Scilab Textbook Companion for Sears And Zemansky'S University Physics With Modern Physics by Hugh D. Young, Roger A. Freedman[1]

Created by
Mohammad Usaid
BTech
Electrical Engineering
Aligarh Muslim University
Cross-Checked by
Scilab TBC Team

June 3, 2020

[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: Sears And Zemansky'S University Physics With Modern Physics

Author: Hugh D. Young, Roger A. Freedman

Publisher: Addison-Wesley, Usa

Edition: 13

Year: 2010

ISBN: 9780321696861

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

Chapter 2

Motion along a straight line

Scilab code Exa 2.1 `Average and Instantaneous velocities`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 2: MOTION ALONG A STRAIGHT A LINE
//EX 2.1: AVERAGE AND INSTANTANEOUS VELOCITIES
clear;
clc;
deff('[X]=x(t)','X=20+5*t^2'); //function of displacement
t1=1; //given time in sec
t2=2; //given time in sec
x1=x(t1); //displacement at t=1s
x2=x(t2); //displacement at t=2s
delta_x=x2-x1; //displacement in meters
mprintf('(a)Displacement between t1=1s and t2=2s: %d m',delta_x);
Vav_x=(x2-x1)/(t2-t1); //average velocity in m/s
mprintf('\n(b)Average speed between t1=1s and t2=2s: %d m/s',Vav_x);
x2=x(t1+0.001);
Vav_x=(x2-x1)/((t1+0.001)-t1); //instantaneous speed
```

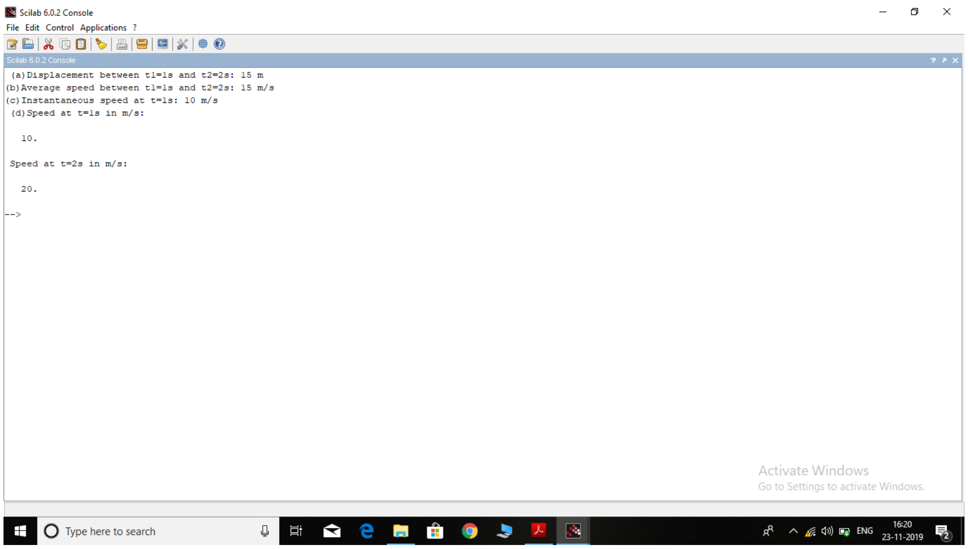

Figure 2.1: Average and Instantaneous velocities

```
    at t=1s
mprintf('\n(c)Instantaneous speed at t=1s: %d m/s',
    Vav_x); //answer vary due to roudoff error
disp(numderivative(x,1),'(d)Speed at t=1s in m/s: ')
    ; //Vx=dx/dt at t=1 sec
disp(numderivative(x,2),'Speed at t=2s in m/s: ');
    //Vx=dx/dt at t=2 sec
```

Scilab code Exa 2.3 Average and Instantaneous accelerations

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 2: MOTION ALONG A STRAIGHT A LINE
//EX 2.3 AVERAGE AND INSTANTANEOUS ACCELERATIONS
clear;
clc;
```

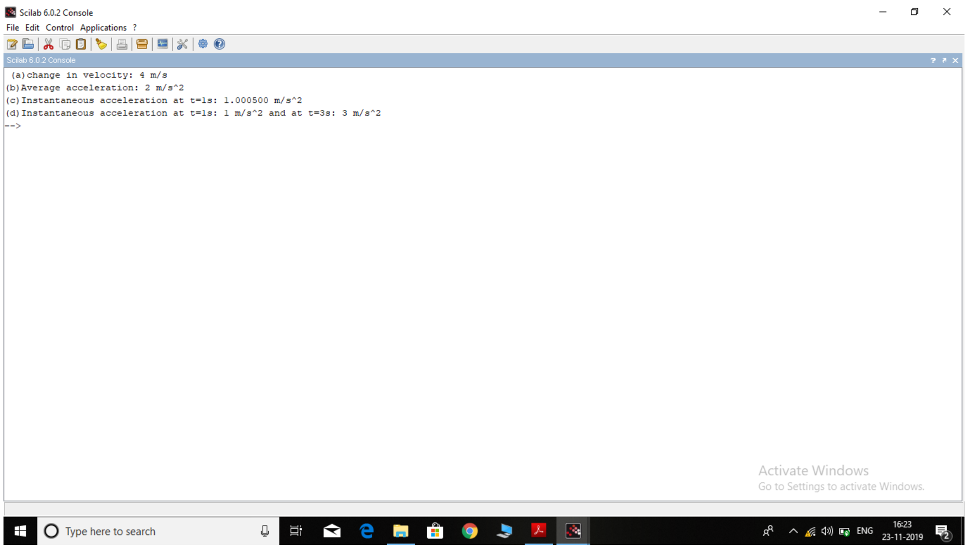

Figure 2.2: Average and Instantaneous accelerations

```
deff('[V_x]=v_x(t)','V_x=60+0.5*t^2'); //given function of velocity
t1=1; //given time instant in sec
t2=3; //given time instant in sec
v_1x=v_x(t1);
v_2x=v_x(t2);
delta_v_x=v_2x-v_1x;
mprintf('(a)change in velocity: %d m/s',delta_v_x);
a_av_x=(v_2x-v_1x)/(t2-t1); //average acceleration b/w t=1 and t=3 sec
mprintf('\n(b)Average acceleration: %d m/s^2',a_av_x);
t2=t1+0.001;
v_2x=v_x(t2); //velocity at t=2 sec
v_1x=v_x(t1); //velocity at t=1 sec
a_av_x=(v_2x-v_1x)/(t2-t1); //acceleration at t=1 sec
mprintf('\n(c)Instantaneous acceleration at t=1s: %f m/s^2',a_av_x);
a_x=numderivative(v_x,1); //instantaneous x
```

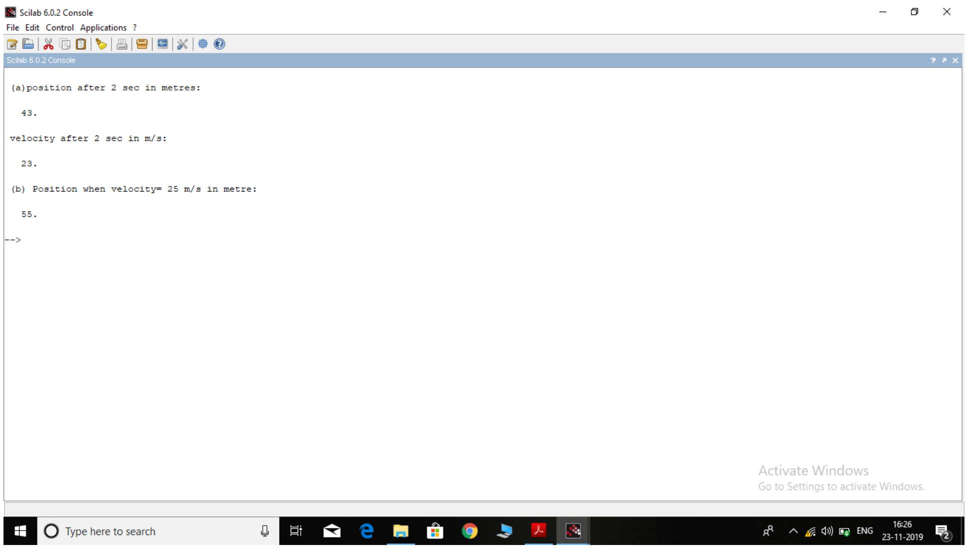

Figure 2.3: Constant acceleration calculations

```
acceleration at t=1 sec using a_x=dv_x/dt
mprintf('\n(d)Instantaneous acceleration at t=1s: %d m/s^2',a_x);
a_x=numderivative(v_x,3); //instantaneous x acceleartion at t=3 sec using a_x=dv_x/dt
mprintf('\n(d)Instantaneous acceleration at t=1s: %d m/s^2',a_x);
```

Scilab code Exa 2.4 `Constant acceleration calculations`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 2: MOTION ALONG A STRAIGHT LINE
//EX 2.4: CONSTANT ACCELERATION CALCULATIONS
clc;
clear;
a_x=4; //acceleration in ms^-2
```

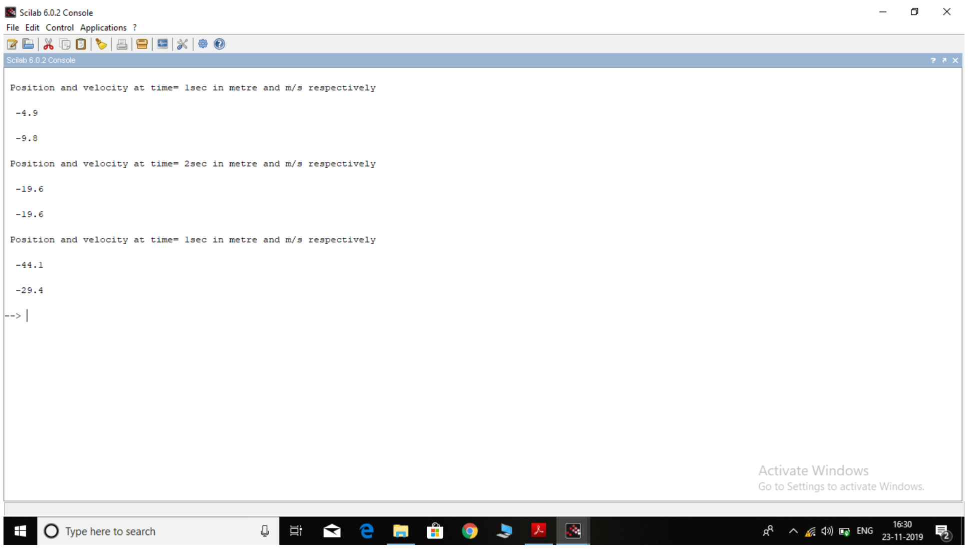

Figure 2.4: A freely falling coin

```
x_0=5; //initial displacement in m
v_0x=15; //initial velocity in m/s
t=2; //time in sec
x= x_0+v_0x*t+0.5*a_x*t^2; //displacement after 2 sec
v_x=v_0x+a_x*t; //velocity after 2 sec
disp(x,'(a)position after 2 sec in metres: ');
disp(v_x,'velocity after 2 sec in m/s: ');
v_x=25; //given velocity in ms^-1
x=x_0+(v_x^2-v_0x^2)/(2*a_x); //position when velocity=25 m/s
disp(x,'(b) Position when velocity= 25 m/s in metre: ');
```

Scilab code Exa 2.6 A freely falling coin

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 2: MOTION ALONG A STRAIGHT LINE
//EX 2.6: A FREELY FALLING COIN
clc;
clear;
g=9.8; //acceleration due to gravity in ms^-2
y_0=0; //given initial position in m
t_0=0; //given time instant in sec
t1=1; //given time instant in sec
t2=2; //given time instant in sec
t3=3; //given time instant in sec
v_0y=0; //given initial velocity in m/s
a_y=-g; //given initial acceleration in m/s^2
deff('[Y]=y(t)','Y=y_0+v_0y*t+0.5*a_y*t^2'); //
    kinematic equation for displacement
deff('[V_y]=v_y(t)','V_y=v_0y+a_y*t'); //kinematic
    equation for velocity
y_1=y(t1); //position at t=1 sec
v_1y=v_y(t1); //velocity at t=1 sec
disp(y_1,v_1y,'Position and velocity at time= 1sec
    in metre and m/s respectively');
y_2=y(t2); //position at t=2 sec
v_2y=v_y(t2); //velocity at t=2 sec
disp(y_2,v_2y,'Position and velocity at time= 2sec
    in metre and m/s respectively'); //answer given
    in textbook is wrong
y_3=y(t3); //position at t=3 sec
v_3y=v_y(t3); //velocity at t=3 sec
disp(y_3,v_3y,'Position and velocity at time= 3sec
    in metre and m/s respectively'); //answer given
    in textbook is wrong
```

Scilab code Exa 2.7 `Up and down motion in free fall`

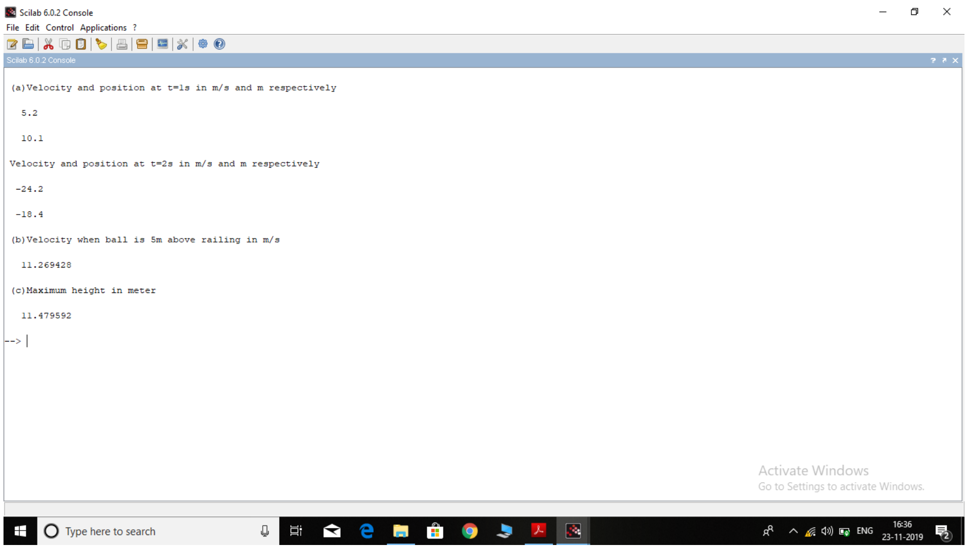

Figure 2.5: Up and down motion in free fall

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 2:MOTION ALONG A STRAIGHT LINE
//EX 2.7: UP AND DOWN MOTION IN FREE FALL
clc;
clear;
g=9.8; //acceleration due to gravity in ms^-2
y_0=0; //initial position in m
v_0y=15; //initial velocity in ms^-1
a_y=-g; //acceleration in y direction in m/s^2
deff('[Y]=y(t)','Y=v_0y*t+0.5*a_y*t^2'); //kinematic
    equation of motion
deff('[V_y]=v_y(t)','V_y=v_0y+a_y*t');  //kinematic
    equation of motion
t1=1; //given time instant in sec
t2=4; //given time instant in sec
disp(y(t1),v_y(t1),'(a)Velocity and position at t=1s
    in m/s and m respectively');
disp(y(t2),v_y(t2),'Velocity and position at t=4s in
    m/s and m respectively');
y=5; //when ball is 5m above origin
```

```
v_y=sqrt(v_0y^2+2*a_y*y);
disp(v_y,'(b)Velocity when ball is 5m above railing in m/s');//answer vary due to roundoff error
y1=(v_0y^2)/(2*g); //position when v_y=0 ( y1-y_0=(v_y^2-v_0^2)/(2*a_y) )
disp(y1,'(c)Maximum height in meter'); //answer vary due to roundoff error
```

Chapter 3

Motion in two or three dimensions

Scilab code Exa 3.6 `A body projected horizontally`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 3: MOTION IN 2 OR 3 DIMENTIONS
//EX 3.6: A BODY PROJECTED HORIZONTALLY
clc;
clear;
v_0x=9; //initial horizontal speed of motorcycle in m/s
v_0y=0; //initial vertical speed of motorcycle in m/s
t=0.5; //given time in sec
g=9.8; //acceleration due to gravity in m/s^2
x=v_0x*t; //horizontal displacement in m
y=v_0y*t-0.5*g*t^2; //vertical displacement in m
r=sqrt(x^2+y^2); //distance of motorcycle from starting point in m
v_x=v_0x; //final horizontal speed of motorcycle at t=0.5s in m/s
```

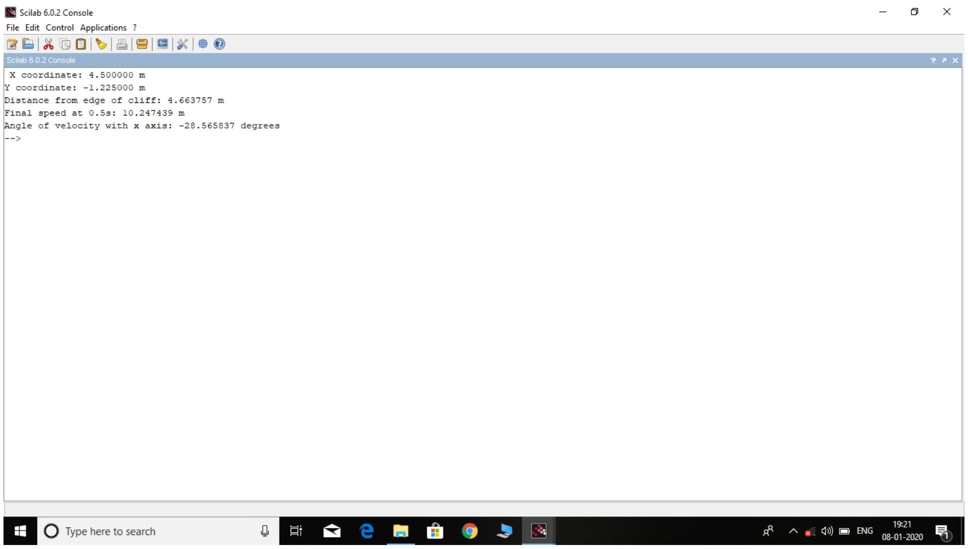

Figure 3.1: A body projected horizontally

```
v_y=-g*t; //final vertical speed of motorcycle at t
    =0.5s in m/s
v=sqrt(v_x^2+v_y^2); //final speed of motorcycle at
    t=0.5s in m/s
alpha=atand(v_y/v_x); //angle of final velocity of
    motorcycle with horizontal axis in degrees
mprintf('X coordinate of motorcycle: %f m',x);
mprintf('\nY coordinate of motorcycle : %f m',y); //
    answer vary due to roundoff error
mprintf('\nDistance of motorcycle from edge of cliff
    : %f m',r); //answer vary due to roundoff error
mprintf('\nFinal speed of motorcycle at 0.5s: %f m/s
    ',v); //answer vary due to roundoff error
mprintf('\nAngle of velocity of motorcycle with x
    axis: %f degrees',alpha); //answer vary due to
    roundoff error
```

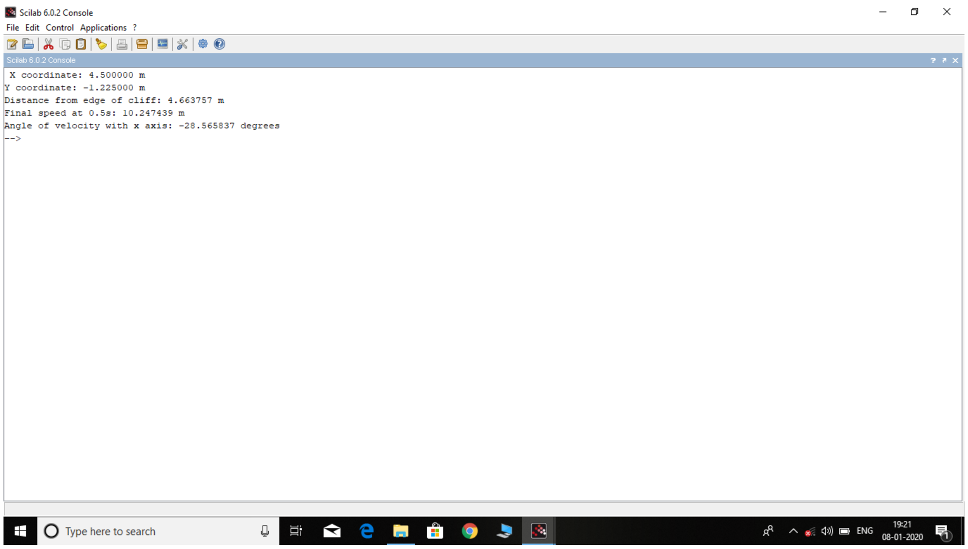

Figure 3.2: Height and range of a projectile 1 A batted baseball

Scilab code Exa 3.7 `Height and range of a projectile 1 A batted baseball`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 3: MOTION IN 1 OR 2 DIMENTIONS
//Ex 3.7: HEIGHT AND RANGE OF A PROJECTILE 1: A
    BATTLED BASEBALL
clear;
clc;
v_0=37; //initial speed of projectile in m/s
alpha_0=53.1; //angle of projection in degrees
g=9.8; //acceleration due to gravity in m/s^2
t=2; //time interval in sec
v_0x=v_0*cosd(alpha_0); //initial horizontal speed
    in m/s
v_0y=v_0*sind(alpha_0); //initial vertical speed in
    m/s
x=v_0x*t; //x component of position in m
```

```
y=v_0y*t-0.5*g*t^2; //y component of position in m
v_x=v_0x; //horizontal velocity after 2s in m/s
v_y=v_0y-g*t; //vertical velocity after 2s in m/s
v=sqrt(v_x^2+v_y^2); //total speed after 2s in m/s
alpha=atand(v_y/v_x);
mprintf('(a)X coordinate after 2sec: %f m',x);
mprintf('\nY coordinate after 2sec: %f m',y);
mprintf('\nTotal speed after 2sec: %f m\nAngle of velocity with x axis in degres: %f',v,alpha);
t1=v_0y/g; //time at v_y=0 in sec
h=v_0y*t1-0.5*g*t1^2; //Maximum height in m
mprintf('\n(b)Time taken to reach maximum height: %f sec',t1);
mprintf('\nMaximum height: %f m',h);
t=poly(0,'t') //polynomial for y
y=(v_0y*t)-0.5*g*t^2; //y=v_0y*t2-(1/2)*g*t^2
t2=roots(y); //solving for two values of t2 when y=0
if(t2(1)~=0) then //considering only non zero value of t2, since t2=0 implies that horizontal range R=0 which is trivial
R=v_0x*t2(1); //horizontal range of projectile in m
else R=v_0x*t2(2);
end
mprintf('\n(c) Horizontal range of projectile: %f m',R); //answer vary due to round off error
```

Scilab code Exa 3.11 Centripetal acceleration on a curved road

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 3: MOTION IN 1 OR 2 DIMENTIONS
//Ex 3.11: CENTRIPETAL ACCELERATION ON CURVED ROAD
clear;
clc;
```

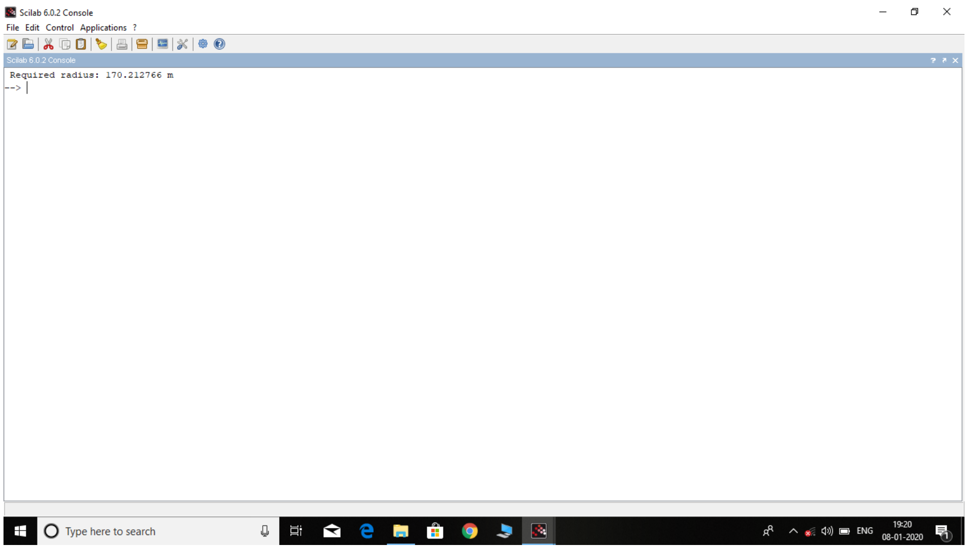

Figure 3.3: Centripetal acceleration on a curved road

```
v=40; //linear velocity in m/s
a_rad=9.4; //centripetal acceleration in m/s^2
R=(v^2)/a_rad; //Radius of acceleration in m/s^2
mprintf('Required radius: %f m',R); //answer vary
   due to roundoff error
```

Scilab code Exa 3.12 `Centripetal acceleration on a carnival ride`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 3: MOTION IN 1 OR 2 DIMENTIONS
//Ex 3.12: CENTRIPETAL ACCELERATION ON A CARNIVAL
   RIDE
clear;
clc;
R=5; //radius in m
T=4; //time taken in 1 revolution
```

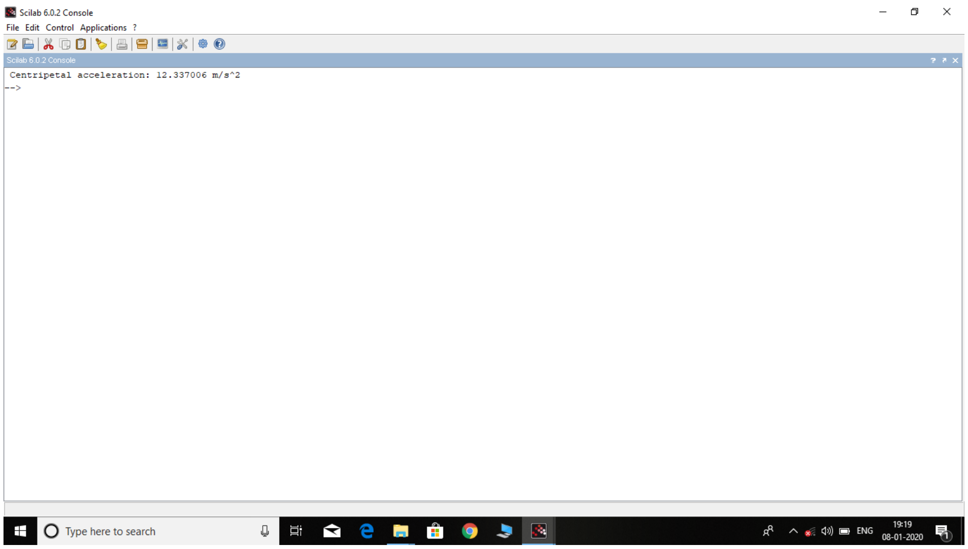

Figure 3.4: Centripetal acceleration on a carnival ride

```
v=(2*%pi*R)/T; //speed in m/s
a_rad=(v^2)/R; //centripetal accleleration in m/s^2
mprintf('Centripetal acceleration: %f m/s^2',a_rad);
    //answer vary due to roundoff error
```

Scilab code Exa 3.14 `Flying in a crosswind`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 3: MOTION IN 1 OR 2 DIMENTIONS
//Ex 3.14: FLYING IN A CROSSWIND
clear;
clc;
v_pa=240; //speed of plane wrt to air in km/h
v_ae=100; //speed of air wrt to Earth in km/h
v_pe=sqrt((v_pa)^2+(v_ae)^2); //speed of plane wrt
    to Earth
```

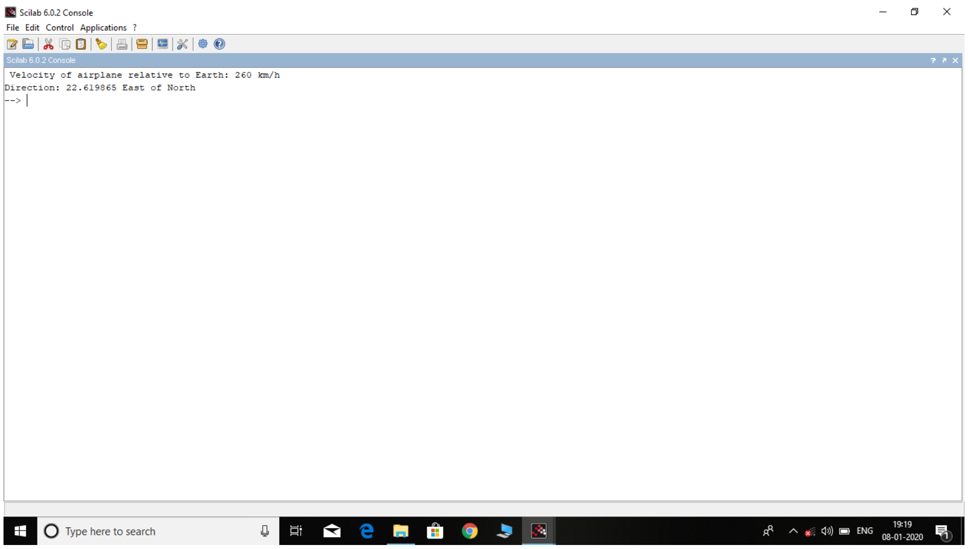

Figure 3.5: Flying in a crosswind

```
alpha=atand(v_ae/v_pa); //angle with vertical axis(North axis) in degrees
mprintf('Velocity of airplane relative to Earth: %d km/h',v_pe);
mprintf('\nDirection: %f degrees East of North',alpha); //answer vary due to roundoff error
```

Chapter 4

Newtons Laws Of Motion

Scilab code Exa 4.1 `Superposition of forces`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 4: NEWTON'S LAWS OF MOTION
//EXAMPLE 4.1: SUPERPOSITION OF FORCES
clc;
clear;
F1=250; //magnitude of force F1 in Newton
F2=50; //magnitude of force F2 in Newton
F3=120; //magnitude of force F3 in Newton
theta_1=127; //angle of force F1 with +ve x axix in degrees
theta_2=0; //angle of force F2 with +ve x axix in degrees
theta_3=270; //angle of force F3 with +ve x axix in degrees
F1x=F1*cosd(theta_1); //x component of force F1 in newton
F1y=F1*sind(theta_1); //y component of force F1 in newton
F2x=F2*cosd(theta_2); //x component of force F2 in
```

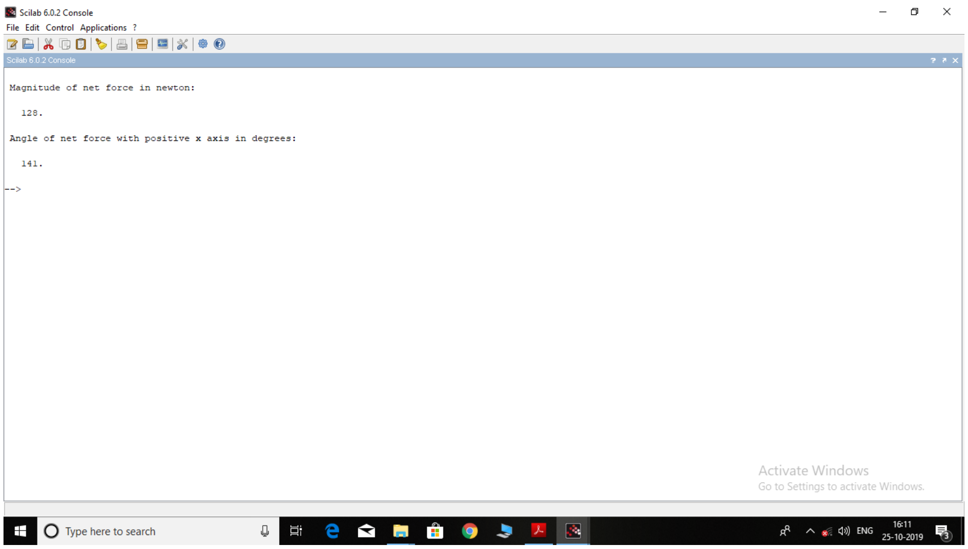

Figure 4.1: Superposition of forces

```
    newton
F2y=F2*sind(theta_2); //y component of force F2 in
    newton
F3x=F3*cosd(theta_3); //x component of force F3 in
    newton
F3y=F3*sind(theta_3); //y component of force F3 in
    newton
Rx=F1x+F2x+F3x; //x component of net force in newton
Ry=F1y+F2y+F3y; //y component of net force in newton
R=sqrt(Rx^2+Ry^2);
theta=atand(Ry/Rx);
disp(floor(R),'Magnitude of net force in newton: ');
disp(180+floor(theta),'Angle of net force with
    positive x axis in degrees: ');
```

Scilab code Exa 4.4 `Determining acceleration from force`

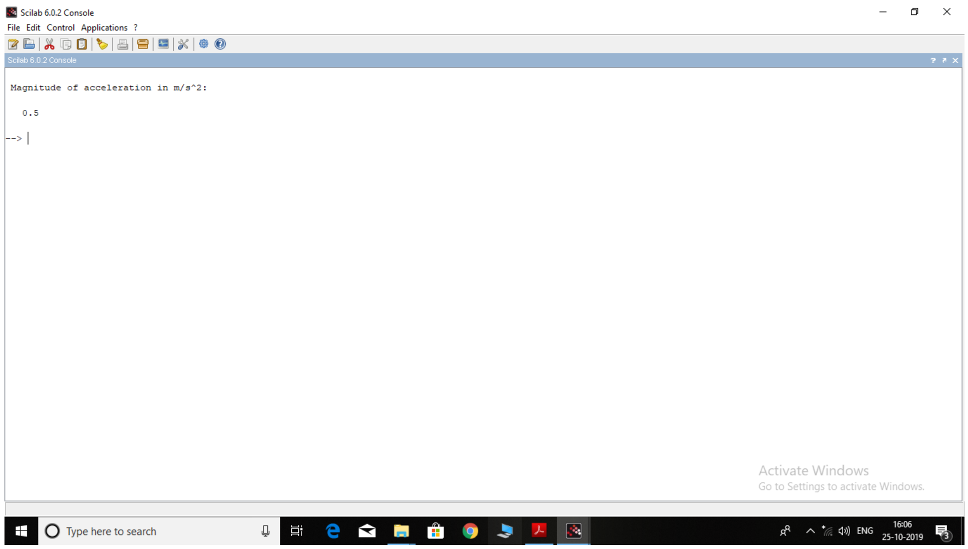

Figure 4.2: Determining acceleration from force

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 4: NEWTONS LAWS OF MOTION
//EXAMPLE 4.4: //DETERMINING ACCELERATION FROM FORCE
clc;
clear;
F_x=20; //magnitude of force in newton
m=40; //mass in kg
a_x=F_x/m; //magnitude acceleration im m/s^2
disp(a_x,'Magnitude of acceleration in m/s^2: ');
```

Scilab code Exa 4.5 Determining force from acceleration

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 4: NEWTON'S LAWS OF MOTION
//EXAMPLE 4.5: DETERMINING FORCE FROM ACCELERATION
clear;
```

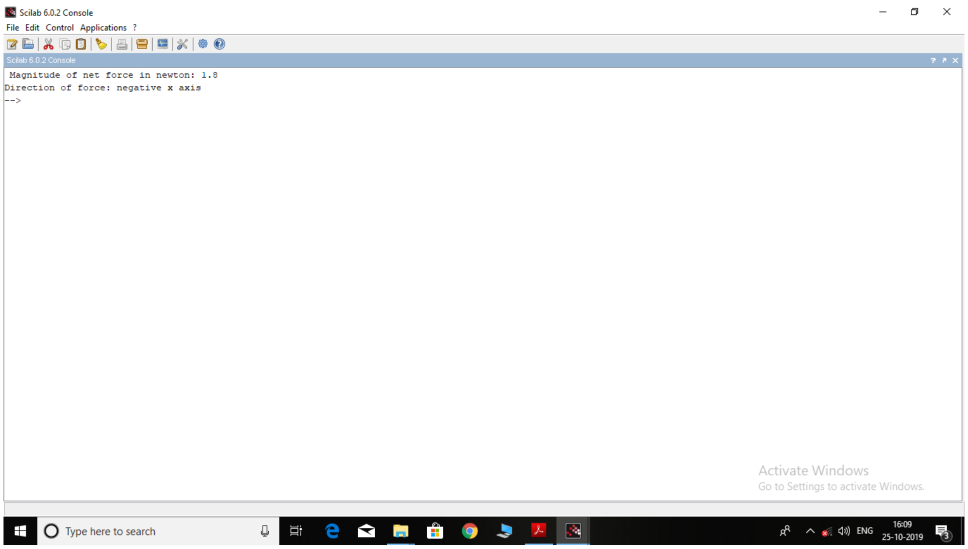

Figure 4.3: Determining force from acceleration

```
clc;
m=0.45 //mass of bottle in kg
v_0x=2.8 //initial velocity of bottle in m/s
x_0=0; //initial position of bottle in m
v_x=0; //final velocity of bottle in m/s
x=1; //displacement of bottle in metres
a_x=(v_x^2-v_0x^2)/(2*(x-x_0)); //kinematic equation for acceleration
f_x=m*a_x; //x component of force in newton(-ve sign because bottle's acceleration is in -ve x dir.)
mprintf('Magnitude of friction force in newton: %0.1f',abs(f_x));
if f_x>0 then mprintf('\nDirection of force: positive x axis');
else mprintf('\nDirection of force: negative x axis');
end
```

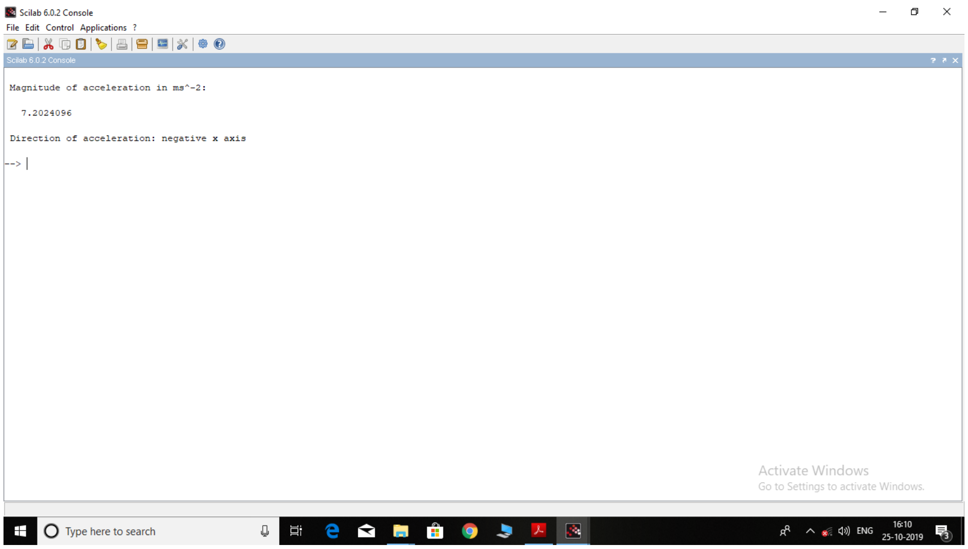

Figure 4.4: Mass and weight

Scilab code Exa 4.7 Mass and weight

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 4: NEWTON'S LAWS OF MOTION
//EXAMPLE 4.7 MASS AND WEIGHT
clear;
clc;
w=2.49*10^4; //weight of car in newton
g=9.8; //acceleration due to gravity in m/s^2;
m=(w/g); // mass of car in kg
F_x=-1.83*10^4; //given retarding force in newton
a_x=(F_x/m); //acceleration of car in m/s^2
disp(abs(a_x),'Magnitude of acceleration in m/s^2: '
    ); //answer vary due to roundoff
if a_x>0 then disp('Direction of acceleration:
    positive x axis');
    else disp('Direction of acceleration: negative x
```

```
            axis');
end
```

Chapter 7

Potential Energy And Energy Conservation

Scilab code Exa 7.1 Height of a baseball from energy conservation

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 7 POTENTIAL ENERGY AND ENERGY CONSERVATION
//EX 7.1: HEIGHT OF BASEBALL FROM ENERGY
    CONSERVATION
clear;
clc;
m=0.145; //mass in kg
v_1=20; //initial velocity in m/sec
v_2=0; //final velocity in m/sec
g=9.8; //acceleration due to gravity in m/s^2
y1=0; //initial position in m
U_grav1=m*g*y1; //initial kinetic energy in J
K1=(1/2)*m*v_1^2; //initial kinetic energy in J
K2=(1/2)*m*v_2^2; //final kinetic energy in J
y2=(K1)/(m*g); //height in m, since(K1 + U_grav1 =
    K2+ U_grav2)
disp(y2,'Height of baseball in meters: ');
```

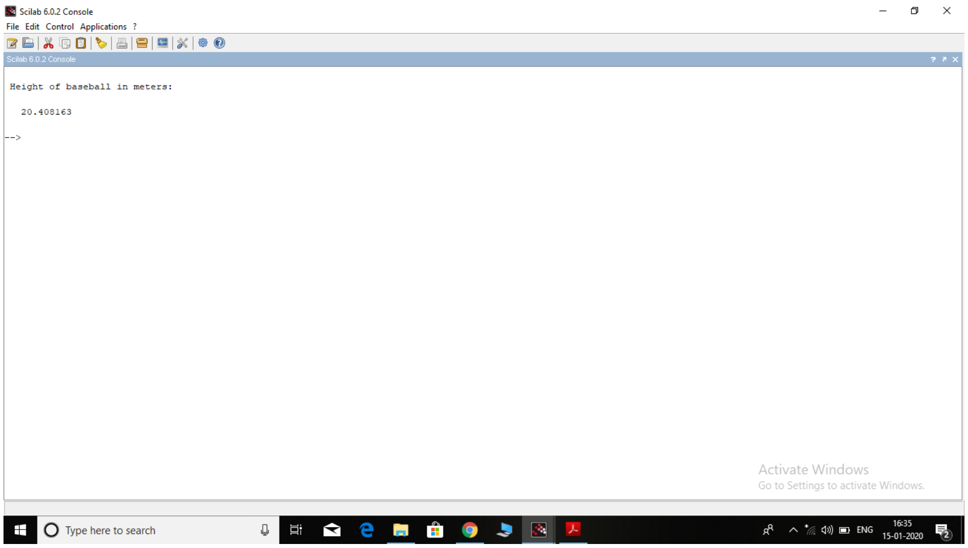

Figure 7.1: Height of a baseball from energy conservation

Scilab code Exa 7.4 Speed at the bottom of a vertical circle

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 7 POTENTIAL ENERGY AND ENERGY CONSERVATION
//EX 7.4: SPEED AT THE BOTTOM OF A VERTICAL CIRCLE
clear;
clc;
m=25; //total mass of throcky and skateboard in kg
R=3; //radius of circle in meters
g=9.8; //acceleration due to gravity in m/s^2
w=m*g //total weight of throcky and skateboard in N
K1=0; //initial kinetic energy in J
U_grav1=m*g*R; //initial gravitational potential
    energy in J
```

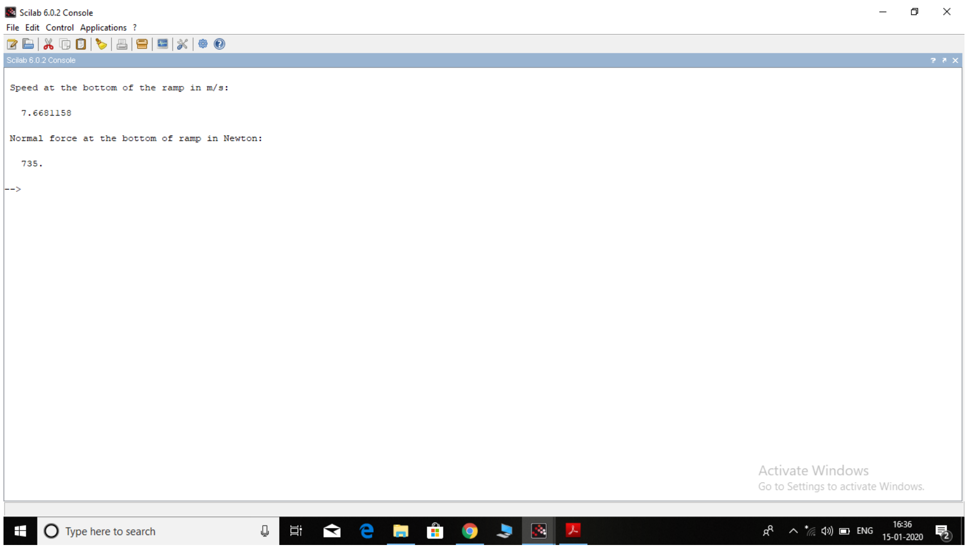

Figure 7.2: Speed at the bottom of a vertical circle

```
U_grav2=0; //final gravitational potential energy in J
v2=sqrt(2*g*R); //from energy conservation law(K1 + U_grav1 = K2 + U_grav2)
a_rad=(v2^2)/R; //centripetal acceleration
n=m*a_rad+w; //normal reaction
disp(v2,'Speed at the bottom of the ramp in m/s: ');
disp(n,'Normal force at the bottom of ramp in Newton : ');
```

Scilab code Exa 7.5 A vertical circle with friction

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 7 POTENTIAL ENERGY AND ENERGY CONSERVATION
//EX 7.5: A VERTICAL CIRCLE WITH FRICTION
clear;
```

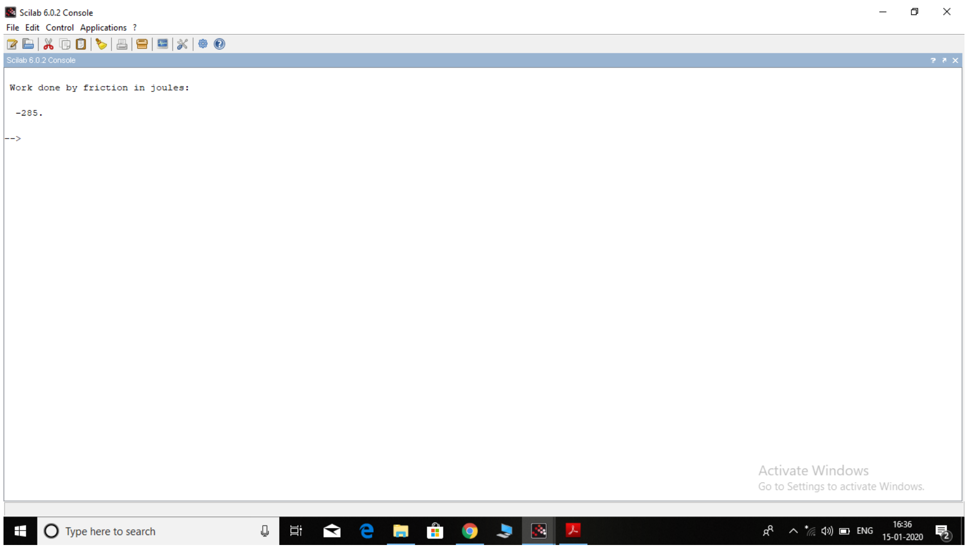

Figure 7.3: A vertical circle with friction

```
clc;
m=25; //mass in kg
v1=0; //initial velocity in m/s^2
v2=6; //final velocity in m/s^2
R=3; //radius in meters
g=9.8; //acceleration due to gravity in m/s^2
K1=(1/2)*m*v1^2; //initial kinetic energy in J
U_grav1=m*g*R; //initial gravitational potential energy in J
K2=(1/2)*m*v2^2; //final kinetic energy in J
U_grav2=0; //final gravitational potential energy in J
W_other=(K2+U_grav2)-(K1+U_grav1); //work done by friction in J
disp(W_other,'Work done by friction in joules: ');
```

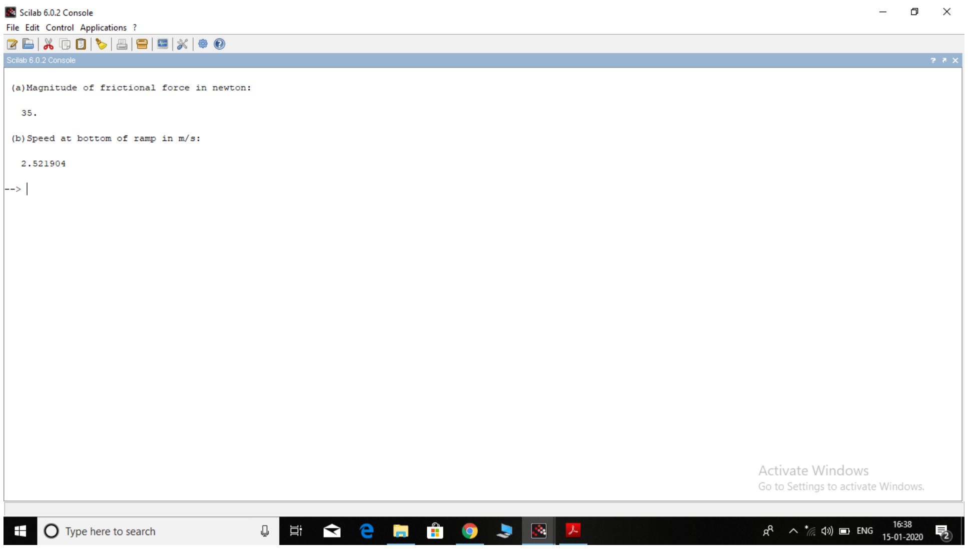

Figure 7.4: An inclined plane with friction

Scilab code Exa 7.6 `An inclined plane with friction`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 7 POTENTIAL ENERGY AND ENERGY CONSERVATION
//EX 7.6: AN INCLINED PLANE WITH FRICTION
clear;
clc;
m=12; //mass in kg
g=9.8; //acceleration due to gravity in ms^-2
v1=5; //initial speed in ms^-1
theta=30; //angle of inclination in degrees
s=1.6; //slant length in meter
y1=0;
y2=s*sind(theta); //vertical height in meter
y3=0;
K1=0.5*m*(v1^2); //initial kinetic energy at bottom
    in J
U_grav1=0; //initial gravitational potential energy
    at bottom in J
U_grav2=m*g*y2; //gravitational potential energy at
```

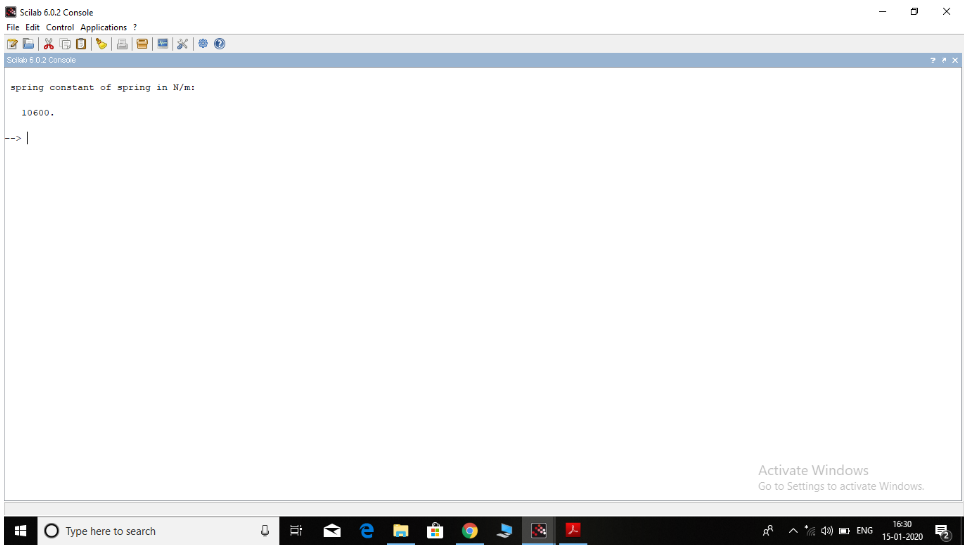

Figure 7.5: Motion with gravitational elastic and frictional forces

```
top in J
K2=0; ////final kinetic energy at top in J
W_other=(K2+U_grav2)-(K1+U_grav1); //work done by friction in J ()
f=W_other/s; //frictional force in N
W_other=(2*W_other); //work done by frictional force in part(b) in J
U_grav3=0;
K3=K1+U_grav1-U_grav3+W_other; //kinetic energy at bottom (K1+U_grav1+W_other=K3+U_grav3)
v3=sqrt((2*K3)/m); //speed at bottom of ramp in ms^-1
disp(abs(round(f)),'(a)Magnitude of frictional force in newton: ');
disp(v3,'(b)Speed at bottom of ramp in m/s: '); //answer vary due to roundoff error
```

Scilab code Exa 7.9 Motion with gravitational elastic and frictional forces

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 7: Potential Energy and Energy
    Conservation
//EXAMPLE: 7.9: Motion with gravitational, elastic,
    and friction forces
clear;
clc;
m=2000; //mass in kg
g=9.8; //accleration due to gravity in m/s^2
v1=4; //initial velocity in ms^-1
K1=0.5*m*v1^2; //initial kinetic energy in J
K2=0; //final kinetic energy in J
y1=0; //initial position in m
Ugrav=0; //initial gravitational potential energy
Uel=0; //initial elastic potential energy in J
U1=Ugrav+Uel; //initial total energy in J
y2=-2; //final position in m
U2=m*g*y2; //final gravitational potential energy in
    J
f=17000; //frictional force in N
W_other=f*y2; //work done by friction in J
k=2*(K1+W_other-m*g*y2)/y2^2; //spring constant in N
    /m (K1+U1+W_other=K2+U2)
disp(k,'spring constant of spring in N/m: ');
```

Chapter 8

Momentum Impulse and Collisions

Scilab code Exa 8.2 A ball hits a wall

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.2: A BALL HITS A WALL
clc;
clear;
m=0.4; //mass in kg
v1x=-30; //initial velocity in -x direction
v2x=20; //final velocity in +x direction
p1x=m*v1x; //initial momentum in -x direction
p2x=m*v2x; //final momentum in -x direction
Jx=p2x-p1x; //impulse in units N.s
delta_t=0.010; //time interval in sec
F_avx=Jx/delta_t; //Average impulsive force in N
mprintf('(a)Magnitude of impulse: %d N.s',Jx);
if Jx>0 then
    mprintf('\nDirection: +x axis')
    else mprintf('\nDirection: -x axis')
```

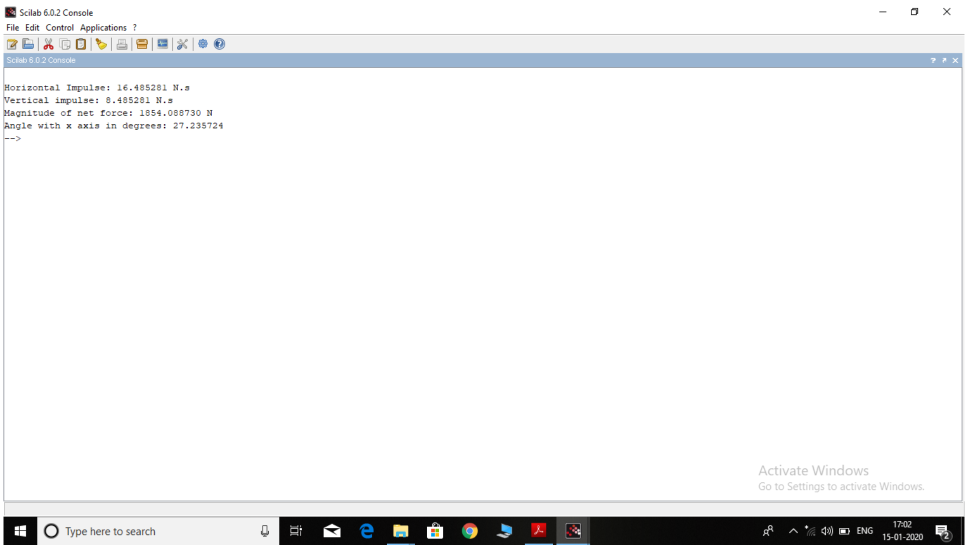

Figure 8.1: A ball hits a wall

```
end
mprintf('\n(b)Magnitude of average impulsive force:
    %d N',F_avx)
if F_avx>0 then
    mprintf('\nDirection: +x axis')
    else mprintf('\nDirection: -x axis')
end
```

Scilab code Exa 8.3 Kicking a soccer ball

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.3: KICKING A SOCCER BALL
clc;
clear;
m=0.4; //mass in kg
```

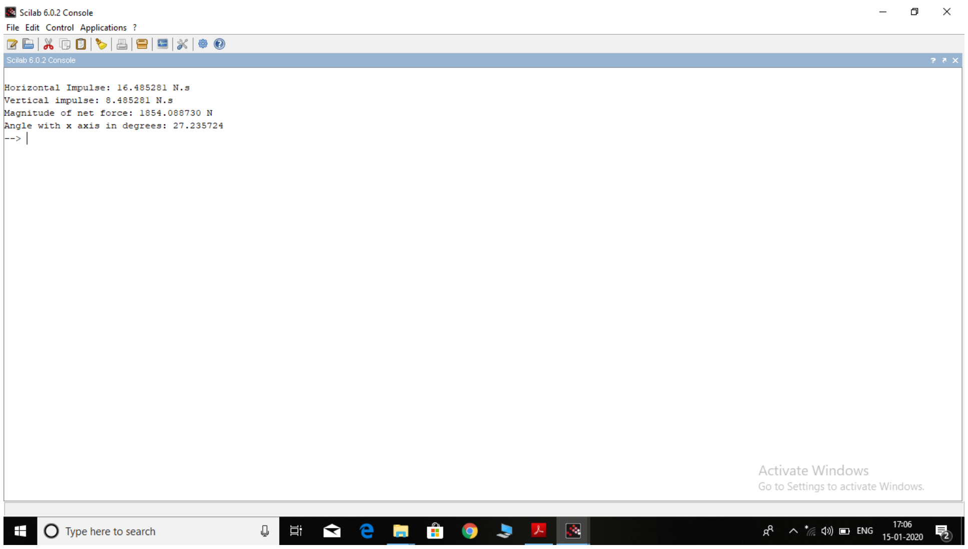

Figure 8.2: Kicking a soccer ball

```
v1x=-20; //initial horizontal velocity in -x dir in m./s
v1y=0; //intial vertical velocity
v2=30; //final speed in m/s
alpha=45; //angle between v2 and +ve x axis
v2x=v2*cosd(alpha),v2y=v2*cosd(alpha); //x and y component of final velocity in m/s
Jx=m*(v2x-v1x); //Horizontal impulse
Jy=m*(v2y-v1y); //Vertical impulse
delta_t=0.010; //time interval in sec
Fav_x=Jx/delta_t; //horizontal force in N
Fav_y=Jy/delta_t; //vertical force in N
Fav=sqrt(Fav_x^2+Fav_y^2); //average force in N
theta=atand(Fav_y/Fav_x); //angle in degrees
mprintf('\nHorizontal Impulse: %f N.s',Jx); //answer vary due to roundoff error
mprintf('\nVertical impulse: %f N.s',Jy); //answer vary due to roundoff error
mprintf('\nMagnitude of net force: %f N',Fav); //answer provided in textbbok is wrong
```

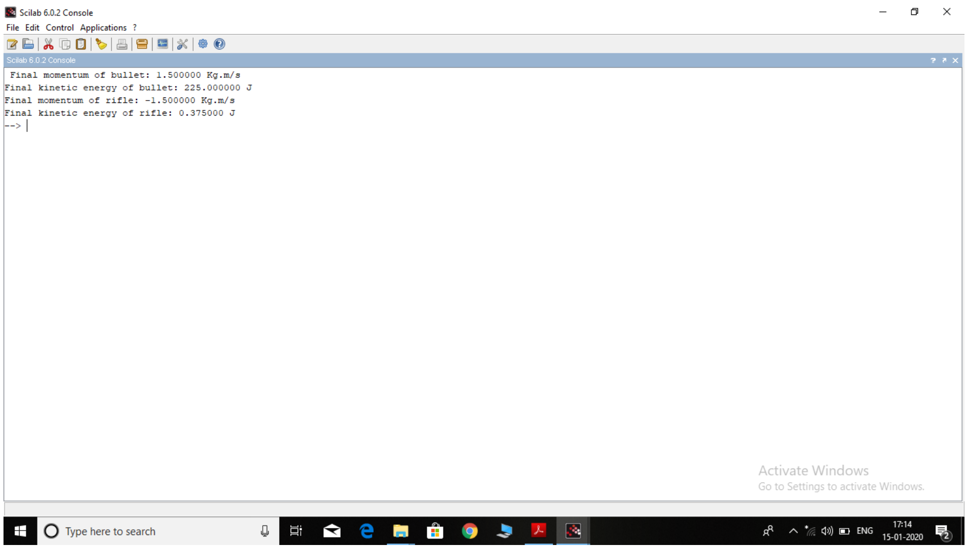

Figure 8.3: Recoil of a rifle

```
mprintf('\nAngle with x axis in degrees: %f',theta);
    //answer provided in textbbok is wrong
```

Scilab code Exa 8.4 Recoil of a rifle

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.4: RECOIL OF A RIFLE
clc;
clear;
m_R=3; //mass of rifle inkg
m_B=0.005; //mass of bullet in kg
v_Bx=300; //velocity of bullet in m/s
v_Rx=-(m_B/m_R)*v_Bx; //recoil velocity of rifle in
   m/s
p_Bx=m_B*v_Bx; //momentum of bullet in Kg.m/s
```

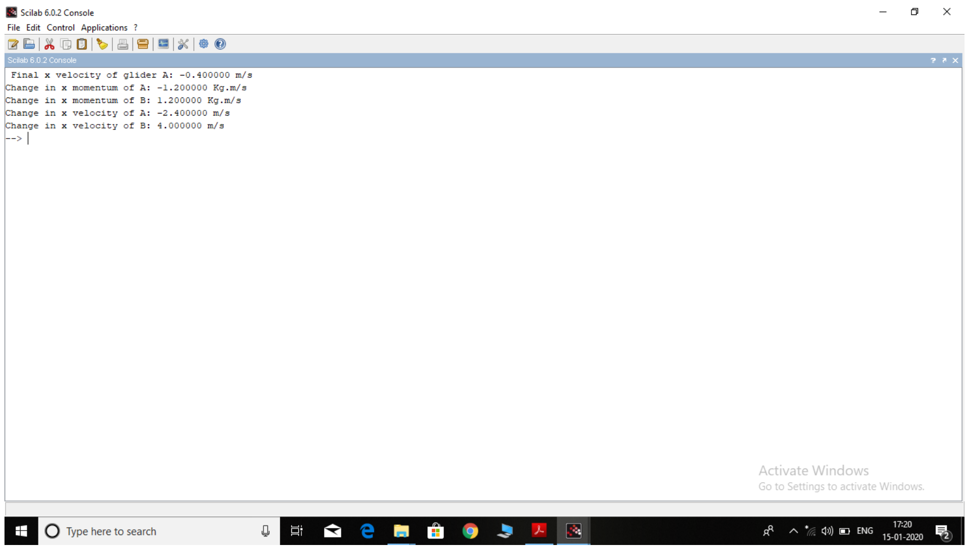

Figure 8.4: Collision along a straight line

```
K_B=0.5*m_B*v_Bx^2; //kinetic energy of bulley in J
p_Rx=m_R*v_Rx; //momentum of rifle in Kg.m/s
K_R=0.5*m_R*v_Rx^2; //kinetic energy of rifle in J
mprintf('Final momentum of bullet: %f Kg.m/s',p_Bx);
mprintf('\nFinal kinetic energy of bullet: %f J',K_B
    );
mprintf('\nFinal momentum of rifle: %f Kg.m/s',p_Rx)
    ;
mprintf('\nFinal kinetic energy of rifle: %f J',K_R)
    ;
```

Scilab code Exa 8.5 `Collision along a straight line`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.5: COLLISION ALONG A STRAIGHT LINE
```

```
clc;
clear;
m_A=0.5; //mass of A in kg
m_B=0.3; //mass of B in kg
v_A1x=2; //initial velocity of A in m/s
v_B1x=-2; //initial velocity of B in m/s
v_B2x=2; //final velocity of B in m/s
Px=(m_A*v_A1x)+(m_B*v_B1x); //initial total momentum in Kg.m/s
v_A2x=(Px-m_B*v_B2x)/m_A; //final velocity of A in m/s
mprintf('Final x velocity of glider A: %f m/s',v_A2x);
mprintf('\nChange in x velocity of A: %f m/s',(v_A2x-v_A1x));
mprintf('\nChange in x velocity of B: %f m/s',(v_B2x-v_B1x));
mprintf('\nChange in x momentum of A: %f Kg.m/s',m_A*(v_A2x-v_A1x));
mprintf('\nChange in x momentum of B: %f Kg.m/s',m_B*(v_B2x-v_B1x));
```

Scilab code Exa 8.6 Collision in a horizontal plane

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.6: COLLISION IN A HORIZONTAL PLANE
clc;
clear;
m_A=20; //mass of A in kg
m_B=12; //mass of B in kg
alpha=30; //angle between v_A1 and v_A2 in degrees
v_A1x=2; //velocity of A before collision in x dir
```

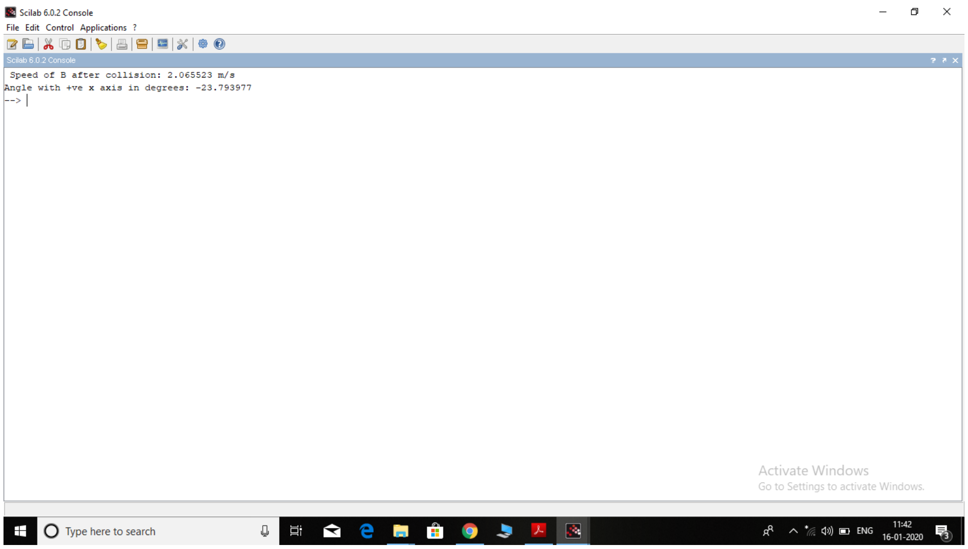

Figure 8.5: Collision in a horizontal plane

```
    in m/s
v_A1y=0; //velocity of A before collision in y dir
    in m/s
v_B1x=0; //velocity of B before collision in x dir
    in m/s
v_B1y=0; //velocity of B before collision in y dir
    in m/s
v_A2=1; //speed of A after collision in m/s
v_A2x=v_A2*cosd(alpha); //velocity of A after
    collision in x dir in m/s
v_A2y=v_A2*sind(alpha); //velocity of A after
    collision in y dir in m/s
v_B2x=(m_A*v_A1x+m_B*v_B1x-m_A*v_A2x)/m_B; //
    velocity of B after collision in x dir in m/s
v_B2y=(m_A*v_A1y+m_B*v_B1y-m_A*v_A2y)/m_B; //
    velocity of B after collision in y dir in m/s
v_B2=sqrt(v_B2x^2+v_B2y^2); //speed of B after
    collision in m/s
Beta=atand(v_B2y/v_B2x); //angle between v_B1 and
    v_B2 in degrees
```

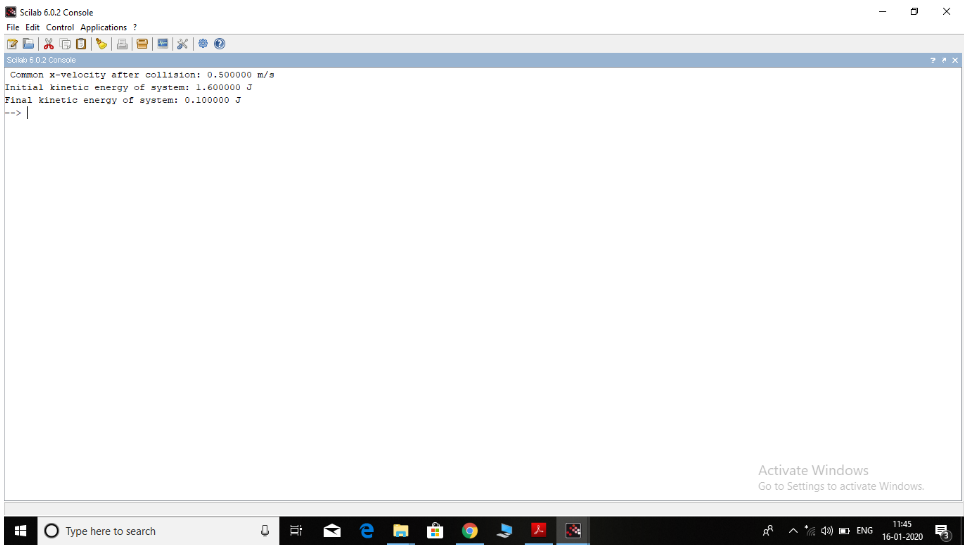

Figure 8.6: A completely inelastic collision

```
mprintf('Speed of B after collision: %f m/s',v_B2);
    //answer given in textbook is wrong
mprintf('\nAngle with +ve x axis in degrees: %f',
    Beta); //answer given in textbook is wrong
```

Scilab code Exa 8.7 A completely inelastic collision

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.7: A COMPLETELY INELASTIC COLLISION
clc;
clear;
m_A=0.5; //mass of A in kg
m_B=0.3; //mass of B in kg
v_A1x=2; //x-velocity of A before collision in m/s
```

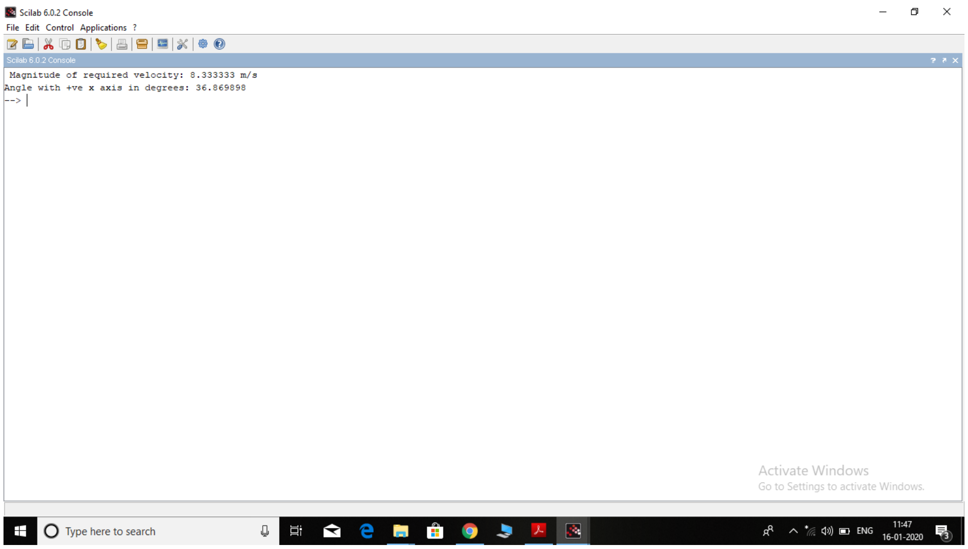

Figure 8.7: An automobile collision

```
K_A=0.5*m_A*v_A1x^2; //kinetic energy of A before
    collision in J
v_B1x=-2; //x-velocity of B before collision in m/s
K_B=0.5*m_B*v_B1x^2; //kinetic energy of B before
    collision in J
K1=K_A+K_B; //total kinetic energy before collision
    in J
v_2x=(m_A*v_A1x+m_B*v_B1x)/(m_A+m_B); //x-velocity
    after collision in m/s
K2=0.5*(m_A+m_B)*v_2x^2; //total kinetic energy
    after collision in J
mprintf('Common x-velocity after collision: %f m/s',
    v_2x);
mprintf('\nInitial kinetic energy of system: %f J',
    K1);
mprintf('\nFinal kinetic energy of system: %f J',K2)
    ;
```

Scilab code Exa 8.9 An automobile collision

```
//OS: WINDOWS 10; SCILAB- 6.0.2
//CHAPTER 8: MOMENTUM AND IMPULSE
//EX 8.9: AN AUTOMOBILE COLLISION
clc;
clear;
m_C=1000; //mass of car in kg
m_T=2000; //mass of truck in kg
M=m_C+m_T; //total combined mass of car and truck in
    kg
v_Cx=0; //velocity of car before collision in x dir
    in m/s
v_Tx=10; //velocity of truck before collision in x
    dir in m/s
v_Cy=15; //velocity of car before collision in y dir
    in m/s
v_Ty=0; //velocity of truck before collision in y
    dir in m/s
P_x=m_C*v_Cx+m_T*v_Tx; //momentum of system before
    collision in x dir in Kg.m/s
P_y=m_C*v_Cy+m_T*v_Ty; //momentum of system before
    collision in y dir in Kg.m/s
P=sqrt(P_x^2+P_y^2); //magnitude of momentum of
    system before collision in Kg.m/s
theta=atand(P_y/P_x); //angle of momentum with +ve x
    axis
V=P/M; //magnitude of velocity after collision in m/
    s
mprintf('Magnitude of required velocity: %f m/s',V);
    //answer vary due to roundoff error
mprintf('\nAngle of velocity with +ve x axis in
    degrees: %f',theta); //answer vary due to
    roundoff error
```

Chapter 9

Rotation of rigid bodies

Scilab code Exa 9.1 Calculating angular velocity

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.1: CALCULATING ANGULAR VELOCITY
clc;
clear;
deff('[Theta]=theta(t)','Theta=2*t^3');
r=0.18; //radius in meter
t1=2; //given time instant in sec
t2=5; //given time instant in sec
theta1=theta(t1);
theta2=theta(t2);
mprintf('(a)Angle at t1=2s: %f radian = %f degrees '
    ,theta1,(180/%pi)*theta(t1)); //answer vary due
    to roundoff error
mprintf('\nAngle at t2=5s: %f radian = %f degrees ',
    theta2,(180/%pi)*theta(t2)); //answer vary due to
     roundoff error
s=r*(theta2-theta1);
mprintf('\n(b)Distance travelled between t=2s and t
```

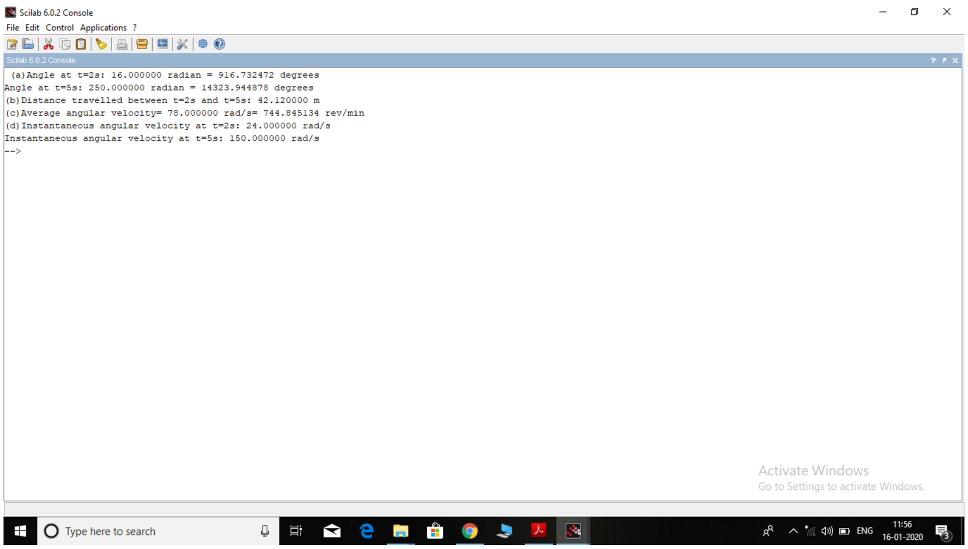

Figure 9.1: Calculating angular velocity

```
    =5s: %f m',s); //answer vary due to roundoff
    error
omega_av_z=(theta2-theta1)/(t2-t1);
mprintf('\n(c)Average angular velocity= %f rad/s= %f
     rev/min',omega_av_z,30*omega_av_z/%pi); //answer
     vary due to roundoff error
omega_1z=numderivative(theta,t1); //Instantaneous
    angular velocity at t=2s (omega_z=d(theta)/dt)
omega_2z=numderivative(theta,t2); //Instantaneous
    angular velocity at t=5s
mprintf('\n(d)Instantaneous angular velocity at t=2s
    : %f rad/s',omega_1z);
mprintf('\nInstantaneous angular velocity at t=5s:
    %f rad/s',omega_2z);
```

Scilab code Exa 9.3 `Rotation with constant angular acceleration`

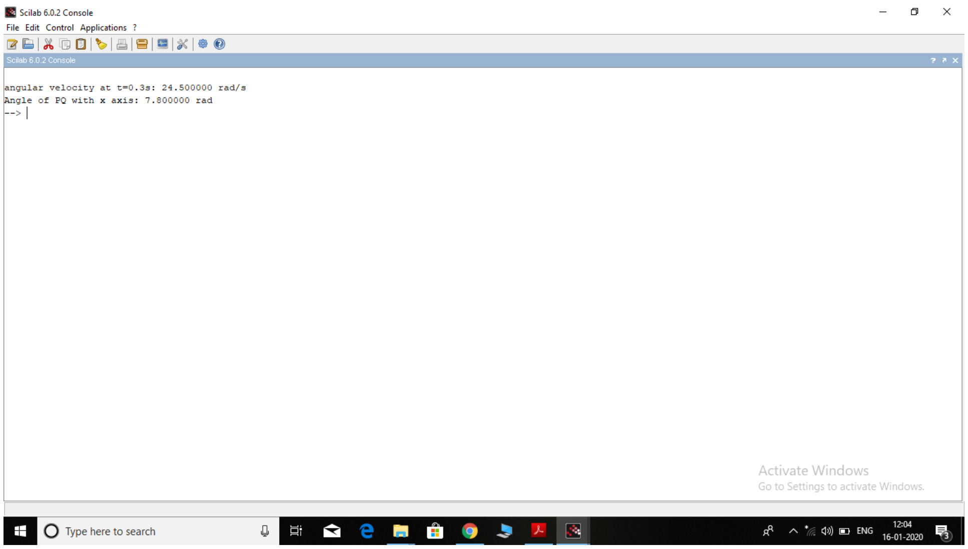

Figure 9.2: Rotation with constant angular acceleration

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.3: ROTATION WITH CONSTANT ANGULAR ACCELERATION
clc;
clear;
omega_0z=27.5; //initial angular velocity in rad/s
alpha_z=-10; //angular acceleration in rad/s^2
t=0.3; //time in sec
theta_0=0; //initial angle of PQ with x axis
omega_z=omega_0z+alpha_z*t;
theta=theta_0+omega_0z*t+0.5*alpha_z*t^2;
mprintf('\nangular velocity at t=0.3s: %f rad/s',omega_z);
mprintf('\nAngle of PQ with x axis: %f rad or %f rev',theta,theta/(2*%pi));
```

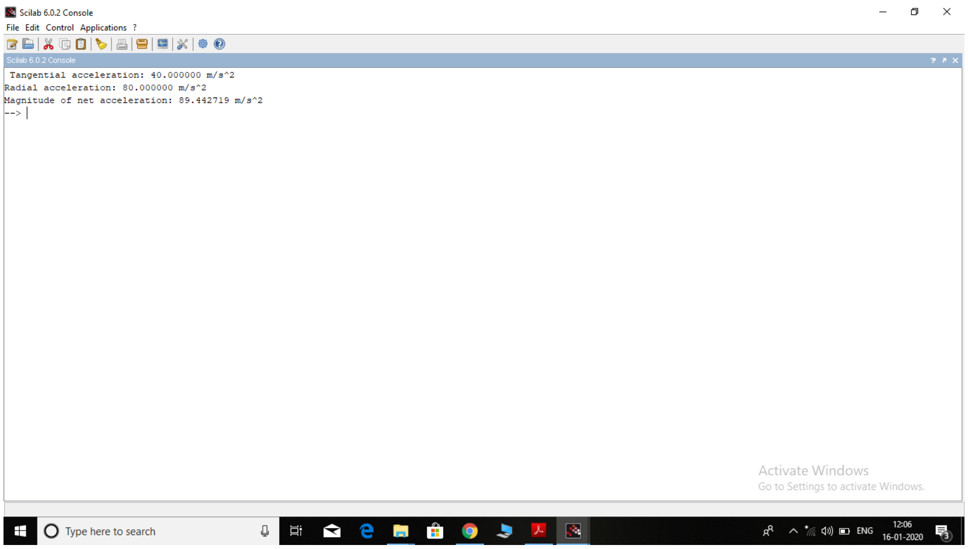

Figure 9.3: Throwing a discus

Scilab code Exa 9.4 Throwing a discus

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.4: THROWING A DISCUS
clc;
clear;
omega=10; //angular velocity in rad/s
alpha=50; //angular acceleration in rad/s^2
r=0.8; //radius in m
a_tan=r*alpha; //tangential acceleration in m/s^2
a_rad=(omega^2)*r; //radial acceleration in m/s^2
a=sqrt(a_tan^2+a_rad^2);
mprintf('Tangential acceleration: %f m/s^2',a_tan);
mprintf('\nRadial acceleration: %f m/s^2',a_rad);
mprintf('\nMagnitude of net acceleration: %f m/s^2',
    a);
```

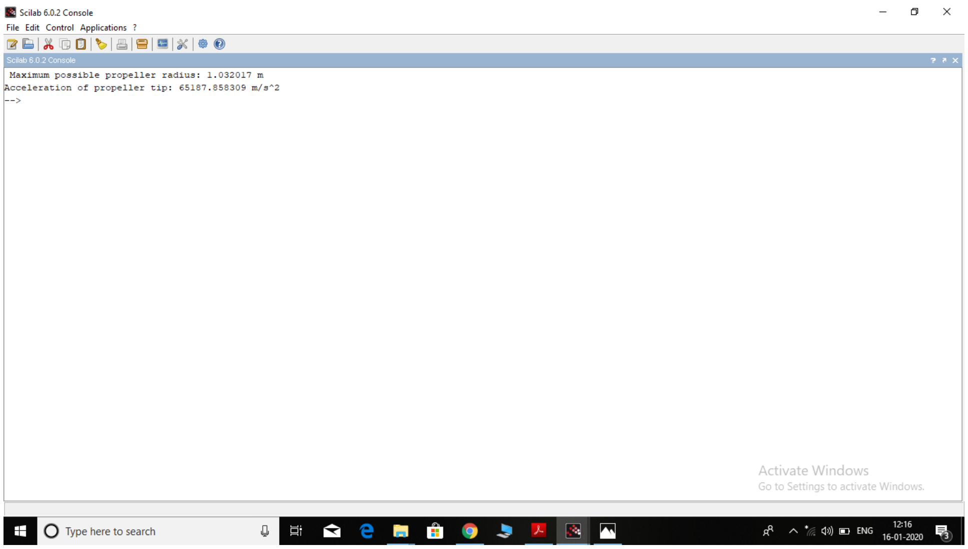

Figure 9.4: Designing a propeller

Scilab code Exa 9.5 Designing a propeller

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.5: DESIGNING A PROPELLER
clc;
clear;
omega=2400; //angular velocity of airplane propeller in rpm
omega=omega*(2*%pi/60);
v_tip=270; //maximum speed of tip of propeller in m/s
v_plane=75; //speed of plane in m/s
r=sqrt(v_tip^2-v_plane^2)/omega; //maximum propeller radius in m
```

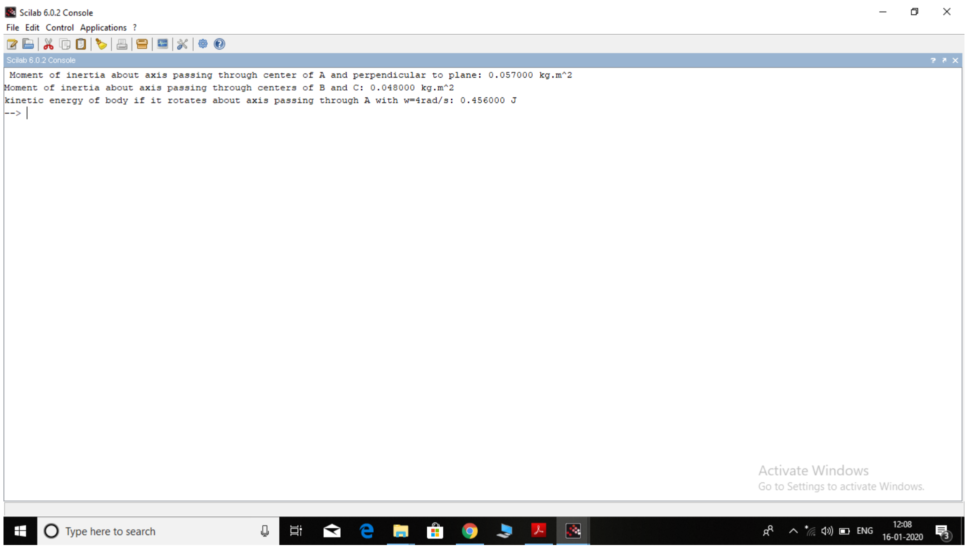

Figure 9.5: Moments of inertia for different rotation axes

```
a_rad=(omega^2)*r; //radial acceleration in m/s^2
mprintf('Maximum possible propeller radius: %f m',r);
mprintf('\nAcceleration of propeller tip: %f m/s^2',a_rad); //answer given in textbook is wrong
```

Scilab code Exa 9.6 `Moments of inertia for different rotation axes`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.6: MOMENT OF INERTIA FOR DIFFERENT ROTATION AXES
clc;
clear;
m_i=[0.3 0.1 0.2] //in the order m_A=0.3, m_B=0.1, m_C=0.2 as given in textbook
```

```
omega=4; //angular velocity in rad/s
r_i=[0; 0.5; 0.4] //perpendicular distance of m_A, m_B, m_C from axis passing through center of A and perpendicular to plane of diagra respectively.
I_A=sum(m_i*r_i^2); //moment of inertia about axis passing through center of A
r_i=[0.4; 0; 0]; //perpendicular distance of m_A, m_B, m_C from axis passing through centers of B and C and perpendicular to plane of diagram respectively.
I_BC=sum(m_i*r_i^2)
K_A=0.5*I_A*omega^2;
mprintf('Moment of inertia about axis passing through center of A and perpendicular to plane: %f kg.m^2',I_A);
mprintf('\nMoment of inertia about axis passing through centers of B and C: %f kg.m^2',I_BC);
mprintf('\nkinetic energy of body if it rotates about axis passing through A with omega=4rad/s: %f J',K_A);
//answer vary due to roundoff error
```

Scilab code Exa 9.7 An unwinding cable 1

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.7: AN UNWINDING CABLE 1
clc;
clear;
m=50; //mass in kg
R=0.12/2; //radius in m
F=9; //force in N
```

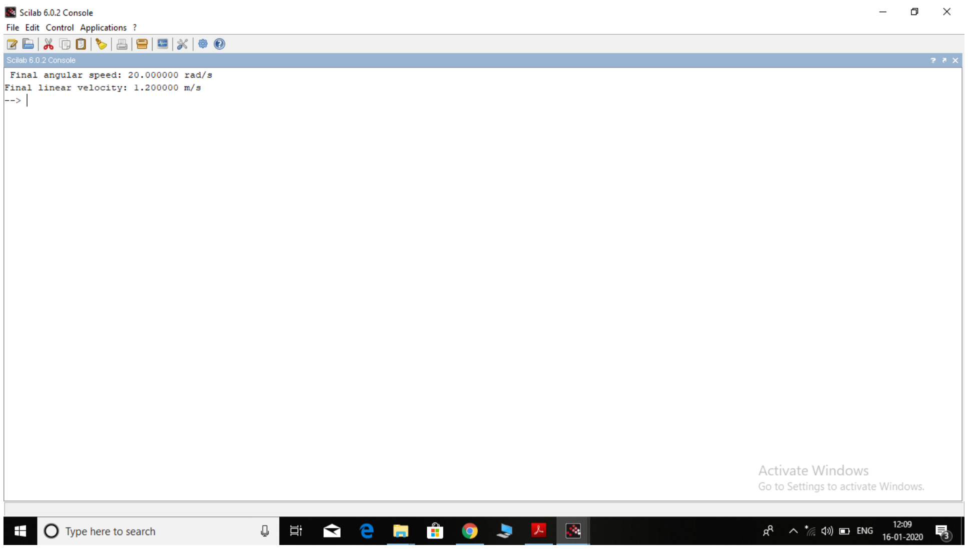

Figure 9.6: An unwinding cable 1

```
s=2; //distance in m
W_other=F*s; //work done in J
I=0.5*m*R^2; //moment of inertia in kg.m^2
omega=sqrt((2*W_other)/I); //final angular velocity
    in rad/s
v=R*omega; //linear velocity in m/s
mprintf('Final angular speed: %f rad/s',omega);
mprintf('\nFinal linear velocity: %f m/s',v);
```

Scilab code Exa 9.9 Using the parallel axis theorem

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 9: ROTATION OF RIGID BODIES
//EX 9.9: USING THE PARALLEL AXIS THEOREM
clc;
clear;
```

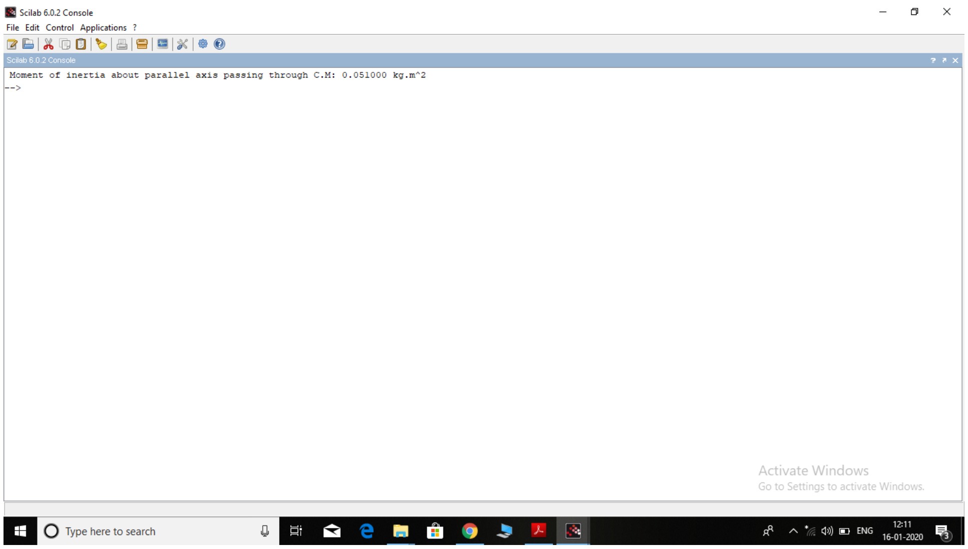

Figure 9.7: Using the parallel axis theorem

```
M=3.6; //mass in kg
I_p=0.132; //moment of inertia about parallel axis
d=0.15; //distance in m
I_cm=I_p-M*d^2; //moment of inertia about center of
    mass
mprintf('Moment of inertia about parallel axis
    passing through C.M: %f kg.m^2',I_cm);
```

Chapter 13

Gravitation

Scilab code Exa 13.1 Calculating gravitational force

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.1: CALCULATING GRAVITATIONAL FORCE
clear;
clc;
m1=0.5; //mass in kg
m2=0.01; //mass in kg
r=0.05; //distance between masses in m
G=6.67*10^-11; //universal gravitational constant in
    Nm^2/kg^2
Fg=(G*m1*m2)/r^2; //Gravitational force in N
mprintf('Gravational force: %e N',Fg);
```

Scilab code Exa 13.2 Acceleration due to gravitational attaraction

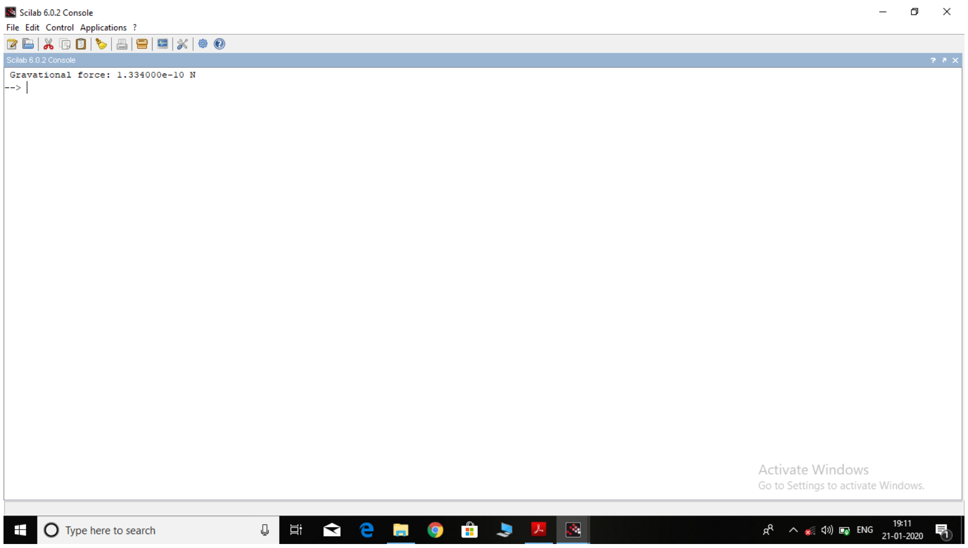

Figure 13.1: Calculating gravitational force

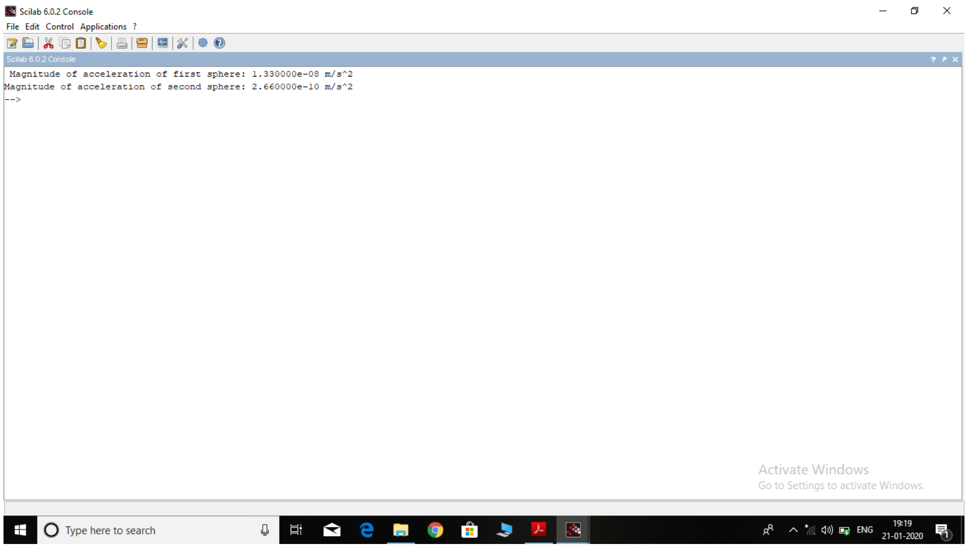

Figure 13.2: Acceleration due to gravitational attaraction

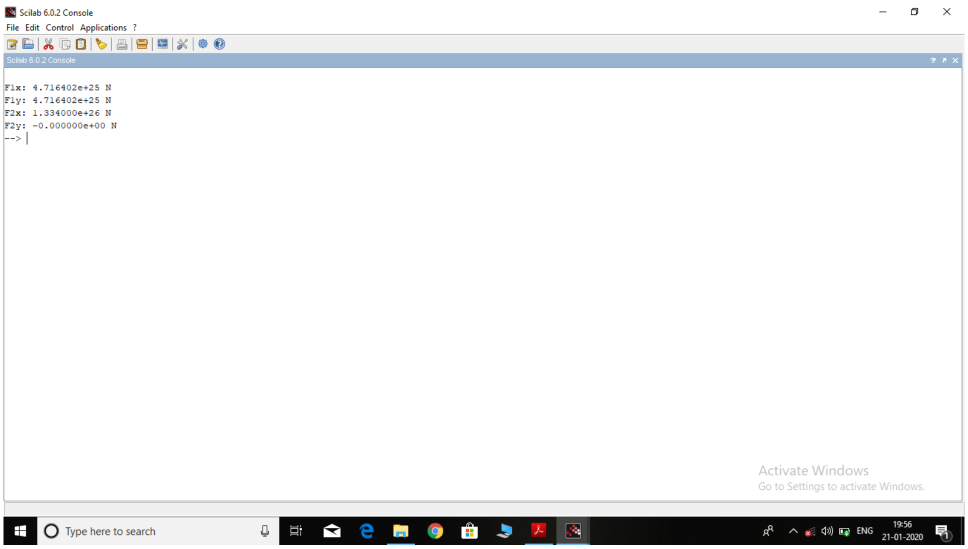

Figure 13.3: Superposition of gravitational forces

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX 13.2: ACCELERATION DUE TO GRAVITATIONAL
    ATTARACTION
clear;
clc;
Fg=1.33*10^-10; //given gravitational force in N (
    refer to Ex13.1)
m1=0.01; //mass of first sphere in kg
m2=0.5; //mass of second sphere in kg
a1=Fg/m1; //acceleration of first sphere in m/s^2
a2=Fg/m2; //acceleration of second sphere in m/s^2
mprintf('Magnitude of acceleration of first sphere:
    %e m/s^2',a1);
mprintf('\nMagnitude of acceleration of second
    sphere: %e m/s^2',a2);
```

Scilab code Exa 13.3 Superposition of gravitational forces

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.3: SUPERPOSITION OF GRAVITATIONAL FORCES
clear;
clc;
m1=1*10^30; //mass of small star in kg
m2=8*10^30; //mass of large stars in kg
m3=m2;
G=6.67*10^-11; //universal gravitational constant in Nm^2/kg^2
theta1=45; //angle of force F1 with + x axis in degrees
theta2=0; //angle of force F2 with + x axis in degrees
r=2*10^12; //length of 2 equal side of right isoceles star triangle system in m
F1=(G*m1*m2)/(2*r^2); //magnitude of gravitational force on small star due to second large star in N
F2=(G*m1*m3)/(r^2); //magnitude of gravitational force on small star due to third large star in N
F1x=F1*cosd(theta1); //x component of force F1 in N
F1y=F1*sind(theta1); //y component of force F1 in N
F2x=F2*cosd(theta2); //x component of force F2 in N
F2y=F2*sind(theta2); //y component of force F2 in N
Fx=F1x+F2x; //x component of net force F in N
Fy=F1y+F2y; //y component of net force F in N
F=sqrt(Fx^2+Fy^2); //magnitude of net force F in N
theta=atand(Fy/Fx); //angle of net force F with +ve x axix
printf('Total gravitational force exerted on small star: %e N',F); //answer vary due to roundoff error
```

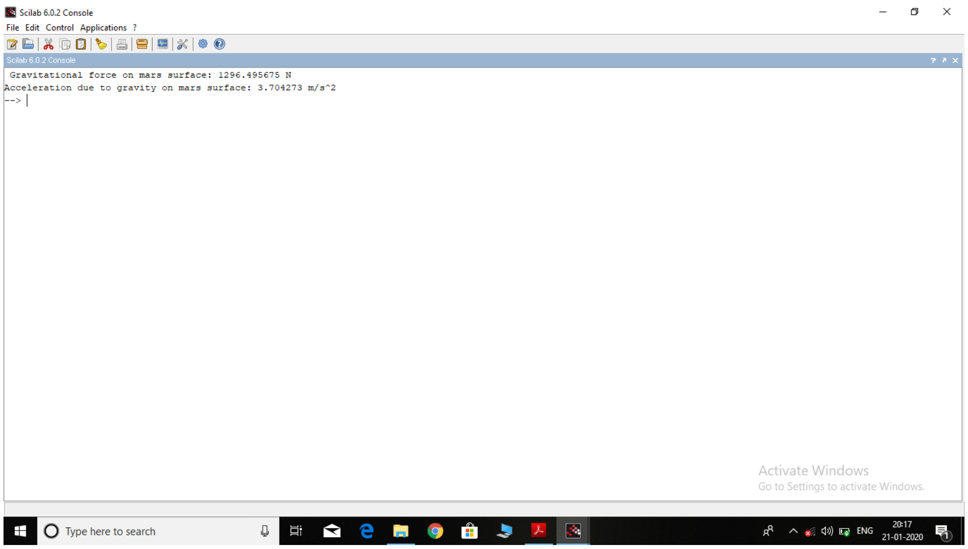

Figure 13.4: Gravity on Mars

```
printf('\nAngle of force F with +ve x axix: %f
    degrees',theta); //answer vary due to roundoff
    error
```

Scilab code Exa 13.4 Gravity on Mars

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.4: GRAVITY ON MARS
clear;
clc;
w=3430; //weight on Earth in N
g=9.8; //acceleration due to gravity on Earth in m/s
    ^2
m=w/g; //mass of lander in kg
m_M=6.42*10^23; //mass of mars in kg
```

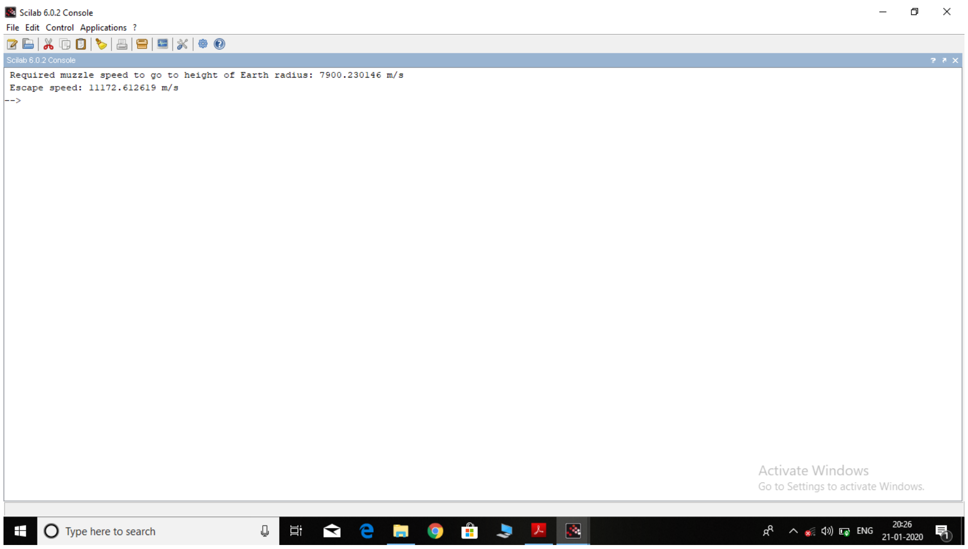

Figure 13.5: From the Earth to the Moon

```
G=6.67*10^-11; //universal gravitational constant in
    Nm^2/kg^2
R_M=3.4*10^6; //radius of mars in m
F_g=(G*m*m_M)/R_M^2; //Gravitational force on mars
    in N
g_M=F_g/m; //acceleration due to gravity on mars
    surface in m/s^2
printf('Gravitational force on mars surface: %f N',
    F_g); //answer given in textbook is wrong
printf('\nAcceleration due to gravity on mars
    surface: %f m/s^2',g_M);
```

Scilab code Exa 13.5 From the Earth to the Moon

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
```

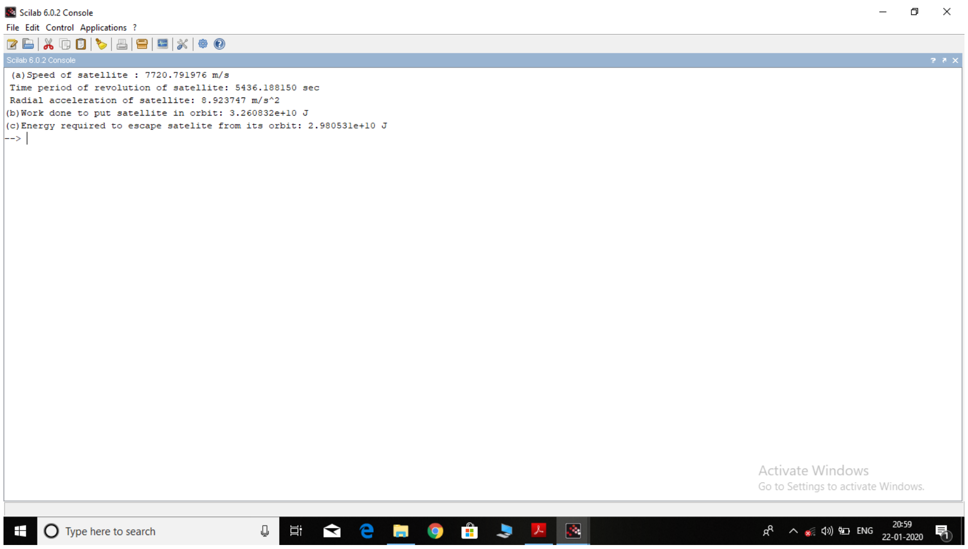

Figure 13.6: A satellite orbit

```
//EX13.5: FROM THE EARTH TO MOON
clear;
clc;
m_E=5.97*10^24; //mass of Earth in kg
G=6.67*10^-11; //universal gravitational constant in
    Nm^2/kg^2
R_E=6.38*10^6; //radius of Earth in m
v_1=sqrt(G*m_E/R_E);
printf('Required muzzle speed to go to height of
    Earth radius: %f m/s',v_1);
v_1=sqrt(2*G*m_E/R_E); //escape speed in m/s (from
    energy conservation K1+U1=K2+U2; in this case K2=
    U2=0)
printf('\n Escape speed: %f m/s',v_1); //answer
    given in textbook is wrong
```

Scilab code Exa 13.6 `A satellite orbit`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.6: A SATELLITE ORBIT
clear;
clc;
m=1000; //mass of satellite in kg
m_E=5.97*10^24; //mass of Earth in kg
G=6.67*10^-11; //universal gravitational constant in Nm^2/kg^2
r_orbit=3*10^5; //radius of orbit of satellite above Earth's surface in m
R_E=6.38*10^6; //radius of Earth in m
r=R_E+r_orbit; //net orbit radius in m
v=sqrt(G*m_E/r); //speed of satellite in m/s
T=(2*%pi*r)/v; //time period of revolution of satellite in sec
a_rad=v^2/r; //Radial acceleration of satellite in m/s^2
printf('(a)Speed of satellite : %f m/s',v); //answer vary due to roundoff error
printf('\n Time period of revolution of satellite: %f sec',T); //answer vary due to roundoff error
printf('\n Radial acceleration of satellite: %f m/s^2',a_rad); //answer vary due to roundoff error
E1=-(G*m_E*m)/R_E; //Initial total energy in J
E2=-(G*m_E*m)/(2*r); //Final total energy in J
W_required=E2-E1; //work done in J
printf('\n(b)Work done to put satellite in orbit: %e J',W_required);
printf('\n(c)Energy required to escape satelite from its orbit: %e J ',-E2);
```

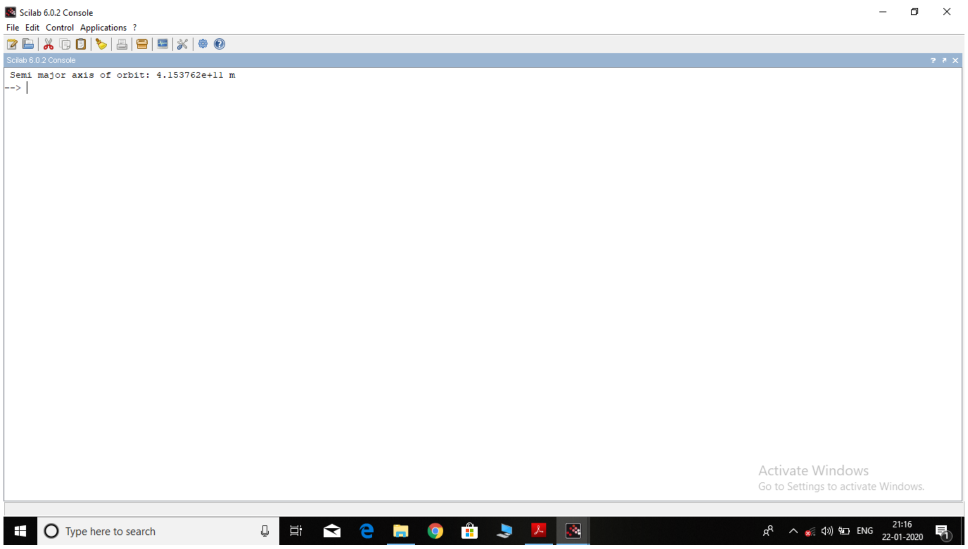

Figure 13.7: Kepler third law

Scilab code Exa 13.8 `Kepler third law`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.8: KEPLER'S THIRD LAW
clear;
clc;
m_S=1.99*10^30; //mass of asteroid in kg
T=4.62; //given time period in years
T=T*365.2425*24*3600; //time period in sec
G=6.67*10^-11; //universal gravitational constant in Nm^2/kg^2
a=((G*m_S*T^2)/(4*%pi^2))^(1/3);
printf('Semi major axis of orbit: %e m',a);
```

Scilab code Exa 13.9 `Comet Halley`

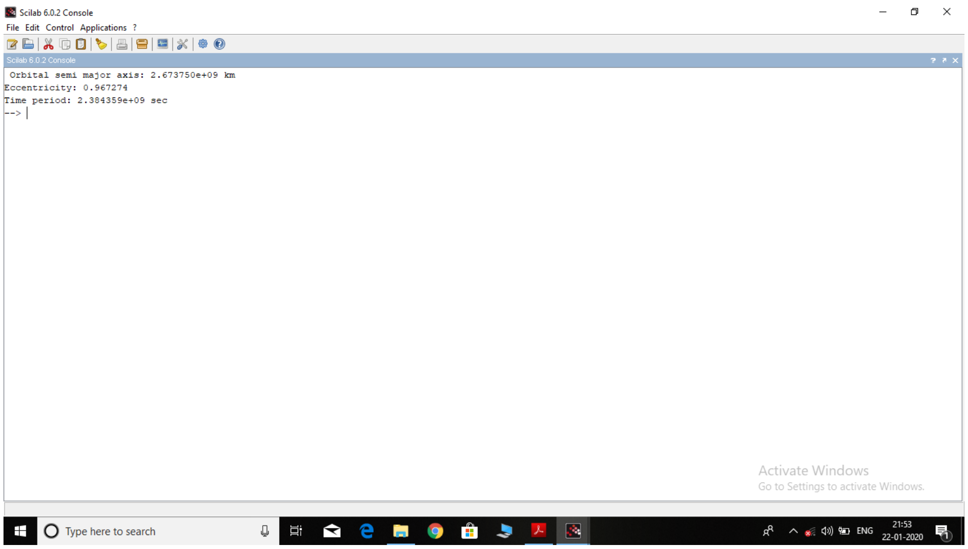

Figure 13.8: Comet Halley

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.9: COMET HALLEY
clear;
clc;
distance_perihelion=8.75*10^7; //distance of comet halley from sun at perihelion in km
distance_aphelion=5.26*10^9; //distance of comet halley from sun at aphelion in km
a=(distance_perihelion+distance_aphelion)/2; //semi major axis in km
printf('Orbital semi major axis: %e km',a);
e=1-distance_perihelion/a; //eccentricity of ellipse
a=a*1000; //semi major axis in m
m_S=1.99*10^30; //mass of comet in kg
G=6.67*10^-11; //universal gravitational constant in Nm^2/kg^2
T=(2*%pi*a^(3/2))/sqrt(G*m_S);
printf('\nEccentricity: %f ',e);
printf('\nTime period: %e sec',T);
```

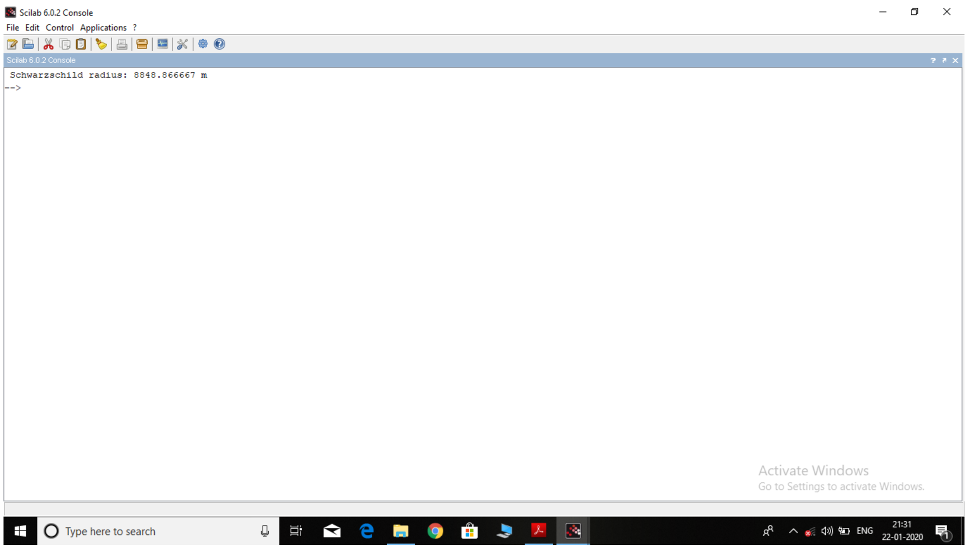

Figure 13.9: Black hole calculations

Scilab code Exa 13.11 `Black hole calculations`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 13: GRAVITATION
//EX13.11: BLACK HOLE CALCULATIONS
clear;
clc;
m=1.99*10^30; //solar mass in kg
M=3*m; //mass of star in kg
G=6.67*10^-11; //universal gravitational constant in
    Nm^2/kg^2
c=3*10^8; //speed of light in m/s
Rs=(2*G*M)/c^2; //Schwarzschild radius in m
printf('Schwarzschild radius: %f m',Rs); //answer
    given in textbook is wrong
```

```
rho=M/((4/3)*%pi*Rs^3); //average density of star in kg/m^3
printf('\nAverage density of star: %e kg/m^3',rho); //answer given in textbook is wrong
```

Chapter 14

Periodic motion

Scilab code Exa 14.1 Period frequency and angular frequency

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 14: PERIODIC MOTION
//EX 14.1: Period, frequency and angular frequency
clear;
clc;
f=6.7*10^6; //frequency of ultrasonic transducer
T=1/f; //time period in sec
omega=2*%pi*f; //angular frequency in rad/s
printf('Time period: %e sec',T);
printf('\nAngular frequency: %e rad/s',omega);
```

Scilab code Exa 14.2 Angular frequency frequency and period in SHM

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 14: PERIODIC MOTION
```

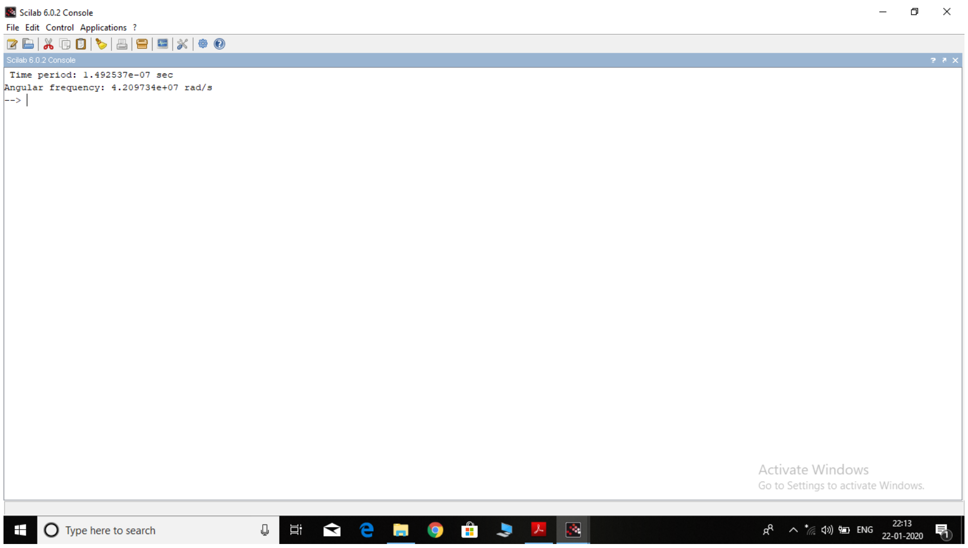

Figure 14.1: Period frequency and angular frequency

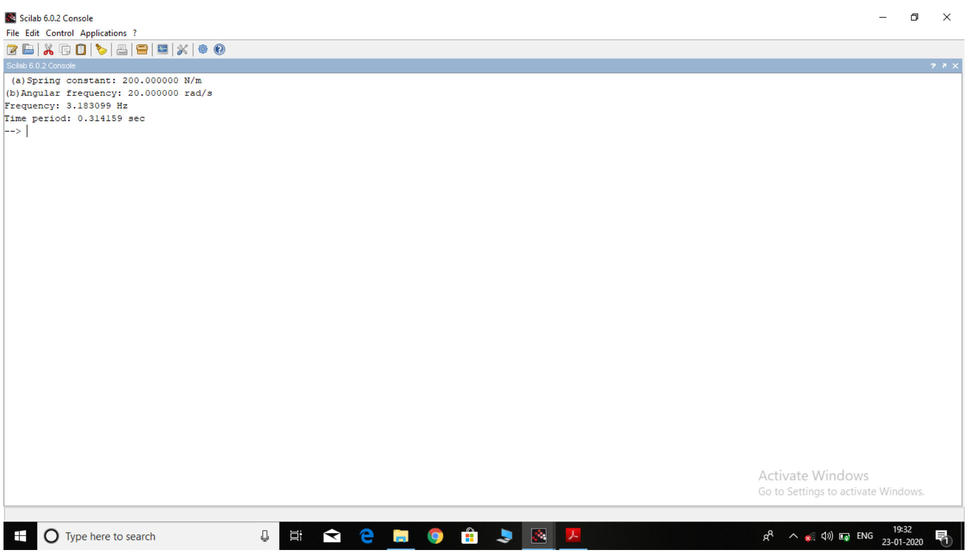

Figure 14.2: Angular frequency frequency and period in SHM

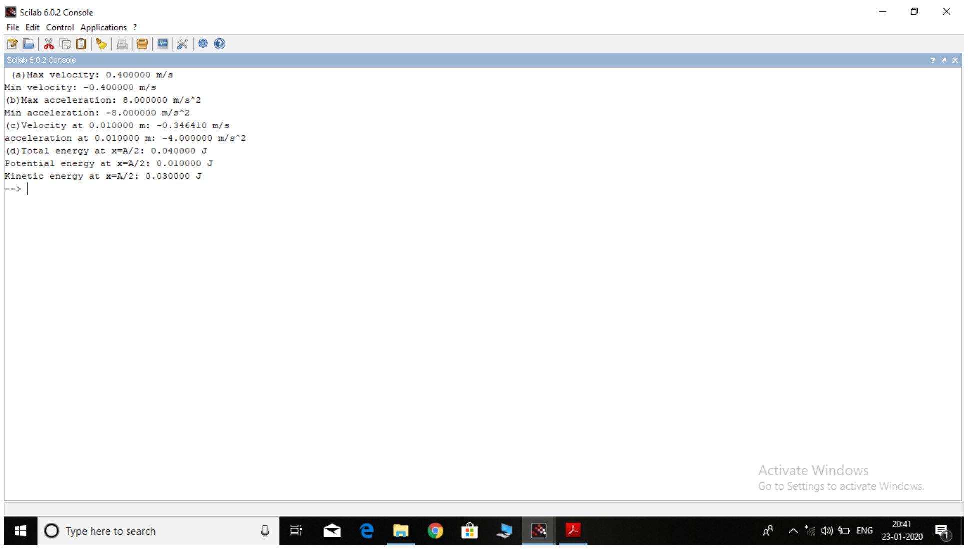

Figure 14.3: Velocity acceleration and energy in SHM

```
//EX 14.2: ANGULAR FREQUENCY, FREQUENCY AND PERIOD
    IN SHM
clear;
clc;
F_x=-6; //given spring force in N
x=0.03; //displacement in m
k=-F_x/x; //spring constant in N/m
m=0.5; //mass of block in kg
omega=sqrt(k/m); //angular frequency in rad/s
f=omega/(2*%pi); //frequency in Hz
T=1/f; //time period of SHM in sec
printf('(a)Spring constant: %f N/m',k);
printf('\n(b)Angular frequency: %f rad/s \nFrequency
    : %f Hz \nTime period: %f sec',omega,f,T); //
    answer vary due to roundoff error
```

Scilab code Exa 14.4 Velocity acceleration and energy in SHM

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 14: PERIODIC MOTION
//EX 14.4: VELOCITY, ACCELERATION AND ENERGY IN SHM
clear;
clc;
k=200; //spring constant in N/m
m=0.5; //mass of block in kg
A=0.02; //amplitude in m
v_max=sqrt(k/m)*A; //max velocity in m/s
v_min=-v_max; //min velocity in m/s
printf('(a)Max velocity: %f m/s\nMin velocity: %f m/
    s',v_max,v_min);
a_max=(-k/m)*(-A); //max acceleration in m/s^2
a_min=-a_max; //min acceleration in m/s^2
printf('\n(b)Max acceleration: %f m/s^2\nMin
    acceleration: %f m/s^2',a_max,a_min);
x=A/2; //displacement of half the amplitude in m
v_x=-sqrt(k/m)*sqrt(A^2-x^2); //velocity at x in m/s
a_x=-(k/m)*x; //acceleration at x in m/s^2
printf('\n(c)Velocity at %f m: %f m/s',x,v_x);
printf('\nacceleration at %f m: %f m/s^2',x,a_x);
E=0.5*k*A^2; //total energy in J
U=0.5*k*x^2; //potential energy at x=A/2 in J
K=0.5*m*v_x^2; //kinetic energy at x=A/2 in J
printf('\n(d)Total energy at x=A/2: %f J',E);
printf('\nPotential energy at x=A/2: %f J',U);
printf('\nKinetic energy at x=A/2: %f J',K);
```

Scilab code Exa 14.6 Vertical SHM in an old car

```
//OS: WINDOWS 10, SCILAB-6.0.2
```

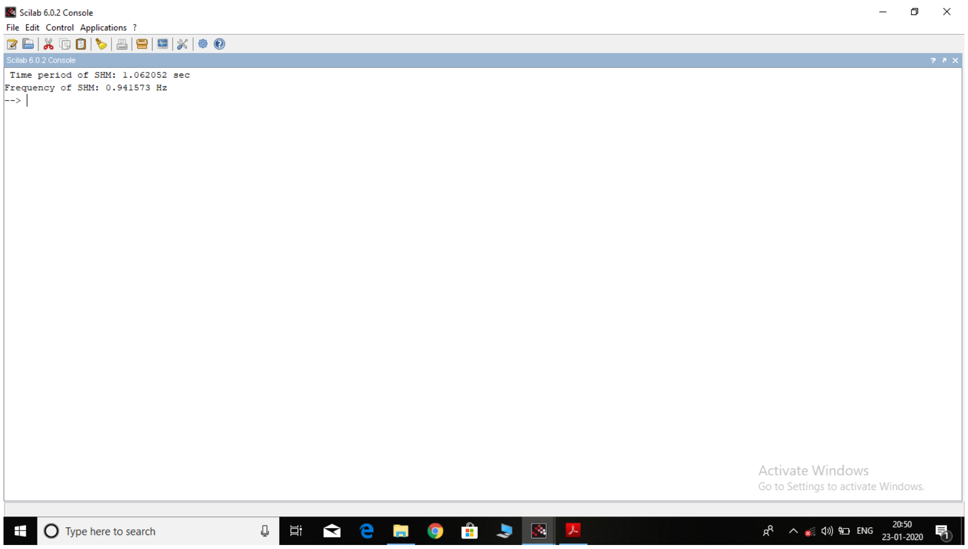

Figure 14.4: Vertical SHM in an old car

```
//CHAPTER 14: PERIODIC MOTION
//EX 14.6: VERTICAL SHM IN AN OLD CAR
clear;
clc;
m=1000; //mass of car in kg
F_x=980; //magnitude of force in N
x=-2.8*10^-2; //given compression of shockers in m
k=-F_x/x; //spring constant of shocker spring in N/m
T=2*%pi*sqrt(m/k); //Time period of SHM
f=1/T; //frequency of SHM
printf('Time period of SHM: %f sec',T); //answer
    given in textbook is wrong
printf('\nFrequency of SHM: %f Hz',f); //answer
    given in textbook is wrong
```

Scilab code Exa 14.7 `Molecular vibration`

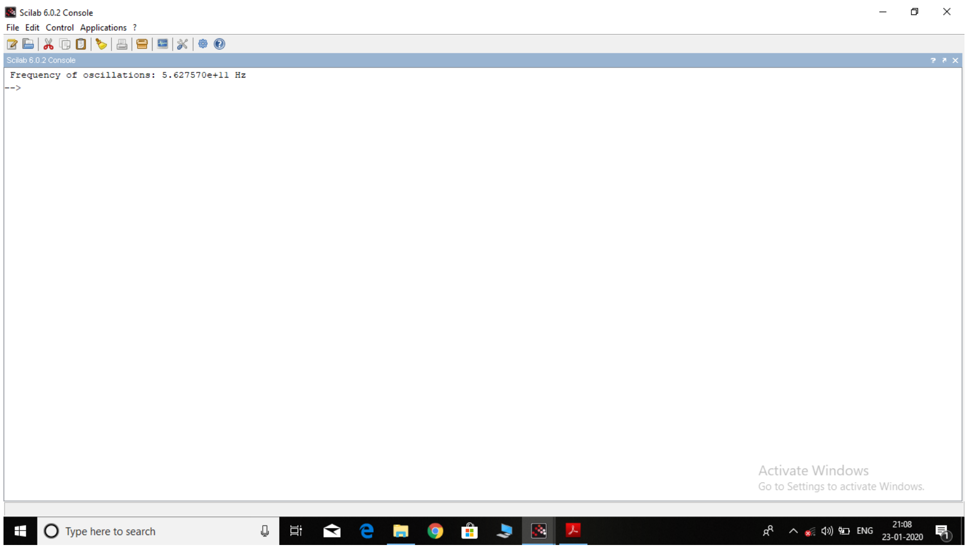

Figure 14.5: Molecular vibration

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 14: PERIODIC MOTION
//EX 14.7: MOLECULAR VIBRATION
clear;
clc;
U_0=1.68*10^-21; //potential energy at equilibrium
    distance in Ar atom in J
R_0=3.82*10^-10; //equilibrium distance on Ar atom
    in m
m=6.63*10^-26; //mass of Ar atom in kg
k=(72*U_0)/R_0^2; //force constant in N/m
f=(1/(2*%pi))*sqrt(k/m); //frequency of SHM
printf('Frequency of oscillations: %e Hz',f);
```

Scilab code Exa 14.8 A simple pendulum

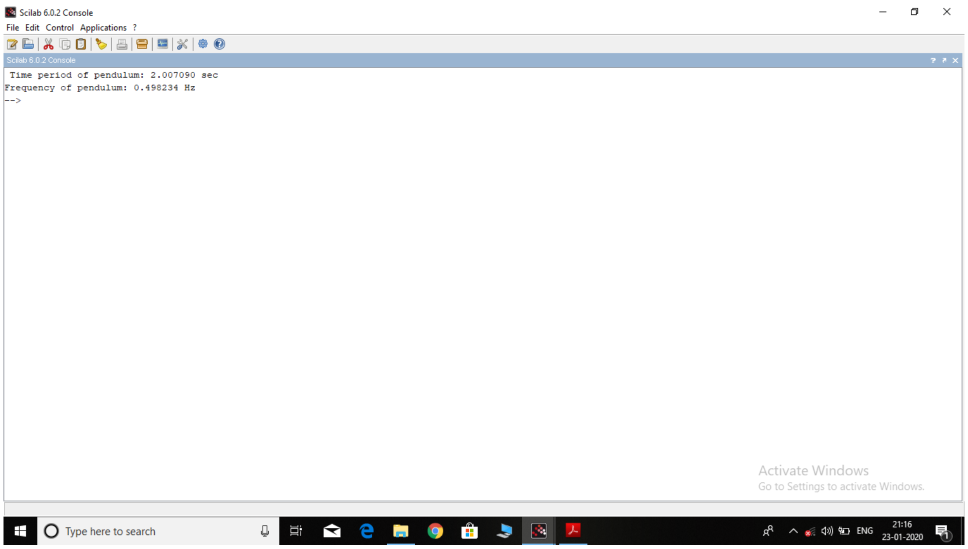

Figure 14.6: A simple pendulum

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 14: PERIODIC MOTION
//EX 14.8: A SIMPLE PENDULUM
clear;
clc;
g=9.8; //acceleration due to gravity on Earth in m/s
    ^2
L=1; //length of pendulum in m
T=2*%pi*sqrt(L/g); //time period of pendulum
f=1/T; //frequency of pendulum in Hz
printf('Time period of pendulum: %f sec ',T);
printf('\nFrequency of pendulum: %f Hz',f);
```

Scilab code Exa 14.9 Physical pendulum VS simple pendulum

```
//OS: WINDOWS 10, SCILAB-6.0.2
```

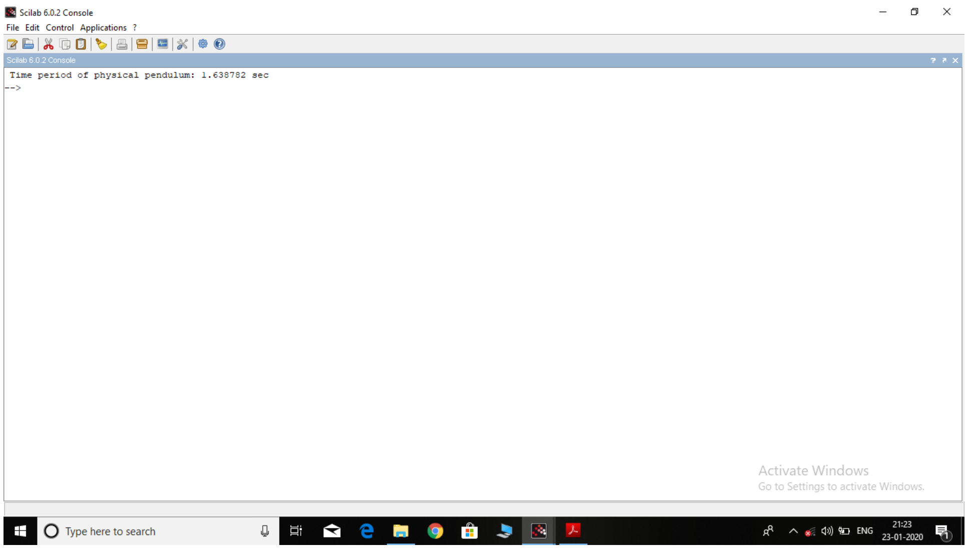

Figure 14.7: Physical pendulum VS simple pendulum

```
//CHAPTER 14: PERIODIC MOTION
//EX 14.9: PHYSICAL PENDULUM VS PHYSICAL PENDULUM
clear;
clc;
L=1; //length of stick in m
g=9.8; //acceleration due to gravity on Earth in m/s^2
T=2*%pi*sqrt((2*L)/(3*g)); //time period of physical pendulum in sec
printf('Time period of physical pendulum: %f sec',T);
```

Chapter 16

Sound and Hearing

Scilab code Exa 16.1 Amplitude of a sound wave in air

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.1: AMPLITUDE OF A SOUND WAVE IN AIR
clc;
clear;
p_max=3*10^-2; //max pressure variation in Pa
f=1000; //frequency of sound wave in Hz
v=344; //speed of sound in m/s
Beta=1.42*10^5; //bulk modulus in Pa
omega=2*%pi*f; //angular frequency in rad/s
k=omega/v; //wave number
A=p_max/(Beta*k); //maximum displacement in m
printf('Amplitude of sound wave in air: %e m',A); //
    answer given in textbook is wrong
```

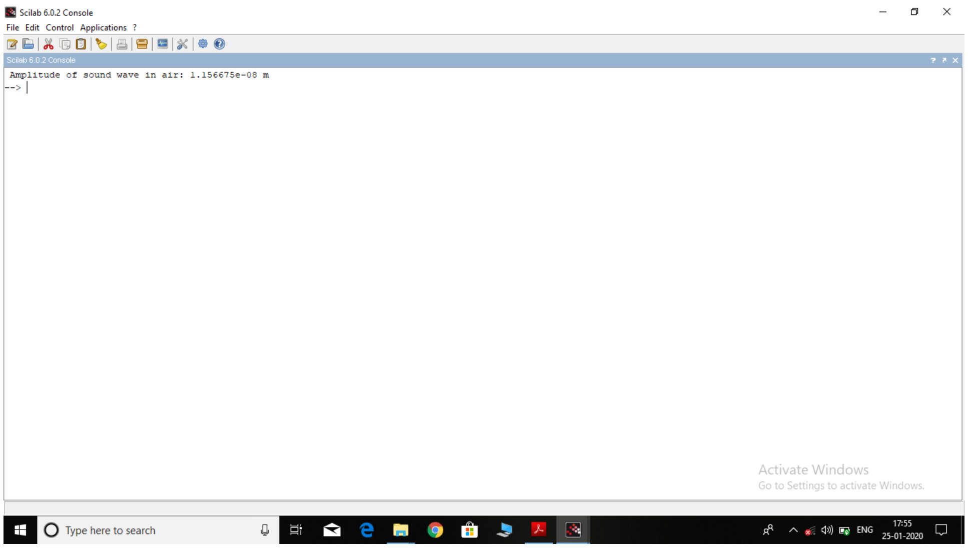

Figure 16.1: Amplitude of a sound wave in air

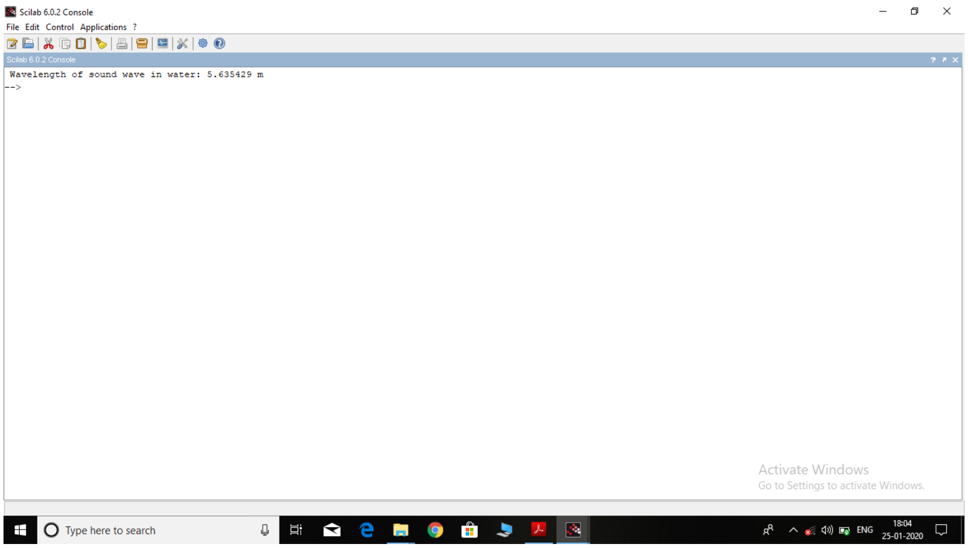

Figure 16.2: Wavelength of sonar waves

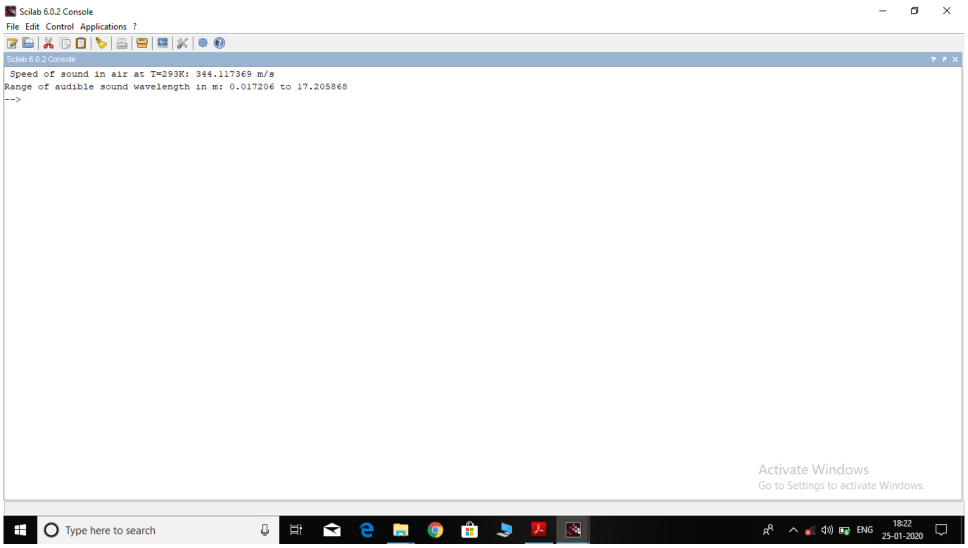

Figure 16.3: Speed of sound in air

Scilab code Exa 16.3 `Wavelength of sonar waves`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.3: WAVELENGTH OF SONAR WAVES
clc;
clear;
rho=1000; //density of water in kg/m^3
Beta=2.18*10^9; //bulk modulus of water in Pa
f=262; //frequency of sound wave in Hz
v=sqrt(Beta/rho); //speed of sound wave in water in
    m/s
lambda=v/f; //wavelength of sound wave in water in m
printf('Wavelength of sound wave in water: %f m',
    lambda); //answer vary due to roundoff error
```

Scilab code Exa 16.4 Speed of sound in air

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.4: SPEED OF SOUND IN AIR
clc;
clear;
T=293; //temperature of air in K
Gamma=1.4; //ratio of heat capacities in air
M=28.8*10^-3; //mean molar mass of air in kg/mol
R=8.314; //gas constant in J/K.mol
v=sqrt((Gamma*R*T)/M); //speed of sound in air in m/s
printf('Speed of sound in air at T=293K: %f m/s',v); //answer vary due to roundoff error
f=20; //min audible sound frequency in Hz
lambda=v/f; //max audible sound wavelength in m
printf('\nMaximum audible wavelength: %f m',lambda); //answer vary due to roundoff error
f=20000; //max audible sound frequency in Hz
lambda=v/f; //min audible sound wavelength in m
lambda=lambda*100; //min audible sound wavelength in cm
printf('\nMinimum audible wavelength: %f cm',lambda); //answer vary due to roundoff error
```

Scilab code Exa 16.5 Intensity of a sound wave in ear

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.5: INTENSITY OF SOUND WAVE IN AIR
clc;
clear;
```

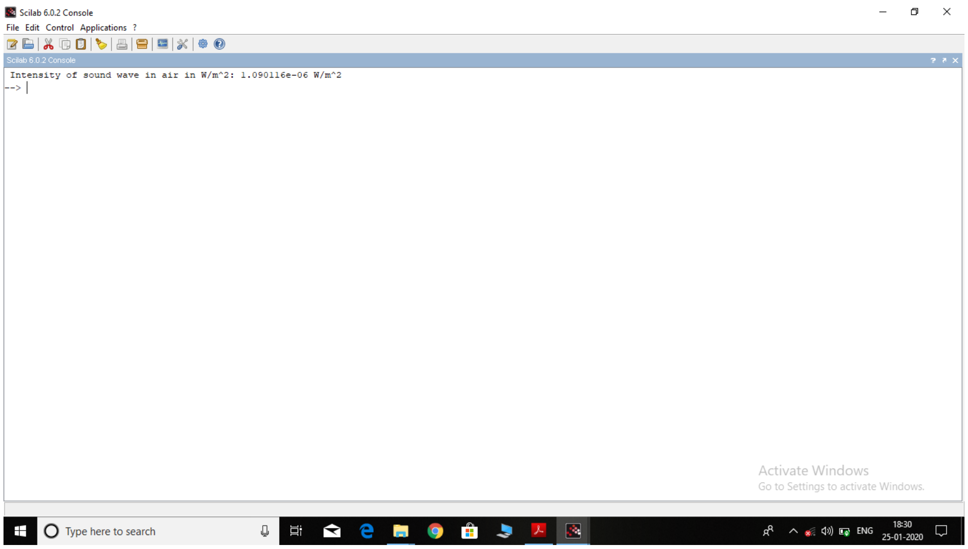

Figure 16.4: Intensity of a sound wave in ear

```
T=293; //temperature of air in K
rho=1.2; //density of air in kg/m^3
p_max=3*10^-2; //pressure amplitude of wave in m
v=344; //speed of sound in air in m/s
I=(p_max^2)/(2*rho*v); //intensity of sound wave in air in W/m^2
printf('Intensity of sound wave in air in W/m^2: %e W/m^2',I); //answer vary due to roundoff error
```

Scilab code Exa 16.6 Same intensities different frequencies

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.6: SAME INTENSITIES, DIFFERENT FREQUENCIES
clc;
clear;
```

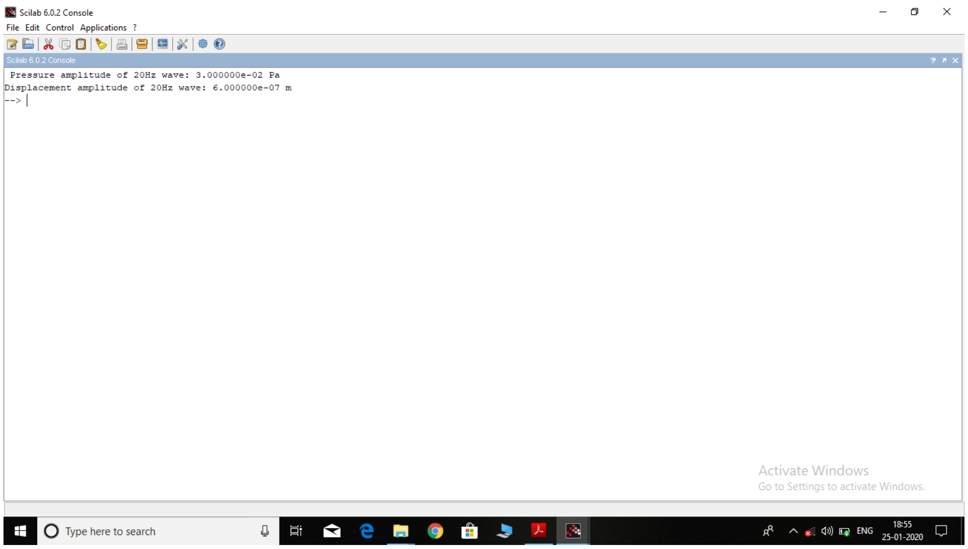

Figure 16.5: Same intensities different frequencies

```
f_1000=1000; //frequency of first sound wave in Hz
f_20=20; //frequency of second sound wave in Hz
A_1000=1.2*10^-8; //displacement amplitude of first sound wave in m
A_20=(f_1000/f_20)*A_1000; //displacement amplitude of second sound wave in m (since fA=constant)
p_max=3*10^-2; //given maximum pressure of 1000 Hz or 20 Hz sound wave (since it is independent of frequency)
printf('Pressure amplitude of 20Hz wave: %e Pa', p_max);
printf('\nDisplacement amplitude of 20Hz wave: %e m',A_20);
```

Scilab code Exa 16.8 `Temporary or permanent hearing loss`

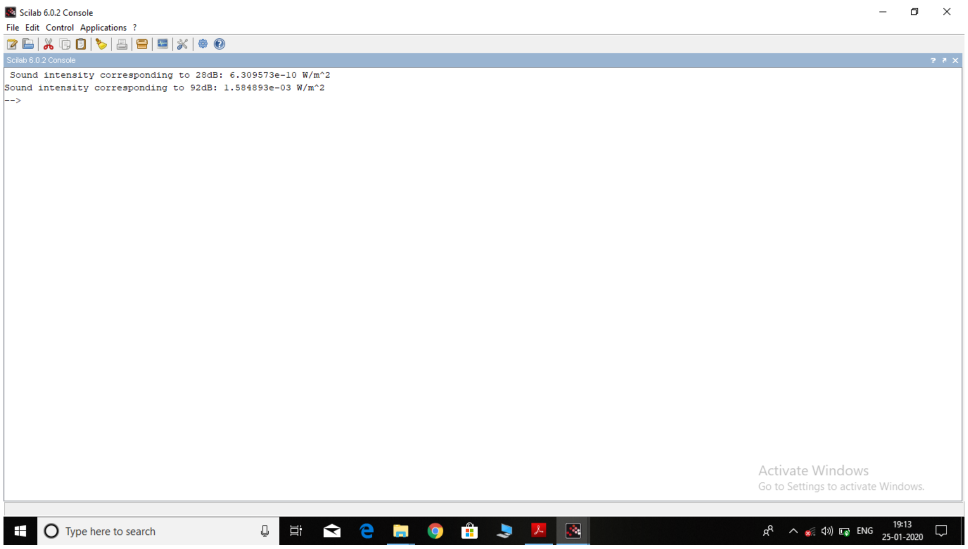

Figure 16.6: Temporary or permanent hearing loss

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.8: TEMPORARY/ PERMANENT HEARING LOSS
clc;
clear;
I_0=1*10^-12; //reference intensity in W/m^2
Beta=28; //first sound intensity level in dB
I_28_dB=I_0*10^(Beta/10); //sound intensity
    corresponding to 28dB in W/m^2
Beta=92; //second sound intensity level in dB
I_92_dB=I_0*10^(Beta/10); //sound intensity
    corresponding to 92dB in W/m^2
printf('Sound intensity corresponding to 28dB: %e W/
   m^2',I_28_dB);
printf('\nSound intensity corresponding to 92dB: %e
   W/m^2',I_92_dB); //answer vary due to roundoff
    error
```

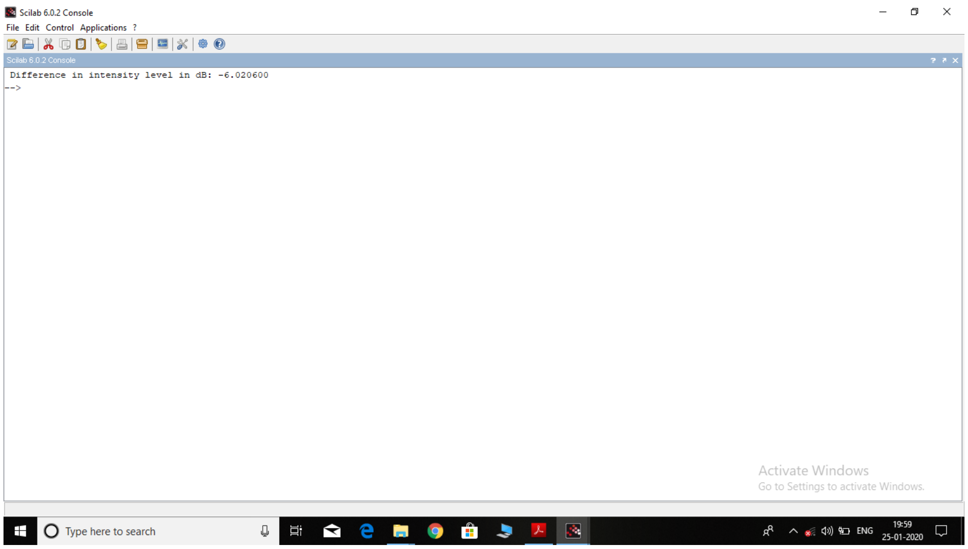

Figure 16.7: A bird sings in a meadow

Scilab code Exa 16.9 `A bird sings in a meadow`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.9: A BIRD SINGS IN A MEADOW
clc;
clear;
r1=1; //distances from sound source in m
r2=2;
I2_by_I1=(r1^2)/(r2^2); //ratio of intensities of sound wave (since I2/I1=r1^2/r2^2)
beta2_minus_beta1=10*log10(I2_by_I1); //Difference in intensity level in dB (B_diff=B2-B1)
printf('Difference in intensity level in dB: %f ', beta2_minus_beta1); //answer vary due to roundoff error
```

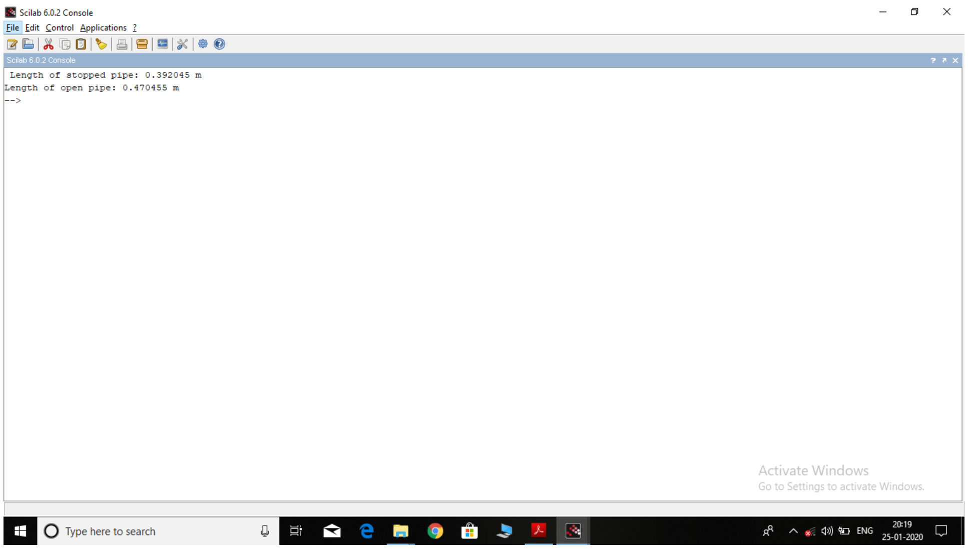

Figure 16.8: A tale of two pipes

Scilab code Exa 16.11 `A tale of two pipes`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.11: A TALE OF TWO PIPES
clc;
clear;
f1=220; //fundamental frequency of stopped organ pipe in Hz
v=345; //speed of sound in m/s
L_stopped=v/(4*f1); //length of stopped organ pipe in m
f5=5*f1; //frequency of second overtone of stopped pipe in Hz
L_open=(3*v)/(2*f5); //length of open pipe having
```

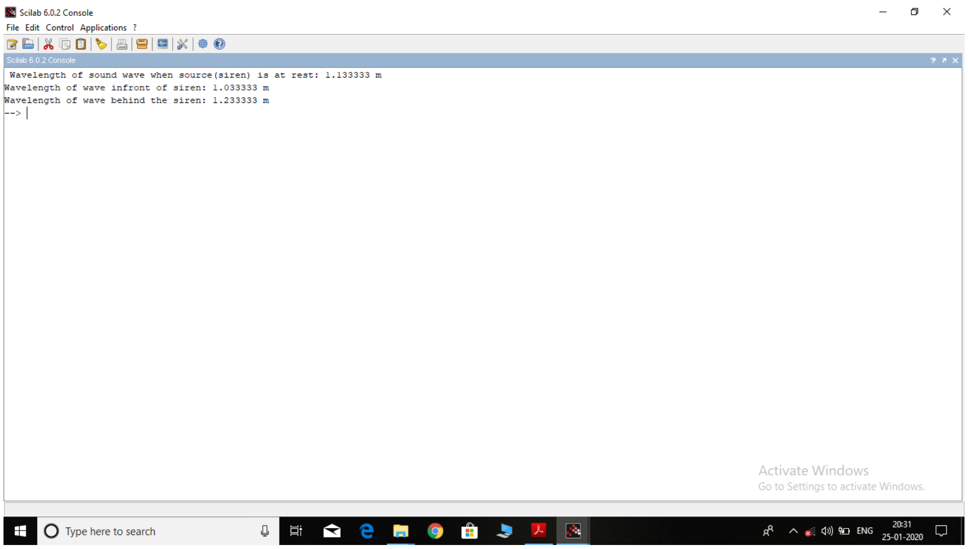

Figure 16.9: Doppler effect 1 wavelengths

```
    third harmonic same as f5
printf('Length of stopped pipe: %f m',L_stopped);
printf('\nLength of open pipe: %f m',L_open);
```

Scilab code Exa 16.14 Doppler effect 1 wavelengths

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.14: DOPPLER'S EFFECT 1: WAVELENGTHS
clc;
clear;
v=340; //speed of sound in air in m/s
f_s=300; //frquency of sound source(siren) in Hz
lambda=v/f_s; //wavelength of wave when source(siren
    ) is at rest in m
```

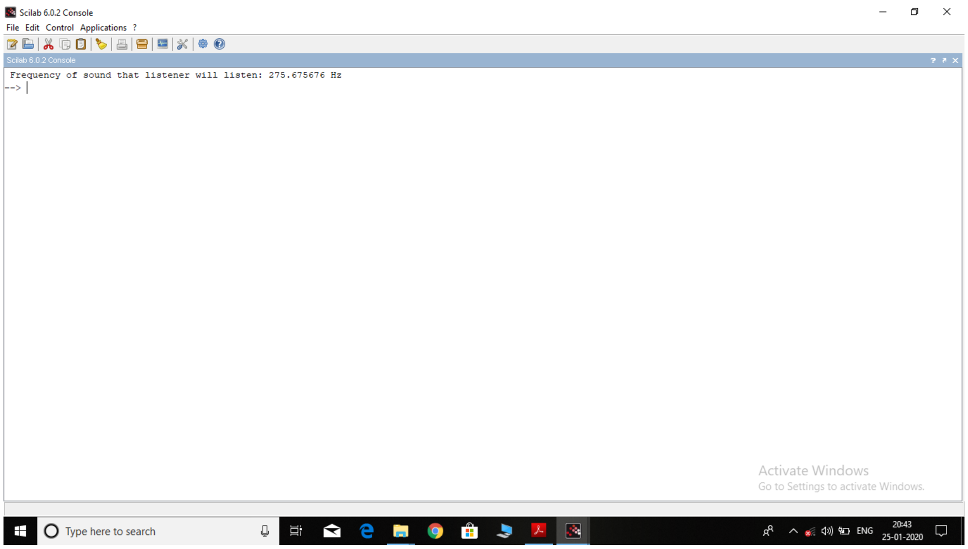

Figure 16.10: Doppler effect 2 frequencies

```
printf('Wavelength of sound wave when source(siren)
    is at rest: %f m',lambda);
v_s=30; //speed of siren in m/s
lambda_infront=(v-v_s)/f_s; //wavelength of wave
    infront of siren in m
lambda_behind=(v+v_s)/f_s; //wavelength of wave
    behind of siren in m
printf('\nWavelength of wave infront of siren: %f m'
    ,lambda_infront);
printf('\nWavelength of wave behind the siren: %f m'
    ,lambda_behind);
```

Scilab code Exa 16.15 Doppler effect 2 frequencies

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
```

```
//EX 16.15: DOPPLER'S EFFECT 2: FREQUENCIES
clc;
clear;
f_s=300; //frequency of sound wave in Hz
v=340; //speed of sound in air in m/s
v_s=30; //speed of siren in m/s
f_l=(v/(v+v_s))*f_s; //frequency of sound that
    listener will listen in Hz
printf('Frequency of sound that listener will listen
    : %f Hz',f_l); //answer vary due to roundoff
```

Scilab code Exa 16.16 Doppler effect 3 A moving listener

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.16: DOPPLER'S EFFECT 3: A MOVING LISTENER
clc;
clear;
f_s=300; //frequency of sound wave in Hz
v=340; //speed of sound in air in m/s
v_l=-30; //speed of siren in m/s
f_l=((v+v_l)/v)*f_s; //frequency of sound that
    listener will listen in Hz
printf('Frequency of sound that listener will listen
    : %f Hz',f_l); //answer in textbook is wrong
```

Scilab code Exa 16.17 Doppler effect 4 moving source moving listener

```
//OS: WINDOWS 10, SCILAB-6.0.2
```

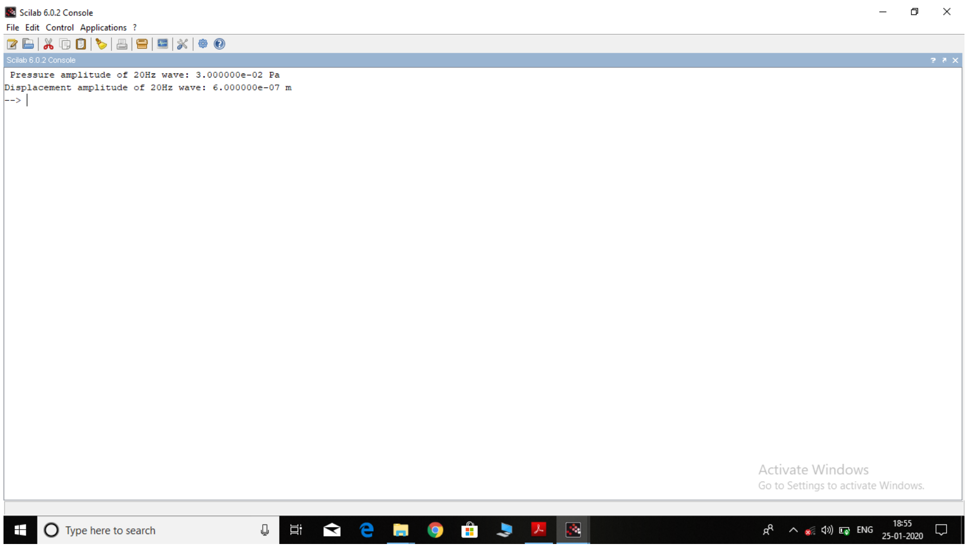

Figure 16.11: Doppler effect 3 A moving listener

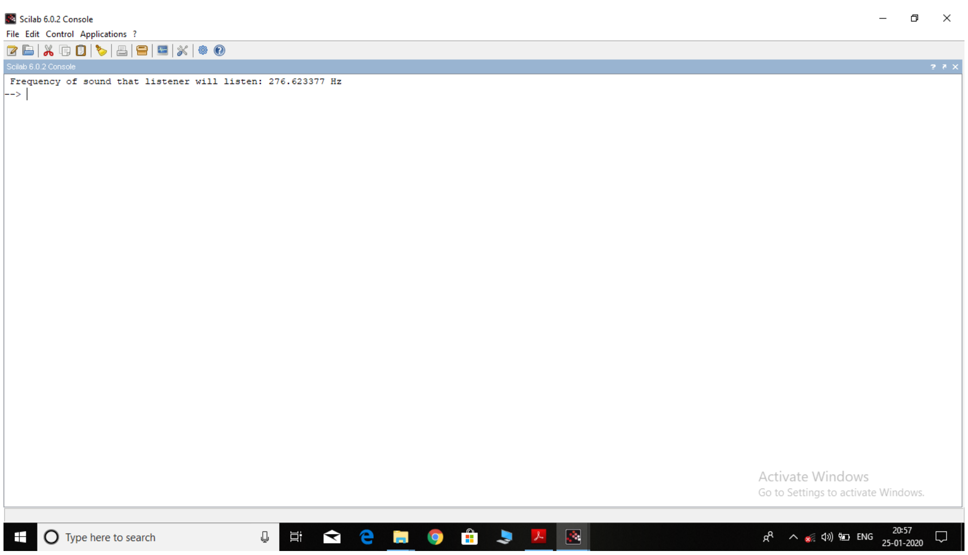

Figure 16.12: Doppler effect 4 moving source moving listener

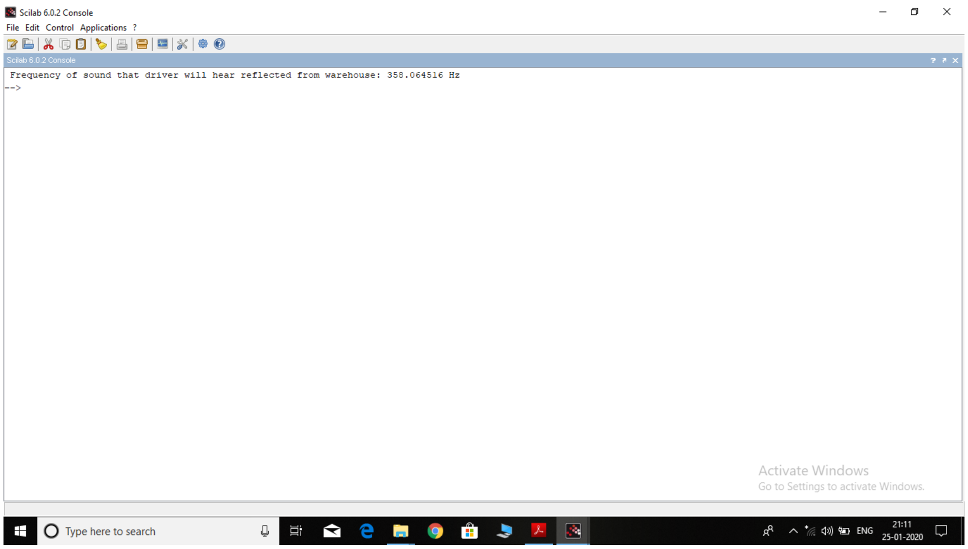

Figure 16.13: Doppler effect 5 A double doppler shift

```
//CHAPTER 16: SOUND AND HEARING
//EX 16.17: DOPPLER'S EFFECT 4: MOVING SOURCE,
    MOVING LISTENER
clc;
clear;
f_s=300; //frequency of sound wave in Hz
v=340; //speed of sound in air in m/s
v_l=15; //speed of listener in m/s
v_s=45; //speed of source in m/s
f_l=((v+v_l)/(v+v_s))*f_s; //frequency of sound that
    listener will listen in Hz
printf('Frequency of sound that listener will listen
    : %f Hz',f_l); //answer vary due to roundoff
```

Scilab code Exa 16.18 Doppler effect 5 A double doppler shift

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.18: DOPPLER'S EFFECT 5: A DOUBLE DOPPLER
    SHIFT
clc;
clear;
f_s=300; //frequency of sound wave in Hz
v=340; //speed of sound in air in m/s
v_s=-30; //speed of police car in m/s
f_w=(v/(v+v_s))*f_s //frequency reaching the
    warehouse in Hz
v_l=30; //speed of listener in m/s
f_l=((v+v_l)/v)*f_w; //frequency heard by driver in
    Hz
printf('Frequency reaching the warehouse: %f Hz',f_w
    ); //answer vary due to roundoff error
printf('\nFrequency of sound that driver will hear
    reflected from warehouse: %f Hz',f_l); //answer
    vary due to roundoff error
```

Scilab code Exa 16.19 Sonic boom from a supersonic airplane

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 16: SOUND AND HEARING
//EX 16.19: SONIC BOOM FROM A SUPERSONIC AIRPLANE
clc;
clear;
M=1.75; //Mach number
alpha=asin(1/M); //angle of shock cone
v=320; //speed of sound in air in m/s
v_s=v*M; //speed of supersonic plane in m/s
altitude=8000; //altitude of plane in m
t=altitude/(tan(alpha)*v_s); //time after which we
```

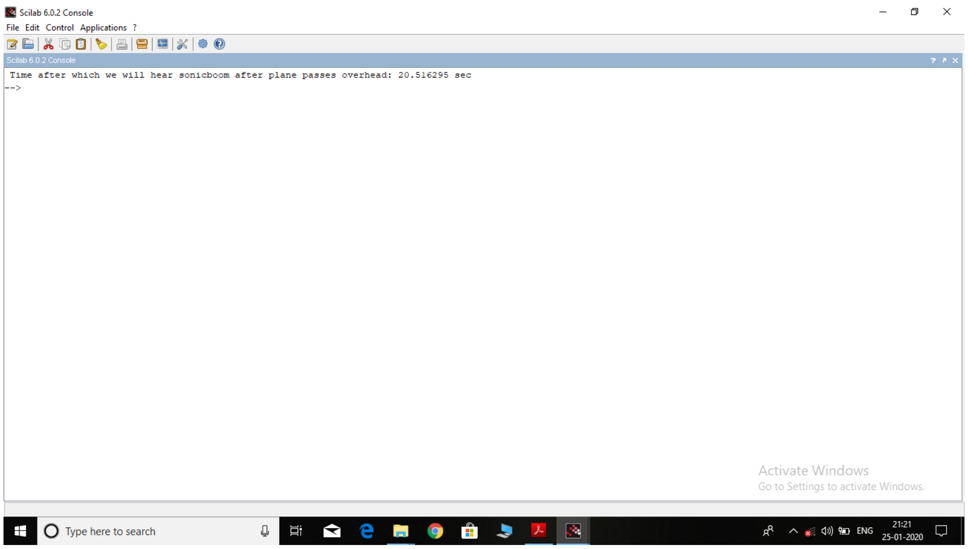

Figure 16.14: Sonic boom from a supersonic airplane

```
will hear sonicboom after plane passes overhead
 in sec
printf('Time after which we will hear sonicboom
 after plane passes overhead: %f sec',t);
```

Chapter 17

Temperature and Heat

Scilab code Exa 17.1 Body temperature

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.1: BODY TEMPERATURE
clear;
clc;
T1=32; //given temperature in fahrenheit
T2=98.6; //given temperature in fahrenheit
T1_celsius=(5/9)*(T1-32); //temperature in celsius
   Tc=(5/9)*(Tf-32)
T2_celsius=(5/9)*(T2-32); //temperature in celsius
deltaT=T2_celsius-T1_celsius;
printf('Temperature difference in celsius: %f
   celsius',deltaT);
T1_kelvin=273.15+T1_celsius; //temperature in kelvin
    (Tk=Tc+273.15)
T2_kelvin=273.15+T2_celsius; //temperature in kelvin
deltaT=T2_kelvin-T1_kelvin;
printf('\nTemperature difference in kelvin: %f
   kelvin',deltaT);
```

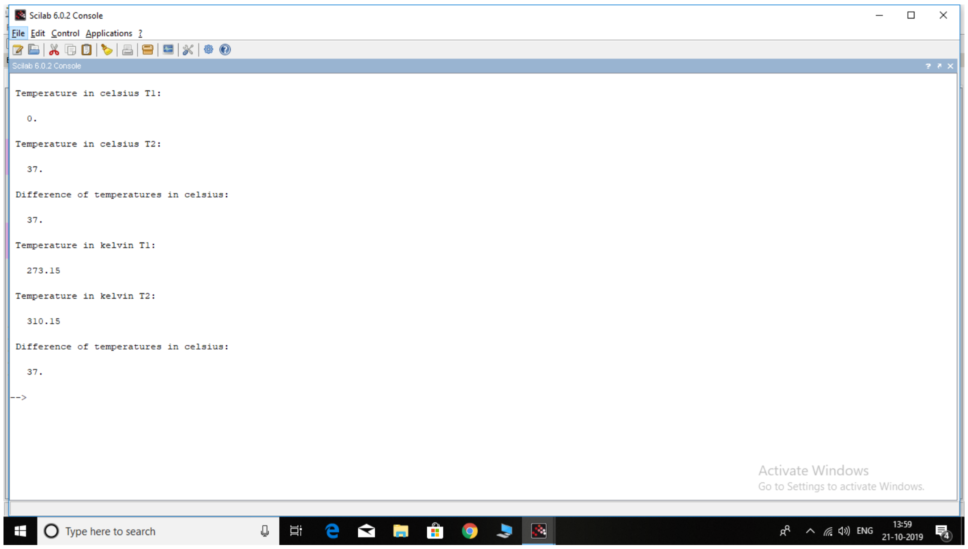

Figure 17.1: Body temperature

Scilab code Exa 17.2 Length change due to temperature change

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.2: LENGTH CHANGE DUE TO TEMPERATURE
    CHANGE
clear;
clc;
L_0=50; //initial length in meters
T_0=293; //inital temperature in kelvin
T=308; //Final temperature in kelvin
deltaT=T-T_0; //temperature difference
alpha=1.2*10^-5; //coefficient of linear expansion
    in unit per kelvin
```

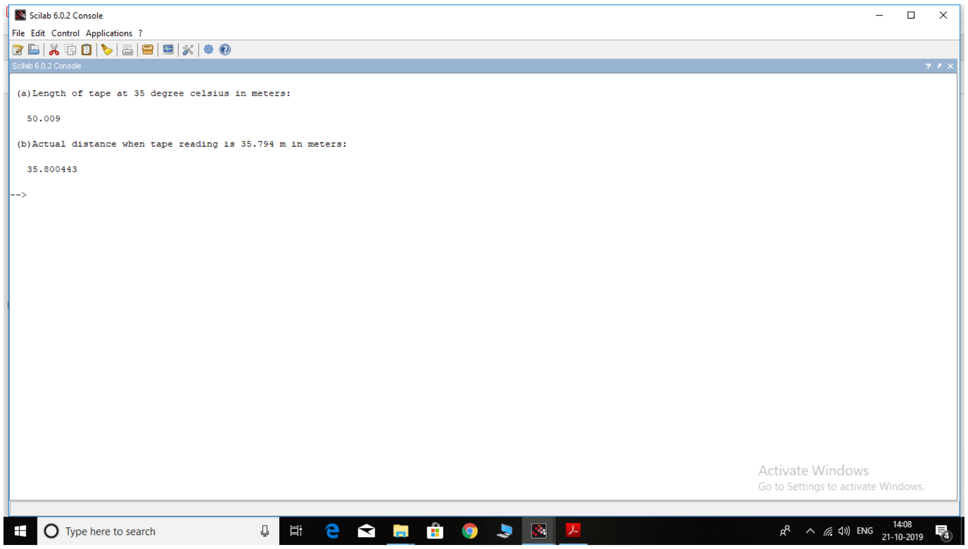

Figure 17.2: Length change due to temperature change

```
deltaL=alpha*L_0*deltaT; //change in length due to temperature change
L=L_0+deltaL;
printf('Length of the tape when the temperature is 35 degree celsius: %f m',L);
Factor=L/L_0; //true distance= factor*measured reading
measured_reading=35.794; //given measured reading in m
true_distance=Factor*measured_reading;
printf('\nActual distance: %f m',true_distance);
```

Scilab code Exa 17.3 volume change due to temperature change

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
```

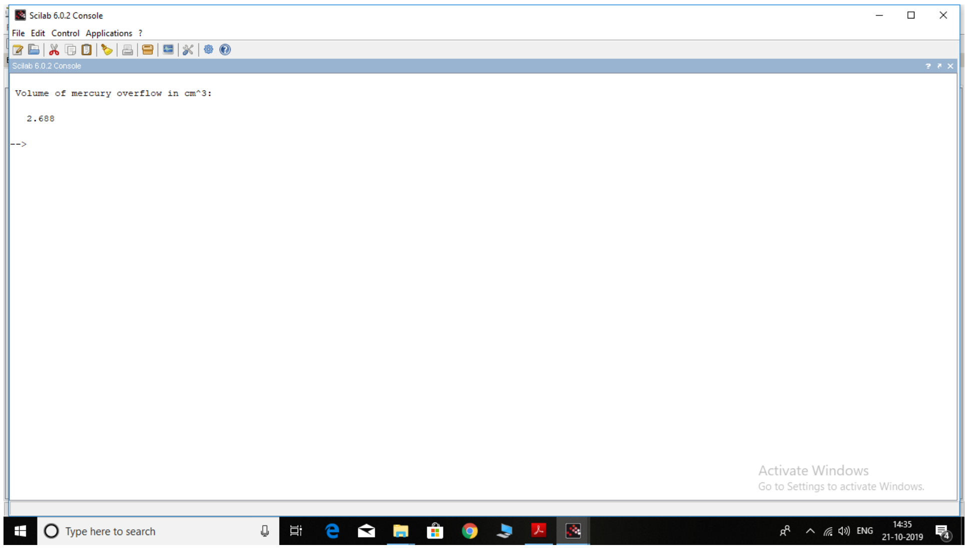

Figure 17.3: volume change due to temperature change

```
//EXAMPLE 17.3 VOLUME CHANGE DUE TO TEMPERATURE CHANGE
clear;
clc;
V_0=200; //volume in cm^3
T1=20; // initial temperature in degree celsius
T2=100; //final temperature in degree celsius
deltaT=T2-T1;
Beta_Hg=18*10^-5; //coefficient of volume expansion of mercury in K^-1
Beta_glass=1.2*10^-5; //coefficient of volume expansion of glass in K^-1
V=V_0*deltaT*(Beta_Hg-Beta_glass);
disp(V,'Volume of mercury overflow in cm^3: '); // answer vary due to roundoff error
```

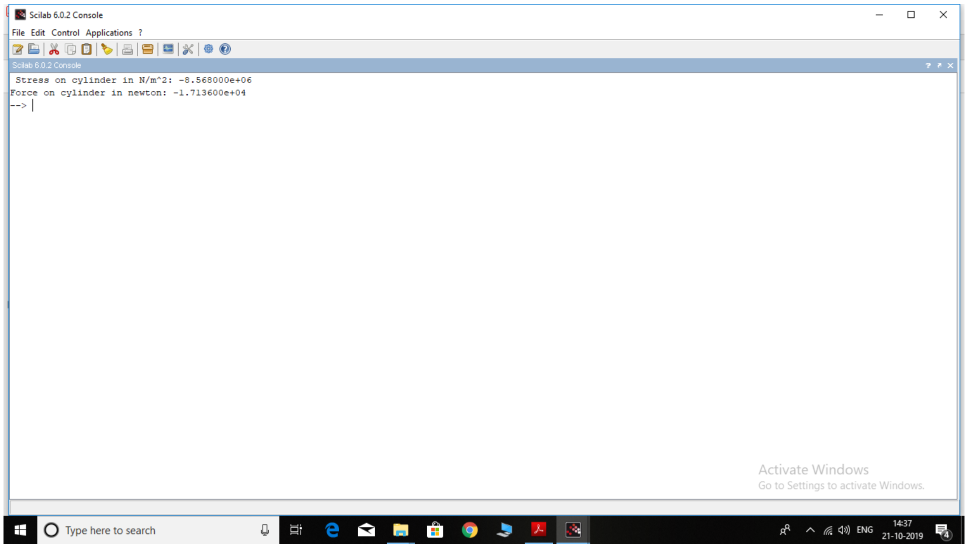

Figure 17.4: Thermal stress

Scilab code Exa 17.4 Thermal stress

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.4 THERMAL STRESS
clear;
clc;
A=20*10^-4; //Area in m^2
T1=17.2; //initial temperature in degree celsius
T2=22.3; //final temperature in degree celsius
deltaT=T2-T1; //temperature difference in degree
    celsius
Gamma_Al=7*10^10; //young's modulus of aluminium
alpha=2.4*10^-5; //coefffficient of linear expansion
    of aluminium
F_by_A=-Gamma_Al*alpha*deltaT; //magnitude of stress
     in N/m^2
F=F_by_A*A; //force in N (Force=stress*Area)
mprintf('Stress on cylinder in N/m^2: %e',F_by_A);
    //answer vary due to round off error
```

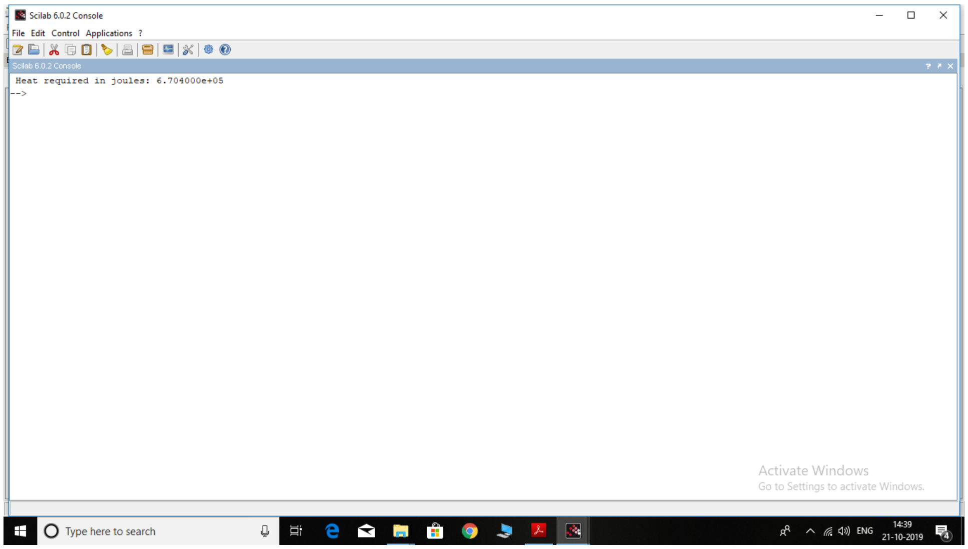

Figure 17.5: Feed a cold starve a fever

```
mprintf('\nForce on cylinder in newton: %e',F); //
    answer vary due to round off error
```

Scilab code Exa 17.5 Feed a cold starve a fever

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.5 FEED A COLD, STARVE A FEVER
clear;
clc;
m=80; //mass of body in kg
T_0=37; //initial body temperature in celsius
T=39; //final body temperature in celsius
deltaT=T-T_0; //difference in temperature in celsius
    OR kelvin
c=4190; //specific heat capacity of water in J/kg.K
```

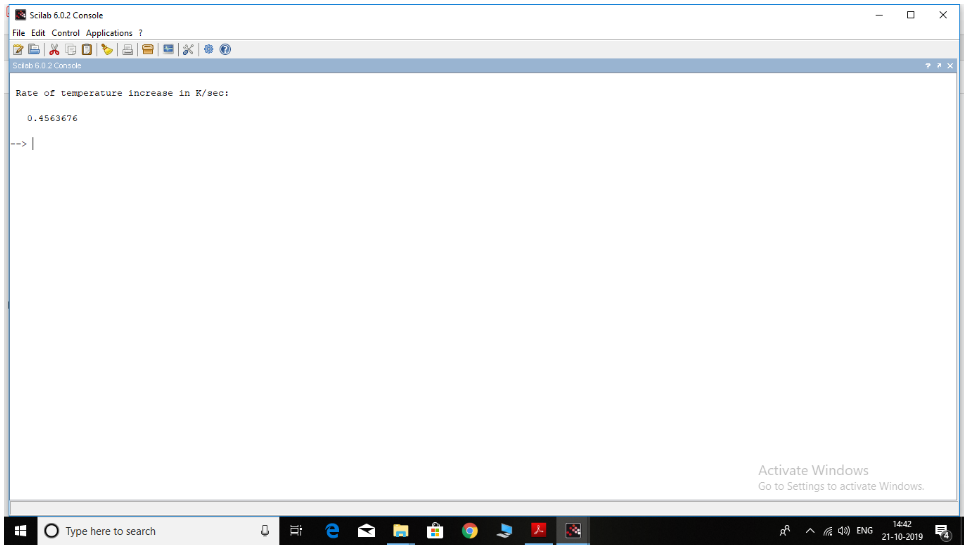

Figure 17.6: Overheating electronics

```
Q=m*c*deltaT; //heat required in J
mprintf('Heat required in joules: %e',Q);
```

Scilab code Exa 17.6 Overheating electronics

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.6 OVERHEATING ELECTRONICS
clear;
clc;
m=23*10^-3; //mass of silicon in grams
dQ_by_dT=7.4*10^-3; //rate of energy transfer in J/sec
c=705*10^-3; //specific heat capacity of silicon in J/g.K
```

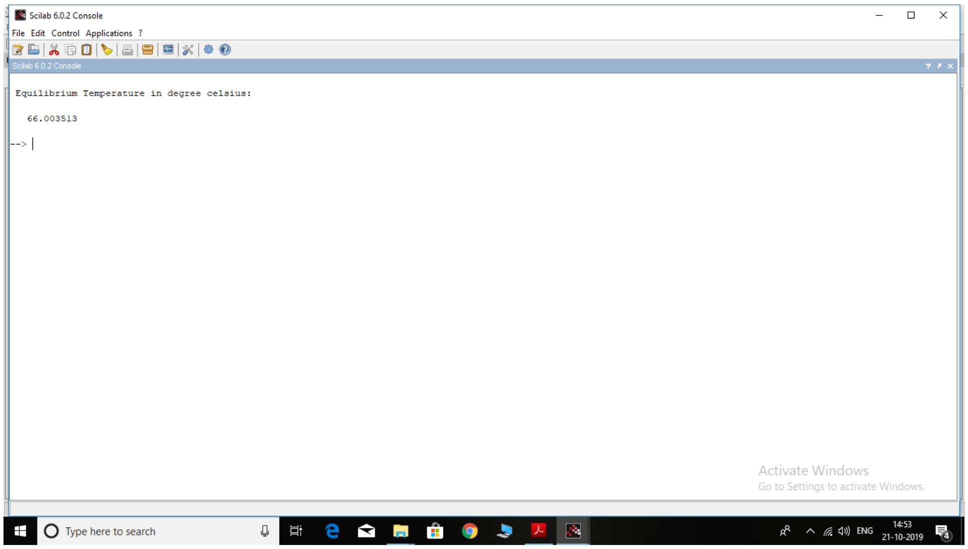

Figure 17.7: A temperature change with no phase change

```
dT_by_dt=(dQ_by_dT)/(m*c); //rate of temperature
    change in K/sec
disp(dT_by_dt,'Rate of temperature increase in K/sec
    : '); //answer vary due to roundoff error
```

Scilab code Exa 17.7 `A temperature change with no phase change`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.7: A TEMPERATURE CHANGE WITH NO PHASE
    CHANGE
clear;
clc;
m_C=0.3; //mass of coffee in kg
m_Al=0.12; //mass of aluminium cup in kg
c_W=4190; //heat capacity of water in J/kg.k
```

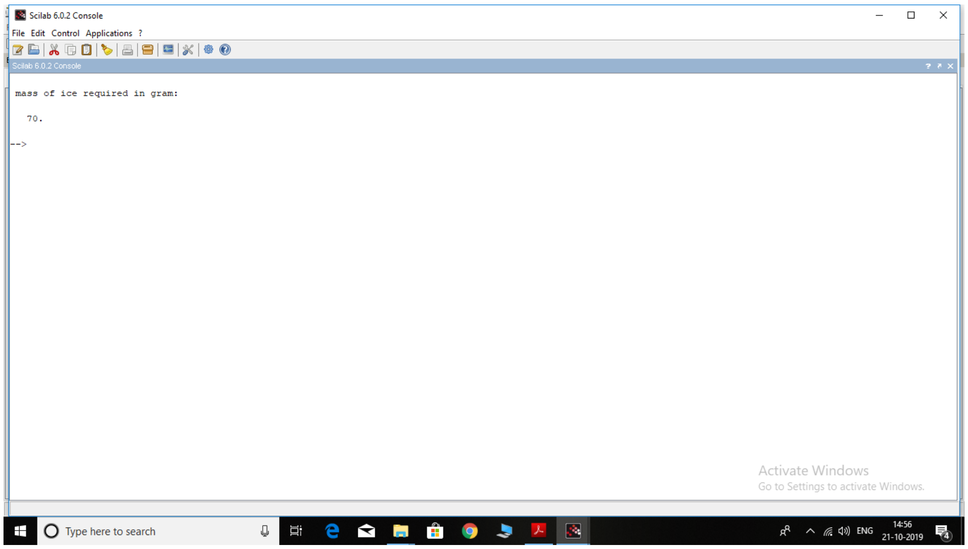

Figure 17.8: Changes in both temperature and phase

```
c_Al=910; //heat capacity of aluminium in J/kg.k
T_0C=70; //initial temperature of coffee in degree celsius
T_0Al=20; //initial temperature of aluminium cup in degree celsius
T=((m_C*c_W*T_0C)+(m_Al*c_Al*T_0Al))/(m_C*c_W+m_Al*c_Al);
disp(T,'Equilibrium Temperature in degree celsius: ');
```

Scilab code Exa 17.8 Changes in both temperature and phase

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.8 CHANGES IN BOTH TEMPERATURE AND HEAT
clear;
```

```
clc;
m_c=0.25; //mass of Omni-cola(mostly water) in kg
c_W=4190; //specific heat capacity of water in J/kg.K
c_I=2100; //specific heat capacity of ice(near 0 degree celsius) in J/kg.K
T=0; //final temperature in celsius
T=273.15+T; //final temperature in kelvin
T_0I=-20; //initial temperature of ice in celsius
T_0I=273.15+T_0I; //initial temperature of ice in kelvin
T_0C=25; //initial temperature of omni-cola in celsius;
T_0C=273.15+T_0C; //initial temperature of omni-cola in kelvin;
L_f=3.34*10^5; //latent heat of fusion of ice in J/kg
m_I=(m_c*c_W*(T_0C-T)/(c_I*(T-T_0I)+L_f))*1000; //mass of ice required in grams
disp(round(m_I),'mass of ice required in gram: ');
```

Scilab code Exa 17.9 Whats cooking

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER TEMPERATURE AND HEAT
//EXAMPLE 17.9 WHAT'S COOKING
clear;
clc;
m_W=0.1; //mass of water in kg
m_cu=2; //mass of copper pot in kg
T_0W=25; //initial temperature of water in celsius
T_0cu=150; //initial temperature of copper pot in celsius
```

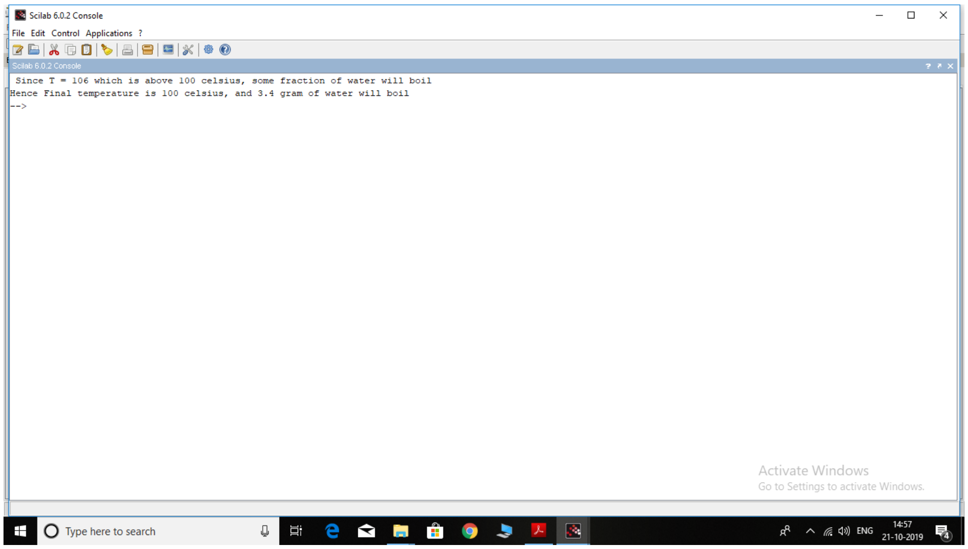

Figure 17.9: Whats cooking

```
c_W=4190; //specific heat capacity of water in J/kg.K
c_cu=390; //specific heat capacity of copper in J/kg.K
L_v=2.256*10^6; //latent heat of vapourization of water in J/kg
T=(m_W*c_W*T_0W+m_cu*c_cu*T_0cu)/(m_W*c_W+m_cu*c_cu); //final temperature in celsius
mprintf('Since T = %d which is above 100 celsius, some fraction of water will boil',T);
x=(-m_cu*c_cu*(100-T_0cu)-m_W*c_W*(100-T_0W))/(m_W*L_v); //fraction of water that will boil from energy conservation
m=x*m_W*1000; //mass of water that will boil in grams
mprintf('\nHence Final temperature is 100 celsius, and %0.1f gram of water will boil',m);
```

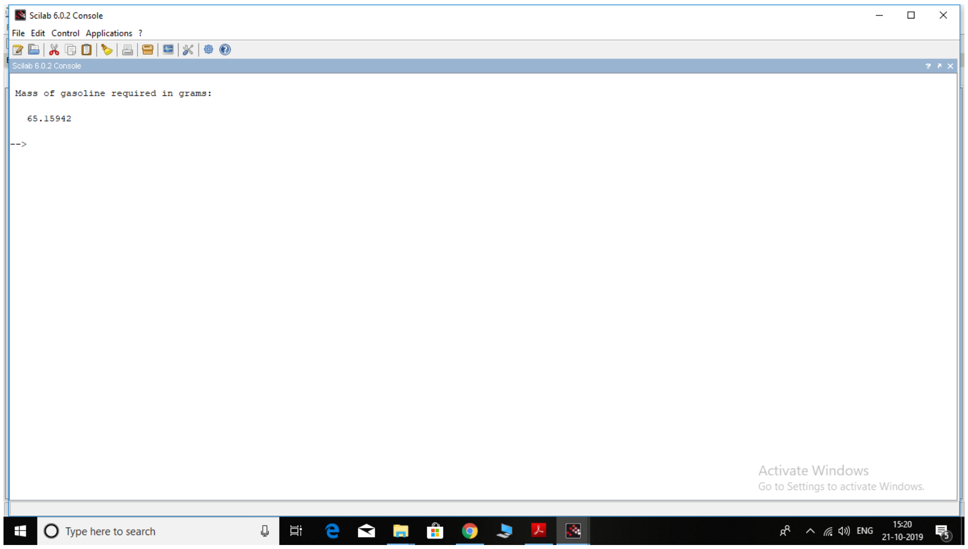

Figure 17.10: Combustion temperature change and phase change

Scilab code Exa 17.10 `Combustion temperature change and phase change`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.10: COMBUSTION,TEMPERATURE CHANGE, PHASE CHANGE
clear;
clc;
m=1; //mass of water in kg
c=4190; //specific heat capacity of water in J/kg.K
T1=20; //initial temperature of water in celsius
T2=100; //final temperature of water in celsius
Q1=m*c*(T2-T1); //heat required to change temp of water from T1 to T2
m=0.25; //mass of water in kg that will boil
L_v=2.256*10^6; //latent heat of vapourization of water in J/kg
```

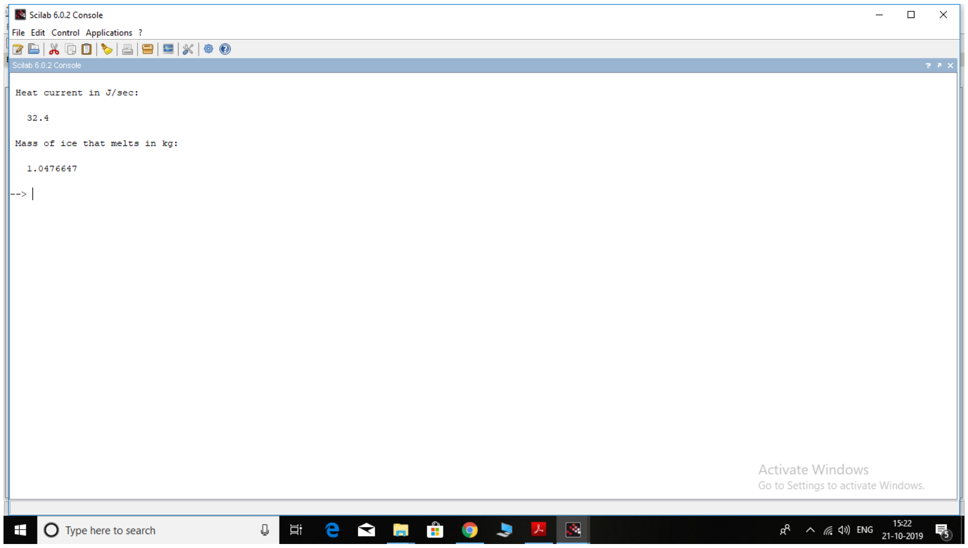

Figure 17.11: Conduction into a picnic cooler

```
Q2=m*L_v; //heat required for boiling
Q=Q1+Q2; //total heat required
Q_combus=46000; //given heat of combustion of gasoline in J/g
Q_gasoline=Q/0.3; //total heat of combustion of gasoline
M_gas=Q_gasoline/Q_combus; //mass of gasoline required
disp(M_gas,'Mass of gasoline required in grams: '); //answer vary due to roundoff error
```

Scilab code Exa 17.11 Conduction into a picnic cooler

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17: TEMPERATURE AND HEAT
//EXAMPLE 17.11 CONDUCTION INTO A PICNIC COOLER
```

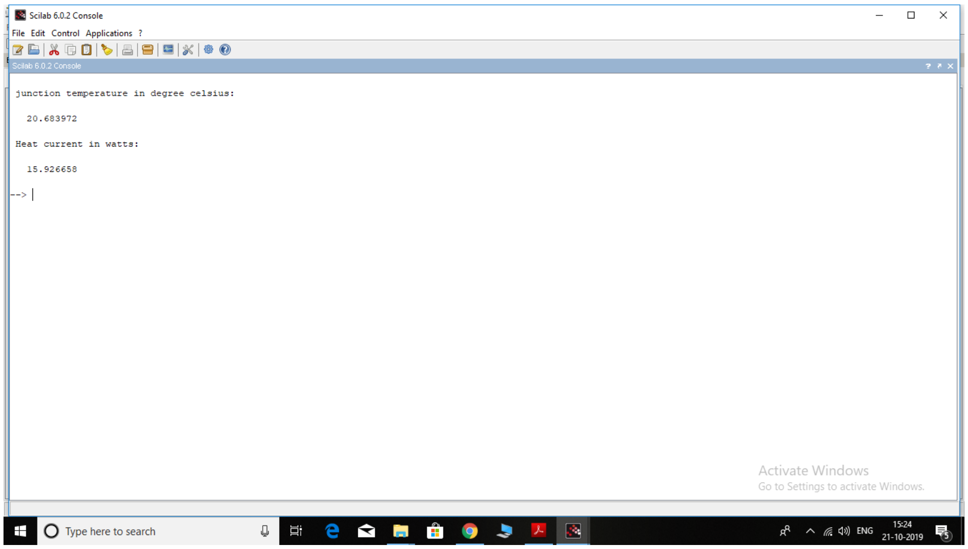

Figure 17.12: Conduction through two bars I

```
clear;
clc;
k=0.027; //heat conductivity in W/m.k
L=0.02; //length in meters
A=0.8; //area in m^2
T_H=30; //higher temperature in celsius
T_C=0; //lower temperature in celsius
H=k*A*(T_H-T_C)/L; //heat current in J/s
disp(H,'Heat current in J/sec: ');
t=10800; //time in second
Q=H*t; //heat transfer in 3 hrs
L_f=3.34*10^5; //heat of fusion in J/kg
m=Q/L_f; //mass of ice melts
disp(m,'Mass of ice that melts in kg: '); //answer
    vary due to roundoff
```

Scilab code Exa 17.12 Conduction through two bars I

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.12 CONDUCTION THROUGH TWO BARS-1
clear;
clc;
A=0.02^2; //Area of bar in m^2
L_s=0.1; //length of steel in meters
L_cu=0.2; //length of copper in meters
T_H=100; //higher temperature in celsius
T_c=0; //lower temperature in celsius
K_s=50.2; //heat conductivity of steel in W/m.K
K_cu=385; //heat conductivity of copper in W/m.K
T=((K_s/L_s)*T_H+(K_cu/L_cu)*T_c)/((K_s/L_s)+(K_cu/
    L_cu)); //junction temperature in degree celsius
H_s=K_s*A*(T_H-T)/L_s; //total heat current in watts
H_cu=K_cu*A*(T-T_c)/L_cu; //total heat current in
    watts
disp(T,'junction temperature in degree celsius: ');
    //answer vary due to roundoff
disp(H_s,'Heat current in watts: '); //since (H_s=
    H_cu) //answer vary due to roundoff
```

Scilab code Exa 17.13 Conduction through two bars II

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.13 CONDUCTION THROUGH TWO BARS-II
clear;
clc;
L=0.02; //length of each bar in m
A=L*L; //area of cross section of each bar in m^2
```

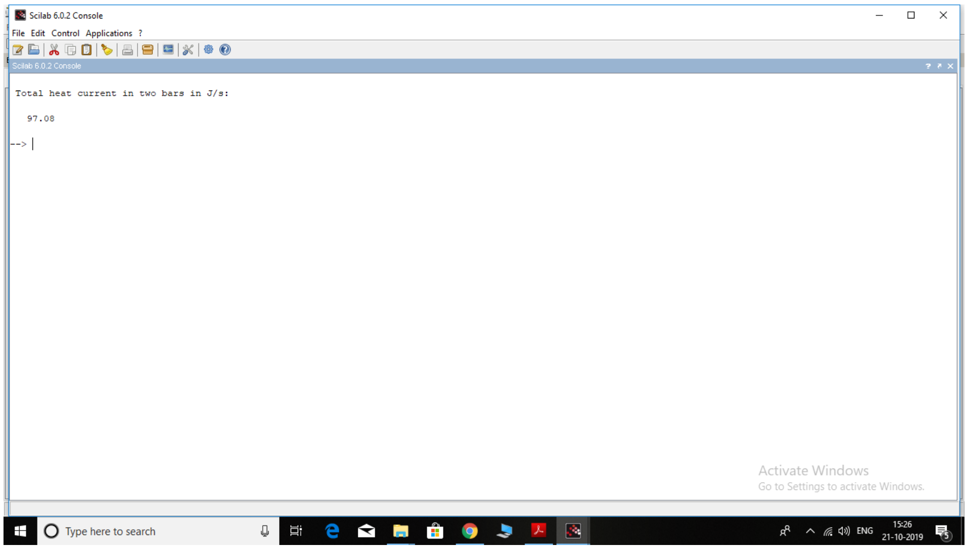

Figure 17.13: Conduction through two bars II

```
T_H=100; //higher temperature in celsius
T_c=0; //lower temperature in celsius
L_s=0.1; //length of steel bar in meters
L_cu=0.2; //length of copper bar in meters
K_s= 50.2; //thermal conductivity of steel in W/m.K
K_cu=385; //thermal conductivity of copper in W/m.K
H_s=K_s*A*(T_H-T_c)/L_s; //heat current in steel in
    J/s
H_cu=K_cu*A*(T_H-T_c)/L_cu; //heat current in cupper
    in J/s
H=H_s+H_cu; //total heat current in J/s
disp(H,'Total heat current in two bars in J/s: ');
```

Scilab code Exa 17.14 Heat transfer by radiation

```
//OS: WINDOWS 10, SCILAB-6.0.2
```

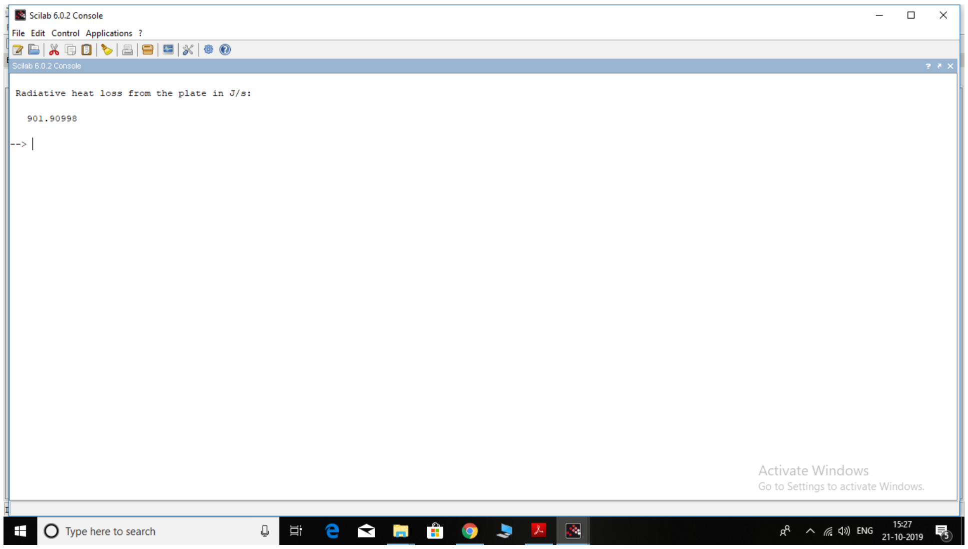

Figure 17.14: Heat transfer by radiation

```
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.14 HEAT TRANSFER BY RADIATION
clc;
clear;
L=0.1; //length of plate in meters
T=1073; //temperature of plate in K
e=0.6; //emissivity of plate
A=2*(L^2); //total surface area of plate in m^2;
sigma=5.67*10^-8; //stefan boltzmann constant in W/m^2.K^4
H=A*e*sigma*T^4; //radiative heat loss in J/sec
disp(H,'Radiative heat loss from the plate in J/s: '); //answer vary due to roundoff error
```

Scilab code Exa 17.15 `Radiation from the human body`

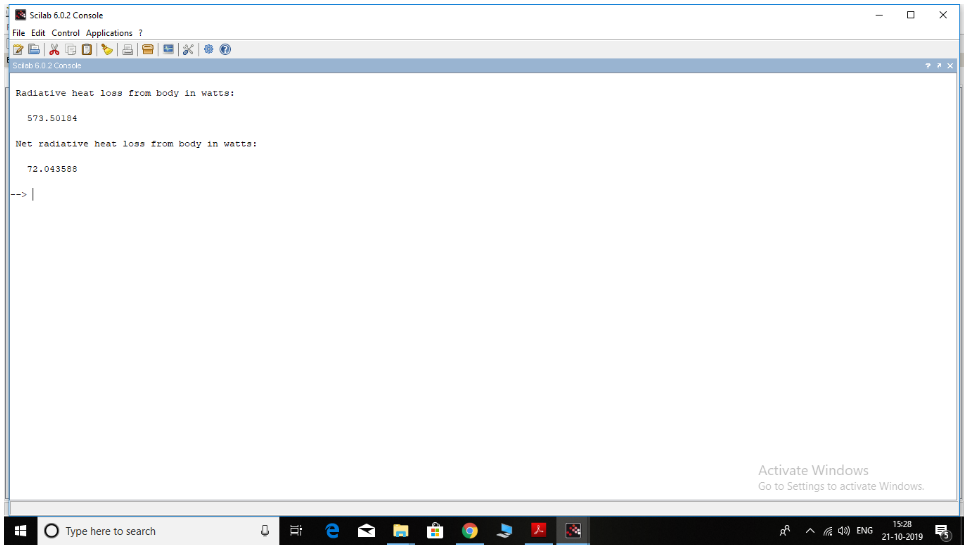

Figure 17.15: Radiation from the human body

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 17 TEMPERATURE AND HEAT
//EXAMPLE 17.15 RADIATION FROM HUMAN BODY
clear;
clc;
A=1.2; //surface area of body in m^2
T=303; //temperature of body in K
T_s=293; //surrounding temperature in K
e=1; // emissivity of human body
sigma=5.67*10^-8; //stefan boltzmann constant in W/m^2.K^4
H=A*e*sigma*T^4; //radiative heat loss from body in watts
H_net=A*e*sigma*(T^4-T_s^4); //net radiative heat loss from body in watts
disp(H,'Radiative heat loss from body in watts: '); //answer vary due to roundoff error
disp(H_net,'Net radiative heat loss from body in watts: ');
```

Chapter 18

Thermal properties of matter

Scilab code Exa 18.1 Volume of an ideal gas at STP

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 18: THERMAL PROPERTIES OF MATTER
//EX 18.1: VOLUME OF AN IDEAL GAS AT STP
clc;
clear;
T=273.15; //given temperature of ideal gas in K
P=1.013*10^5; //given pressure of ideal gas in Pa
n=1; //given no. of moles of ideal gas
R=8.314; //universal gas constant in J/K.mol
V=(n*R*T)/P; //volume of ideal gas at STP in m^3
V=V*1000; //volume of ideal gas at STP in litres
printf('Volume of an ideal gas at STP: %f litres',V)
    ;
```

Scilab code Exa 18.5 Atomic and molecular mass

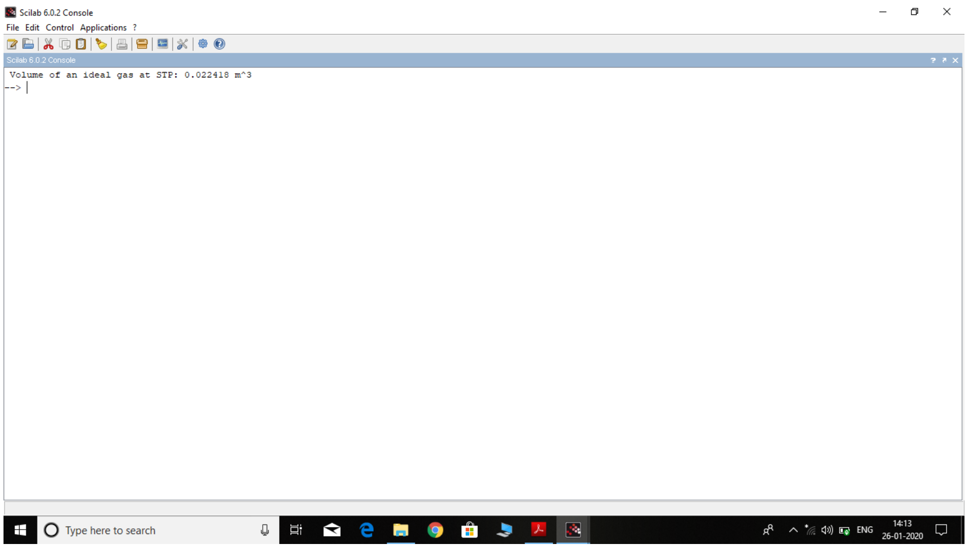

Figure 18.1: Volume of an ideal gas at STP

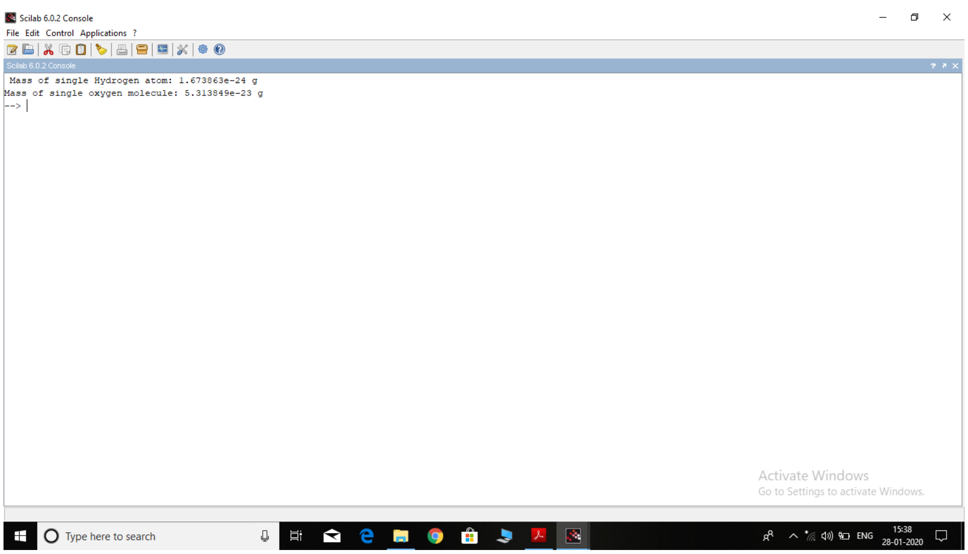

Figure 18.2: Atomic and molecular mass

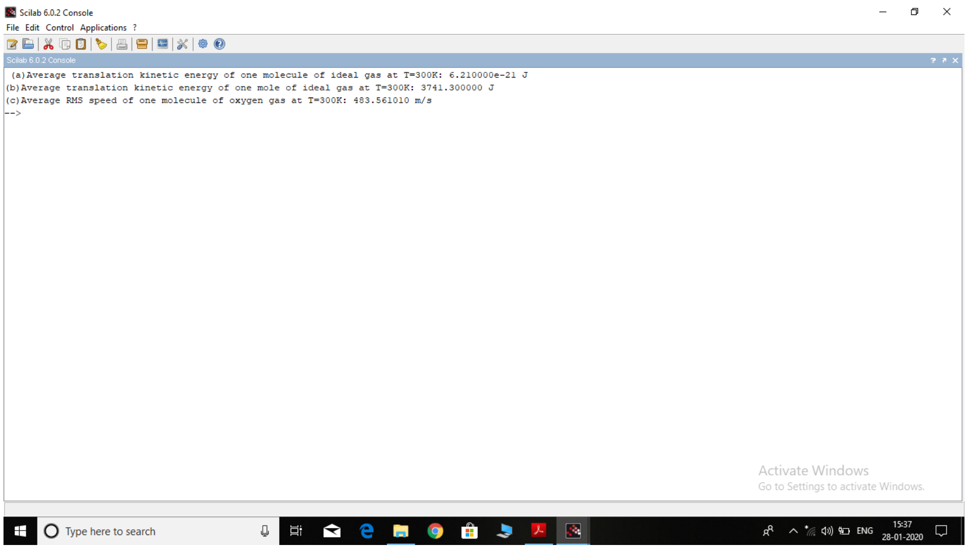

Figure 18.3: Molecular kinetic energy and Vrms

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 18: THERMAL PROPERTIES OF MATTER
//EX 18.5: ATOMIC AND MOLECULAR MASS
clc;
clear;
Na=6.022*10^23; //avogadro's number
M_H=1.008; //molar mass of Hydrogen atom in g/mol
m_H=M_H/Na; //mass of single Hydrogen atom in g
M_o2=32; //molar mass of Oxygen molecule in g/mol
m_o2=M_o2/Na; //mass of single Oxygen molecule in g
printf('Mass of single Hydrogen atom: %e g',m_H);
printf('\nMass of single oxygen molecule: %e g',m_o2
    );
```

Scilab code Exa 18.6 `Molecular kinetic energy and Vrms`

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 18: THERMAL PROPERTIES OF MATTER
//EX 18.6: MOLECULAR KINETIC ENERGY AND Vrms
clc;
clear;
T=300; //temperature of ideal gas molecule in kelvin
k=1.38*10^-23; //boltzman constant in J/K
Kavg_molecule=(3/2)*k*T; //Average translation
    kinetic energy of one molecule in J
n=1; //no. of moles of ideal gas
R=8.314; //universal gas constant in J/K.mol
Ktr=(3/2)*n*R*T; //average translation kinetic
    energy of 1 mole of ideal gas in J
M=32*10^-3; //Molar mass of oxygen molecule in kg/
    mol
v_rms=sqrt((3*R*T)/M); //rms speed of oxygen
    molecule in m/s (as calculated in second method
    in book: Vrms=sqrt(3RT/M))
printf('(a)Average translation kinetic energy of one
     molecule of ideal gas at T=300K: %e J',
    Kavg_molecule);
printf('\n(b)Average translation kinetic energy of
    one mole of ideal gas at T=300K: %f J',Ktr); //
    answer vary due to roundoff error
printf('\n(c)Average RMS speed of one molecule of
    oxygen gas at T=300K: %f m/s',v_rms); //answer
    vary due to roundoff error
```

Scilab code Exa 18.7 Calculating rms and average speeds

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 18: THERMAL PROPERTIES OF MATTER
//EX 18.7: CALCULATING RMS AND AVERAGE SPEEDS
```

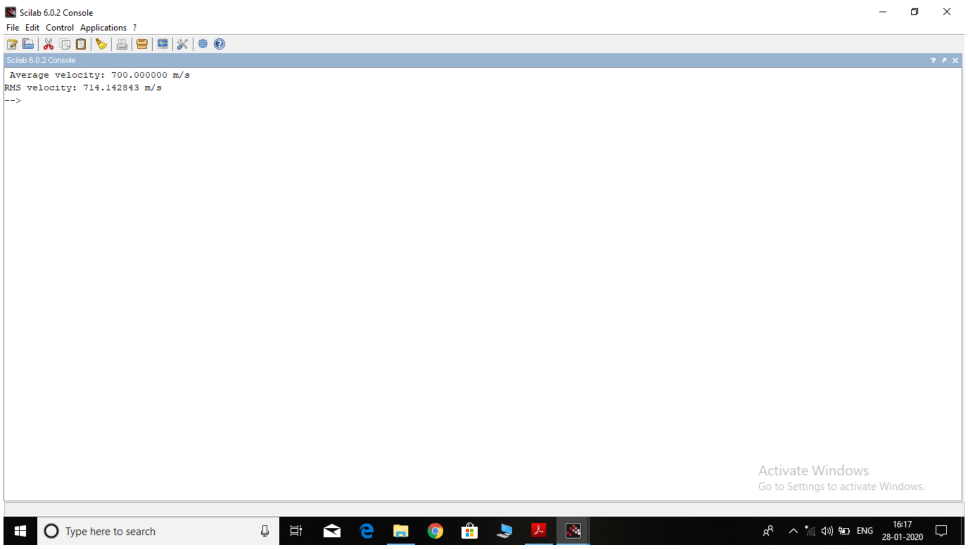

Figure 18.4: Calculating rms and average speeds

```
clc;
clear;
v=[500 600 700 800 900]; //given velocities of molecules in m/s
v_av=mean(v); //average velocity in m/s
printf('Average velocity: %f m/s',v_av);
v_square_av=mean(v^2); //mean square velocity in m/s
v_rms=sqrt(v_square_av); //RMS velocity in m/s
printf('\nRMS velocity: %f m/s',v_rms); //answer vary due to round off error
```

Scilab code Exa 18.8 Calculating mean free path

```
//OS: WINDOWS 10, SCILAB-6.0.2
//CHAPTER 18: THERMAL PROPERTIES OF MATTER
//EX 18.8: CALCULATING MEAN FREE PATH
```

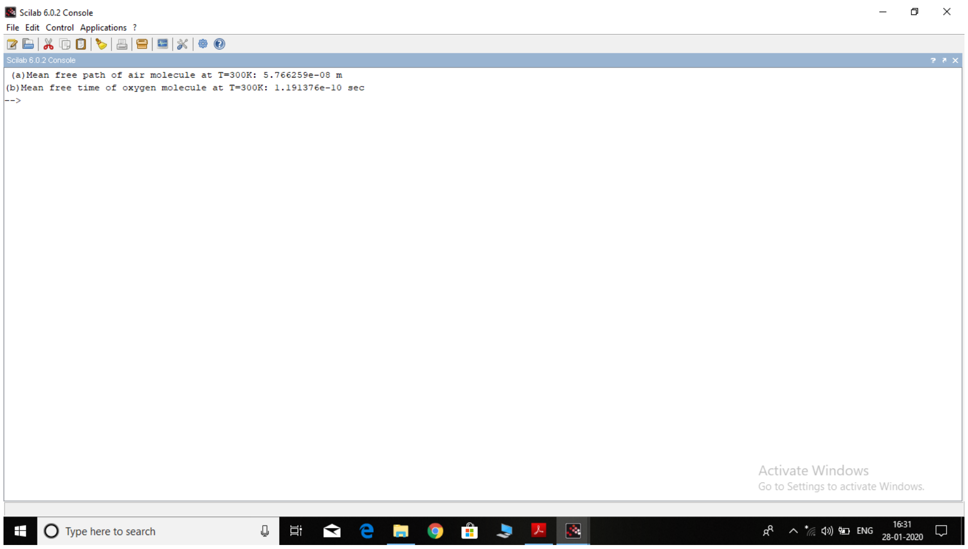

Figure 18.5: Calculating mean free path

```
clc;
clear;
k=1.38*10^-23; //boltzman constant in J/K
T=300; //temperature of air in K
p=1.01*10^5; //pressure of air in Pa
r=2*10^-10; //radius of air molecule in m
lambda=(k*T)/(4*%pi*sqrt(2)*(r^2)*p); //mean free path of air molecule in m
v=484; //RMS speed of air molecule at T=300K (from Ex18.6)
t_mean=lambda/v; //mean free time of oxygen molecule in sec
printf('(a)Mean free path of air molecule at T=300K: %e m',lambda); //answer vary due to roundoff error
printf('\n(b)Mean free time of oxygen molecule at T=300K: %e sec',t_mean); //answer vary due to roundoff error
```