# Matlab-Code Demonstrations

## Linear solver: cmd

```
% n x m matrix
K=[1 2 3;4 5 6]
% n x n submatrix
k=K(1:2,2:3)
%  right hand side (make sure its a column vector)
f=[10;11]
% solve k*u=f for u
u=fsolvelineqs(k,f)
% check solution
u_check=linsolve(k,f)
% second check
f_check=k*u
```

### fsolvelineqs.m

```
function [sol] = fsolvelineqs(matrix,righthandside)
%
% computation of the solution of a linear system of equations
sol=matrix\righthandside;
```

## Plot 2D: plt_trig_demo.m

```
% open figure window
figure(1)
% clear previous figures
clf
% t-values
t=0:pi/20:2*pi;
% plot sin, plotstyle: dashdot-red-stars
plot(t,sin(t),'-.r*')
% hold on necessary for multiple plots in one window
 hold on % leave out first and show what happens
% plot -sin, plotstyle: doubledash-magenta-circles
plot(t,(sin(t-pi/2)),'--mo')
% plot -cos, plotstyle:
x_line = t;
y_line = zeros(size(t,1),size(t,2));
% x_line = [min(t) max(t)];
```

```
% y_line = [0 0]
line(x_line,y_line,'linestyle',':','color','b','marker','s')
% hold off necessary so that possible following plots appear in new window
 hold off % leave out first and show what happens
% plot(t,2*sin(t))
% close all
```

## Solution: t_rotating_cantilever.m

```
clear all; clc; close all;
%--- Geometrical Parameter ---
% Height of vertical rod
H = 10;
% Length of cantilever arm
L = 5;
% Radius of sphere
R = 1;
% Angle between vertical rod and cantilever arm
alpha = 90*pi/180;
% Series of angle of rotation
phi = linspace(0,2*pi,100);
%--- Basic Coordinates ---
% Uniform sphere (r=1)
[xk yk zk] = sphere;
% Scaling of sphere by R
xk = R.*xk;
yk = R.*yk;
zk = R.*zk;
% Points of veritcal rod's ends
P1 = [0 ; 0 ; 0];
P2 = P1 + [0 ; 0 ; H];
% Point of sphere's center in inital position
P30 = [ L * sin(alpha) ; 0 ; -L * cos(alpha)];
%--- Plot Procedure ---
% Create new figure with number 1
figure(1)
% Limits of axis
axis([-5  5 -5 5 0 15])
% Equal aspect ratio to all axis
axis equal
% Show grid
grid on
```

```
% Plot of vertical rod
line([P1(1) P2(1)],[P1(2) P2(2)],[P1(3) P2(3)] ,...
     'linewidth', 2 , 'color' , 'k');
%--- Begin of Animation ---
for istep = 1 : size(phi,2)
  % Delete previous plot objects if necessary
  if istep > 1
      % Sphere
      delete(id_kugel)
      % cantilever arm
      delete(id_line)
  end
  % Rotary matrix
  R = [ cos(phi(istep)) -sin(phi(istep)) 0;
        sin(phi(istep))  cos(phi(istep)) 0;
            0                  0         1]
  % Actual point of sphere's center
  P3 = P1 + P2 + R*P30;
  % Plot of sphere
  id_kugel = surface(P3(1)+xk,P3(2)+yk,P3(3)+zk);
  % Hold current graph
  hold on
  % Plot of cantilever arm
  id_line = line([P2(1) P3(1)],[P2(2) P3(2)],[P2(3) P3(3)] ,...
                 'linewidth', 2 , 'color' , 'k');
  % Update of screen
  drawnow
end
```

## Solution: t_rotating_cantilever_with_function.m

```
clear all; clc; close all;
%--- Geometrical Parameter ---
% Height of vertical rod
H = 10;
% Length of cantilever arm
L = 5;
% Radius of sphere
R = 1;
% Angle between vertical rod and cantilever arm
alpha = 90*pi/180;
% Series of angle of rotation
```

```
phi = linspace(0,2*pi,300);
%--- Basic Coordinates ---
% Uniform sphere (r=1)
[xk yk zk] = sphere;
% Scaling of sphere by R
xk = R.*xk;
yk = R.*yk;
zk = R.*zk;
% Points of veritcal rod's ends
P1 = [0 ; 0 ; 0];
P2 = P1 + [0 ; 0 ; H];
% Point of sphere's center in inital position
P30 = [ L * sin(alpha) ; 0 ; -L * cos(alpha)];
%--- Plot Procedure ---
% Create new figure with number 1
figure(1)
% Limits of axis
axis([-5  5 -5 5 0 15])
% Equal aspect ratio to all axis
axis equal
% Show grid
grid on
% Plot of vertical rod
line([P1(1) P2(1)],[P1(2) P2(2)],[P1(3) P2(3)] ,...
  'linewidth', 2 , 'color' , 'k');
%--- Begin of Animation ---
for istep = 1 : size(phi,2)
  % Delete previous plot objects if necessary
  if istep > 1
      % Sphere
      delete(id_kugel)
      % cantilever arm
      delete(id_line)
  end
  %--- Using of new function frotation_z ---
  % Actual point of sphere's center
  P3 = P1 + P2 + frotation_z(phi(istep),P30);
  %--------------------------------------------
  % Plot of sphere
  id_kugel = surface(P3(1)+xk,P3(2)+yk,P3(3)+zk);
  % Hold current graph
  hold on
  % Plot of cantilever arm
```

```
  id_line = line([P2(1) P3(1)],[P2(2) P3(2)],[P2(3) P3(3)] ,...
    'linewidth', 2 , 'color' , 'k');
  % Update of screen
  drawnow
end
```

### t_frotation_z.m

```
function [xyz_new] = rotation_z(phi,xyz_old)
% Rotation of a point with the coordinates 'xyz'
% around the z-axis with angle 'phi'
%
% input:  phi     [1x1] ... angle
%         xyz_old [3x1] ... vector of coordinates
% output: xyz_new [3x1] ... vector of rotated coordinates

% Rotary matrix
R = [ cos(phi) -sin(phi) 0;
      sin(phi)  cos(phi) 0;
          0        0     1];
% Compute rotated coordinates
xyz_new = R * xyz_old;
```