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 16, 2014

[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

```
clc,clear
//example 1.1
//To determine unknown angle in 3 given triangles

//Triangle ABC
A = 35//angle at vertex A in degrees
C = 20//angle at vertex C in degrees
B=180- (A+C) //unknown angle
printf('Triangle ABC: B = %.0f degree\n',B)

//Triangle DEF
```

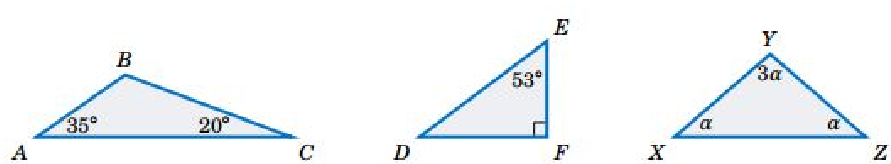

Figure 1.1: To determine unknown angle in 3 given triangles

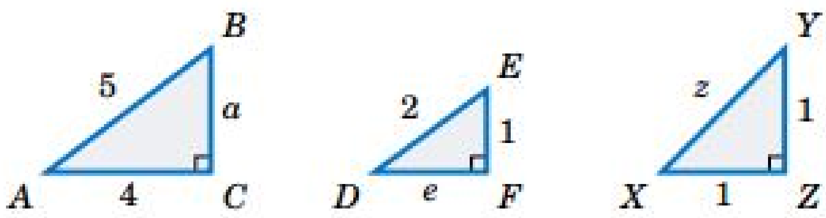

Figure 1.2: To determine length of unknown side in 3 given right triangles

```
E = 53//angle at vertex E in degree
//F = 90, DEF is right triangle
//So angles E and D are complimentary
D = 90- E //unknown angle
printf(' Triangle DEF: D = %.0f degree\n',D)

//Triangle XYZ
sum_multiple= 1+3+1 //for solvong for alpha
alpha = 180/ sum_multiple
X= alpha //unknown angle
Y= 3* alpha //unknown angle
Z= alpha //unknown angle
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

```
clc,clear
//example 1.3
//To determine length of unknown side in 3 given
    right triangles

//Triangle ABC
```

```
AB=5 //given
AC=4 //given
a=sqrt(AB^2- AC^2) //by pythagoras theorem
printf('Triangle ABC: a=%f units \n',a)

//Triangle DEF
DE=2 //given
EF=1 //given
e=sqrt(DE^2- EF^2) //by pythagoras theorem
printf(' Triangle DEF: e=%f units = sqrt(%f) units\n',e,e^2)

//Triangle XYZ
XZ=1 //given
YZ=1 //given
z=sqrt(XZ^2+YZ^2)//by pythagoras theorem
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

```
clc,clear
//example 1.4
//To determine height of the top of ladder touching the wall

ladder = 17 //length of ladder or hypotenuse in feet
base = 8 //distance between lower tip of ladder and wall in feet
//Using pythagoras theorem
h=sqrt(ladder^2 - base^2) //required height
printf('Required height of top of ladder in contact
```

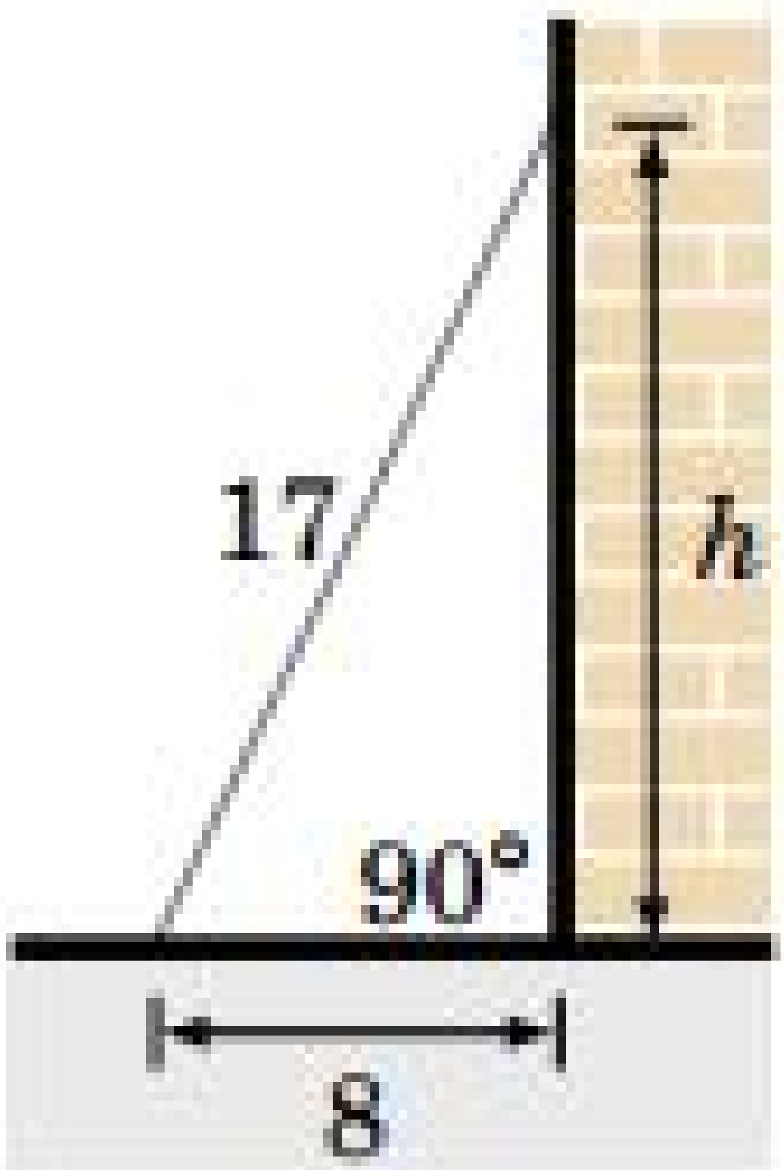

Figure 1.3: To determine height of the top of ladder touching the wall

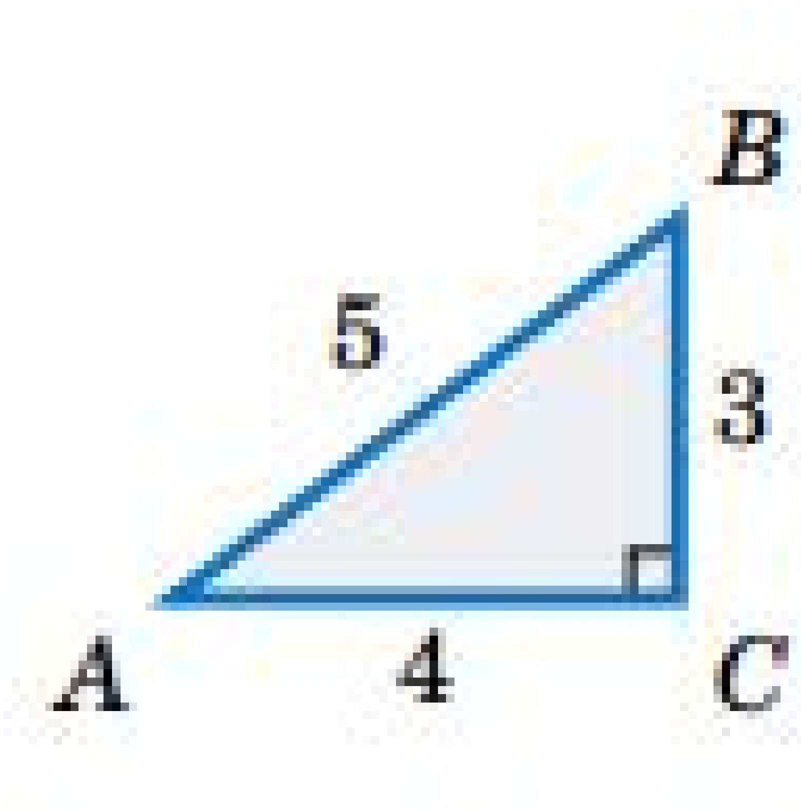

Figure 1.4: To find values of all trigonometric functions for angles A and B

```
    with wall = %.0f ft',h)
```

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

```
clc,clear
//example 1.5
//To find values of all trigonometric functions for
    angles A and B
```

```
//Angle at vertex A
opposite = 3;
adjacent = 4;
hypotenuse=5;

sin_A = opposite / hypotenuse;
cos_A = adjacent / hypotenuse;
tan_A = opposite / adjacent;
csc_A = hypotenuse/opposite;
sec_A = hypotenuse/adjacent;
cot_A = adjacent / opposite;
printf('ANGLE A')
printf('\nsin(A)=  %.1f    ;  cos(A)= %.2f;    tan(A)= %.2f;\n',sin_A,cos_A,tan_A)
printf('csc(A)=  %.3f  ;  sec(A)= %.2f;    cot(A)= %.2f;',csc_A,sec_A,cot_A)

//Angle at vertex B
opposite = 4;
adjacent = 3;
hypotenuse=5;

sin_B = opposite / hypotenuse;
cos_B = adjacent / hypotenuse;
tan_B = opposite / adjacent;
csc_B = hypotenuse/opposite;
sec_B = hypotenuse/adjacent;
cot_B = adjacent / opposite;
printf('\n\nANGLE B')
printf('\nsin(B)=  %.1f    ;  cos(B)= %.2f;    tan(B)= %.2f;\n',sin_B,cos_B,tan_B)
printf('csc(B)=  %.2f   ;  sec(B)= %.2f;    cot(B)= %.2f;',csc_B,sec_B,cot_B)
```

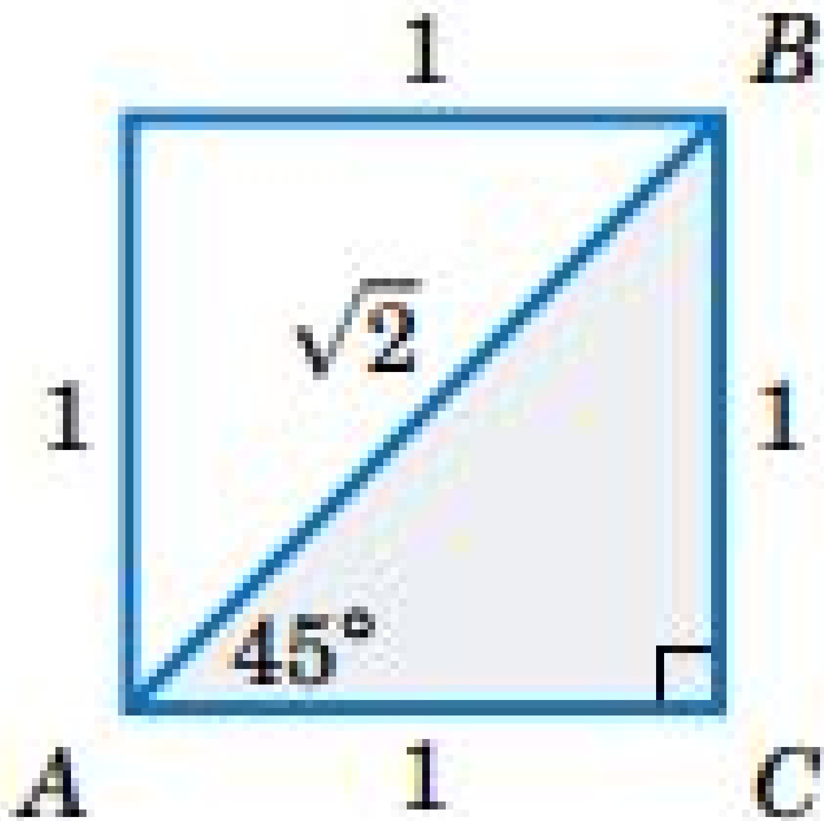

Figure 1.5: To find values of all trigonometric functions for 45 degree

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)
```

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')
```

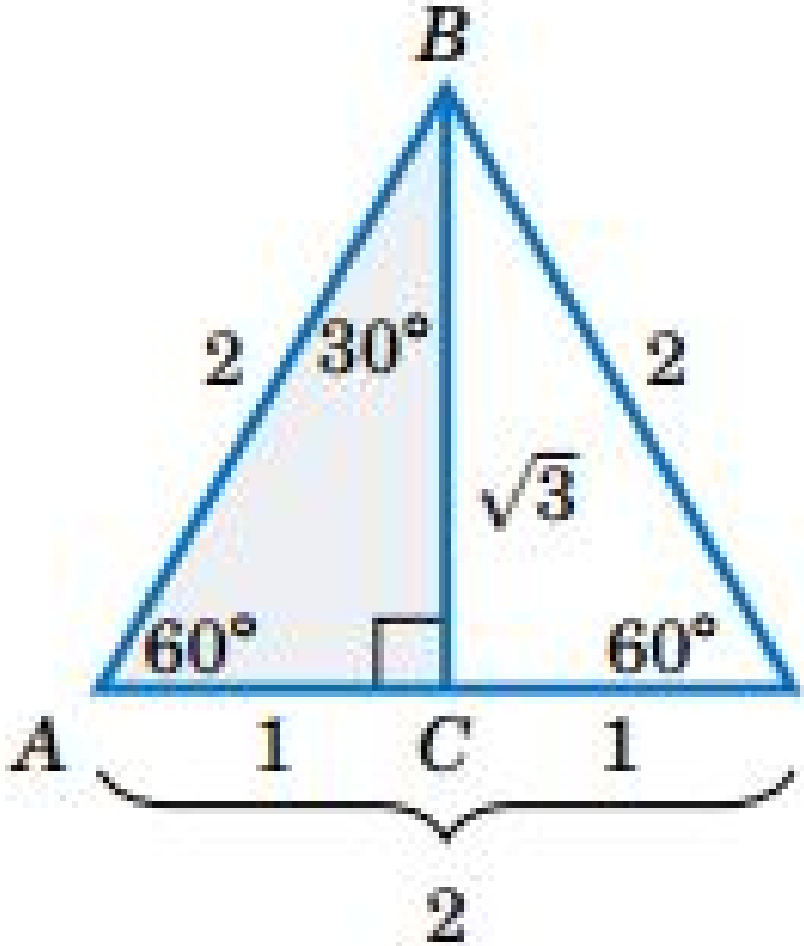

Figure 1.6: To find values of all trigonometric functions for 60 degree

```
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;
adjacent = a;
hypotenuse = c;
sin_30 = opposite / hypotenuse;
cos_30 = adjacent / hypotenuse;
tan_30 = opposite / adjacent;
csc_30 = hypotenuse/opposite;
sec_30 = hypotenuse/adjacent;
cot_30 = adjacent / opposite;
printf('\n\nANGLE = 30 degree')
printf('\nsin(30)=  %.4f  ;  cos(30)= %.4f;     tan(30)= %.4f;\n',sin_30,cos_30,tan_30)
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 given

```
clc,clear
//example 1.8
//To find all trigonometric functions when sine functions is given

sin_A=2/3 //given
//since sine function  is opposite/hypotenuse and
//T-ratios are defined interms of ratio of sided of right triangle
```

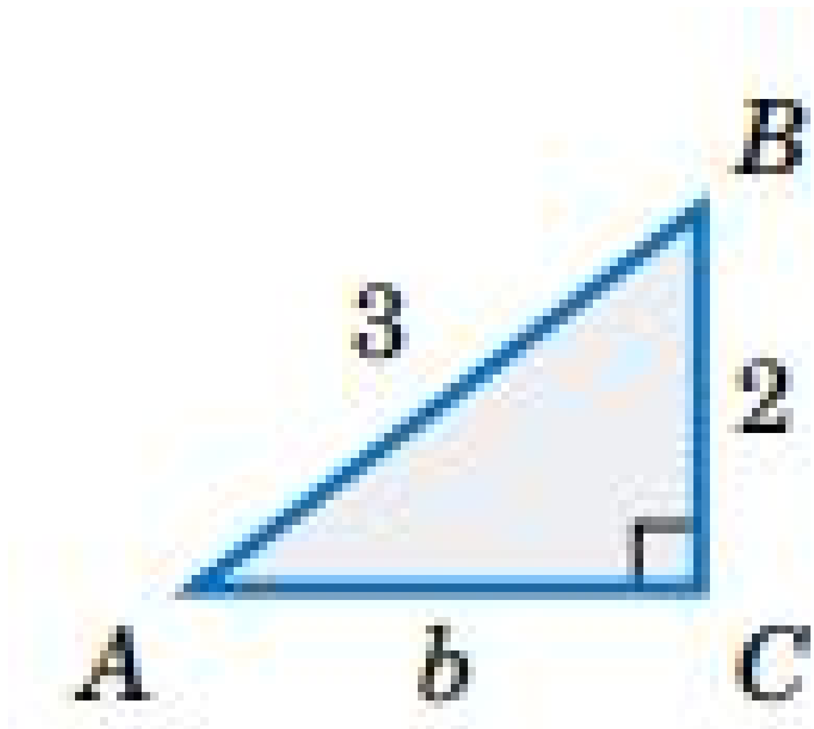

Figure 1.7: To find all trigonometric functions when sine functions is given

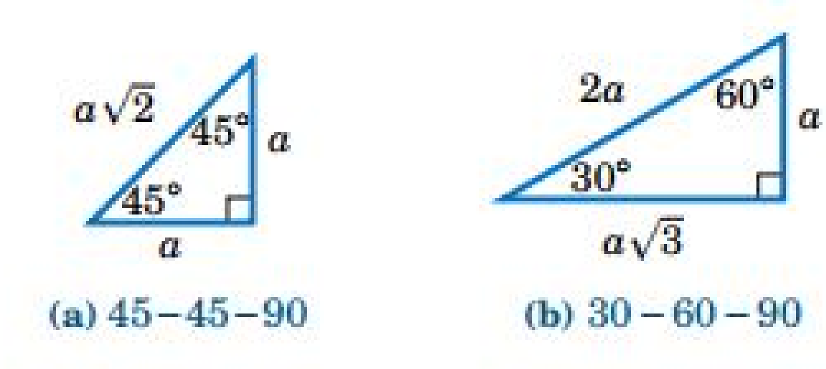

Figure 1.8: To convert given function into function of angle less than 45

```
opposite=2;
hypotenuse=3;
BC = opposite;
AB = hypotenuse;
b  = sqrt(hypotenuse^2- opposite^2) //by pythagoras
    theorem
adjacent = b;

cos_A = adjacent / hypotenuse;
tan_A = opposite / adjacent;
csc_A = hypotenuse/opposite;
sec_A = hypotenuse/adjacent;
cot_A = adjacent / opposite;

printf('for ANGLE A')
printf('\nsin(A)=  %.4f  ;  cos(A)= %.4f;     tan(A)= %
    .4f;\n',sin_A,cos_A,tan_A)
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

```
clc,clear
//example 1.9
//To convert given function into function of angle
   less than 45

//(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
```

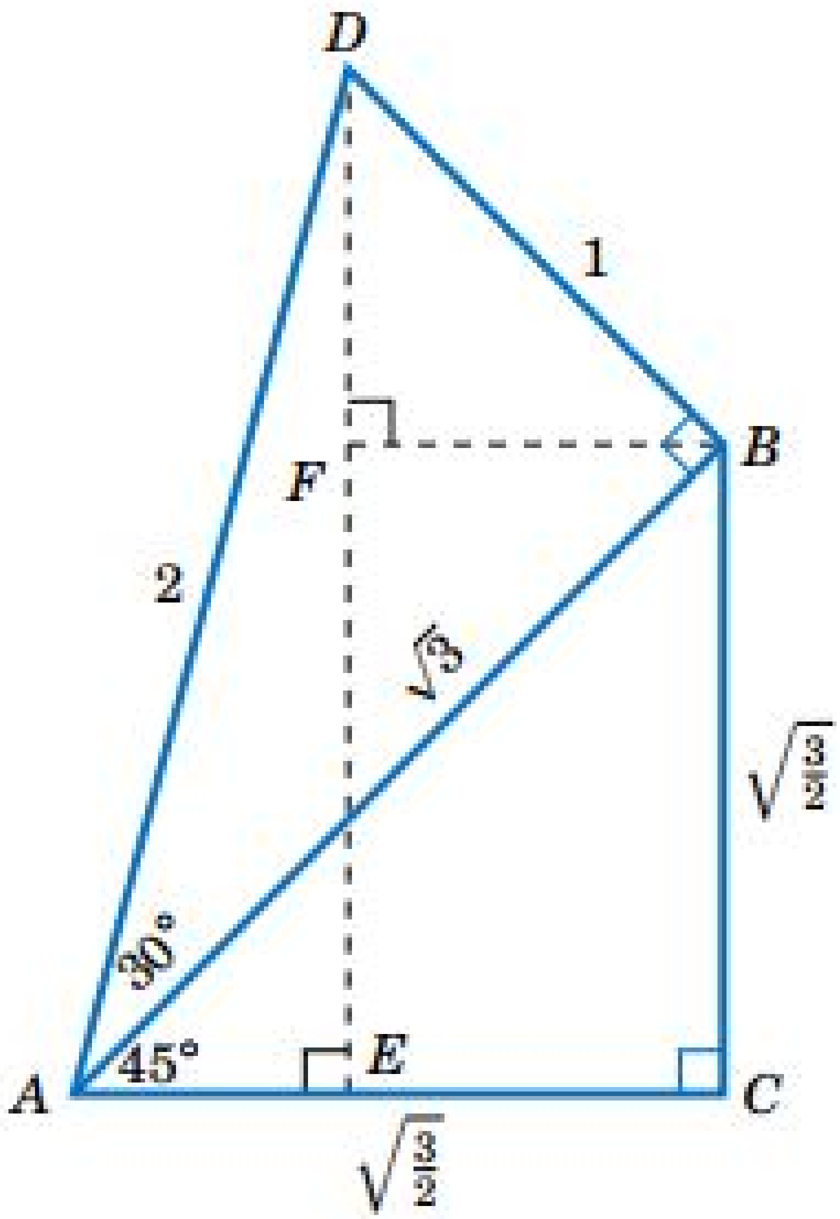

Figure 1.9: To find sine cosine and tangent functions for 75 degree

```
//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
//pythagoras theorem and 45 degrees
AC=AB/sqrt(2);BC=AC;

angle_BAC = 45 ; angle_DAB = 30 ;
angle_DAE = angle_BAC + angle_DAB ;//required angle
angle_ADE = 90 - angle_DAE ;//complement of DAE
angle_ADB = 90 - angle_DAB ;//complement of DAB

//Draw BF perpendicular to DE
angle_BDF = angle_ADB - angle_ADE;
angle_DBF = 90 - angle_BDF; //complement of BDF
//By pythagoras theorem and 45 degree
DF=sqrt(BD/2);FB=DF;

EC=FB;//parallel sides of rectangle
FE= BC; //parallel sides of rectangle
DE=DF+FE;//from the figure
AE=AC-EC;//from the figure

sin_DAE = DE/AD;
cos_DAE = AE/AD;
tan_DAE = DE/AE;
csc_DAE = AD/DE;
sec_DAE = AD/AE;
cot_DAE = AE/DE;

printf('sin(%d)=%f\n',angle_DAE,sin_DAE);
printf('cos(%d)=%f\n',angle_DAE,cos_DAE);
```

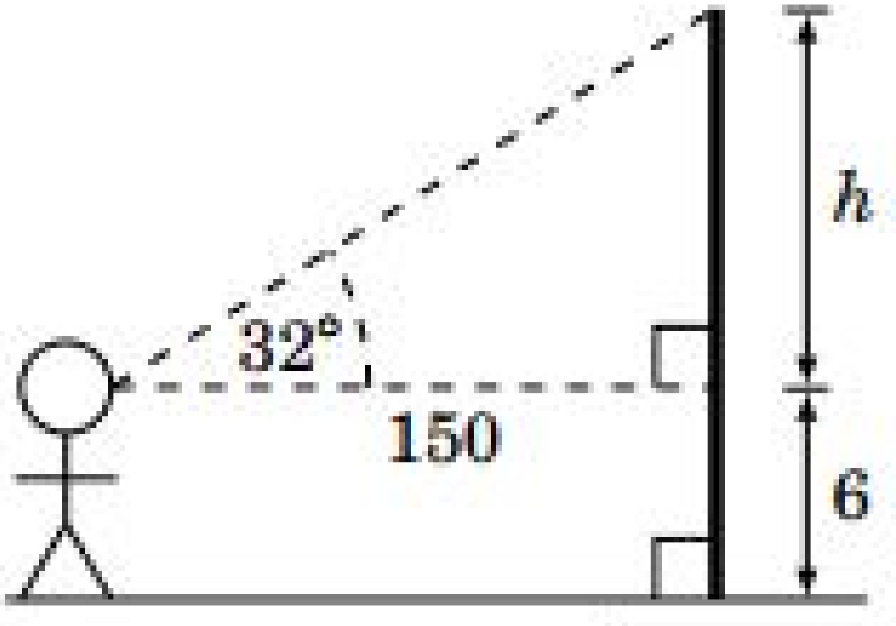

Figure 1.10: To find the height of the flagpole

```
printf('tan(%d)=%f\n',angle_DAE,tan_DAE);
printf('csc(%d)=%f\n',angle_DAE,csc_DAE);
printf('sec(%d)=%f\n',angle_DAE,sec_DAE);
printf('cot(%d)=%f\n',angle_DAE,cot_DAE);
```

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
```

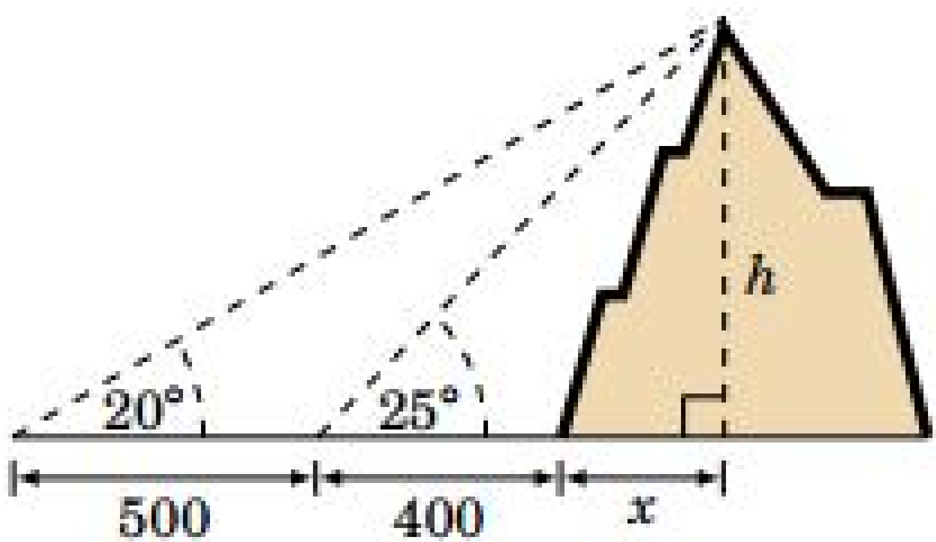

Figure 1.11: To find the height of mountain

```
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
//Example 1.13
//To find the horizontal distance from blimp to
   house

//consider the figure attached
angle_dep = 24 //angle of depression in degrees
theta = angle_dep  //angle of elevation
height_blimp = 4280 //height of blimp from ground in
    feet
x = height_blimp  / tand(theta) //required distance
```

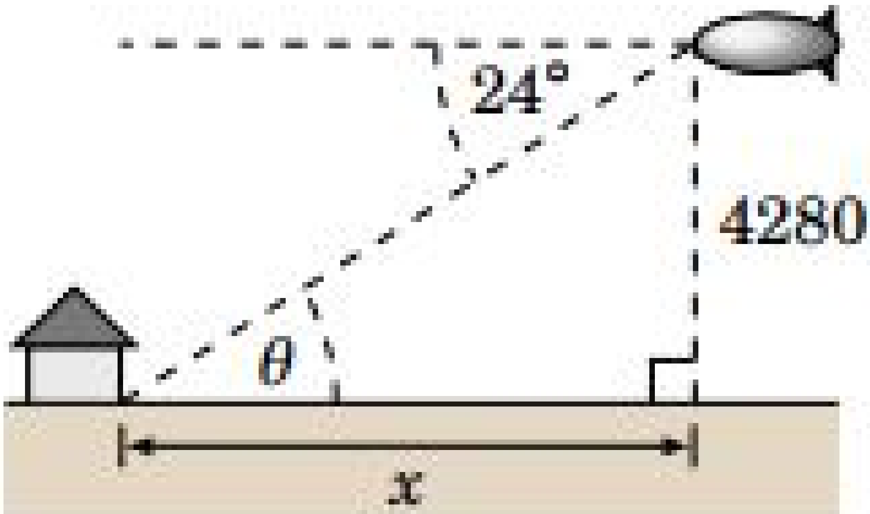

Figure 1.12: To find the horizontal distance from blimp to house

```
    in feet

printf('The house is %.0f ft far from blimp along
    the ground',x)
```

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
```

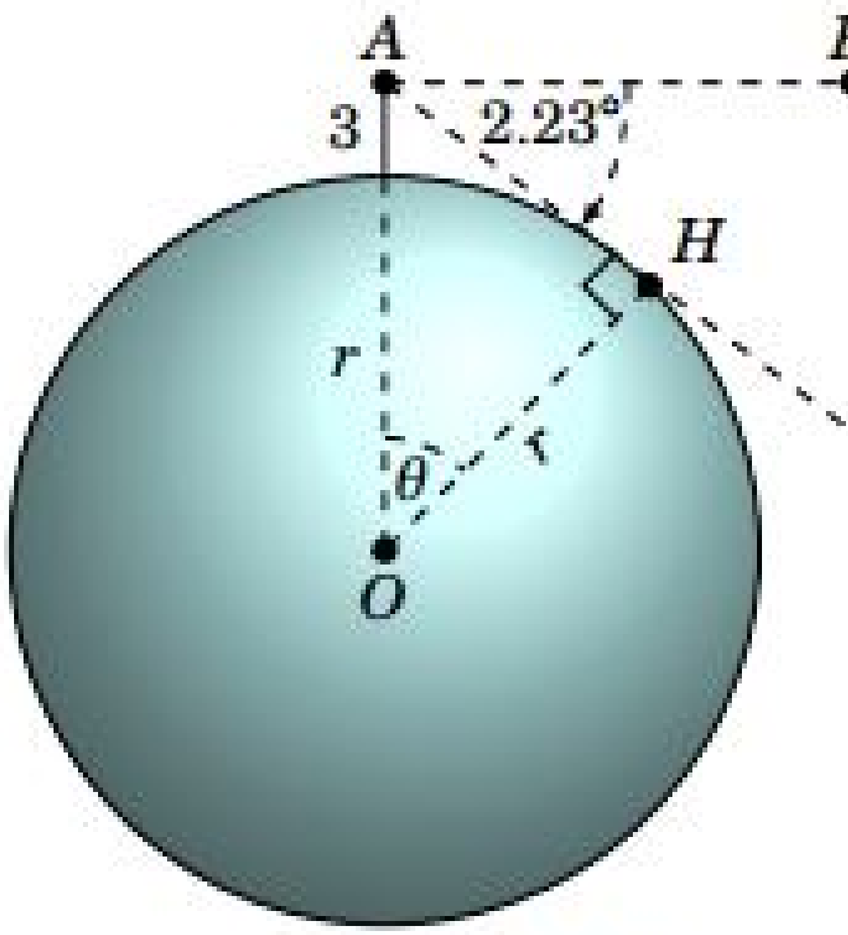

Figure 1.13: To estimate radius of earth when angle of depression is known

```
//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

```
clc,clear
//Example 1.15
// To find the distance from centre of earth to sun

alpha = 0.00244; // equitorial paralalx in degree
OA = 3956.6; //radius of earth
angle_OAB = 90;

OB = OA / sind(alpha) ;
printf('Distance is obtained as %.0f miles = %.0f
    million miles',OB,OB/10^6)
```

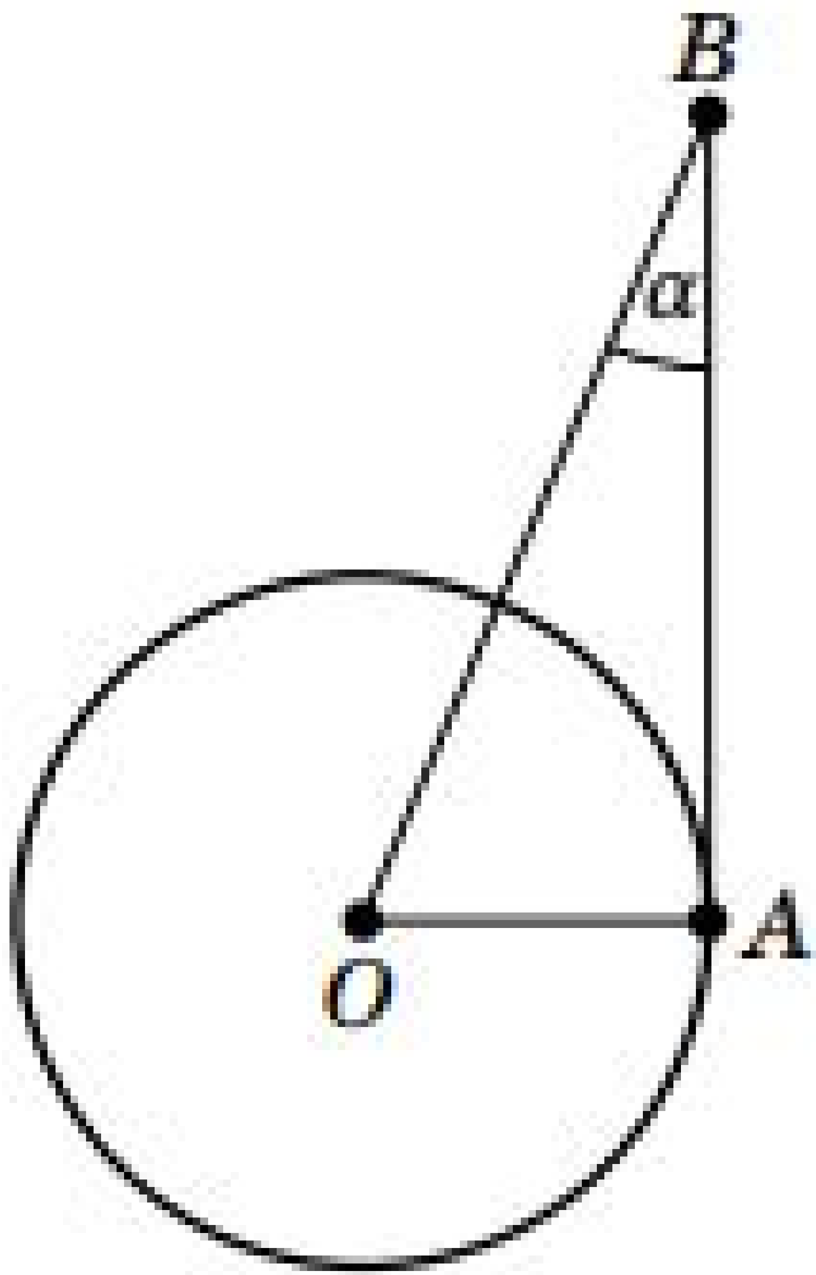

Figure 1.14: To find the distance from centre of earth to sun

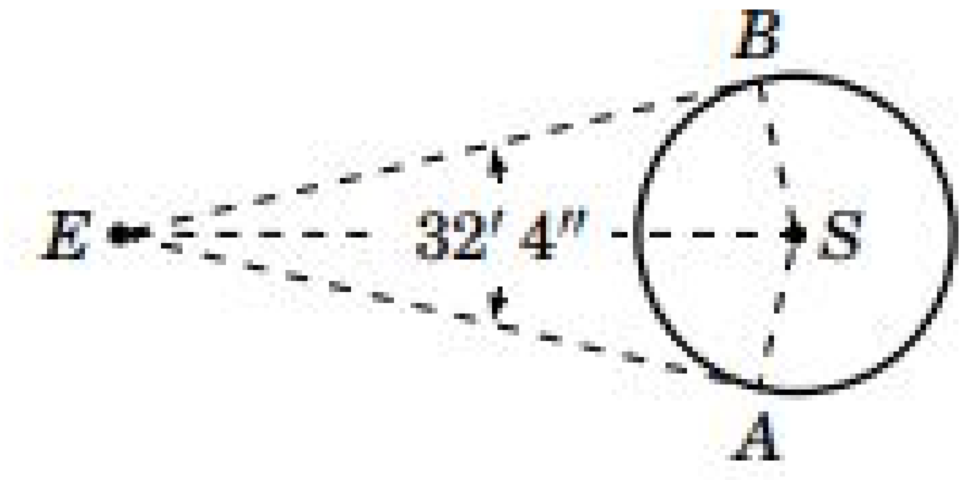

Figure 1.15: To determine the radius of sun

Scilab code Exa 1.16 To determine the radius of sun

```
clc,clear
//Example 1.16
// To determine the radius of sun

angle_AEB =0 +(32/60)+ (4/60)/60//converting to degrees

//Triangle BES and AES are similar
//BS=AS as they are radius
//ES is common to both triangles
//angle_EBS=angle_ABS =90 as tangents are perpendicualar to radius
// angle_AES = angle_BES
angle_AES= angle_AEB /2;
```

```
angle_BES= angle_AEB /2;

//to find ditance from sun to centre of earth
//obtained from previous example
alpha = 0.00244; // equitorial paralalx in degree
OA = 3956.6 ;//radius of earth
angle_OAB = 90 ;//radius perpendicular to tangent
OB = OA / sind(alpha) ;

//ES is from earth surface to sun centre
//centre of earth to sun is  OB
//we initially treated sun as point
//that ditance is distance between their centres
radius_earth=3956.6 ;//in miles
ES = OB - radius_earth ;//in miles
AS=ES * sind(angle_AES) ;//in miles
printf('Required radius of sun = %.0f miles \n',AS)
printf('Answer might vary due to approximations in
    book and scilab precision')
```

Scilab code Exa 1.17 To determine the diameter of larger roller

```
clc,clear
//Example 1.17
//To find the diameter of larger roller

//since radius perpendicular to tangent
angle_ODA=90;
angle_PEC=90
angle_OAD=37 ;//by symmetry
ED=1.38 ;//given
//since DOA is right triangle , DOA and OAD are
    complementary angles
```

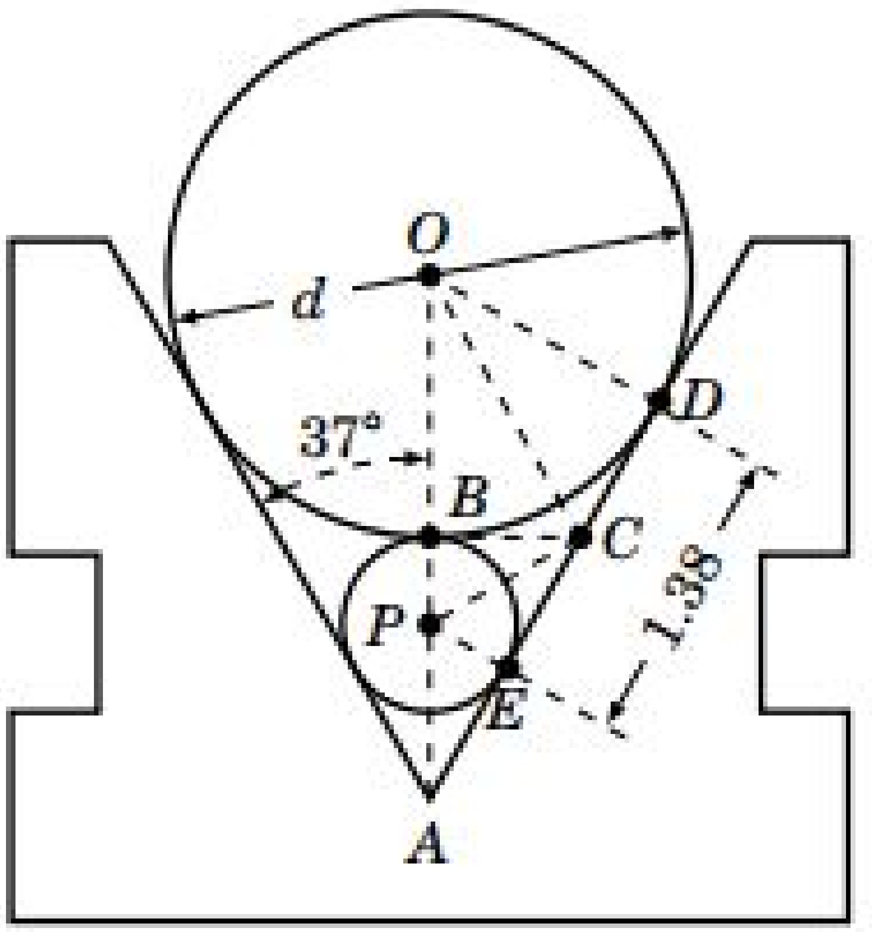

Figure 1.16: To determine the diameter of larger roller

```
angle_DOA=90 - angle_OAD ;

//since radius perpendicular to tangent
angle_OBC=90;
angle_PBC=90;

//since ODA and ODC are right triangle ,
//OD = OB as radius and BC=DC by pythagoras
//OBC and ODC are now congruent
//angle_BOC = angle_DOC
//angle_BOC + angle_DOC =(90-angle_OAD)
angle_BOC= (90-angle_OAD)/2;
angle_DOC= (90-angle_OAD)/2;

//BP=EP as radius
//since radius perpendicular to tangent
angle_PBC=90;
angle_PEC=90;
//Thus,BPC and EPC are congruent triangles
//Therefore,BC=DC and BC+DC = ED
BC = ED /2;
DC = ED /2;
OB = BC / tand(angle_BOC); //radius of large roller
diameter= 2* OB ;
printf('Diameter of larger roller = %.3f units',
    diameter)
```

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

```
clc,clear
//Example 1.19
//To solve the right triangle with given information

```

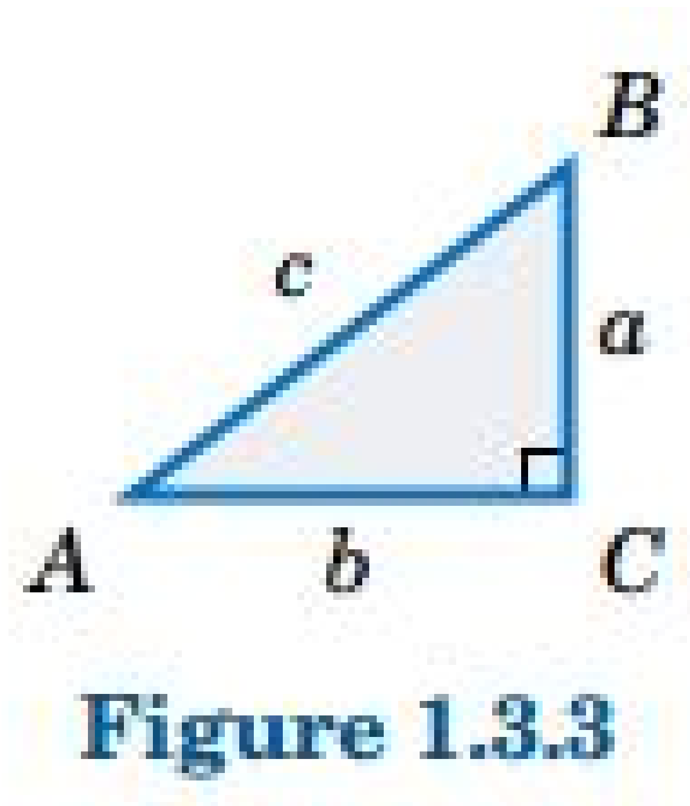

Figure 1.17: To solve the right triangle with given information

```
//part(a)
c=10 ;//side opposite to vertex C
A=22 ;//Angle at vertex A
a=c*sind(A);
b=c*cosd(A) ;
B=90 - A ;//since C is 90, A and B are complimentary
printf('(a)a= %.2f units ; b= %.2f  units; B = %.0f
    degree\n',a,b,B)

//part(b)
b=8 ;//side opposite to vertex B
A=40 ;//Angle at vertex A
a=b*tand(A);
c=b/cosd(A) ;
B=90 - A ;//since C is 90, A and B are complimentary
printf(' (b)a= %.2f units ; c= %.2f units; B = %.0f
    degree\n',a,c,B)

//part(c)
a=3 ;//side opposite to vertex A
b=4 ;//side opposite to vertex B
c=sqrt(a^2+b^2) ;//by pythagoras theorem
A = atand(a/b) ;//angle at vertex A
B=90 - A ;//since C is 90, A and B are complimentary
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 120

```
clc,clear
//Example 1.20
//To find values of all trigonometric values of
```

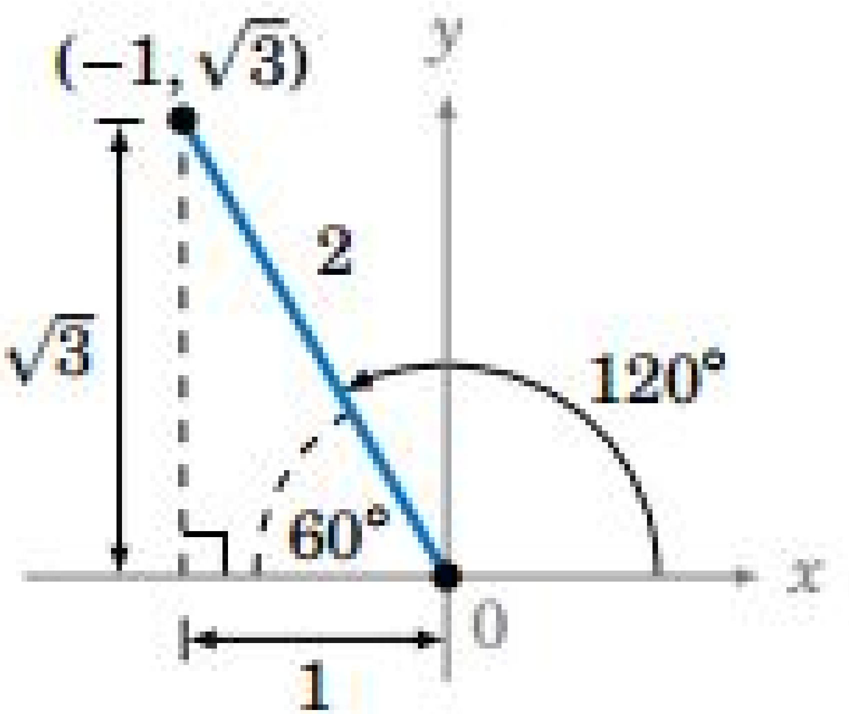

Figure 1.18: To find values of all trigonometric values of given angle of 120

```
given angle of 120

theta=120; //given angle in degree
//Consider a point (-1,sqrt(3)) in 2nd quadrant
//This point can be used on terminal side of 120
//Thus for a basic right angled triangle formed in second quadrant
adjacent = 1;
opposite = sqrt(3);
//by pythagoras theorem
hypotenuse = sqrt(adjacent^2 + opposite^2);

//since its third quadrant
x=-adjacent;
y= opposite;
r=hypotenuse;

sin_120 =y/r ;
cos_120 =x/r ;
tan_120 =y/x ;
csc_120 =r/y ;
sec_120 =r/x ;
cot_120 =x/y ;

printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;',theta,sin_120,theta,cos_120,theta,tan_120)
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 225

```
clc,clear
```

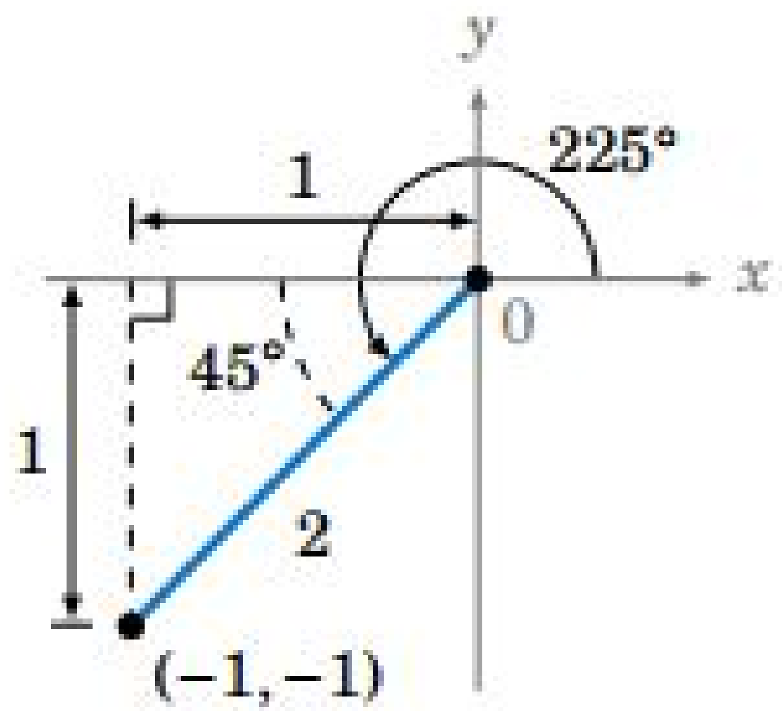

Figure 1.19: To find values of all trigonometric values of given angle of 225

```
//Example 1.21
//To find values of all trigonometric values of
   given angle of 225

theta=225 ;//given angle in degree
//Consider a point (−1,−1) in 3rd quadrant
//This point can be used on terminal side of 225
//Thus for a basic right angled triangle formed in 3
   rd quadrant
adjacent = 1 ;
opposite = 1 ;
//by pythagoras theorem
hypotenuse = sqrt(adjacent^2 + opposite^2) ;

//since its third quadrant
x=-adjacent ;
y=-opposite ;
r=hypotenuse ;

sin_225 =y/r ;
cos_225 =x/r ;
tan_225 =y/x ;
csc_225 =r/y ;
sec_225 =r/x ;
cot_225 =x/y ;

printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;'
   ,theta,sin_225,theta,cos_225,theta,tan_225)
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 330

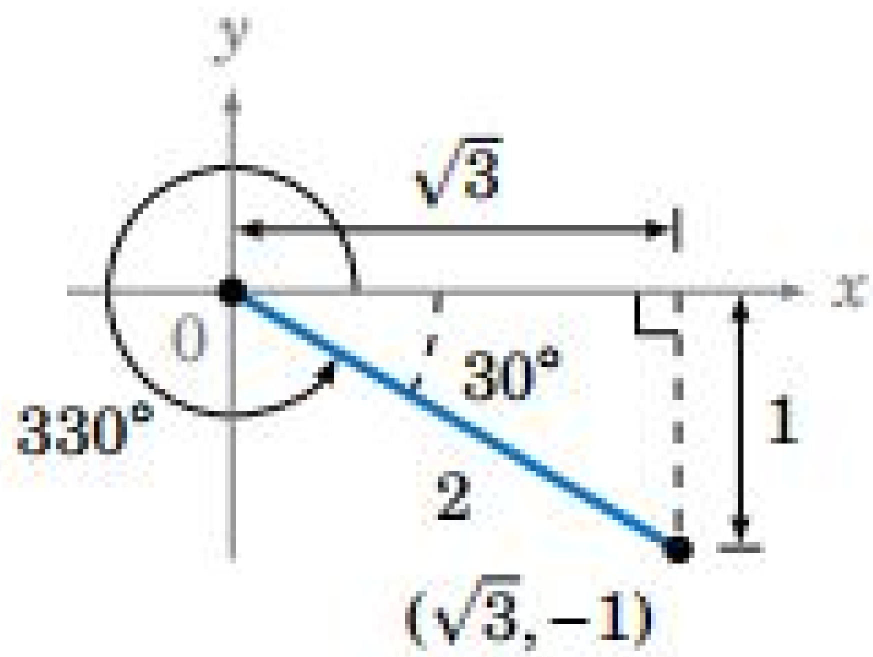

Figure 1.20: To find values of all trigonometric values of given angle of 330

```
clc,clear
//Example 1.22
//To find values of all trigonometric values of given angle of 330

theta=330 ; //given angle in degree
//Consider a point (sqrt(3),-1) in 4th quadrant
//This point can be used on terminal side of 330
//Thus for a basic right angled triangle formed in 4th quadrant
adjacent = sqrt(3);
opposite = 1 ;
hypotenuse = sqrt(adjacent^2 + opposite^2);
//by pythagoras theorem

//since its 4th quadrant
x=adjacent ;
y=-opposite ;
r=hypotenuse ;

sin_330 =y/r ;
cos_330 =x/r ;
tan_330 =y/x ;
csc_330 =r/y ;
sec_330 =r/x ;
cot_330 =x/y ;

printf('\nsin(%d)= %f ; cos(%d)= %f ; tan(%d)= %f ;',theta,sin_330,theta,cos_330,theta,tan_330)
printf('\ncsc(%d)= %f ; sec(%d)= %f ; cot(%d)= %f ;',theta,csc_330,theta,sec_330,theta,cot_330)
```

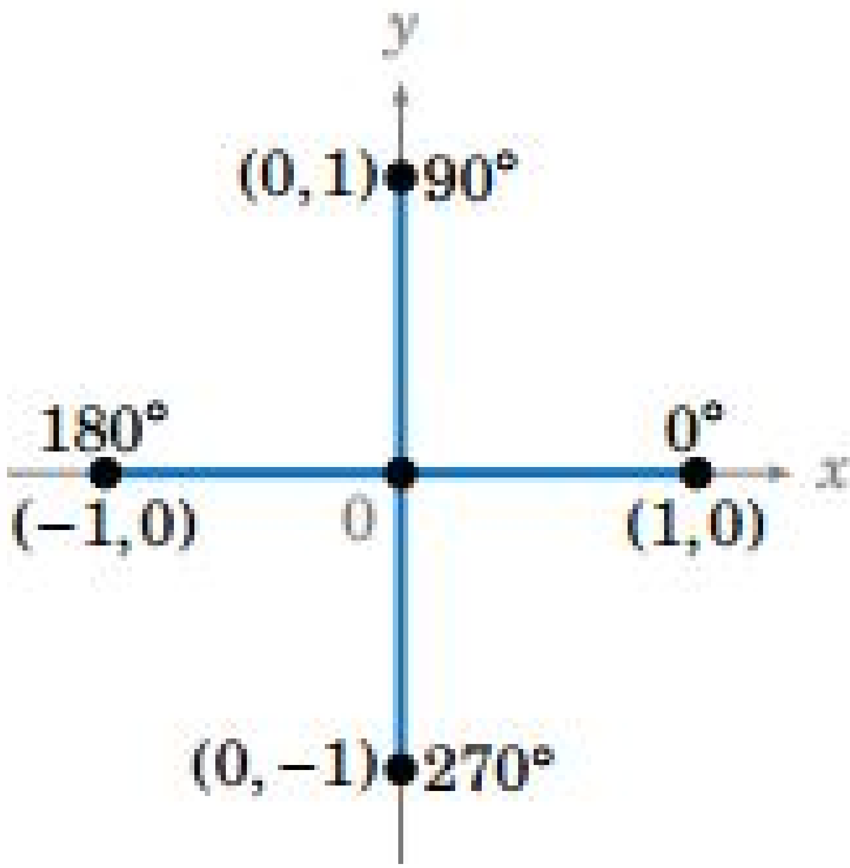

Figure 1.21: To find trigonometric ratios of 0 90 180 and 270 degrees

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

```
clc,clear
//Example 1.23
//To find trigonometric functions of various angles
//Note: Undefined ratios are commented to avoid
   divide by zero error

//zero degrees
//consider a point (1,0)
//Line segment joining (0,0) and (1,0) can be
   treated as triangle

x=1//base
y=0//height
r=1 //hypotenuse
sin_0 = y/r;
cos_0 = x/r;
tan_0 = y/x;
//csc_0 =r/y;
sec_0 = r/x;
//cot_0=x/y;
printf('\nZERO DEGREES:\n')
printf('sin(0)= %f;\n',sin_0)
printf('cos(0)= %f;\n',cos_0)
printf('tan(0)= %f;\n',tan_0)
printf('csc(0)= undefined = (1/0);\n')
printf('sec(0)= %f;\n',sec_0)
printf('cot(0)= undefined = (1/0);\n')

//90 degrees
//consider a point (0,1)
//Line segment joining (0,0) and (0,1) can be
   treated as triangle

x=0//base
y=1//height
```

```
r=1 //hypotenuse
sin_90 = y/r;
cos_90 = x/r;
//tan_90 = y/x;
csc_90 =r/y;
//sec_90 = r/x;
cot_90=x/y;
printf('\n90 DEGREES:\n')
printf('sin(90)= %f;\n',sin_90)
printf('cos(90)= %f;\n',cos_90)
printf('tan(90)= undefined = (1/0);\n')
printf('csc(90)= %f;\n',csc_90)
printf('sec(90)= undefined = (1/0);\n')
printf('cot(90)= %f;\n',cot_90)

//180 degrees
//consider a point (-1,0)
//Line segment joining (0,0) and (-1,0) can be treated as triangle

x=-1//base
y=0//height
r=1 //hypotenuse
sin_180 = y/r;
cos_180 = x/r;
tan_180 = y/x;
//csc_180 =r/y;
sec_180 = r/x;
//cot_180=x/y;
printf('\n180 DEGREES:\n')
printf('sin(180)= %f;\n',sin_180)
printf('cos(180)= %f;\n',cos_180)
printf('tan(180)= %f;\n',tan_180)
printf('csc(180)= undefined = (1/0);\n')
printf('sec(180)= %f;\n',sec_180)
printf('cot(180)= undefined = (-1/0);\n')
```

```
//270 degrees
//consider a point (0,-1)
//Line segment joining (0,0) and (0,-1) can be
    treated as triangle

x=0//base
y=-1//height
r=1 //hypotenuse
sin_270 = y/r;
cos_270 = x/r
//tan_90 = y/x;
csc_270 =r/y;
//sec_90 = r/x;
cot_270=x/y;
printf('\n270 DEGREES:\n')
printf('sin(270)= %f;\n',sin_270)
printf('cos(270)= %f;\n',cos_270)
printf('tan(270)= undefined = (-1/0);\n')
printf('csc(270)= %f;\n',csc_270)
printf('sec(270)= undefined = (1/0);\n')
printf('cot(270)= %f;\n',cot_270)
```

Scilab code Exa 1.24 To determine reference angle and angle between 0 to 360 with same terminal side as given angle

```
clc,clear
//Example 1.24
//To determine reference angle and angle ( 0 to 360
    )with same terminal side as given angle

theta = 928 ;//given angle in degrees

//The while loop works for ALL VALUES OF theta
```

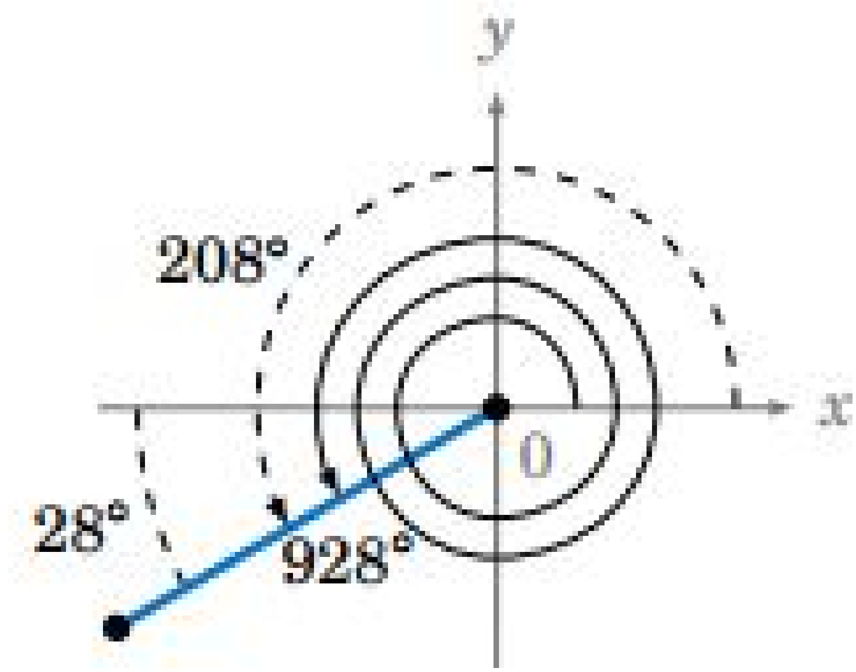

Figure 1.22: To determine reference angle and angle between 0 to 360 with same terminal side as given angle

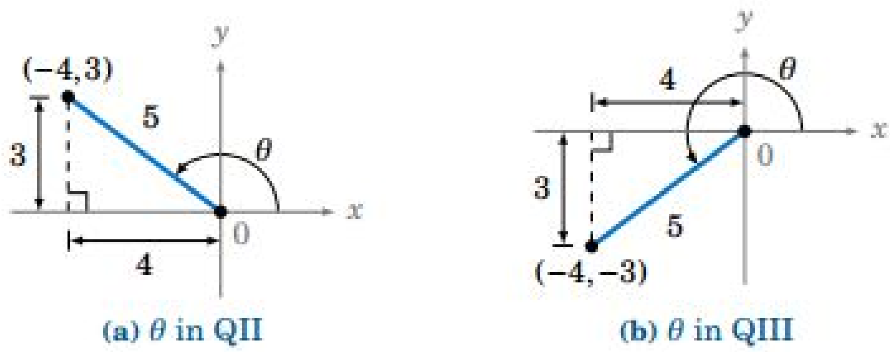

Figure 1.23: To find sin theta and tan theta when cos theta is given

```
//It keeps subtracting 360 till a value in (0 to
   360) is obtained
 result = theta ;
while 1==1,
    if result<360 then
        printf('(a)Required angel between 0 and 360
           is %.0f degree',result);
        break
    end
    result = result - 360 ;
end

//928 and 208 has same terminal side in 3rd quadrant
//so their reference angle is same
ref_angle_928 = result - 180 ; //required reference
   angle
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

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

cos_theta = -4/5;
adjacent =4 ; hypotenuse =5 ;
opposite = sqrt(hypotenuse ^2 - adjacent ^2) //by pythagoras theorem

//minus sign of cos_theta implies 2nd or 3rd quadrant
//Possibility 1 : 2nd quadrant
x= -adjacent ;
y= opposite ;
r= hypotenuse ;
sin_theta = y/r ;
tan_theta = y/x ;
printf('POSSIBILITY 1:Theta in 2nd quadrant\n')
printf('sin(theta)= %.2f ; tan(theta) = %.2f; \n\n',sin_theta,tan_theta)

//Possibility 2 : 3rd quadrant
x=-adjacent ;
y=-opposite ;
r=hypotenuse ;
sin_theta = y/r ;
tan_theta = y/x ;
printf('POSSIBILITY 2:Theta in 3rd quadrant\n')
printf('sin(theta)= %.2f ; tan(theta) = %.2f; ',sin_theta,tan_theta)
```

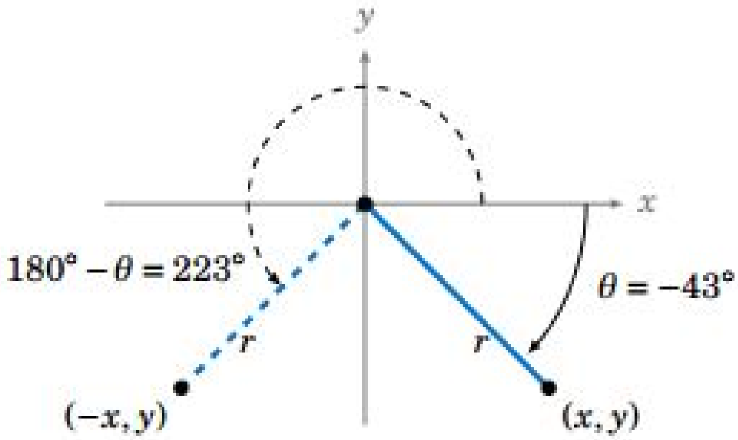

Figure 1.24: To find all the angles with a given sine function value

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

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

sin_theta = -0.682;
theta=asind(sin_theta);
//This results -43 degree which isnt in 0 to 360
   range
//And theta exists in 4th quadrant
//Angles in 1st and 2nd quadrant have +ve sine
   values

theta1 = 180 - theta ;//reflection of theta in 3rd
   quadrant
```

```
theta2 = 360 + theta ;//both theta n theta 2 have
    same trigonometric values
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

```
//Example 2.1
//To solve the triangle when one side and 2 angles
   are given
clc,clear

a=10 //side opposite to vertex A
A=41 //angle at vertex A
C=75 //angle at vertex C

B=180- (A+C)
b=a*sind(B)/sind(A) //law of sines
c=a*sind(C)/sind(A) //law of sines
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)
```

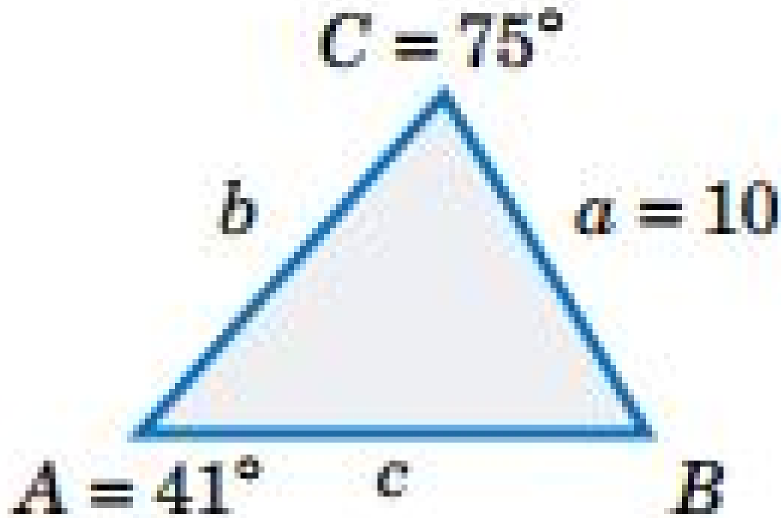

Figure 2.1: To solve the triangle when one side and 2 angles are given

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 given

```
//Example 2.2
//To solve the triangle when 2 sides and one
   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

sin_B=(b/a)*sind(A) //law of sines

//case 1
B=asind(sin_B)   //law of sines
C=180-(A+B)
c=a*sind(C)/sind(A) //law of sines
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)

//case 2
B=180 - asind(sin_B)  //law of sines
C=180-(A+B)
c=a*sind(C)/sind(A)  //law of sines
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

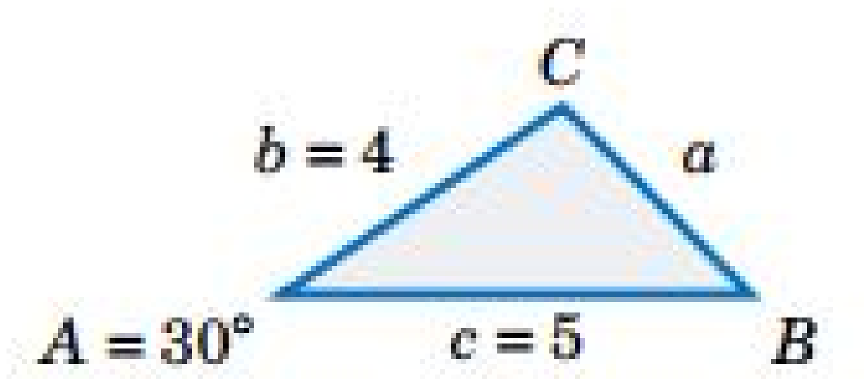

Figure 2.3: To solve the triangle when 2 sides and angle between them is given

```
//Example 2.3
//To solve the triangle when 2 sides and opposite angle is given
clc,clear

a=5 //side oposite to vertex A
A=30 //angle at vertex A
b=12 //side opposite to vertex B

sin_B=(b/a)*sind(A) //law of sines
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 given

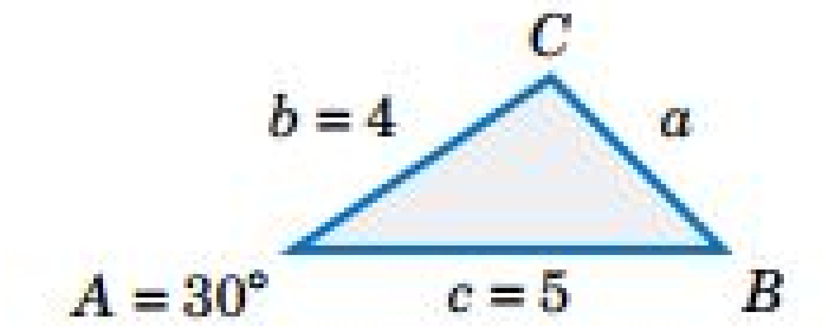

Figure 2.4: To solve the triangle when 2 sides and angle between them is given

```
//Example 2.4
//To solve the triangle when 2 sides and angle
   between them is given
clc,clear

c=5 //side oposite to vertex C
A=30 //angle at vertex A
b=4 //side opposite to vertex B

printf('By law of sines:\n')
printf('a/sin(30) = 4/sin(B) = 5 / sin(C)\n')
printf('\nEach of the equations has 2 unknowns.')
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')
printf('Now we have a in terms of C which is unknown
   \n')
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 given

```
//Example 2.5
//To solve the triangle when 2 sides and angle
   between them is given
clc,clear

c=5 //side oposite to vertex C
A=30 //angle at vertex A
b=4 //side opposite to vertex B

a = sqrt( b^2 + c^2 -2*b*c*cosd(A) ) //from law of
   cosines
printf('Length of a= %.2f units\n',a)
cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
B=acosd(cos_B)
printf('Angle B=%.1f degrees\n',B)
C=180-(A+B)
printf('Angle C=%.1f degrees\n',C)
```

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

```
//Example 2.6
//To solve the triangle when 3 sides are given
clc,clear

c=4 //side oposite to vertex C
a=2 //side opposite to vertex A
b=3 //side opposite to vertex B
```

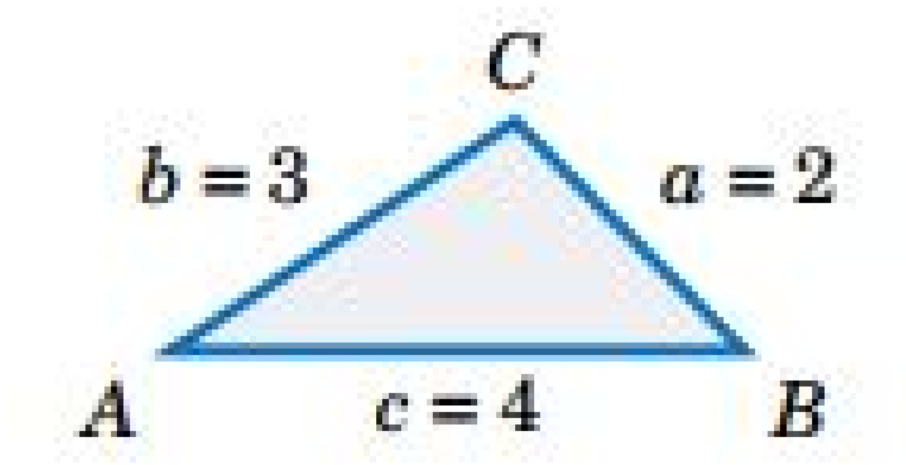

Figure 2.5: To solve the triangle when 3 sides are given

```

cos_B = (c^2+a^2-b^2)/(2*c*a) //from law of cosines
B=acosd(cos_B)
printf('Angle B=%.1f degrees\n',B)
cos_C = (b^2+a^2-c^2)/(2*b*a) //from law of cosines
C=acosd(cos_C)
printf('Angle C=%.1f degrees\n',C)

A=180-(C+B)
printf('Angle A=%f degrees',A)
```

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

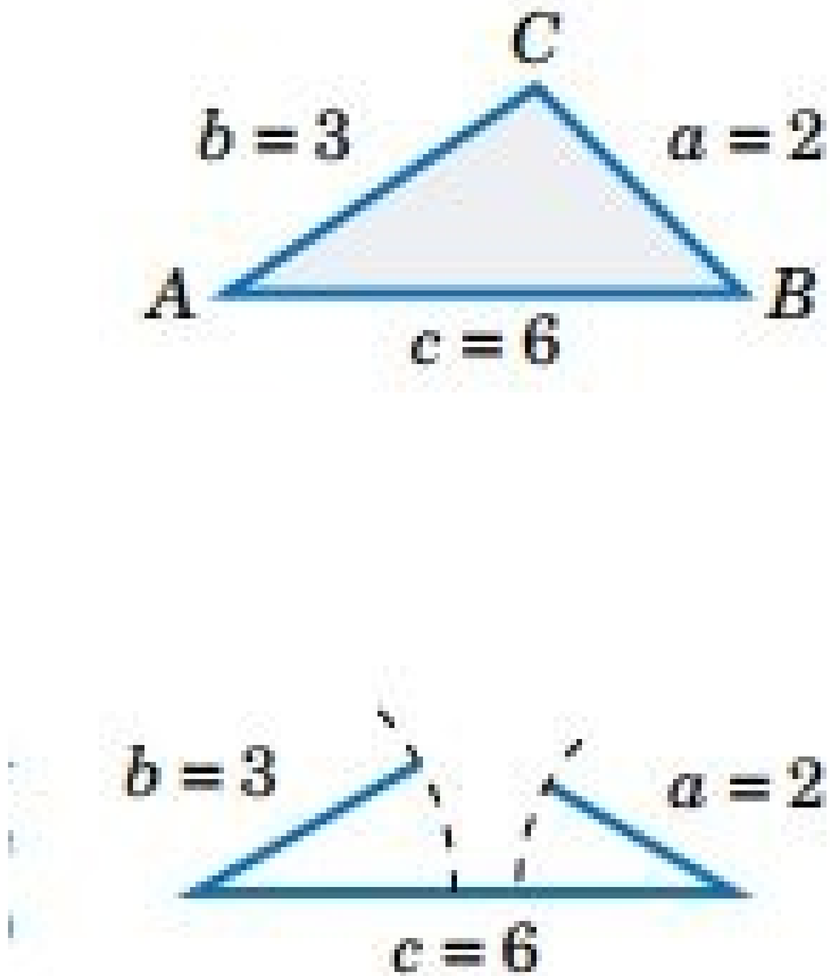

Figure 2.6: To determine solution of a triangle when 3 sides are given

```
//Example 2.7
//To determine solution of a triangle when 3 sides 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)
C=180-(A+B)
printf('1st possible answer set\nAngle B=%.1f degree
   \nAngle C=%.1f degree\nlength of c=%.0f units\n\n
   ',B,C,c)

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

```
//Example 2.10
//To solve the triangle when 2 sides and included
   angle is given
clc,clear

a=5 //side oposite to vertex a
b=3 //side opposite to vertex b
C=96 //angle at vertex C

ApB=180-C //A + B
//using law of tangents
 AmB =2* atand(  tand(ApB/2)*(a-b)/(a+b) )  //A-B

```

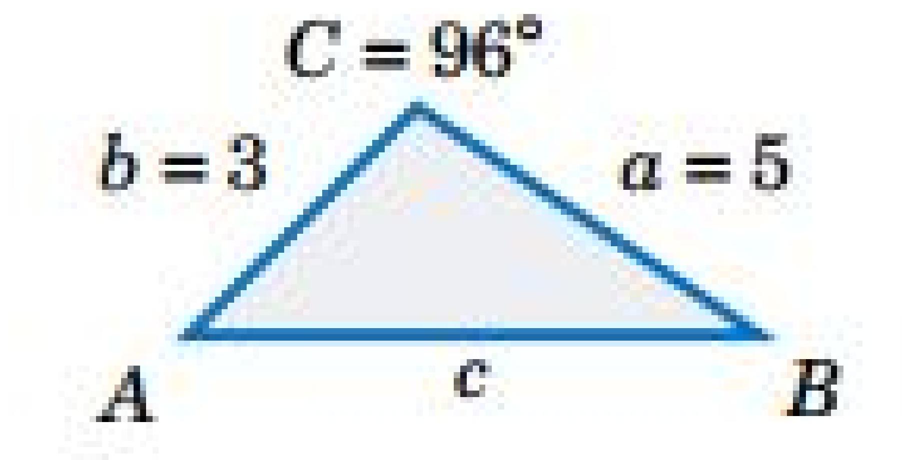

Figure 2.7: To solve the triangle when 2 sides and included angle is given

```
//solving for A and B using AmB and ApB
A= (AmB + ApB)/2
B=  ApB - A

c=a*sind(C)/sind(A) //law of sines
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

```
//Example 2.11
//To check the solution of triangle using Mollweide equation
clc,clear

```

```
c=6.09 //side oposite to vertex C
a=5 //side opposite to vertex A
b=3 //side opposite to vertex B

A=54.7 //angle at vertex A
B=29.3 //angle at vertex B
C=96 //angle at vertex C

LHS = (a-b)/c
RHS = sind((A-B)/2)/cosd(C/2)
printf(' LHS =  (a-b)/c                    = %.4f\n',LHS)
printf('  RHS =  sin((A-B)/2)/cos(C/2) = %.4f\n\n',
   RHS)

printf('Small difference in LHS and RHS is due to
   rounding off.\ni.e.Mollweides equation is holding
    true.\n')
printf('THE SOLUTION OF TRIANGLE IS CORRECT')
```

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

```
//Example 2.12
//To determine if a triangle can be formed with
   given dimension
clc,clear

c=9 //side oposite to vertex C
a=6 //side opposite to vertex A
b=7 //side opposite to vertex B

A=55 //angle at vertex A
B=60 //angle at vertex B
C=65 //angle at vertex C
printf('Sum of angles=180\n')
```

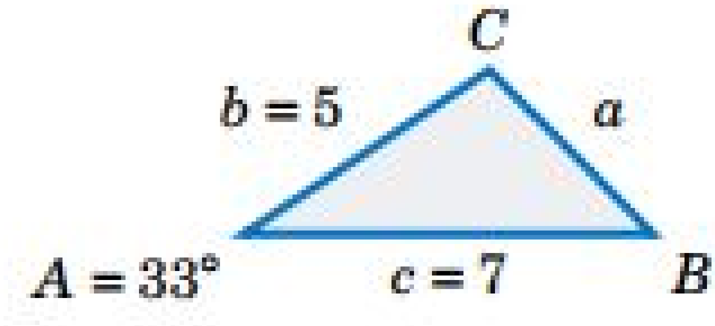

Figure 2.8: To determine area of triangle when 2 sides and an angle is given

```
printf('Smallest and largest sides are opposite to
    smallest and largest angle respectively\n\n')

LHS = (a+b)/c
RHS = cosd((A-B)/2)/sind(C/2)
printf(' LHS =  (a+b)/c                 = %.2f\n',LHS)
printf(' RHS =  cos((A-B)/2)/sin(C/2) = %.2f\n\n',
    RHS)

printf('As we can see, LHS is not equal to RHS.\ni.e
    .Mollweides equation is not holding true.\n')
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 given

```
//Example 2.13
```

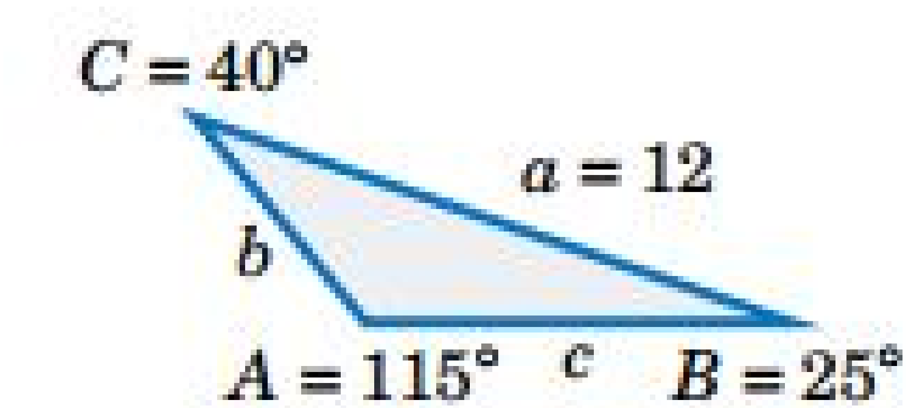

Figure 2.9: To determine area of triangle when 3 angles and a side is given

```
//To determine area of triangle when 2 sides and an angle is given
clc,clear

c=7 //side oposite to vertex C
A=33 //angle at vertex A
b=5 //side opposite to vertex B

area_K = b*c*sind(A)/2
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 given

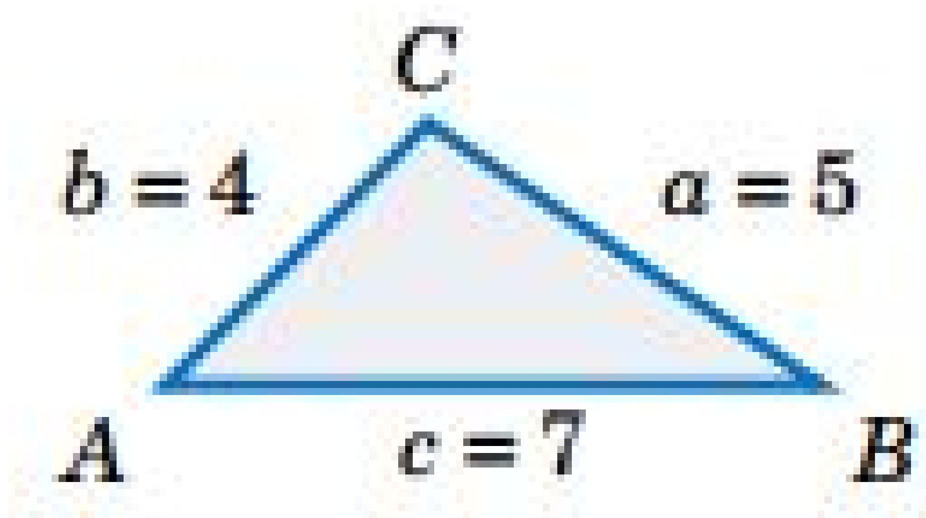

Figure 2.10: To determine area of triangle when 3 sides are given

```
//Example 2.14
//To determine area of triangle when 3 angles and a
    side is given
clc,clear

A=115 //angle at vertex A
a=12 //side opposite to vertex A
B=25 //angle at vertex B
C=40 //angle at vertex C

area_K = a^2*sind(B)*sind(C)/(2*sind(A))
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

```
//Example 2.15
//To determine area of triangle when 3 sides are given
clc,clear

c=7 //side oposite to vertex C
a=5 //side opposite to vertex A
b=4 //side opposite to vertex B

s= (a+b+c)/2  //semi perimeter
area_K = sqrt(s*(s-a)*(s-b)*(s-c)) //using herons formula
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

```
//Example 2.16
//To determine area of triangle when 3 sides are given
clc,clear

c=0.0000029 //side oposite to vertex C
a=1000000 //side opposite to vertex A
b=999999.9999979 //side opposite to vertex B

s= (a+b+c)/2  //semi perimeter
area_K = sqrt(s*(s-a)*(s-b)*(s-c)) //using herons formula
printf('Area of triangle ABC = %.3f square units\n\n',area_K)

printf('Note:\n')
```

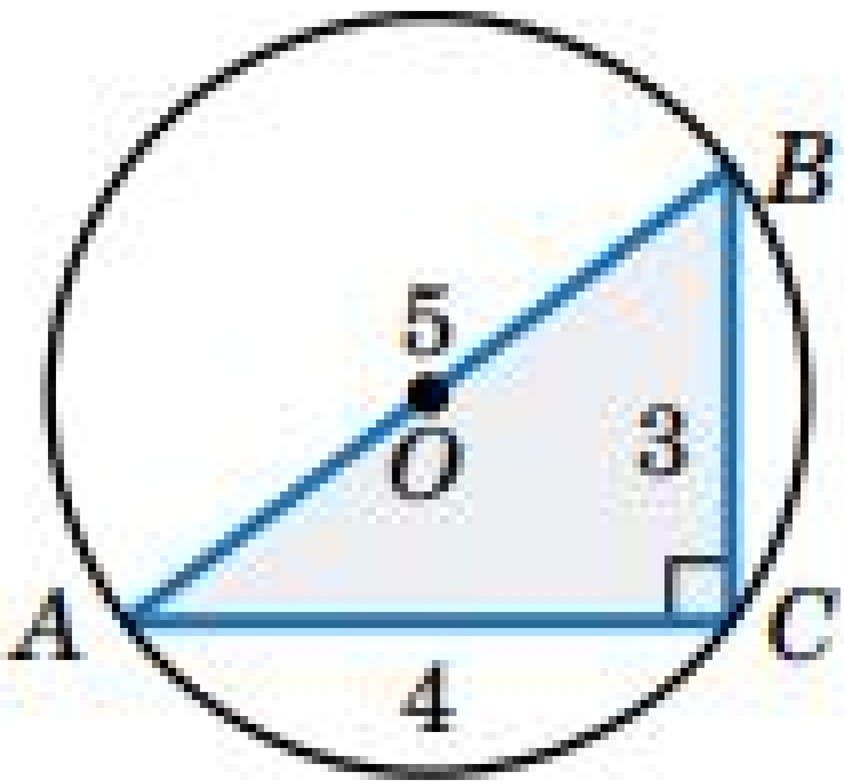

Figure 2.11: To find radius of circumscribed circle for triangle ABC

```
printf('In calculators like TI-83 plus, due to
    rounding off etc s will be 1000000\n')
printf('Therefore (s-a) is zero. And area will be
    zero according to herons formula\n')
printf('Due to large number of digits in scilab,(s-a
    ) is not zero. Thus, area is non-zero above.')
```

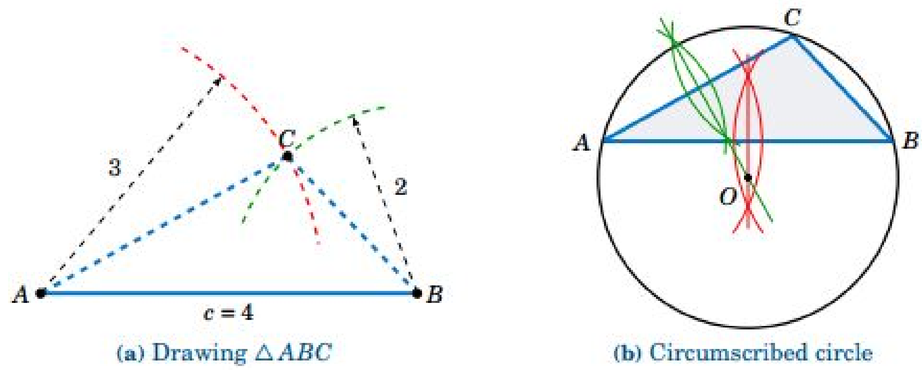

Figure 2.12: To find the radius of circumscribed circle for triangle ABC

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

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

c=5//side oposite to vertex C
a=3//side opposite to vertex A
b=4//side opposite to vertex B

cos_A = (c^2+b^2-a^2)/(2*c*b) //from law of cosines
A= acosd(cos_A)
diameter=(a/sind(A))
radius = diameter/2
printf('Radius of circumscribed circle = %.1f units
   \n',radius)
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

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

c=4 //side oposite to vertex C
a=2 //side opposite to vertex A
b=3 //side opposite to vertex B

cos_A = (c^2+b^2-a^2)/(2*c*b) //from law of cosines
A= acosd(cos_A)
diameter=(a/sind(A))
radius = diameter/2
printf('Radius of circumscribed circle = %.2f units
   \n\n',radius)

//To draw the triangle
printf('NOTE:\nPROCEDURE TO DRAW THE TRIANGLE ABC\n'
   )
printf('Use a ruler to draw the longest side AB of
   length c = 4.\n')
printf('Use a compass to draw arcs of radius 3 and 2
    centered at A and B respectively.\n')
printf('The intersection of the arcs is the vertex C
   .\n\n')

//To draw the circumscribed circle
printf('PROCEDURE TO DRAW CIRCUMSCRIBED CIRCLE\n')
printf('Draw the perpendicular bisectors of AB and
   AC.\n')
printf('Their intersection is the center O of the
```

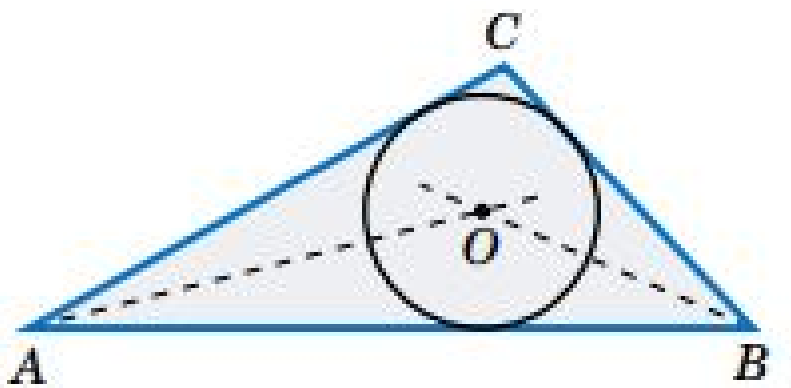

Figure 2.13: To determine radius of inscribed circle ABC

```
    circle.\n')
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

```
//Example 2.19
//To determine radius of inscribed circle ABC
clc,clear

c=4 //side oposite to vertex C
a=2 //side opposite to vertex A
b=3 //side opposite to vertex B

s= (a+b+c)/2  //semi perimeter
radius_r = sqrt((s-a)*(s-b)*(s-c)/s)
```

```
printf('Radius of inscribed circle is %.3f units=
    sqrt(5/12)units ',radius_r)
printf('\n\nNote:To obtain inscribed circle:\n(1)
    Intersect perpendicular bisectors of A and B\n')
printf('(2)Point of intersection is the centre of
    circle\n')
printf('(3)The radius is 0.645 as caculated above\n'
    )
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 functions of 2 angles are given

```
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

```
clc,clear
//Example 4.1
//To convert a degree measure to radians

deg=18  //degree measure
radian=deg*(%pi/180) //radian measure
printf('Radian measure is %f rad\n(or)\n',radian)
printf('Radian measure is (pi/%.0f)rad',1/(radian/
    %pi))
```

Scilab code Exa 4.2 To convert a radian measure to degree

```
clc,clear
//Example 4.2
//To convert a radian meeasure to degree

radian=%pi/9  //radian measure
deg=radian/(%pi/180) //degree measure
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
```

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

Scilab code Exa 4.6 To determine the length of the rope

```
clc,clear
//Example 4.6
//To determine the length of the rope

d=8 //distance between places in feet
r=2 //radius of cylinder in feet
//from the figure
DA=d/2,BE=r
DE=3 //distance from centre of container to wall

AE=sqrt(DE^2 + DA^2) //pythagoras theorem
AB=sqrt(AE^2 - BE^2) //pythagoras theorem

//all angles below are in radians
angle_AED = atan((d/2)/DE)
angle_AEB = acos(BE/AE)
angle_BEC = %pi - (angle_AED + angle_AEB)
arc_BC = BE*angle_BEC //length of arc BC
L = 2*(AB + arc_BC) //length of rope
printf('Length of the rope = %.1f ft',L)
```

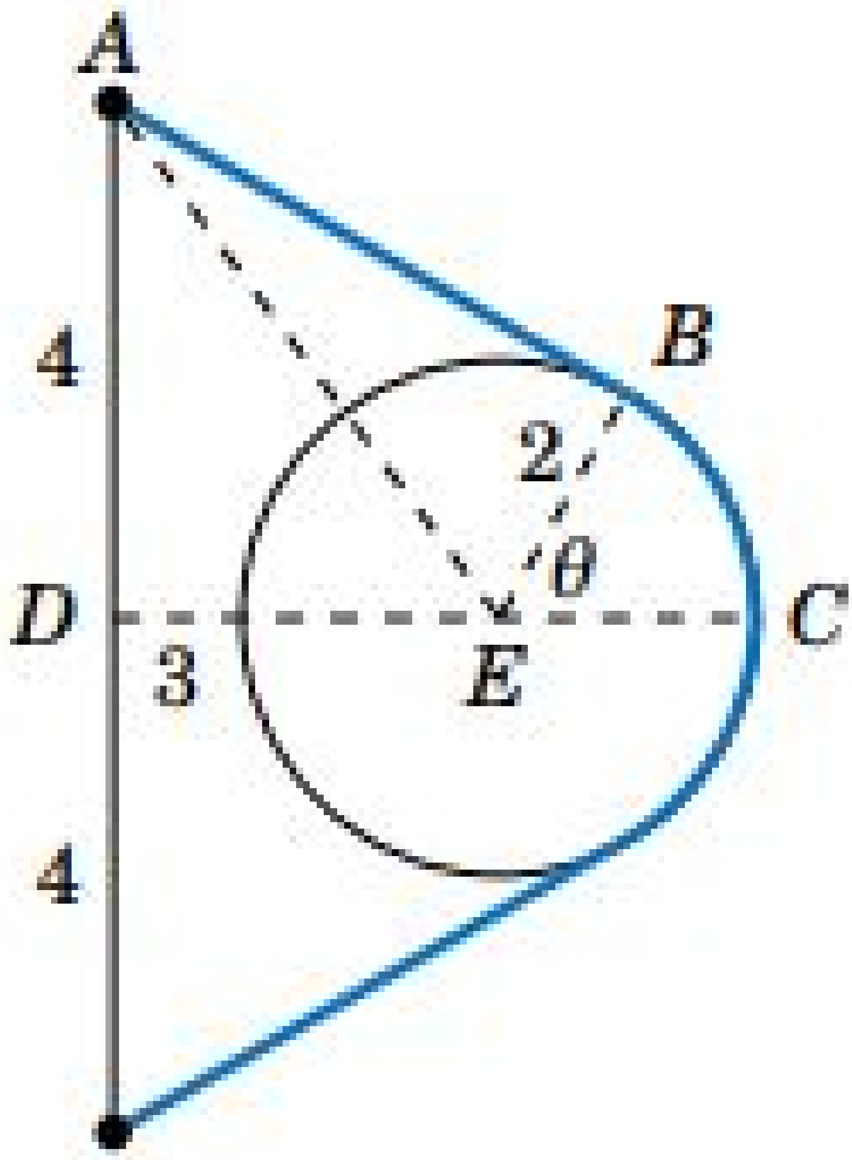

Figure 4.1: To determine the length of the rope

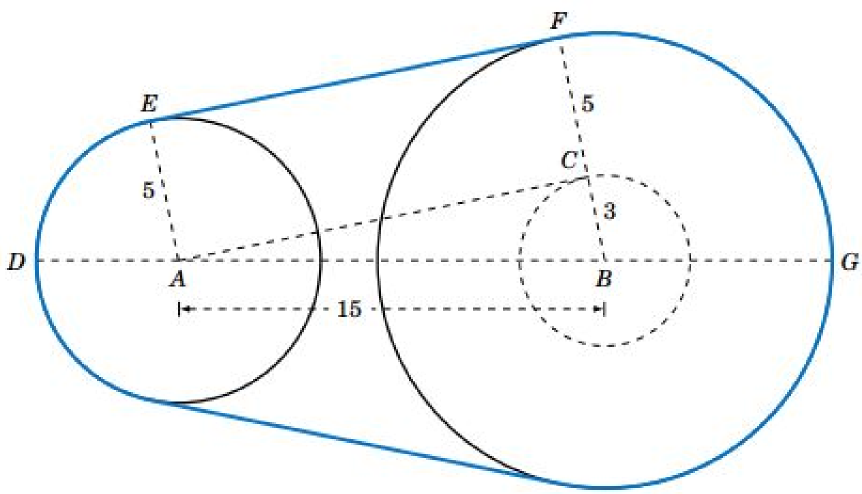

Figure 4.2: To determine the length of the belt around the pulleys

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

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

arc_DE=AE*angle_ABC //length of arc DE
arc_FG=BF*angle_GBF //length of arc FG
L=2*(arc_DE + EF + arc_FG) //length of belt
printf('Length of belt around pulley = %f cm',L)
```

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

```
clc,clear
//Example 4.8
//To find the area of sector of circle

theta= %pi/5 //angle in radian
r=4 //radius in cm
A=r*r*theta/2 //Area of sector
printf('Area of sector = %.1f*pi cm^2\n(or)\n',A/%pi
  )
printf('Area of sector = %f cm^2',A)
```

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

```
clc,clear
//Example 4.9
//To determine area of sector of a circle

theta= 117*(%pi/180) //angle in radian
```

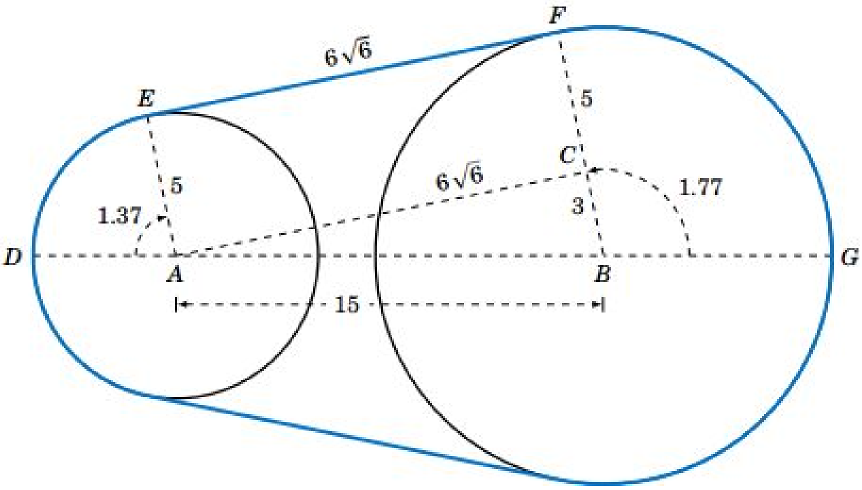

Figure 4.3: To determine area insude belt pulley system

```
r=3.5 //radius in m
A=r*r*theta/2 //Area of sector
printf('Area of sector = %.2f m^2',A)
```

Scilab code Exa 4.10 To determine area of sector of circle

```
clc,clear
//Example 4.10
//To determine area of sector of circle

s=6 //arc length in cm
r=9 //radius in cm
A=r*s/2 //Area of sector
printf('Area of sector = %.0f cm^2\n\n',A)
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')
```

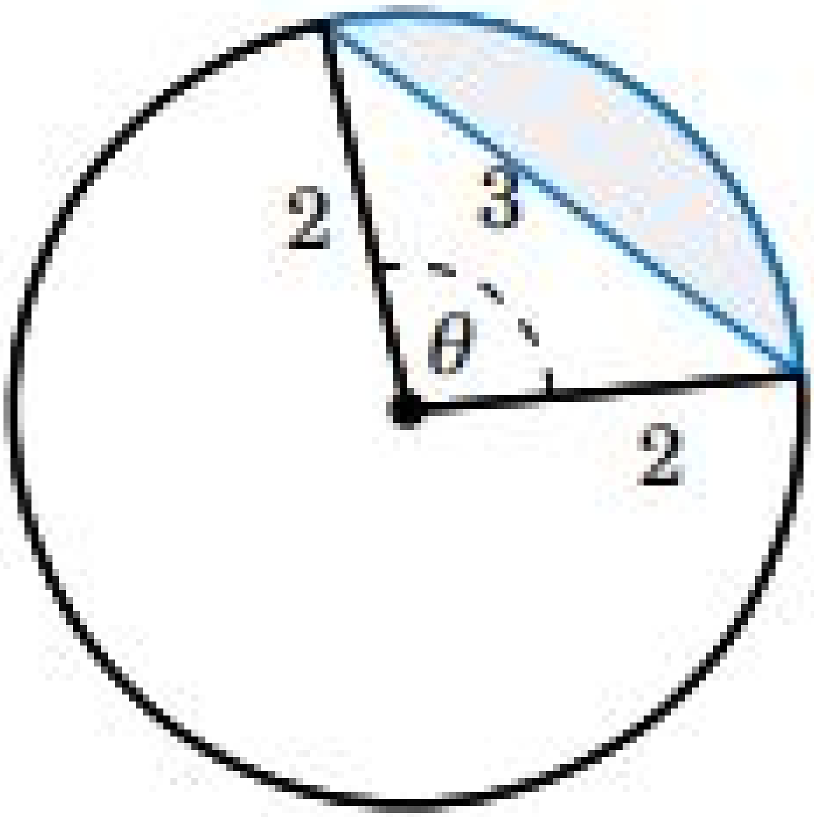

Figure 4.4: To determine area of segment formed by a chord in circle

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

```
clc,clear
//Example 4.12
```

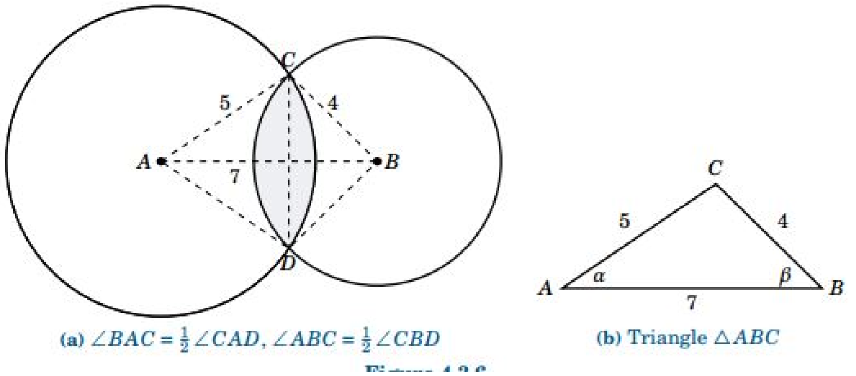

Figure 4.5: To determine area of intersection of 2 circles

```
//To determine area of segment formed by a chord in
   circle

radius = 2
chord = 3
//Use law of cosines
cos_theta = (radius^2+radius^2-chord^2)/(2*radius*
   radius)
theta=acos(cos_theta) //subtended central angle in
   radians

area_K=radius^2* (theta-sin(theta))/2
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

t=0.5 //time in second
r= 3 //radius in m of the circle
theta = %pi/3 // central angle in radian
```

```
w = theta/t //angular speed in rad /sec
v=w*r//linear speed in m/sec

printf('Angular speed= %f radian/sec\n',w)
printf('Linear speed = %f m/sec',v)

printf('\n\n(or)\n\nAngular speed= %f*pi radian/sec\n',w/%pi)
printf('Linear speed = %f*pi m/sec',v/%pi)
```

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

```
clc,clear
//Example 4.15
//To find linear and angular speed of a moving object

t=2.7 //time in second
r= 2 //radius in ft of the circle
s=35 //distance in feet

v=s/t //linear speed in ft/sec
w=v/r //angular speed in rad /sec

printf('Linear speed = %.2f ft/sec\n',v)
printf('Angular speed= %.2f radian/sec\n',w)
```

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

```
clc,clear
//Example 4.16
//To find the central angle swept by a moving object

```

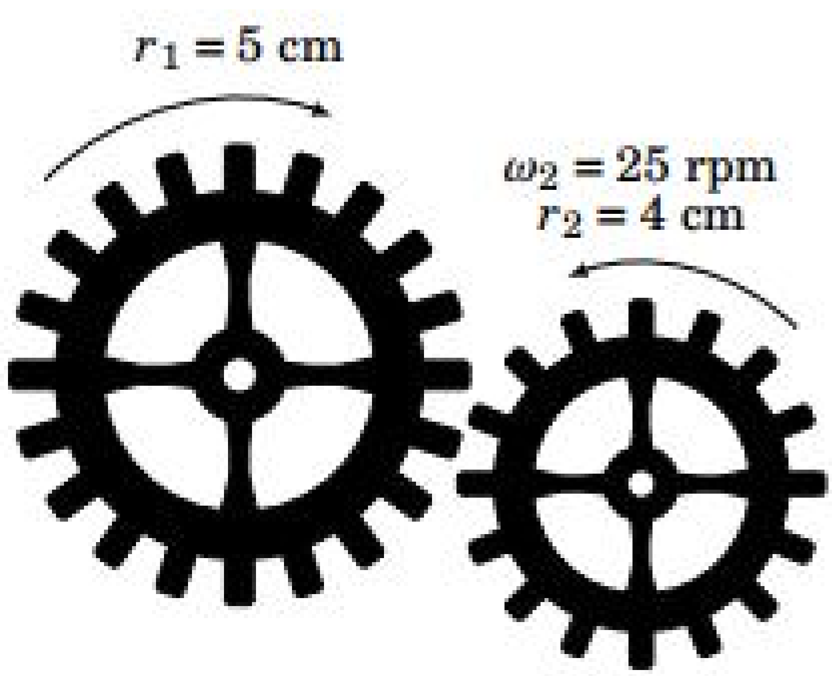

Figure 4.6: To find the angular speed of larger gear interlocked with smaller gear

```
t=3.1 //time in second
v= 10 //linear speed in m/sec
r= 4 //radius in m of the circle
s=v*t //distance in m

theta = s/r //central angle swept
printf('central angle swept = %.2f radian',theta)
```

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

```
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

```
//Example 5.1
//To sketch the graph of minus sinx in a given interval
clear,clc;

x = linspace(-0,2*%pi,50);
y = -sin(x) ;
set(gca(),"grid",[5 5]);
plot(x,y);
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red");
ylabel("$y(x)=-sin(x)$","fontsize",4,"color","red");
title("Example 5.1","color","blue","fontsize",9);
```

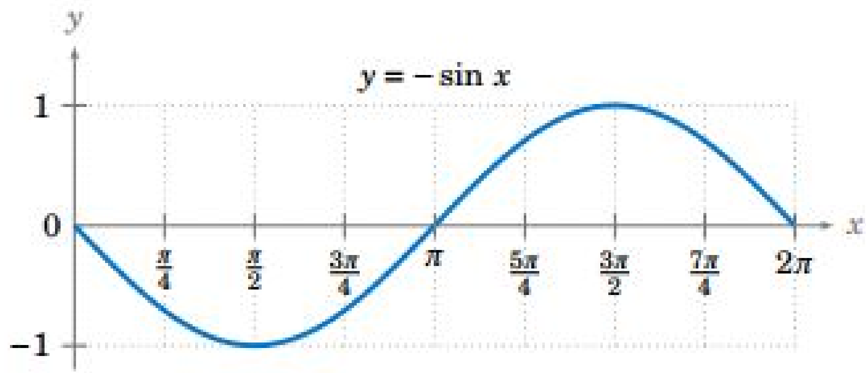

Figure 5.1: To sketch the graph of minus sinx in a given interval

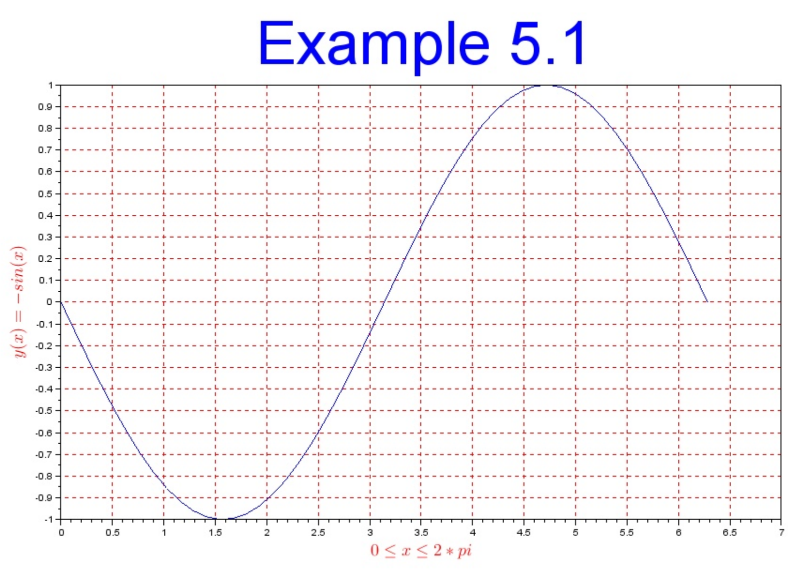

Figure 5.2: To sketch the graph of minus sinx in a given interval

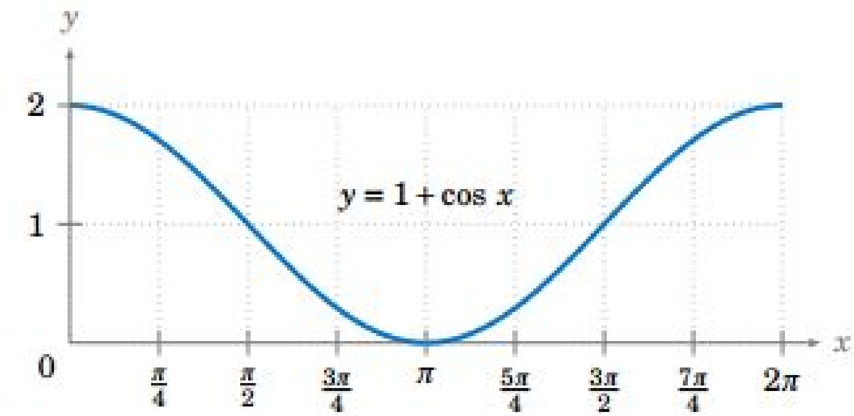

Figure 5.3: To sketch the graph of given function of in given interval

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

```
//Example 5.2
//To sketch the graph of function of 1+cos(x) in
   given interval
clear,clc;

x = linspace(-0,2*%pi,50);
y = 1+cos(x) ;
set(gca(),"grid",[5 5]);
plot(x,y);
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
   );
ylabel("$y(x)=1+cos(x)$","fontsize",4,"color","red")
   ;
title("Example 5.2","color","blue","fontsize",9);
```

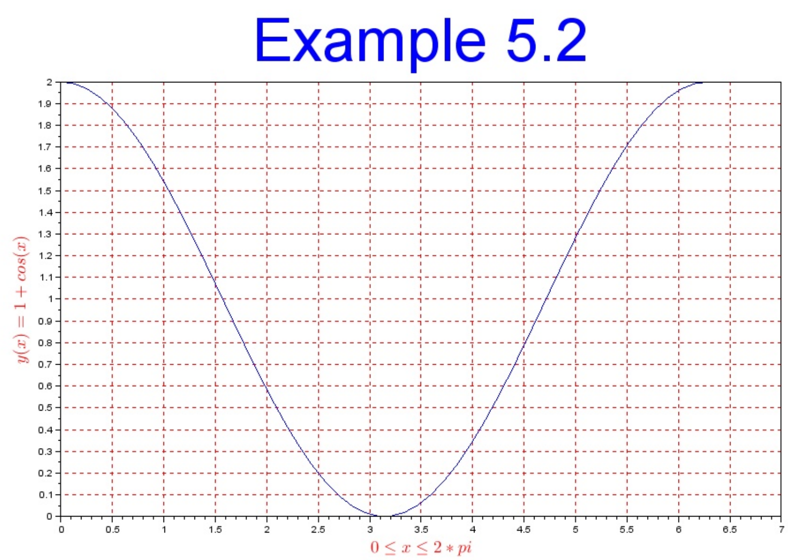

Figure 5.4: To sketch the graph of given function of in given interval

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

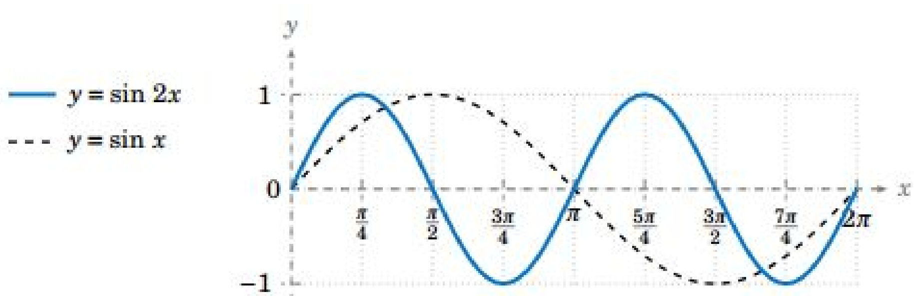

Figure 5.5: To determine the period of given sinusoidal function

```
//Example 5.4
//To determine the period of given sinusoidal
    function
clear,clc;

multiple = 2; //multiplicity of angle
period_sinx = 2*%pi ;//period of sin(x) in radians
required_period = period_sinx / multiple;
printf('Required period is %f radians',
    required_period);

//Note that sin 2x  goes twice as fast  as sin x
    .
//While sin x takes a full 2*pi radians to go
    through an entire cycle
//sin 2x goes through an entire cycle in just pi
    radians

x = linspace(-0,2*%pi,100);
y = sin(2*x) ;
z = sin(x) ;
set(gca(),"grid",[4 4]);
plot(x,y,'r-');
plot(x,z,'b-');
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
    );
ylabel("$y(x)=sin(2x)$","fontsize",4,"color","red");
title("Example 5.4","color","blue","fontsize",9);
legend(["sin(2x)";"sin(x)"]);
```

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

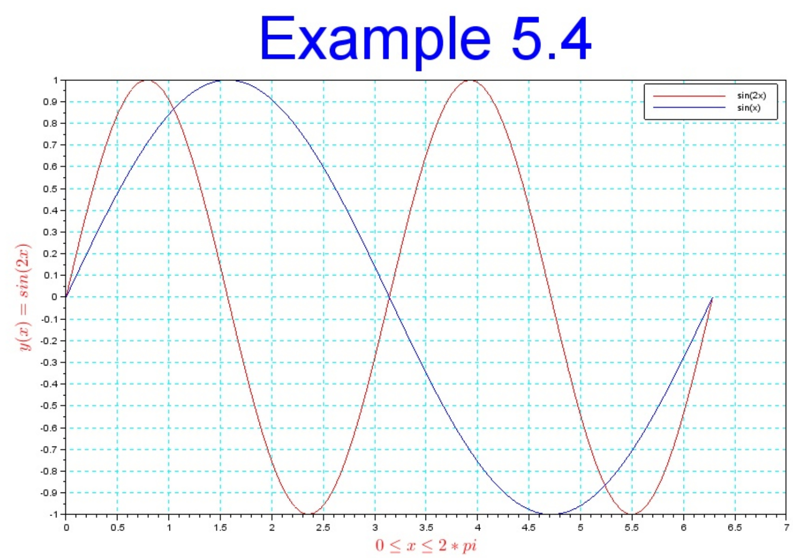

Figure 5.6: To determine the period of given sinusoidal function

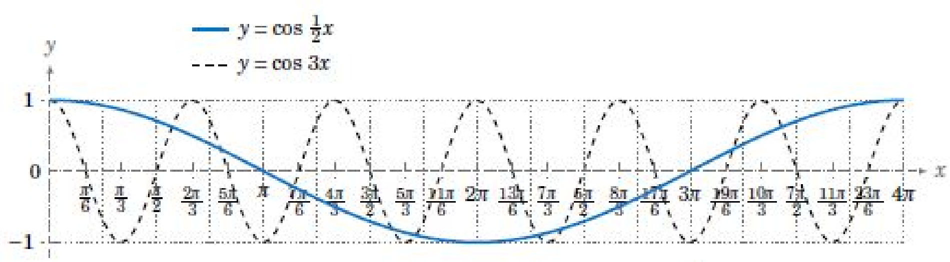

Figure 5.7: To determine the period of 2 given cosine functions

```
//Example 5.5
//To determine the period of 2 given cosine functions
clear,clc;

//y=cos(3*x)
multiple = 3; //multiplicity of angle
period_cosx = 2*%pi; //period of sin(x) in radians
required_period = period_cosx / multiple;
printf('Period of cos(3*x)is %f radians\n', required_period);

//y=cos(0.5*x)
multiple = 1/2; //multiplicity of angle
period_cosx = 2*%pi; //period of sin(x) in radians
required_period = period_cosx / multiple;
printf('Period of cos(x/2)is %f radians', required_period);

x = linspace(-0,4*%pi,200);
y = cos(3*x) ;
z = cos(x/2) ;
set(gca(),"grid",[4 4]);
plot(x,y,'r-');
plot(x,z,'b-');
xlabel("$0\le x\le 4*pi$","fontsize",4,"color","red");
title("Example 5.5","color","blue","fontsize",9);
legend(["y = cos(3x)";"y = cos(x/2)"]);
```

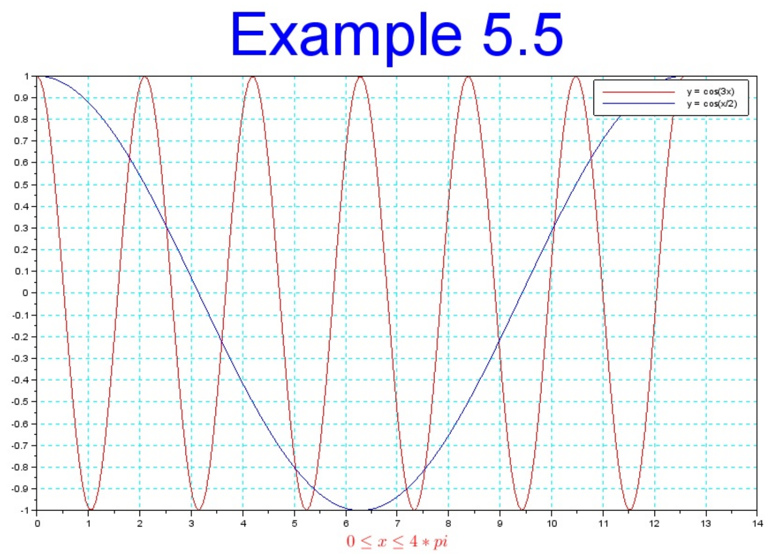

Figure 5.8: To determine the period of 2 given cosine functions

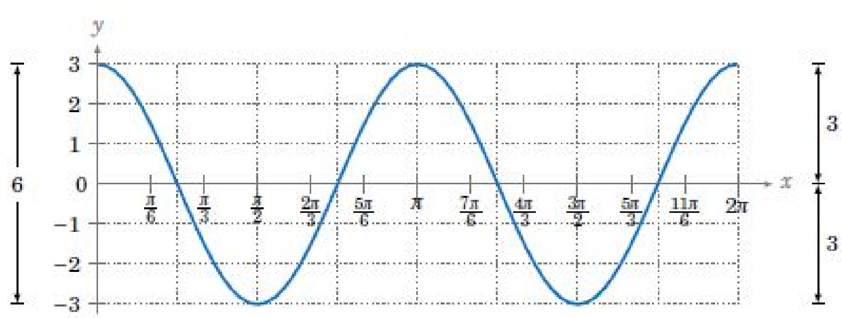

Figure 5.9: To determine the amplitude and period of given function

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

```
//Example 5.6
//To determine the amplitude and period of given function
clear,clc;

x = linspace(-0,4*%pi,200);
y = 3*cos(2*x) ; //given function
amplitude = y/cos(2*x);
printf('Amplitude = %f',amplitude);

multiple = 2; //multiplicity of angle
period_cosx=2*%pi; //period od cos(x)
period_required = period_cosx / multiple;
printf('\nPeriod = %f radians',period_required);

x = linspace(-0,2*%pi,50);
y = 3*cos(2*x) ;
set(gca(),"grid",[5 5]);
plot(x,y);
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red");
ylabel("$y(x)=3*cos(2*x)$","fontsize",4,"color","red");
title("Example 5.6","color","blue","fontsize",9);
```

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

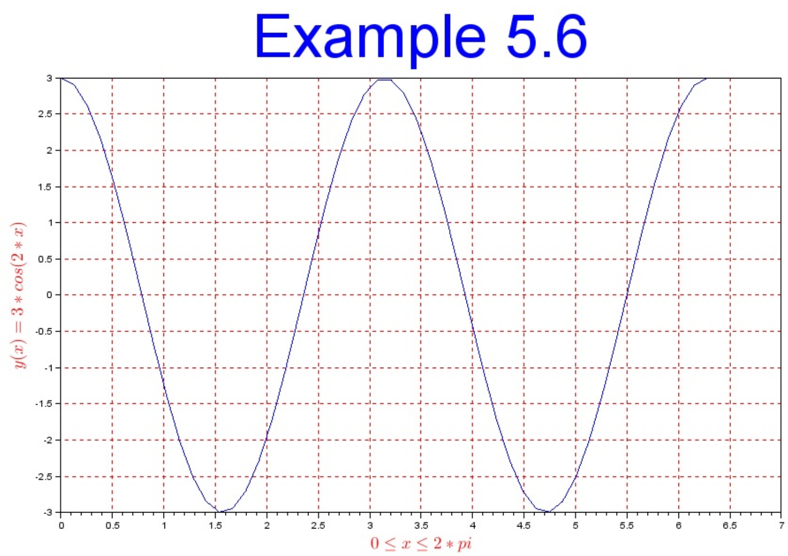

Figure 5.10: To determine the amplitude and period of given function

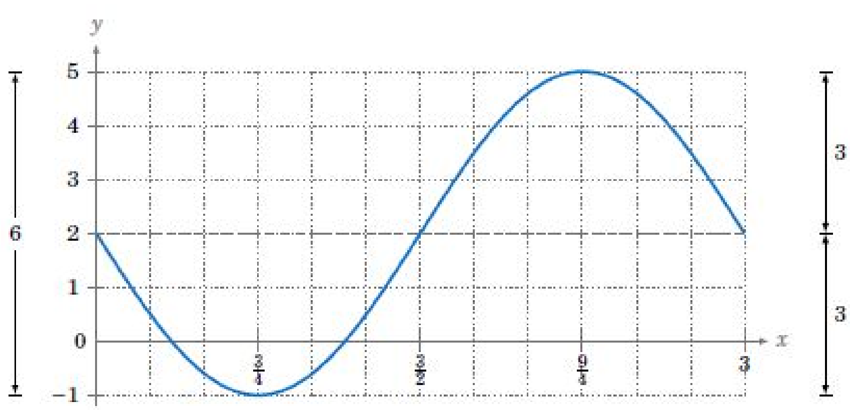

Figure 5.11: To find amplitude and period of given composite function

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

x = linspace(0,3,200);
y1=2 ;//1st part of given function
amplitude1=y1 ;//amplitude numerically same for constant function
y2= -3*sin((2*%pi/3)*x); //second part of given function
amplitude2 = abs(y2/sin((2*%pi/3)*x)) ;//amplitude of part 2
//Note: adding 2 doesnt change ampitude
//It just causes the upward shift of graph
maax =amplitude1 + amplitude2; //altered maximum due to adding of 2
minn =amplitude1 - amplitude2;//altered minimum due to adding of 2
amplitude = (maax-minn)/2;//required amplitude
printf('Amplitude = %f',amplitude);

multiple=2*%pi/3 ;//multiplicity of angle
period_sinx=2*%pi;//period of sin_x
period_required = period_sinx/ multiple;
printf('\nRequired period is %f radians',period_required);

x = linspace(0,3,200);
y = 2 -3*sin((2*%pi/3)*x)
set(gca(),"grid",[5 5]);
plot(x,y);
xlabel("$0\le x\le 3$","fontsize",4,"color","red");
ylabel("$y(x)= 2 -3*sin((2*%pi/3)*x)$","fontsize",4,"color","red");
title("Example 5.7","color","blue","fontsize",9);
```

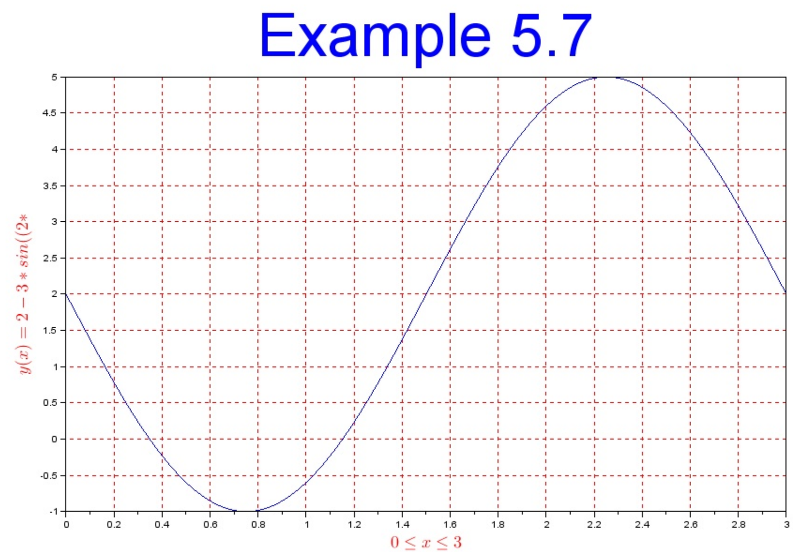

Figure 5.12: To find amplitude and period of given composite function

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

```
//Example 5.8
//To find the amplitude and period of given function
clear,clc;

//Period
printf('PERIOD:\n')
printf('This isnt a periodic function as x^2 is
   linearly related to x \n')
printf('and hence period doesnt exist\n')

```

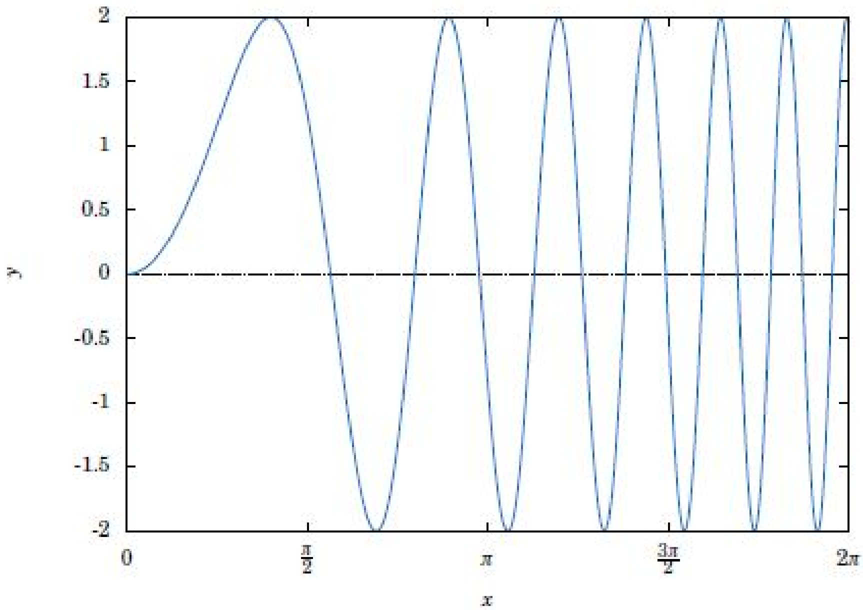

Figure 5.13: To find the amplitude and period of given function

```
//Amplitude
x = linspace(0,2*%pi,200);
y = 2*sin(x^2);
amplitude= y/sin(x^2);
printf('AMPLITUDE:\n')
printf('Amplitude exists unlike period\n')
printf('Because sine component of the given function never exceeds 1\n')
printf('Function value is always ;ess than constant factor adjacent to sine\n')
printf('Hence amplitude is the constant factor multiplied with sine component\n\n')
printf('Amplitude = %f as calculated\n',amplitude)


x = linspace(0,2*%pi,200);
y = 2*sin(x^2) ;
set(gca(),"grid",[5 5]);
```

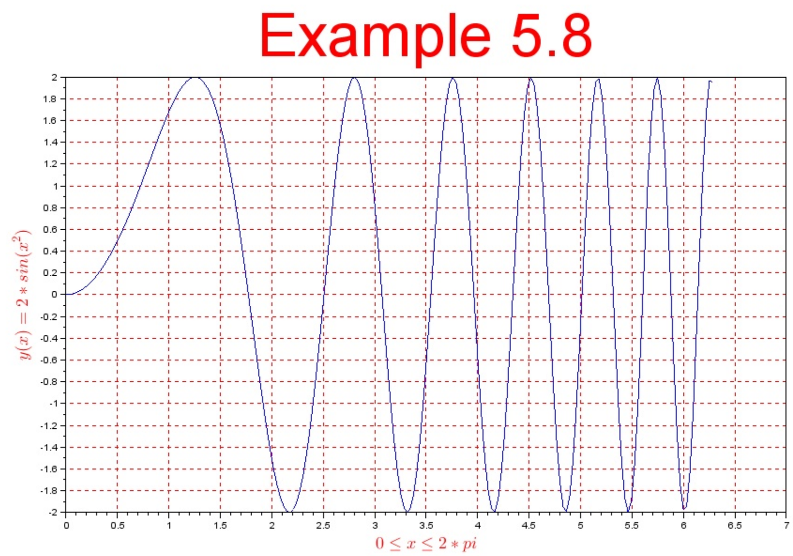

Figure 5.14: To find the amplitude and period of given function

```
plot(x,y,'b');
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
    );
ylabel("$y(x)= 2*sin(x^2)$","fontsize",4,"color","
    red");
title("Example 5.8","color","red","fontsize",9);
```

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

```
//Example 5.9
//To find the amplitude and period of given function
clear,clc;

```

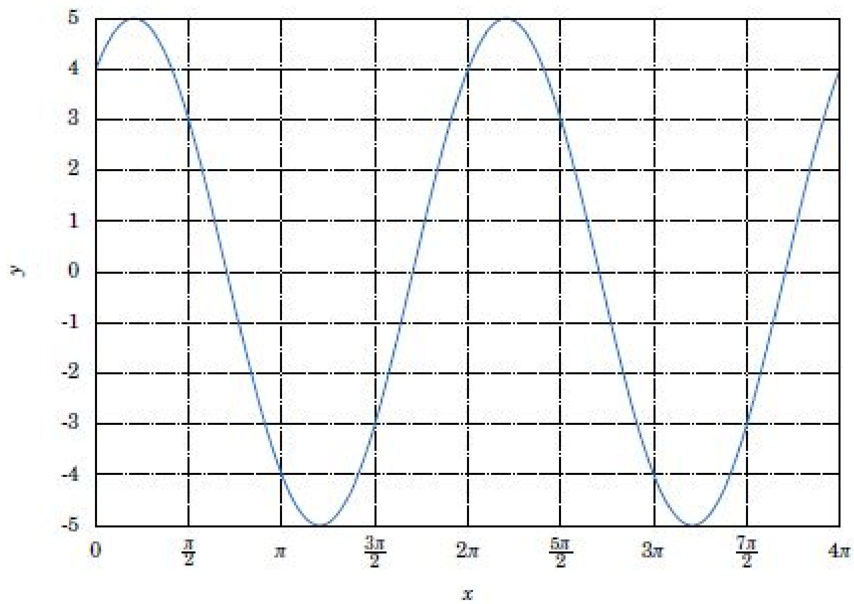

Figure 5.15: To find the amplitude and period of given function

```
//amplitude
x = linspace(-0,4*%pi,200);
y1=3*sin(x); //1st part of given function
amplitude1=y1/sin(x); //amplitude of part 1
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
```

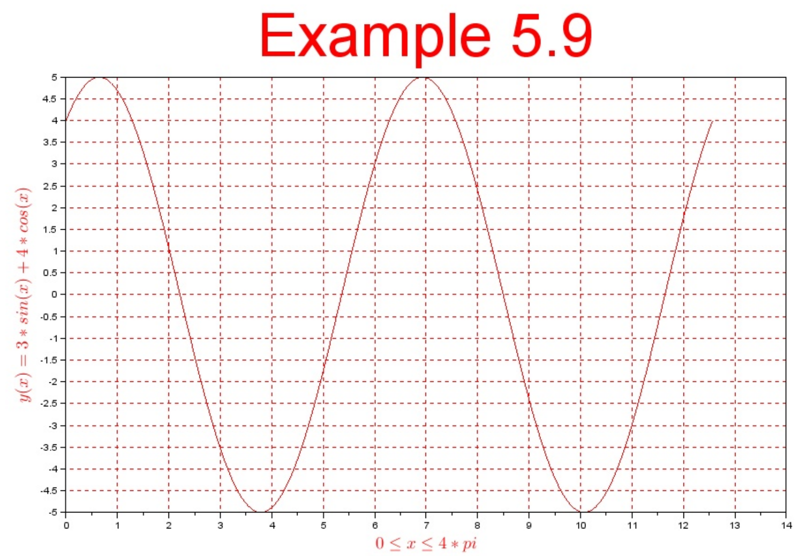

Figure 5.16: To find the amplitude and period of given function

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

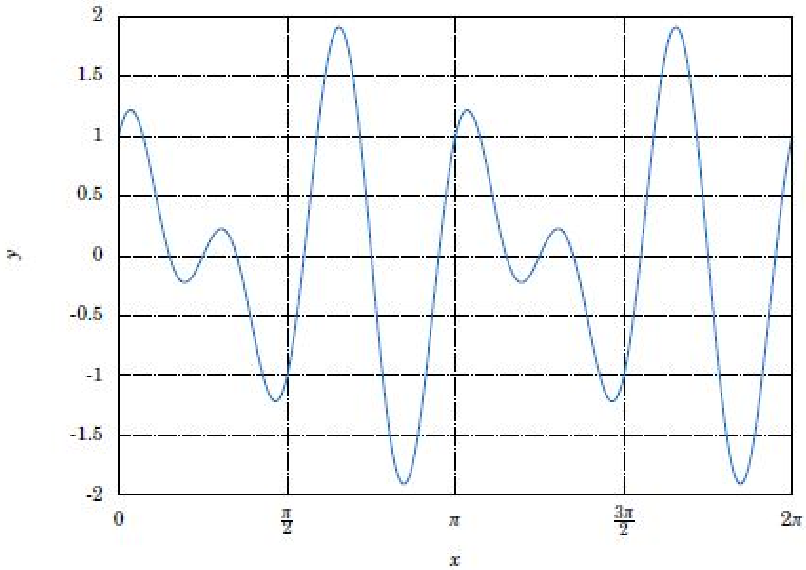

Figure 5.17: To find the period of given function

Scilab code Exa 5.10 To find the period of given function

```
//Example 5.10
//To find the period of given function
clear,clc;

x = linspace(-0,2*%pi,200);
y1=cos(6*x); //1st part of given function
multiple1=6;//multiplicity of angle
period_cosx=2*%pi ;//period of cos(x) is 2 pi
period1= period_cosx/ multiple1;
printf('Note: Period of cos(%d*x)= %f radians\n',
    multiple1,period1);
y2= sin(4*x); //second part of given function
multiple2=4; //multiplicity of angle
```

```
period_sinx=2*%pi ;//period of sin(x) is 2 pi
period2= period_sinx/multiple2;
printf('Period of sin(%d*x)= %f radians\n',multiple2
    ,period2);

locm = %pi ;//LCM of period1 and period 2
period = locm ;//final period
printf('\nRequired period is %f radians',period);
x = linspace(0,2*%pi,200);
y = cos(6*x)+ sin(4*x);//given function
set(gca(),"grid",[5 5]);
plot(x,y,'r');
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
    );
ylabel("$y(x)= sin(4*x) +cos(6*x)$","fontsize",4,"
    color","red");
title("Example 5.10","color","red","fontsize",9);
```

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

```
//Example 5.11
//To find the amplitude phase shift and period of
    given function
clear,clc;

x = linspace(-0,2*%pi,200);
deviation = %pi; //deviation from multiples of x
y = 3*cos(2*x- deviation) ;//given function
amplitude = y/ cos(2*x- %pi) ;
printf('Amplitude = %f\n',amplitude);
```

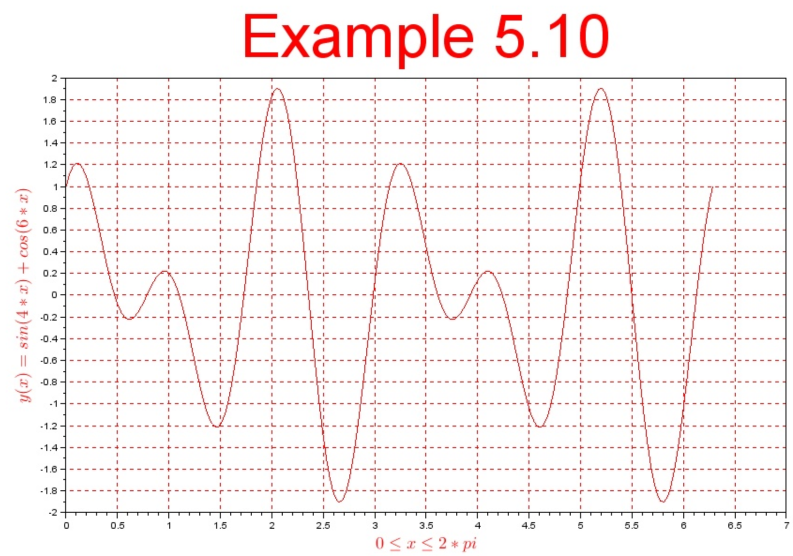

Figure 5.18: To find the period of given function

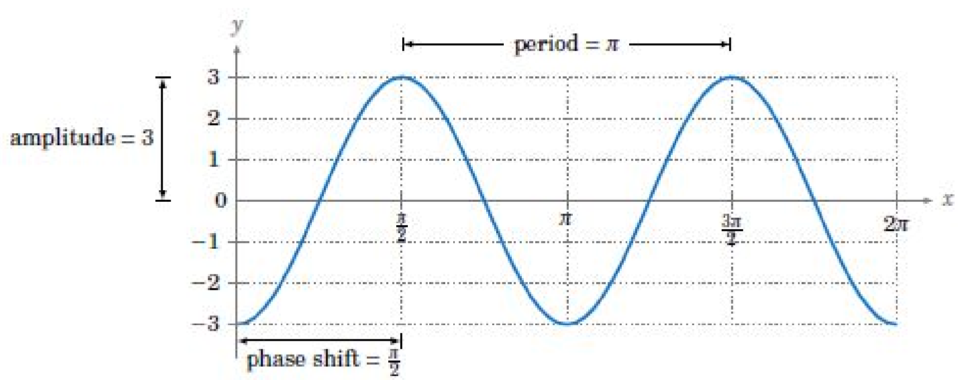

Figure 5.19: To find the amplitude phase shift and period of given function

```
multiple = 2; //multiplicity of angle
period_cosx = 2*%pi; //period of sin(x) in radians
required_period = period_cosx / multiple;
printf('Required period = %f radians\n', required_period);
phase_shift = deviation / multiple;
printf('Phase shift = %f radians',phase_shift);

x = linspace(-0,2*%pi,200);
y = 3*cos(2*x- %pi) ;
set(gca(),"grid",[5 5]);
plot(x,y);
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red");
ylabel("$y(x)=3*cos(2*x- pi)$","fontsize",4,"color","red");
title("Example 5.11","color","blue","fontsize",9);
```

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

```
//Example 5.12
//To find the amplitude phase shift and period of given function
clear,clc;

x = linspace(-%pi/6,4*%pi/3,200);
deviation = -%pi/2;//deviation from multiples of x
y = -2*sin(3*x- deviation) ;// given function
amplitude = abs(y/(sin(3*x- deviation)) );
printf('Amplitude = %f\n',amplitude);
```

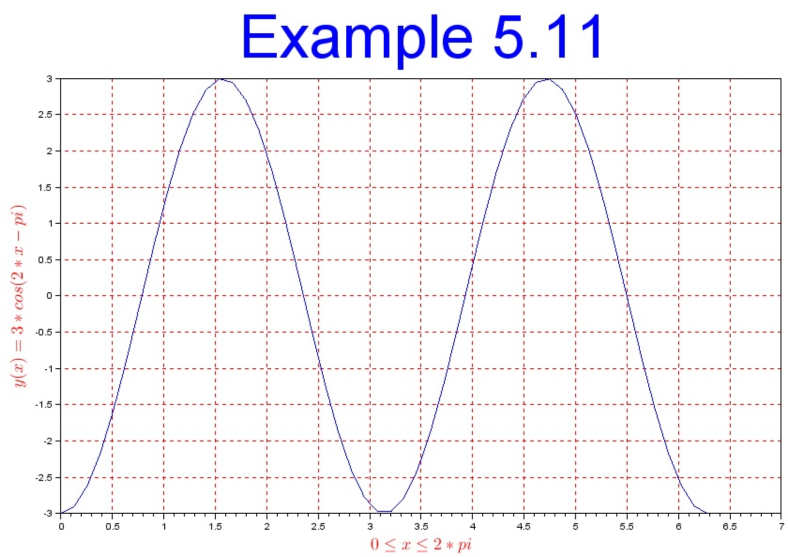

Figure 5.20: To find the amplitude phase shift and period of given function

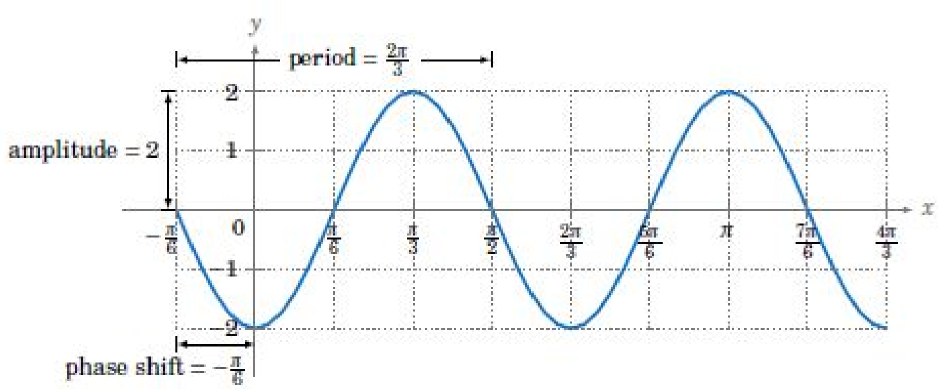

Figure 5.21: To find the amplitude phase shift and period of given function

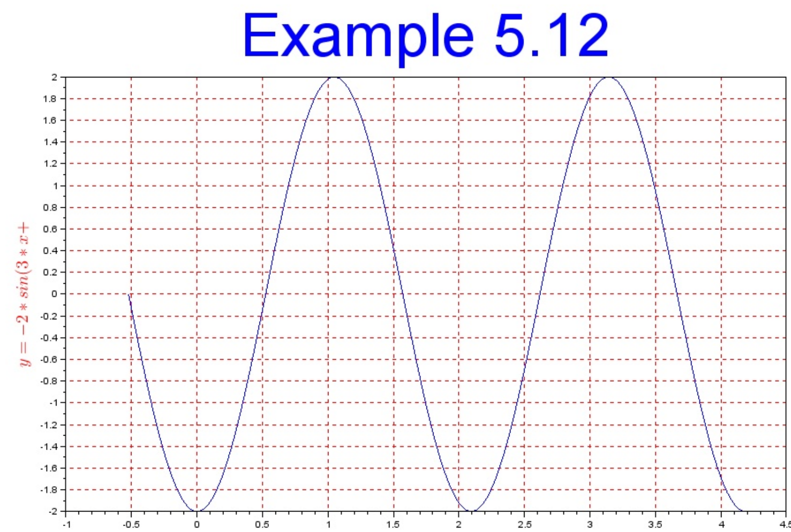

Figure 5.22: To find the amplitude phase shift and period of given function

```
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

```
clc,clear;
//Example 5.13
//To determine inverse sine function of a given
   value

given =  sin(%pi/4); //given value
answer= asin(given); //final answer

printf('Required answer is %f radians',answer);
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

```
clc,clear;
//Example 5.14
//To determine inverse sine function of a given
   value

given =  sin(5*%pi/4); //given value
answer= asin(given); //final answer

printf('Required answer is %f radians',answer);
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

```
clc,clear;
//Example 5.15
//To determine inverse cosine function of a given
   value

given =  cos(%pi/3); //given value
answer= acos(given); //final answer

printf('Required answer is %f radians',answer);
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

```
clc,clear;
//Example 5.16
//To determine inverse cosine function of a given
   value

given =  cos(4*%pi/3); //given value
answer= acos(given); //final answer

printf('Required answer is %f radians',answer);
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

```
clc,clear;
//Example 5.17
//To determine inverse tan function of a given value

given =  tan(%pi/4); //given value
```

```
answer= atan(given); //final answer

printf('Required answer is %f radians',answer);
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

```
clc,clear;
//Example 5.18
//To determine inverse tan function of a given value

given =  tan(%pi); //given value
answer= atan(given);//final answer

printf('Required answer is %f radians',answer);
```

Scilab code Exa 5.19 To determine exact value of given expression involving inverse trigonometric functions

```
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

```
clc,clear
//Example 6.3
//To solve the given equation

sec_theta = 1/2
cos_theta = 1 / sec_theta
printf('cos(theta) = %f as calculated\n',cos_theta)
printf('But value of cos function can never exceed
    unity\n')
printf('Thus, NO SOLUTION exists')
```

Scilab code Exa 6.4 To solve the given equation

```
clc,clear
//Example 6.4
//To solve the given equation

//Given equation is cos_theta = tan_theta
```

```
//simplyfying given equation, we get
//(sin_theta)^2 + sin_theta - 1 = 0
//Solve for sin_theta as follows
p=[1 1 -1]
sin_theta= roots(p)
printf('Values of sin(theta) after simplifying and solving = %f and %f\n',sin_theta(1),sin_theta(2))
printf('Eliminate %f as sin_theta cant be below -1',sin_theta(1))

//Since sin_theta is +ve, 2 solutions exist. in 1st and 2nd quadrant
theta_1=asin(sin_theta(2));  //in 1st quadrant
theta_2=%pi-asin(sin_theta(2));//the reflection in 2nd quadrant

printf('\n\nSOLUTIONS:\n')
printf('%f radians\n%f radians',theta_1,theta_2)

printf('\n\nGENERAL SOLUTIONS:\n')
printf('%f + integer multiples of 2pi \n',theta_1)
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 numbers

```
clc,clear
//Example 6.9
//To find the result of basic operations on 2 given complex numbers

z1 = complex(-2,3)
z2 = complex(3,4)

summ = z1+z2
```

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

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

```
clc,clear
//Example 6.10
//To represent given complex number in trigonometric
    form

z=-2 + -1*%i ;//given number
x=real(z) ;//real part
y=imag(z) ;//imaginary part

//theta is in third quadrant as x and y are -ve
theta=180 + atand(y/x);
r=sqrt(x^2+y^2) ;//modulus of z
printf('z= %f + i* %f  can be written as: \n',real(z
    ),imag(z))
```

```
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 formula

```
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 theorem

```
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

cube_root_1 = r1 *(cosd(theta1)+ %i*sind(theta1))
cube_root_2 = r2 *(cosd(theta2)+ %i*sind(theta2))
cube_root_3 = r3 *(cosd(theta3)+ %i*sind(theta3))

printf('cuberoot 1: %f + %f*i\n',real(cube_root_1),
    imag(cube_root_1))
printf('cuberoot 2: %f + %f*i\n',real(cube_root_2),
    imag(cube_root_2))
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

```
clc,clear
//Example 6.15
//To convert from polar to cartesian coordinates

//part(a)
r=2 ;
theta=30 ;
x=r*cosd(theta) ;
y=r*sind(theta) ;
printf('(a)(x,y)= (%f,%f)\n',x,y) ;

//part(b)
r=3 ;
theta=3*%pi/4 ;
x=r*cos(theta) ;
y=r*sin(theta) ;
```

```
printf('(b)(x,y)= (%f,%f)\n',x,y) ;

//part(c)
r=-1 ;
theta=5*%pi/3 ;
x=r*cos(theta) ;
y=r*sin(theta) ;
printf('(c)(x,y)= (%f,%f)',x,y) ;
```

Scilab code Exa 6.16 To convert from cartesian to polar coordinates

```
clc,clear
//Example 6.16
//To convert from cartesian to polar coordinates

//part(a)
x=3 ;
y=4 ;

//53.13 is in same quadrant as(3,4)
r=sqrt(x^2+y^2) ;
theta=atand(y/x) ;
printf('PART A\n(r,theta)= %f,%f',r,theta) ;
printf('\nOR\n') ;
r=-sqrt(x^2+y^2) ;
//tan theta is +ve in 3rd quadrant
//so 180 + 53.33 is also a permissible value
theta=180 + atand(y/x) ;
printf('(r,theta)= %f,%f',r,theta) ;

//part(b)
x=-5 ;
y=-5 ;

//225 is in same quadrant as(-5,-5)
```

```
//tan theta is +ve in 3rd quadrant
r=sqrt(x^2+y^2) ;
theta=180+ atand(y/x) ;
printf('\n\nPART B\n(r,theta)= %f,%f',r,theta) ;
printf('\nOR\n') ;
r=-sqrt(x^2+y^2) ;
theta= atand(y/x) ;
printf('(r,theta)= %f,%f',r,theta) ;
```

Scilab code Exa 6.17 To express an equation in polar coordinates

```
clc,clear
//Example 6.17
//to express an equation in polar coordinates

RHS=9 ;
//Note that LHS is basically an equation of circle
//But at any instant , it is numberically same as 9
LHS_numerically=RHS  ;
r=sqrt(LHS_numerically) ;

printf('The equation in terms of polar coordinates
    is : r =%.0f',r)
```

Scilab code Exa 6.19 To express an equation in polar coordinates

```
clc,clear
//Example 6.19
//to express an equation in polar coordinates

//Given equation is : y=x
y_by_x =1; //ratio of y and x
tan_theta = y_by_x;
```

```
theta=atand(tan_theta); //azimuth angle

printf('The given equation in polar coordinates is :
     theta = %.0f degree\n',theta)
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

```
//Example 8.1
//To plot the function of sin(x)
clear,clc;

x = linspace(-0,2*%pi,50);
y = sin(x) ;

//For grid, uncomment below line
//set(gca(),"grid",[5 5]);

printf('NOTE:\nTo enable the grid , check the code')
plot(x,y,'r');
xlabel("$0\le x\le 2*pi$","fontsize",4,"color","red"
    );
ylabel("$y(x)=sin(x)$","fontsize",4,"color","red");
title("Example 8.1","color","blue","fontsize",9);
legend("sin(x)");
```

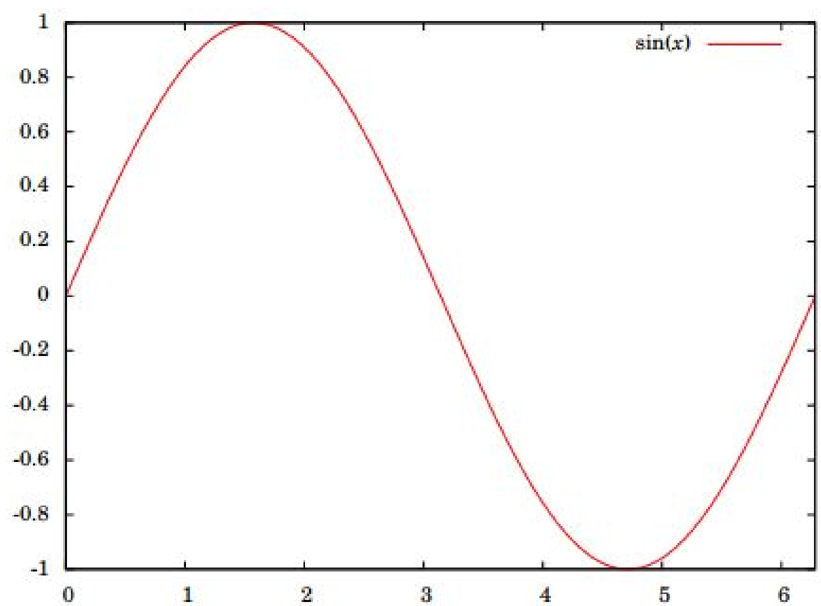

Figure 8.1: To plot the function of sin x

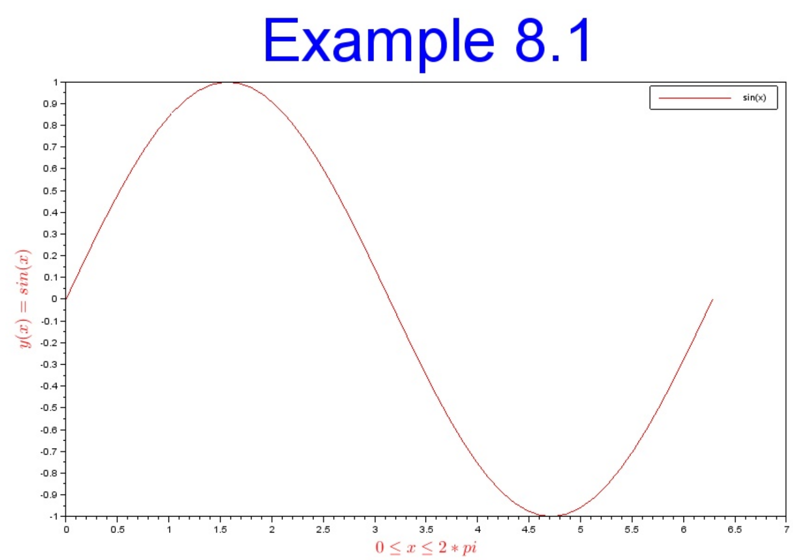

Figure 8.2: To plot the function of sin x