BF- CF
AC = sqrt(AB^2 - BC^2)
//from the figure
angle_EAC = %pi/2
angle_BAC = asin(BC/AB)
angle_DAE = %pi -  angle_EAC - angle_BAC
angle_ABC = angle_DAE //AE and BF are parallel
angle_GBF= %pi - angle_ABC

area_DAE = AE^2*angle_DAE/2 //area of sector DAE
area_GBF = BF^2*angle_GBF/2 //area of sector GBF
area_AEFC = AE*AC //area of rectangle AEFC
area_ABC = AC*BC/2 //area of triangle ABC

area_K=2*(area_DAE + area_AEFC + area_ABC + +
   area_GBF )
printf('Area enclosed by belt pulley system = %.2f
   cm^2',area_K)
printf('\n\nNote: answer differs from book due to
   approximations by them')
```

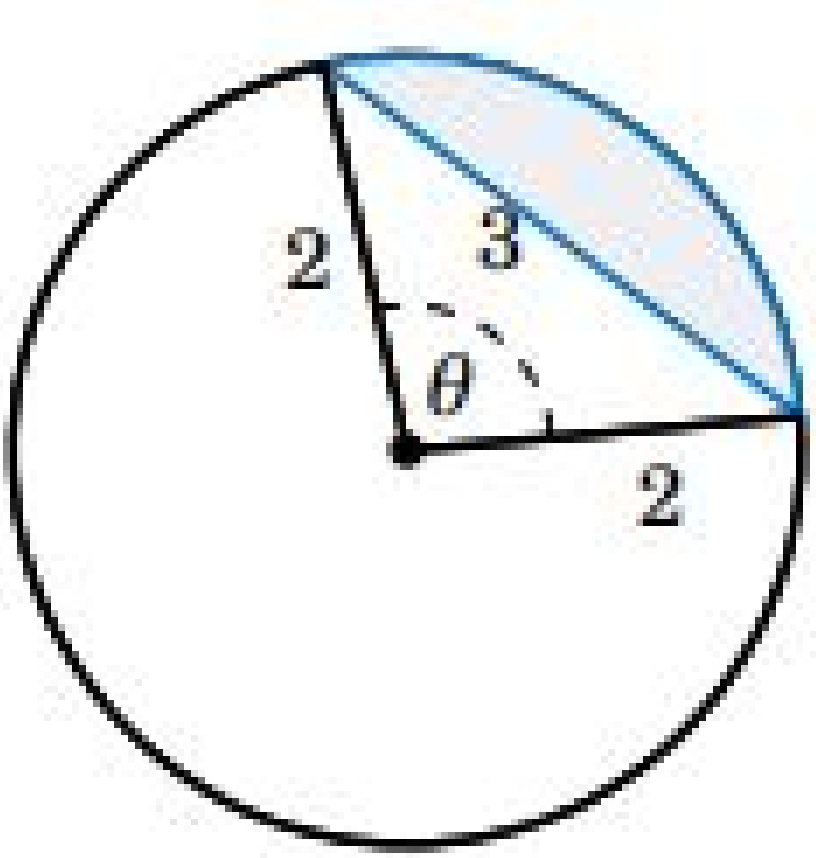

Figure 4.4: To determine area of segment formed by a chord in circle

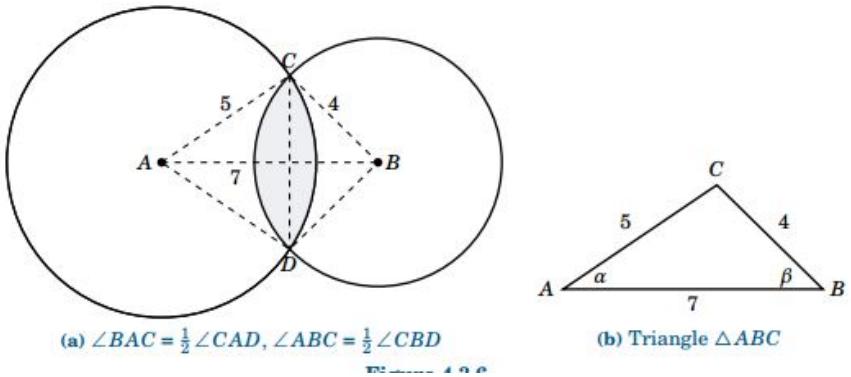

Figure 4.5: To determine area of intersection of 2 circles

Scilab code Exa 4.12 To determine area of segment formed by a chord in circle

```
1 clc,clear
2 //Example 4.12
3 //To determine area of segment formed by a chord in
     circle
4
5 radius = 2
6 chord = 3
7 //Use law of cosines
8 cos_theta = (radius^2+radius^2-chord^2)/(2*radius*
     radius)
9 theta=acos(cos_theta) //subtended central angle in
     radians
10
11 area_K=radius^2* (theta-sin(theta))/2
12 printf('Required area of segment = %.3f square units
     ',area_K)
```

Scilab code Exa 4.13 To determine area of intersection of 2 circles

```
clc,clear
//Example 4.13
//To determine area of intersection of 2 circles

d=7 //distance between centres in cm
r1= 5 //radius of first circle in cm
r2= 4 //radius of second circle in cm

//use  law of cosines
cos_alpha=(d^2+ r1^2 - r2^2 ) /(2*d*r1)
cos_beeta=(d^2+ r2^2 - r1^2 ) /(2*d*r2)

//from the geometry of the figure
//all the angles below are in radians
alpha= acos(cos_alpha)
beeta= acos(cos_beeta)
angle_BAC = alpha
angle_ABC = beeta
angle_CAD =2* angle_BAC
angle_CBD =2* angle_ABC

//required area = area at segment CD in circle at A
   and at B
area_K = r1^2*(angle_CAD-sin(angle_CAD))/2   + r2
   ^2*(angle_CBD-sin(angle_CBD))/2
printf('Area of intersection of 2 circles = %.2f cm
   ^2',area_K)
```

Scilab code Exa 4.14 To find linear and angular speed of a moving object

```
clc,clear
//Example 4.14
//To find linear and angular speed of a moving
   object

```

```
5 t=0.5 //time in second
6 r= 3 //radius in m of the circle
7 theta = %pi/3 // central angle in radian
8 w = theta/t //angular speed in rad /sec
9 v=w*r//linear speed in m/sec
10
11 printf('Angular speed= %f radian/sec\n',w)
12 printf('Linear speed = %f m/sec',v)
13
14 printf('\n\n(or)\n\nAngular speed= %f*pi radian/sec\
   n',w/%pi)
15 printf('Linear speed = %f*pi m/sec',v/%pi)
```

Scilab code Exa 4.15 To find linear and angular speed of a moving object

```
1 clc,clear
2 //Example 4.15
3 //To find linear and angular speed of a moving
   object
4
5 t=2.7 //time in second
6 r= 2 //radius in ft of the circle
7 s=35 //distance in feet
8
9 v=s/t //linear speed in ft/sec
10 w=v/r //angular speed in rad /sec
11
12 printf('Linear speed = %.2f ft/sec\n',v)
13 printf('Angular speed= %.2f radian/sec\n',w)
```

Scilab code Exa 4.16 To find the central angle swept by a moving object

```
1 clc,clear
```

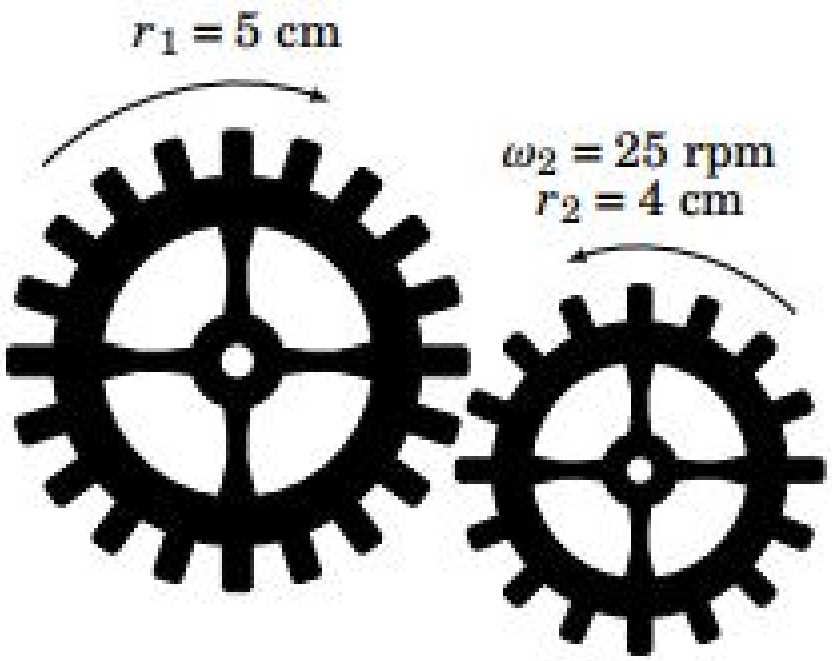

Figure 4.6: To find the angular speed of larger gear interlocked with smaller gear

```
2  //Example 4.16
3  //To find the central angle swept by a moving object
4
5  t=3.1 //time in second
6  v= 10 //linear speed in m/sec
7  r= 4 //radius in m of the circle
8  s=v*t //distance in m
9
10 theta = s/r //central angle swept
11 printf('central angle swept = %.2f radian',theta)
```

Scilab code Exa 4.17 To find the angular speed of larger gear interlocked with sma

```
clc,clear
//Example 4.17
//To find the angular speed of larger gear
   interlocked with smaller gear

r1=5 //radius of larger gear
r2=4 //radius smaller gear
w2=25 //angular speed of smaller gear

// Imagine a particle on outer radii of each gear
//At any time, for every rotation, circular
   displacement of each particle is same
// (or) s1=s2   implies  v1*t=v2*t
//v1= v2 implies w1*r1=w2*r2

w1=(w2*r2)/r1  //angular speed of larger gear
printf('Angular speed of larger gear= %.0f rpm ',w1)
```

Chapter 5

Graphing and inverse functions

Scilab code Exa 5.1 To sketch the graph of minus sinx in a given interval

```
1 //Example 5.1
2 //To sketch the graph of minus sinx in a given
     interval
3 clear,clc;
4
5 x = linspace(-0,2*%pi,50);
6 y = -sin(x) ;
7 set(gca(),"grid",[5 5]);
8 plot(x,y);
9 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
     );
10 ylabel("$y(x)=-sin(x)$","fontsize",4,"color","red");
11 title("Example 5.1","color","blue","fontsize",9);
```

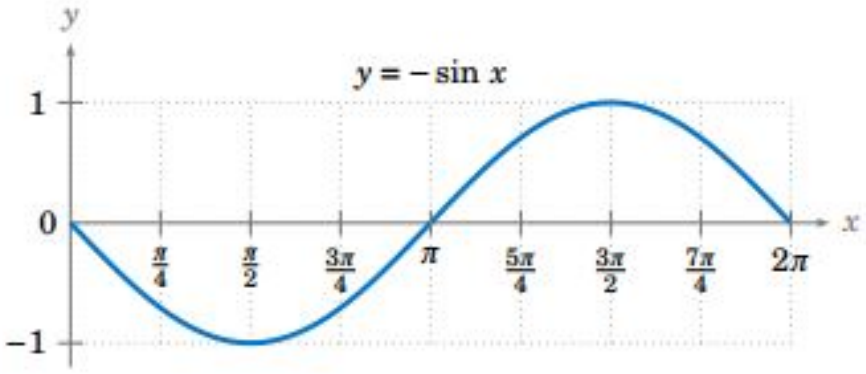

Figure 5.1: To sketch the graph of minus sinx in a given interval

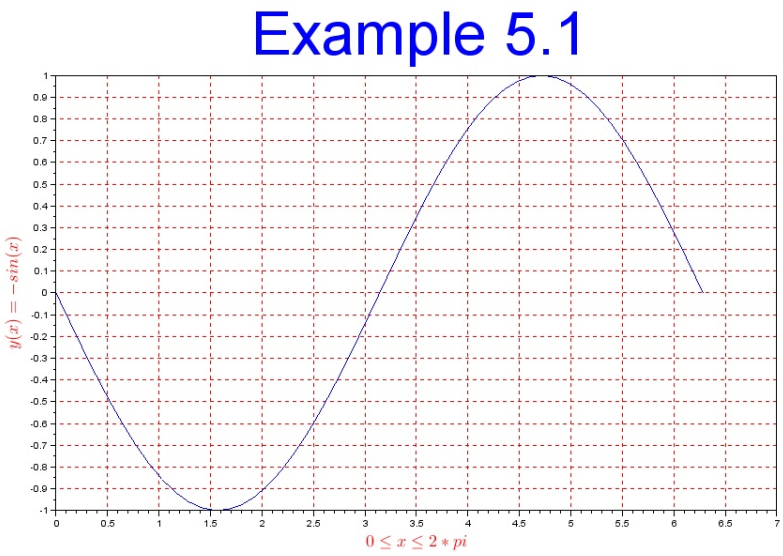

Figure 5.2: To sketch the graph of minus sinx in a given interval

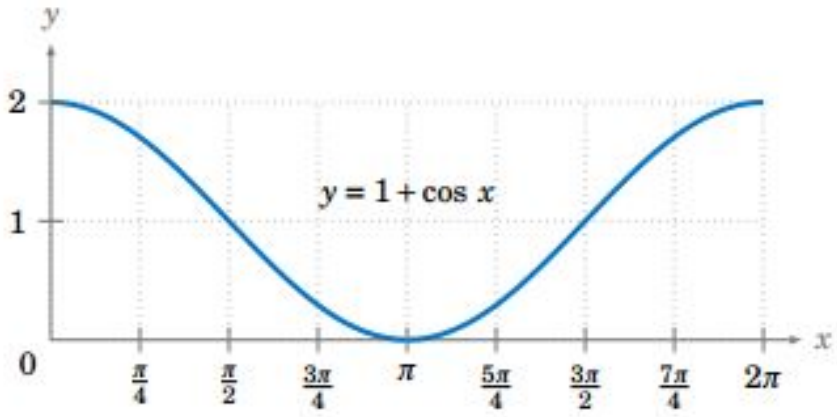

Figure 5.3: To sketch the graph of given function of in given interval

Scilab code Exa 5.2 To sketch the graph of given function of in given interval

```
1 //Example 5.2
2 //To sketch the graph of function of 1+cos(x) in
     given interval
3 clear,clc;
4
5 x = linspace(-0,2*%pi,50);
6 y = 1+cos(x) ;
7 set(gca(),"grid",[5 5]);
8 plot(x,y);
9 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
     );
10 ylabel("$y(x)=1+cos(x)$","fontsize",4,"color","red")
     ;
11 title("Example 5.2","color","blue","fontsize",9);
```

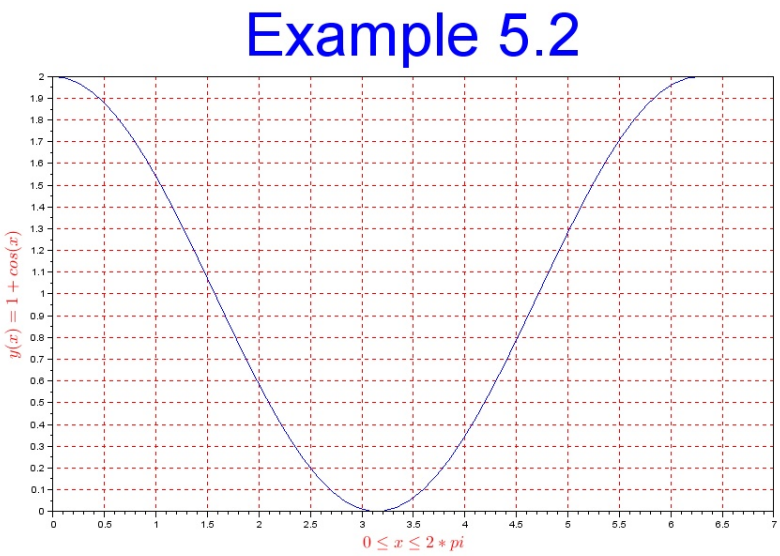

Figure 5.4: To sketch the graph of given function of in given interval

Scilab code Exa 5.4 To determine the period of given sinusoidal function

```
1 //Example 5.4
```

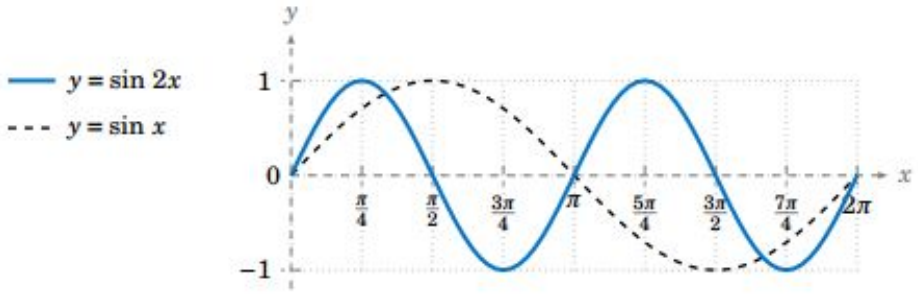

Figure 5.5: To determine the period of given sinusoidal function

```
2 //To determine the period of given sinusoidal
    function
3 clear,clc;
4
5 multiple = 2; //multiplicity of angle
6 period_sinx = 2*%pi ;//period of sin(x) in radians
7 required_period = period_sinx / multiple;
8 printf('Required period is %f radians',
    required_period);
9
10 //Note that sin 2x  goes  twice as fast  as sin x
    .
11 //While sin x takes a full 2*pi radians to go
    through an entire cycle
12 //sin 2x goes through an entire cycle in just pi
    radians
13
14 x = linspace(-0,2*%pi,100);
15 y = sin(2*x) ;
16 z = sin(x) ;
17 set(gca(),"grid",[4 4]);
18 plot(x,y,'r-');
19 plot(x,z,'b-');
20 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
    );
21 ylabel("$y(x)=sin(2x)$","fontsize",4,"color","red");
22 title("Example 5.4","color","blue","fontsize",9);
23 legend(["sin(2x)";"sin(x)"]);
```

Scilab code Exa 5.5 To determine the period of 2 given cosine functions

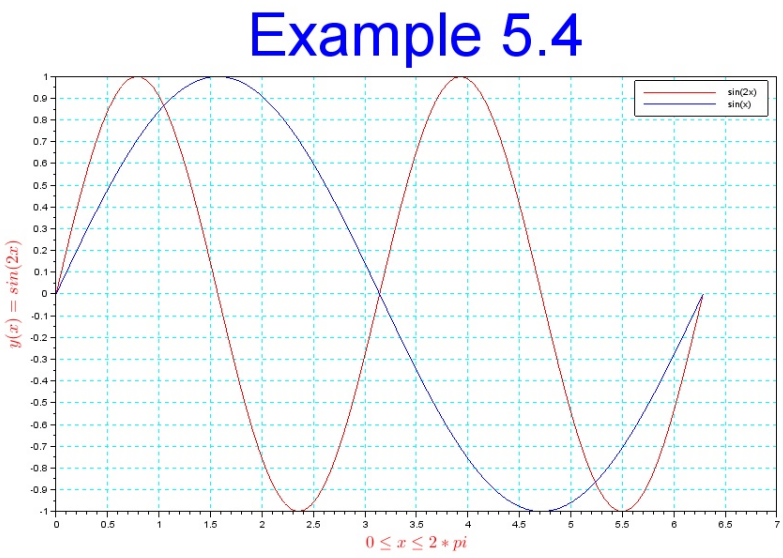

Figure 5.6: To determine the period of given sinusoidal function

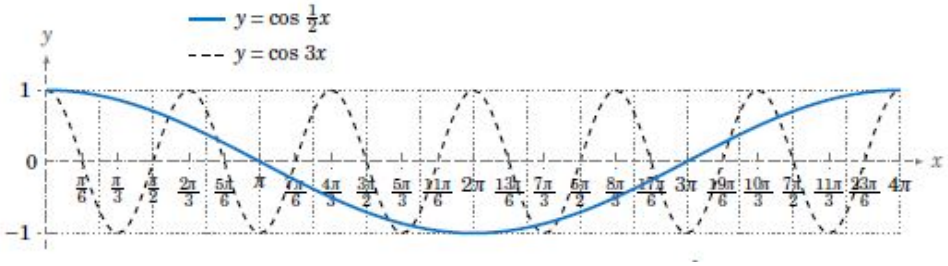

Figure 5.7: To determine the period of 2 given cosine functions

```
1  //Example 5.5
2  //To determine the period of 2 given cosine
       functions
3  clear,clc;
4
5  //y=cos(3*x)
6  multiple = 3; //multiplicity of angle
7  period_cosx = 2*%pi; //period of sin(x) in radians
8  required_period = period_cosx / multiple;
9  printf('Period of cos(3*x)is %f radians\n',
       required_period);
10
11 //y=cos(0.5*x)
12 multiple = 1/2; //multiplicity of angle
13 period_cosx = 2*%pi; //period of sin(x) in radians
14 required_period = period_cosx / multiple;
15 printf('Period of cos(x/2)is %f radians',
       required_period);
16
17 x = linspace(-0,4*%pi,200);
18 y = cos(3*x) ;
19 z = cos(x/2) ;
20 set(gca(),"grid",[4 4]);
21 plot(x,y,'r-');
22 plot(x,z,'b-');
23 xlabel("$0\le x\le 4*pi$","fontsize",4,"color","red"
       );
24 title("Example 5.5","color","blue","fontsize",9);
25 legend(["y = cos(3x)";"y = cos(x/2)"]);
```

Scilab code Exa 5.6 To determine the amplitude and period of given function

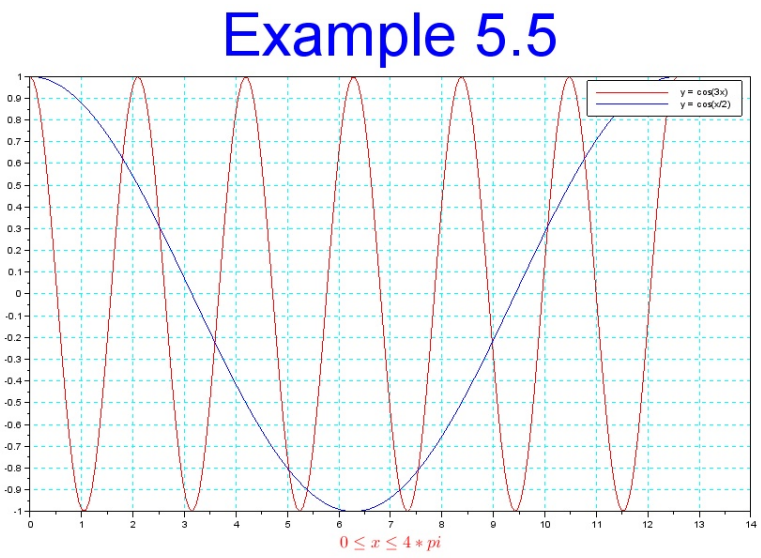

Figure 5.8: To determine the period of 2 given cosine functions

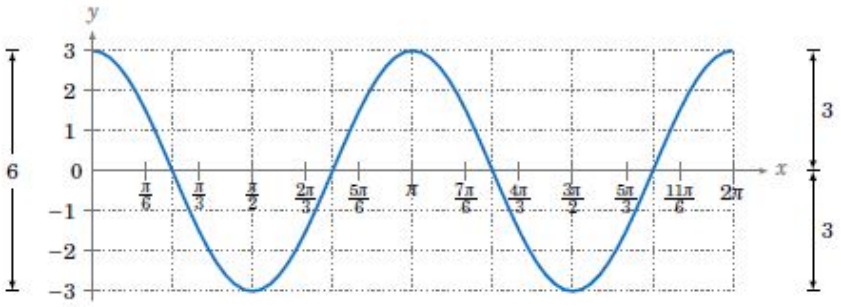

Figure 5.9: To determine the amplitude and period of given function

```
1 //Example 5.6
2 //To determine the amplitude and period of given
     function
3 clear,clc;
4
5 x = linspace(-0,4*%pi,200);
6 y = 3*cos(2*x) ; //given function
7 amplitude = y/cos(2*x);
8 printf('Amplitude = %f',amplitude);
9
10 multiple = 2; //multiplicity of angle
11 period_cosx=2*%pi; //period od cos(x)
12 period_required = period_cosx / multiple;
13 printf('\nPeriod = %f radians',period_required);
14
15 x = linspace(-0,2*%pi,50);
16 y = 3*cos(2*x) ;
17 set(gca(),"grid",[5 5]);
18 plot(x,y);
19 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
     );
20 ylabel("$y(x)=3*cos(2*x)$","fontsize",4,"color","red
     ");
21 title("Example 5.6","color","blue","fontsize",9);
```

Scilab code Exa 5.7 To find amplitude and period of given composite function

```
1 //Example 5.7
2 //To find amplitude and period of given composite
     function
3 clear,clc;
```

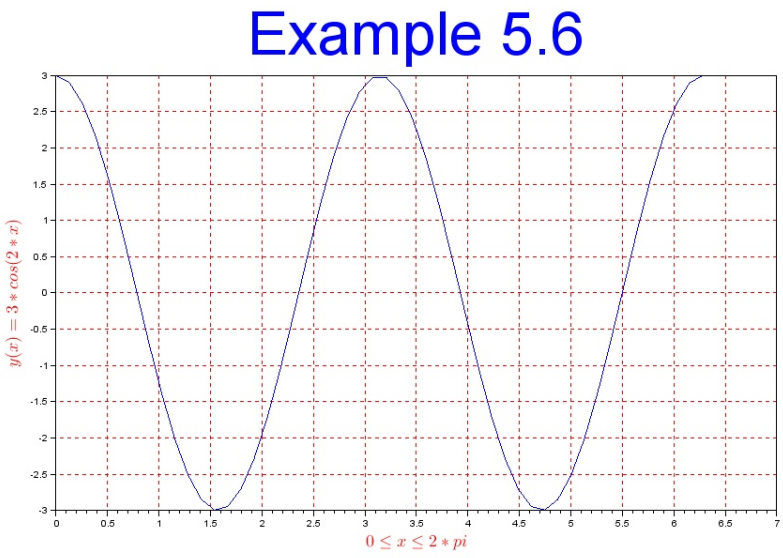

Figure 5.10: To determine the amplitude and period of given function

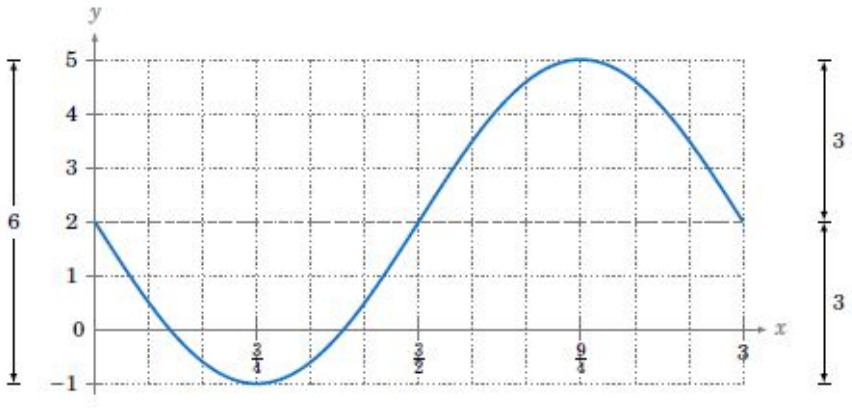

Figure 5.11: To find amplitude and period of given composite function

```
4
5  x = linspace(0,3,200);
6  y1=2 ;//1st part of given function
7  amplitude1=y1 ;//amplitude numerically same for
       constant function
8  y2= -3*sin((2*%pi/3)*x); //second part of given
       function
9  amplitude2 = abs(y2/sin((2*%pi/3)*x)) ;//amplitude
       of part 2
10 //Note: adding 2 doesnt change ampitude
11 //It just causes the upward shift of graph
12 maax =amplitude1 + amplitude2; //altered maximum due
        to adding of 2
13 minn =amplitude1 - amplitude2;//altered minimum due
       to adding of 2
14 amplitude = (maax-minn)/2;//required amplitude
15 printf('Amplitude = %f',amplitude);
16
17 multiple=2*%pi/3 ;//multiplicity of angle
18 period_sinx=2*%pi;//period of sin_x
19 period_required = period_sinx/ multiple;
20 printf('\nRequired period is %f radians',
       period_required);
21
22 x = linspace(0,3,200);
23 y = 2 -3*sin((2*%pi/3)*x)
24 set(gca(),"grid",[5 5]);
25 plot(x,y);
26 xlabel("$0\le x\le 3$","fontsize",4,"color","red");
27 ylabel("$y(x)= 2 -3*sin((2*%pi/3)*x)$","fontsize",4,
       "color","red");
28 title("Example 5.7","color","blue","fontsize",9);
```

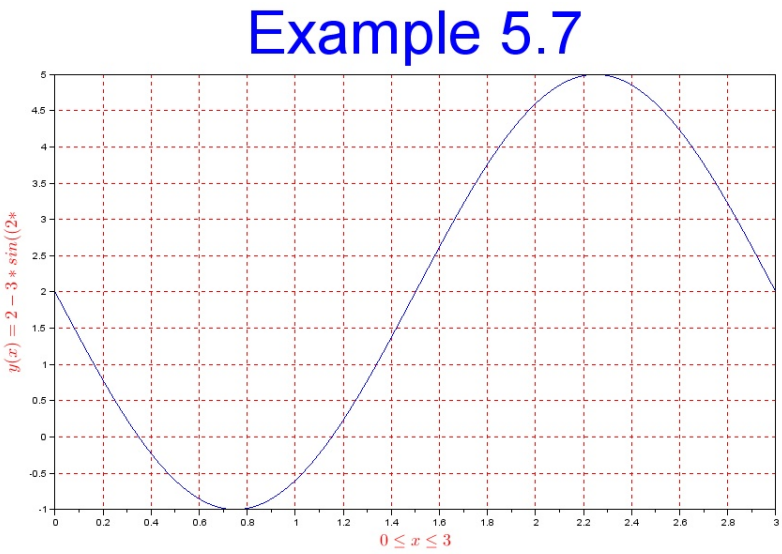

Figure 5.12: To find amplitude and period of given composite function

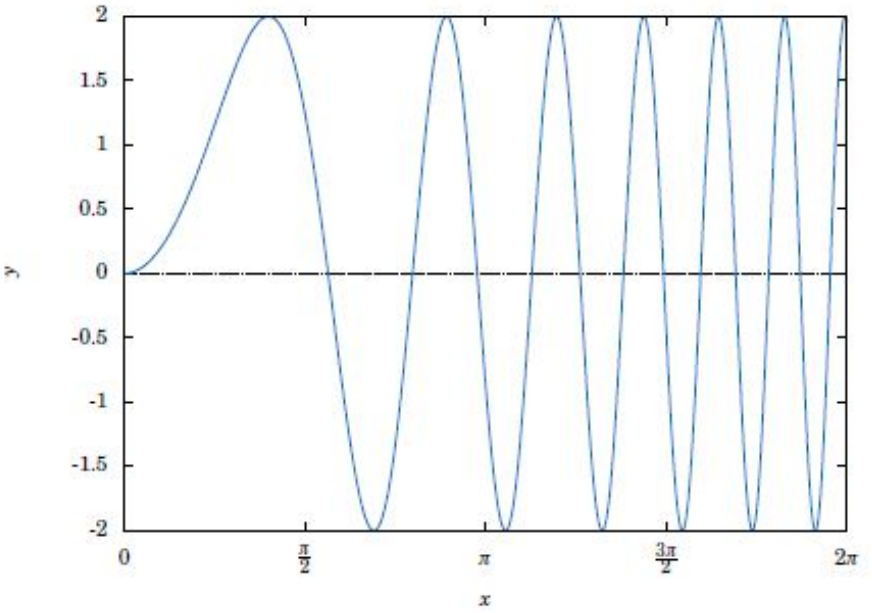

Figure 5.13: To find the amplitude and period of given function

Scilab code Exa 5.8 To find the amplitude and period of given function

```
1 //Example 5.8
2 //To find the amplitude and period of given function
3 clear,clc;
4
5 //Period
6 printf('PERIOD:\n')
7 printf('This isnt a periodic function as x^2 is
     linearly related to x \n')
8 printf('and hence period doesnt exist\n')
9
10 //Amplitude
11 x = linspace(0,2*%pi,200);
12 y = 2*sin(x^2);
13 amplitude= y/sin(x^2);
14 printf('AMPLITUDE:\n')
15 printf('Amplitude exists unlike period\n')
16 printf('Because sine component of the given function
      never exceeds 1\n')
17 printf('Function value is always ;ess than constant
     factor adjacent to sine\n')
18 printf('Hence amplitude is the constant factor
     multiplied with sine component\n\n')
19 printf('Amplitude = %f as calculated\n',amplitude)
20
21
22 x = linspace(0,2*%pi,200);
23 y = 2*sin(x^2) ;
24 set(gca(),"grid",[5 5]);
25 plot(x,y,'b');
26 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
     );
27 ylabel("$y(x)= 2*sin(x^2)$","fontsize",4,"color","
```

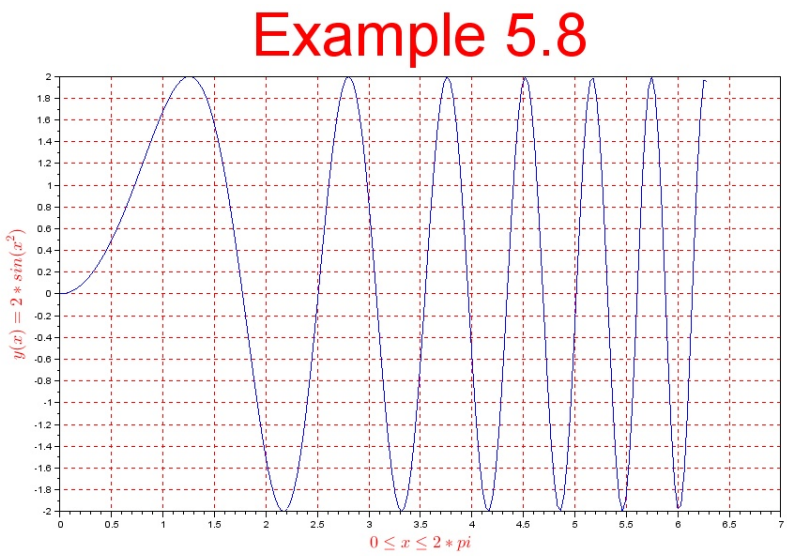

Figure 5.14: To find the amplitude and period of given function

```
    red");
28 title("Example 5.8","color","red","fontsize",9);
```

Scilab code Exa 5.9 To find the amplitude and period of given function

```
1 //Example 5.9
2 //To find the amplitude and period of given function
3 clear,clc;
4
5 //amplitude
6 x = linspace(-0,4*%pi,200);
7 y1=3*sin(x); //1st part of given function
8 amplitude1=y1/sin(x); //amplitude of part 1
```

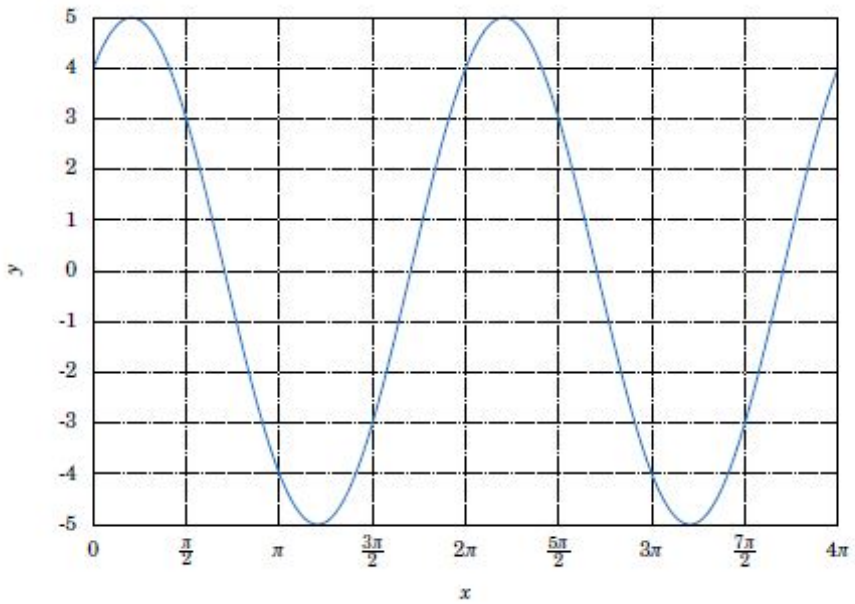

Figure 5.15: To find the amplitude and period of given function

```
y2= 4*cos(x); //second part of given function
amplitude2 =y2/(cos(x)); //amplitude of part 2

//given function is a composition of 2 functions
//Using trigonometric identities, merge them into 1
//the amplitude of resultant is the required
   amplitude
//In this case the merged function can be sine or
   cos
//merging sine and cos into sine,
amplitude = sqrt(amplitude1^2 + amplitude2^2);
printf('Amplitude = %f',amplitude);

//period
period_cosx=2*%pi ;//period of cos(x) is 2 pi
period_sinx=2*%pi ;//period of sin(x) is 2 pi
locm = 2*%pi; //lcm of period_sinx and period_cosx
printf('\nRequired period is %f radians',locm);
```

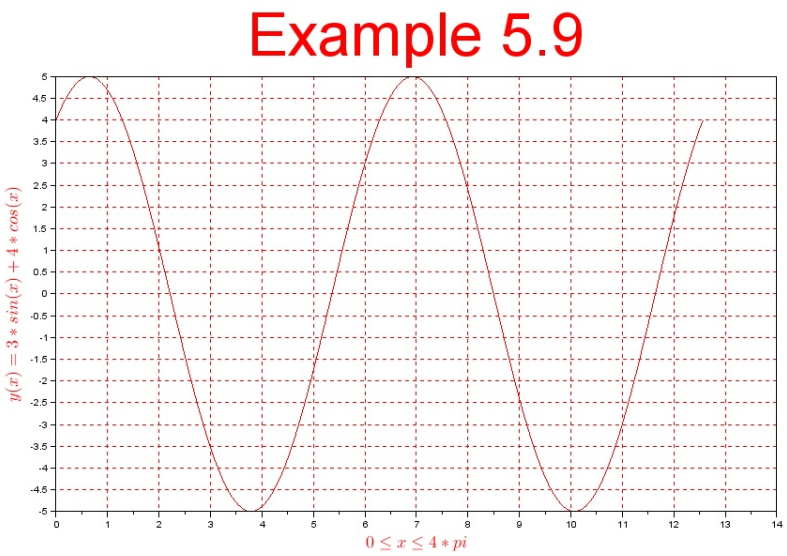

Figure 5.16: To find the amplitude and period of given function

```
25 x = linspace(0,4*%pi,200);
26 y = 3*sin(x) +4*cos(x);
27 set(gca(),"grid",[5 5]);
28 plot(x,y,'r');
29 xlabel("$0\le x\le 4*pi$","fontsize",4,"color","red"
   );
30 ylabel("$y(x)= 3*sin(x) +4*cos(x)$","fontsize",4,"
   color","red");
31 title("Example 5.9","color","red","fontsize",9);
```

Scilab code Exa 5.10 To find the period of given function

```
1 //Example 5.10
```

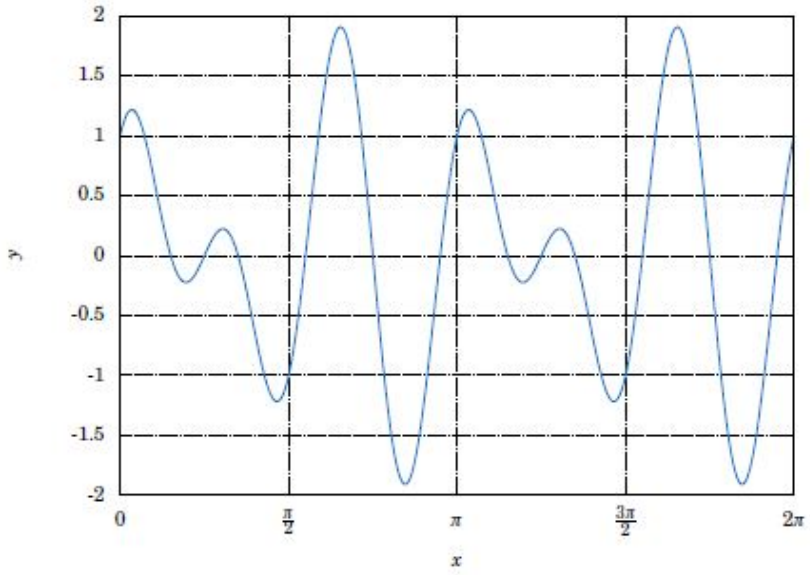

Figure 5.17: To find the period of given function

```
2  //To find the period of given function
3  clear,clc;
4
5  x = linspace(-0,2*%pi,200);
6  y1=cos(6*x); //1st part of given function
7  multiple1=6;//multiplicity of angle
8  period_cosx=2*%pi ;//period of cos(x) is 2 pi
9  period1= period_cosx/ multiple1;
10 printf('Note: Period of cos(%d*x)= %f radians\n',
     multiple1,period1);
11 y2= sin(4*x); //second part of given function
12 multiple2=4; //multiplicity of angle
13 period_sinx=2*%pi ;//period of sin(x) is 2 pi
14 period2= period_sinx/multiple2;
15 printf('Period of sin(%d*x)= %f radians\n',multiple2
     ,period2);
16
17 locm = %pi ;//LCM of period1 and period 2
```

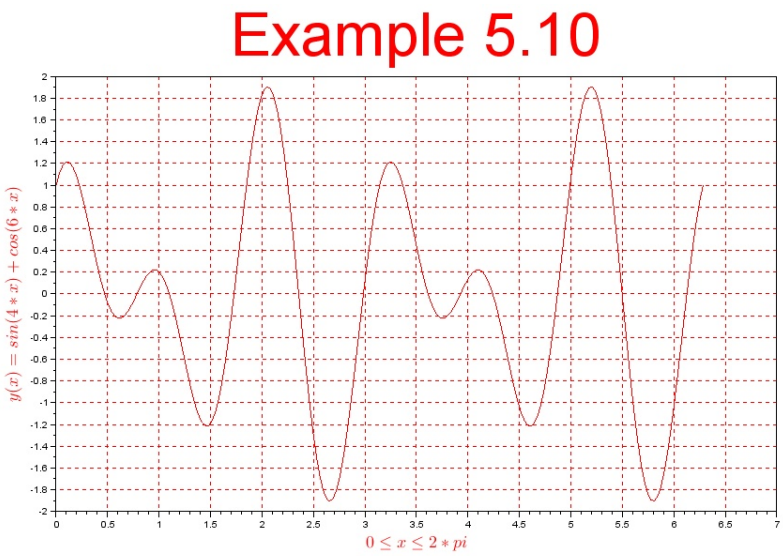

Figure 5.18: To find the period of given function

```
18 period = locm ;//final period
19 printf('\nRequired period is %f radians',period);
20 x = linspace(0,2*%pi,200);
21 y = cos(6*x)+ sin(4*x);//given function
22 set(gca(),"grid",[5 5]);
23 plot(x,y,'r');
24 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
   );
25 ylabel("$y(x)= sin(4*x) +cos(6*x)$","fontsize",4,"
   color","red");
26 title("Example 5.10","color","red","fontsize",9);
```

Scilab code Exa 5.11 To find the amplitude phase shift and period of given functio

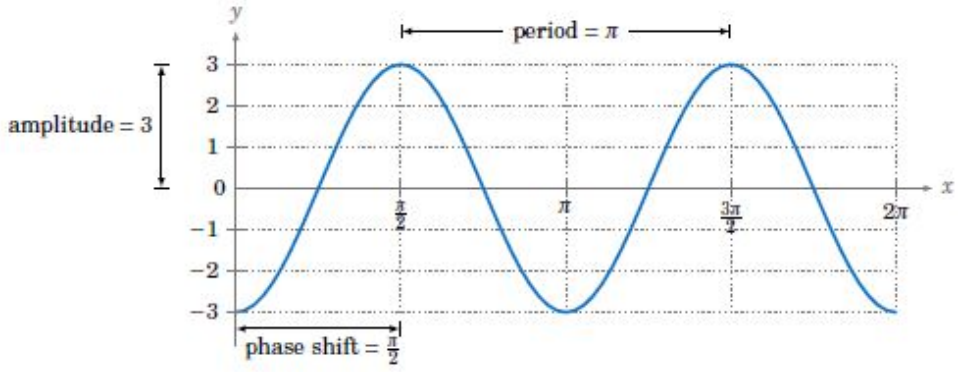

Figure 5.19: To find the amplitude phase shift and period of given function

```
1  //Example 5.11
2  //To find the amplitude phase shift and period of
       given function
3  clear,clc;
4
5  x = linspace(-0,2*%pi,200);
6  deviation = %pi; //deviation from multiples of x
7  y = 3*cos(2*x- deviation) ;//given function
8  amplitude = y/ cos(2*x- %pi) ;
9  printf('Amplitude = %f\n',amplitude);
10 multiple = 2; //multiplicity of angle
11 period_cosx = 2*%pi; //period of sin(x) in radians
12 required_period = period_cosx / multiple;
13 printf('Required period = %f radians\n',
       required_period);
14 phase_shift = deviation / multiple;
15 printf('Phase shift = %f radians',phase_shift);
16
17 x = linspace(-0,2*%pi,200);
18 y = 3*cos(2*x- %pi) ;
19 set(gca(),"grid",[5 5]);
20 plot(x,y);
21 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
       );
```

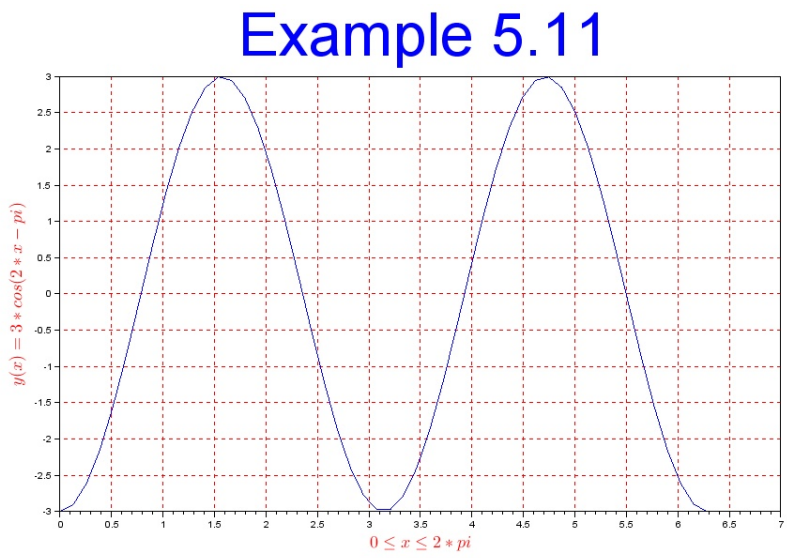

Figure 5.20: To find the amplitude phase shift and period of given function

```
22 ylabel("$y(x)=3*cos(2*x- pi)$","fontsize",4,"color",
   "red");
23 title("Example 5.11","color","blue","fontsize",9);
```

Scilab code Exa 5.12 To find the amplitude phase shift and period of given functio

```
1 //Example 5.12
2 //To find the amplitude phase shift and period of
   given function
3 clear,clc;
4
5 x = linspace(-%pi/6,4*%pi/3,200);
6 deviation = -%pi/2;//deviation from multiples of x
```

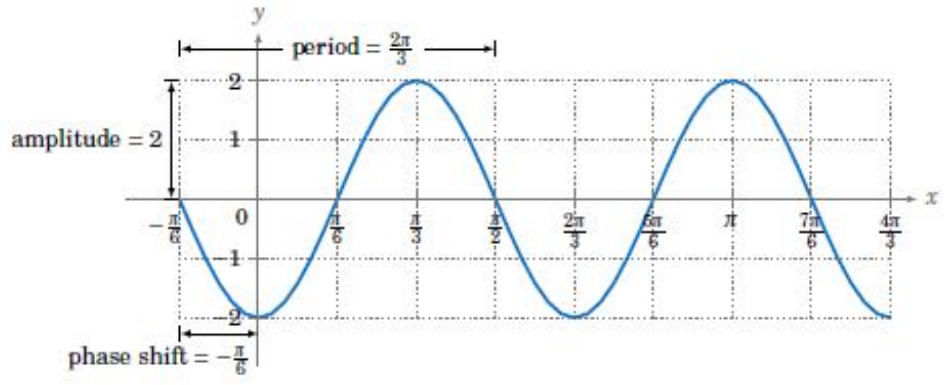

Figure 5.21: To find the amplitude phase shift and period of given function

```
y = -2*sin(3*x- deviation) ;// given function
amplitude = abs(y/(sin(3*x- deviation)) );
printf('Amplitude = %f\n',amplitude);
multiple = 3; //multiplicity of angle
period_sinx = 2*%pi ;//period of sin(x) in radians
required_period = period_sinx / multiple;
printf('Required period = %f radians\n',
   required_period);
phase_shift = deviation / multiple;
printf('Phase shift = %f radians',phase_shift);

x = linspace(-%pi/6,4*%pi/3,200);
y =-2*sin(3*x+ %pi/2) ;
set(gca(),"grid",[5 5]);
plot(x,y);
ylabel("$y =-2*sin(3*x+ %pi/2)$","fontsize",4,"color
   ","red");
title("Example 5.12","color","blue","fontsize",9);
```

Scilab code Exa 5.13 To determine inverse sine function of a given value

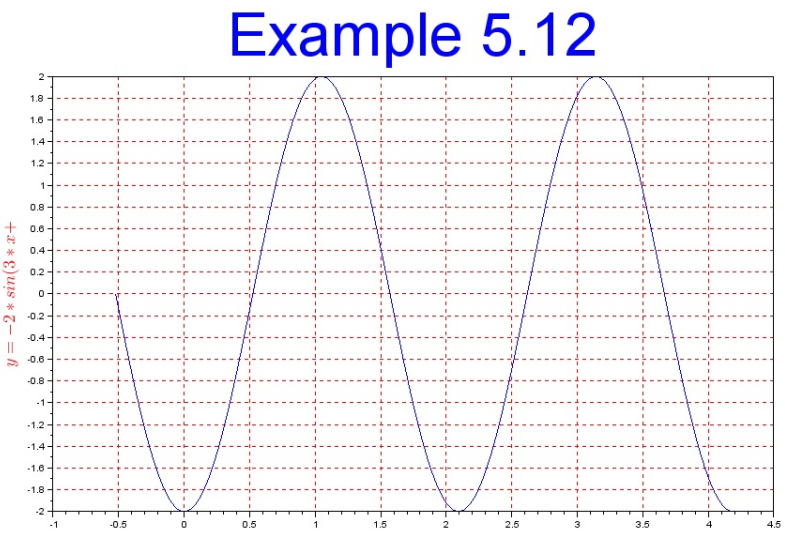

Figure 5.22: To find the amplitude phase shift and period of given function

```
1 clc,clear;
2 //Example 5.13
3 //To determine inverse sine function of a given
     value
4
5 given =  sin(%pi/4); //given value
6 answer= asin(given); //final answer
7
8 printf('Required answer is %f radians',answer);
9 printf('\n\nOR \n\n(pi/4)*%f radians',answer*(4/%pi)
     );
```

Scilab code Exa 5.14 To determine inverse sine function of a given value

```
1 clc,clear;
2 //Example 5.14
```

```
3 //To determine inverse sine function of a given
     value
4
5 given =  sin(5*%pi/4); //given value
6 answer= asin(given); //final answer
7
8 printf('Required answer is %f radians',answer);
9 printf('\n\nOR \n\n(pi/4)*%f radians',answer*(4/%pi)
     );
```

Scilab code Exa 5.15 To determine inverse cosine function of a given value

```
1 clc,clear;
2 //Example 5.15
3 //To determine inverse cosine function of a given
     value
4
5 given =  cos(%pi/3); //given value
6 answer= acos(given); //final answer
7
8 printf('Required answer is %f radians',answer);
9 printf('\n\nOR \n\n(pi/3)*%f radians',answer*(3/%pi)
     );
```

Scilab code Exa 5.16 To determine inverse cosine function of a given value

```
1 clc,clear;
2 //Example 5.16
3 //To determine inverse cosine function of a given
     value
4
5 given =  cos(4*%pi/3); //given value
6 answer= acos(given); //final answer
```

```
7
8 printf('Required answer is %f radians',answer);
9 printf('\n\nOR \n\n(pi/3)*%f radians',answer*(3/%pi)
    );
```

Scilab code Exa 5.17 To determine inverse tan function of a given value

```
1 clc,clear;
2 //Example 5.17
3 //To determine inverse tan function of a given value
4
5 given =  tan(%pi/4); //given value
6 answer= atan(given); //final answer
7
8 printf('Required answer is %f radians',answer);
9 printf('\n\nOR \n\n(pi/4)*%f radians',answer*(4/%pi)
    );
```

Scilab code Exa 5.18 To determine inverse tan function of a given value

```
1 clc,clear;
2 //Example 5.18
3 //To determine inverse tan function of a given value
4
5 given =  tan(%pi); //given value
6 answer= atan(given);//final answer
7
8 printf('Required answer is %f radians',answer);
```

Scilab code Exa 5.19 To determine exact value of given expression involving invers

```
clc,clear;
//Example 5.19
//To determine exact value of given expression
   involving inverse trigonometric functions

expression=  cos(asin(-1/4)); //given expresion

printf('Value of given expression is %f radians',
   expression);
```

Chapter 6

Additional Topics

Scilab code Exa 6.3 To solve the given equation

```
1 clc,clear
2 //Example 6.3
3 //To solve the given equation
4
5 sec_theta = 1/2
6 cos_theta = 1 / sec_theta
7 printf('cos(theta) = %f as calculated\n',cos_theta)
8 printf('But value of cos function can never exceed
     unity\n')
9 printf('Thus, NO SOLUTION exists')
```

Scilab code Exa 6.4 To solve the given equation

```
1 clc,clear
2 //Example 6.4
3 //To solve the given equation
4
5 //Given equation is cos_theta = tan_theta
```

```
6 //simplyfying given equation, we get
7 //(sin_theta)^2 + sin_theta - 1 = 0
8 //Solve for sin_theta as follows
9 p=[1 1 -1]
10 sin_theta= roots(p)
11 printf('Values of sin(theta) after simplifying and
   solving = %f and %f\n',sin_theta(1),sin_theta(2))
12 printf('Eliminate %f as sin_theta cant be below -1',
   sin_theta(1))
13
14 //Since sin_theta is +ve, 2 solutions exist. in 1st
   and 2nd quadrant
15 theta_1=asin(sin_theta(2));  //in 1st quadrant
16 theta_2=%pi-asin(sin_theta(2));//the reflection in 2
   nd quadrant
17
18 printf('\n\nSOLUTIONS:\n')
19 printf('%f radians\n%f radians',theta_1,theta_2)
20
21 printf('\n\nGENERAL SOLUTIONS:\n')
22 printf('%f + integer multiples of 2pi \n',theta_1)
23 printf('%f + integer multiples of 2pi \n',theta_2)
```

Scilab code Exa 6.9 To find the result of basic operations on 2 given complex numb

```
1 clc,clear
2 //Example 6.9
3 //To find the result of basic operations on 2 given
   complex numbers
4
5 z1 = complex(-2,3)
6 z2 = complex(3,4)
7
8 summ = z1+z2
9 difference = z1-z2
```

```
10 product = z1*z2
11 ratio = z1/z2
12 mag_z1= abs(z1) //modulus of z1
13 mag_z2= abs(z2)//modulus of z2
14 //printf('Note: Please go through complex nos scilab
       syntaxes to comprehend this example code\n\n')
15 printf('z1 + z2 = %.0f + %.0f*i\n',real(summ),imag(
   summ))
16 printf('z1 - z2 = %.0f + %.0f*i\n',real(difference),
   imag(difference))
17 printf('z1 * z2 = %.0f + %.0f*i\n',real(product),
   imag(product))
18 printf('z1 / z2 = %f + %f*i\n',real(ratio),imag(
   ratio))
19 printf('|z1|= sqrt(%.0f)= %f \n',mag_z1^2,mag_z1)
20 printf('|z2| = %.0f',mag_z2)
```

Scilab code Exa 6.10 To represent given complex number in trigonometric form

```
1 clc,clear
2 //Example 6.10
3 //To represent given complex number in trigonometric
      form
4
5 z=-2 + -1*%i ;//given number
6 x=real(z) ;//real part
7 y=imag(z) ;//imaginary part
8
9 //theta is in third quadrant as x and y are -ve
10 theta=180 + atand(y/x);
11 r=sqrt(x^2+y^2) ;//modulus of z
12 printf('z= %f + i* %f  can be written as: \n',real(z
   ),imag(z))
13 printf('z = sqrt(%.0f)*(cos(%.1f)+i*sin(%.1f))',r^2,
   theta,theta)
```

Scilab code Exa 6.11 To determine product and ratio of complex numbers using formu

```
clc,clear
//Example 6.11
//To determine product and ratio of complex numbers
   using formula

//given values
z1 = 6*(cosd(70)+ %i*sind(70));
z2 = 2*(cosd(31)+ %i*sind(31));

//arguements of complex numbers
theta1=phasemag(z1);
theta2=phasemag(z2);
//modulus of complex numbers
r1=abs(z1);
r2=abs(z2);
theta_1p2 =theta1 + theta2  ;//theta1 + theta 2
theta_1m2 =theta1 - theta2  ;//theta1 - theta 2
//according to the formula used in book
product = r1*r2*(cosd(theta_1p2)+%i*sind(theta_1p2))
   ;
ratio = (r1/r2)*(cosd(theta_1m2)+%i*sind(theta_1m2))
   ;

printf('z1*z2 = %.0f*(cos(%.0f)+i*sin(%.0f))\n',r1*
   r2,phasemag(product),phasemag(product))
printf('z1/z2 = %.0f*(cos(%.0f)+i*sin(%.0f))\n',r1/
   r2,phasemag(ratio),phasemag(ratio))
```

Scilab code Exa 6.12 To find higher powers of complex number using demoivre theore

```
clc,clear
//Example 6.12
//To find higher powers of complex number using
   demoivre theorem

z= complex(1,1);
r= abs(z);//modulus of z
theta=phasemag(z) ;//arguement of z
power=10;
//using demoivre formula
answer= (r^power)*(cosd(theta*power)+%i*sind(theta*
   power));
//printf('(1+i)^10 = (%.0f)*(cos(%.0f)+ i*sin(%.0f))
   ',r^power,theta*power,theta*power);
printf('\n %.0f + %.0f*i',real(answer),imag(answer))
   ;
printf('\n(OR)\n %.0f*i',imag(answer));
```

Scilab code Exa 6.13 To determine the cube roots of i

```
clc,clear
//Example 6.13
//To determine the cube roots of i

z=%i //given complex number
//modulii for cuberoots
r1=abs(z)^(1/3)
r2=abs(z)^(1/3)
r3=abs(z)^(1/3)

//arguements for cuberoots
theta1= (phasemag(z)+360*0)/3
theta2= (phasemag(z)+360*1)/3
theta3= (phasemag(z)+360*2)/3

```

```
16 cube_root_1 = r1 *(cosd(theta1)+ %i*sind(theta1))
17 cube_root_2 = r2 *(cosd(theta2)+ %i*sind(theta2))
18 cube_root_3 = r3 *(cosd(theta3)+ %i*sind(theta3))
19
20 printf('cuberoot 1: %f + %f*i\n',real(cube_root_1),
      imag(cube_root_1))
21 printf('cuberoot 2: %f + %f*i\n',real(cube_root_2),
      imag(cube_root_2))
22 printf('cuberoot 3: %f + %f*i\n',real(cube_root_3),
      imag(cube_root_3))
```

Scilab code Exa 6.15 To convert from polar to cartesian coordinates

```
1 clc,clear
2 //Example 6.15
3 //To convert from polar to cartesian coordinates
4
5 //part(a)
6 r=2 ;
7 theta=30 ;
8 x=r*cosd(theta) ;
9 y=r*sind(theta) ;
10 printf('(a)(x,y)= (%f,%f)\n',x,y) ;
11
12 //part(b)
13 r=3 ;
14 theta=3*%pi/4 ;
15 x=r*cos(theta) ;
16 y=r*sin(theta) ;
17 printf('(b)(x,y)= (%f,%f)\n',x,y) ;
18
19 //part(c)
20 r=-1 ;
21 theta=5*%pi/3 ;
22 x=r*cos(theta) ;
```

```
23 y=r*sin(theta) ;
24 printf('(c)(x,y)= (%f,%f)',x,y) ;
```

Scilab code Exa 6.16 To convert from cartesian to polar coordinates

```
1  clc,clear
2  //Example 6.16
3  //To convert from cartesian to polar coordinates
4
5  //part(a)
6  x=3 ;
7  y=4 ;
8
9  //53.13 is in same quadrant as(3,4)
10 r=sqrt(x^2+y^2) ;
11 theta=atand(y/x) ;
12 printf('PART A\n(r,theta)= %f,%f',r,theta) ;
13 printf('\nOR\n') ;
14 r=-sqrt(x^2+y^2) ;
15 //tan theta is +ve in 3rd quadrant
16 //so 180 + 53.33 is also a permissible value
17 theta=180 + atand(y/x) ;
18 printf('(r,theta)= %f,%f',r,theta) ;
19
20 //part(b)
21 x=-5 ;
22 y=-5 ;
23
24 //225 is in same quadrant as(-5,-5)
25 //tan theta is +ve in 3rd quadrant
26 r=sqrt(x^2+y^2) ;
27 theta=180+ atand(y/x) ;
28 printf('\n\nPART B\n(r,theta)= %f,%f',r,theta) ;
29 printf('\nOR\n') ;
30 r=-sqrt(x^2+y^2) ;
```

```
31 theta= atand(y/x) ;
32 printf('(r,theta)= %f,%f',r,theta) ;
```

Scilab code Exa 6.17 To express an equation in polar coordinates

```
1  clc,clear
2  //Example 6.17
3  //to express an equation in polar coordinates
4
5  RHS=9 ;
6  //Note that LHS is basically an equation of circle
7  //But at any instant , it is numberically same as 9
8  LHS_numerically=RHS  ;
9  r=sqrt(LHS_numerically) ;
10
11 printf('The equation in terms of polar coordinates
       is : r =%.0f',r)
```

Scilab code Exa 6.19 To express an equation in polar coordinates

```
1  clc,clear
2  //Example 6.19
3  //to express an equation in polar coordinates
4
5  //Given equation is : y=x
6  y_by_x =1; //ratio of y and x
7  tan_theta = y_by_x;
8  theta=atand(tan_theta); //azimuth angle
9
10 printf('The given equation in polar coordinates is :
       theta = %.0f degree\n',theta)
11 printf('\nNote: Polar form is same regardless of
       value of r ')
```

Chapter 8

Appendix B

Scilab code Exa 8.1 To plot the function of sin x

```
1 //Example 8.1
2 //To plot the function of sin(x)
3 clear,clc;
4
5 x = linspace(-0,2*%pi,50);
6 y = sin(x) ;
7
8 //For grid, uncomment below line
9 //set(gca(),"grid",[5 5]);
10
11 printf('NOTE:\nTo enable the grid , check the code')
12 plot(x,y,'r');
13 xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
   );
14 ylabel("$y(x)=sin(x)$","fontsize",4,"color","red");
15 title("Example 8.1","color","blue","fontsize",9);
16 legend("sin(x)");
```

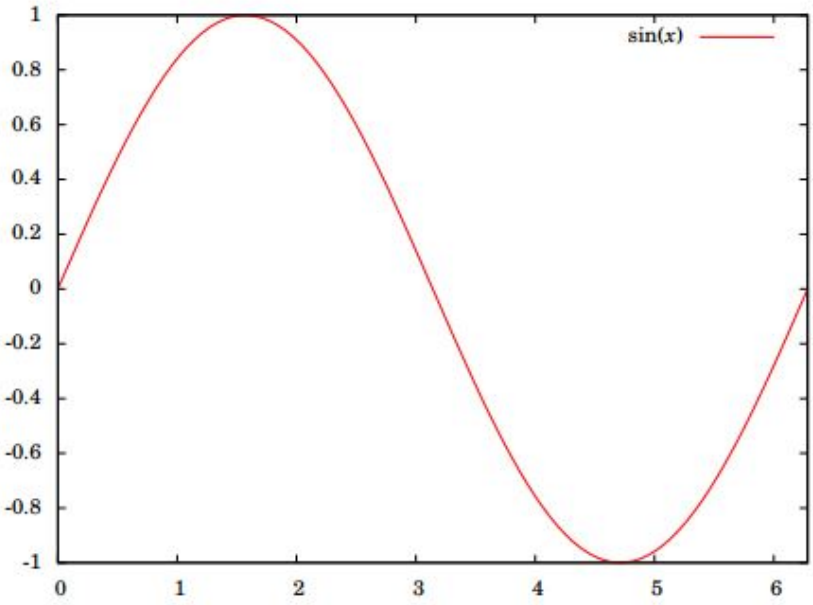

Figure 8.1: To plot the function of sin x

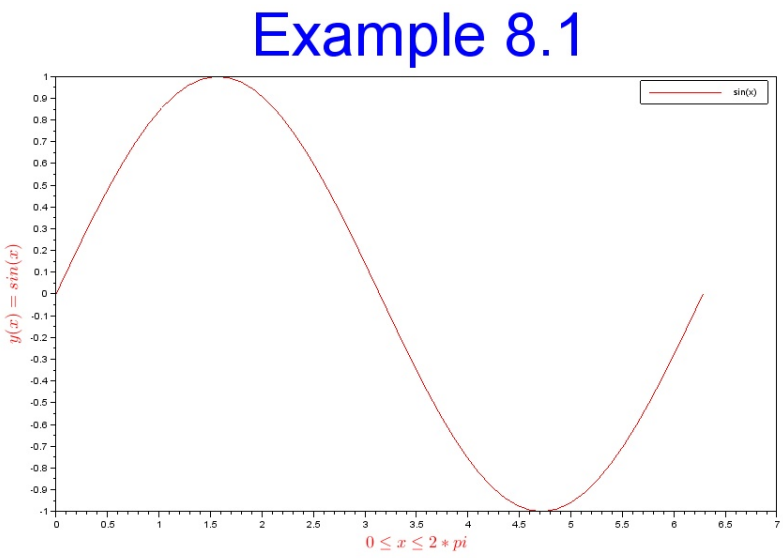

Figure 8.2: To plot the function of sin x