Central Impact Program

Written by: Gregg Zappone

MET 4501 (Engineering Computations using Matlab)
Tuesday, Thursday; 6:00pm

Professor Simin Nasseri

April 24, 2012

Impact

Impact or the collision between two bodies occurs in a very short time while the two bodies exert large forces on each other. There are two types of impact: central impact and oblique impact. In this section, we focus on central impact.

Central impact occurs when the direction of motion of the mass centers is along a line passing through the mass center of the objects. This line is called the line of impact (Figure 1-1).

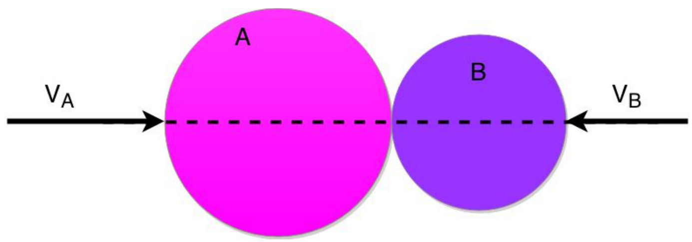

Figure 1-1: Central impact.

Here, we work on a MATLAB program which is specifically designed for the problem shown in Figure 1-2. A sack A with a certain mass is released from rest. This sack can be released at any height with respect to the line of impact. Angle θ is the angle between the line passing through the center of the sack (the rope attached to the sack) and the vertical line. After falling, it strikes chest B which has a known mass. If the coefficient of restitution between the sack and the chest is given, we would like to determine the velocities of the sack before and after the impact, the velocity of the chest after the impact and also the loss of energy during collision.

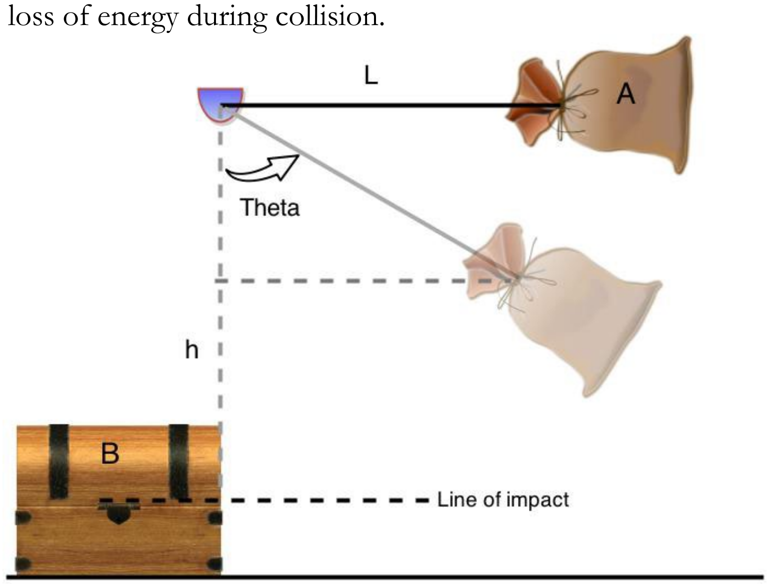

Figure 1-2: The impact problem for which the program central_impact is designed. This image will be shown to the user when the function is run.

Equations:

We first use the conversation of energy to obtain the velocity of the sack right before striking the chest. The line of impact can be considered as the datum. Therefore:

$$KE_1 + PE_1 = KE_2 + PE_2$$

$$0 + m_A gh = \frac{1}{2} m_A V_A^2 + 0 = g\left(L - L\cos(\theta)\right) = \frac{V_A^2}{2}$$

$$V_A = \sqrt{2g\left(L - L\cos(\theta)\right)}$$

Where h is the height of the sack when $0 < \theta \leq 90$ and it is maximum when θ=90. The velocity V_A is the velocity of the sack right before striking the chest and we name it V_{A1}. Conversation of momentum requires that we have:

$$m_A V_{A1} + m_B V_{B1} = m_A V_{A2} + m_B V_{B2}$$

The chest is at rest before the impact, so $V_{B1}=0$. Hence:

$$(1)\ \ m_A V_{A1} = m_A V_{A2} + m_B V_{B2}$$

We also use the coefficient of restitution, to obtain the velocities of the sack and the chest after the impact.

$$(2)\ \ e = \frac{V_{B2} - V_{A2}}{V_{A1} - V_{B1}} = \frac{V_{B2} - V_{A2}}{V_{A1}}$$

Considering equations (1) and (2), we can obtain the velocities V_{A2}, and V_{B2} by rewriting them in matrix format:

$$m_A V_{A2} + m_B V_{B2} = m_A V_{A1}$$

$$V_{B2} - V_{A2} = V_{A1} e$$

or:

$$\begin{bmatrix} m_A & m_B \\ -1 & 1 \end{bmatrix} \begin{bmatrix} V_{A2} \\ V_{B2} \end{bmatrix} = \begin{bmatrix} m_A V_{A1} \\ e V_{A1} \end{bmatrix}$$

Rewriting this matrix equation as AX=B, the solution of this system of equations can be obtained by using X=A\B as explained in Chapter 6.

Input Parameters:

The required input parameters are:
(a) the Coefficient of restitution, (b) an angle of release (theta) between 0 and 90 degrees, (c) the weight or mass of object A, and (d) the weight or mass of object B.

Output Parameters:

The purpose of the central_impact program is to calculate and output:

(a) the velocity of an object (sack A) just before central impact with an object at rest (chest B), (b) the velocity of sack A just after impact, (c) the velocity of chest B just after impact, and (d) the energy lost due to the impact.

<u>Program Operation:</u>

See Figure 1-3 for a flowchart of the program's sequential logic. The program should calculate the velocities of the objects after the impact and also the energy which is lost. When the function is run, the figure of the problem is shown to the user (Figure 1-2). Then user is asked to enter the input parameters one by one. This code does fool proofing by checking if the parameters are inputted correctly. For example mass of each object should be positive and the coefficient of restitution should be between zero and one.

After finding the velocity of the sack right before striking the chest, the matrix of A and the vector of B are formed (as explained before) and the vector of unknowns is formed using the Gaussian elimination. This program works in two unit systems and the user receives the results with appropriate units for each case.

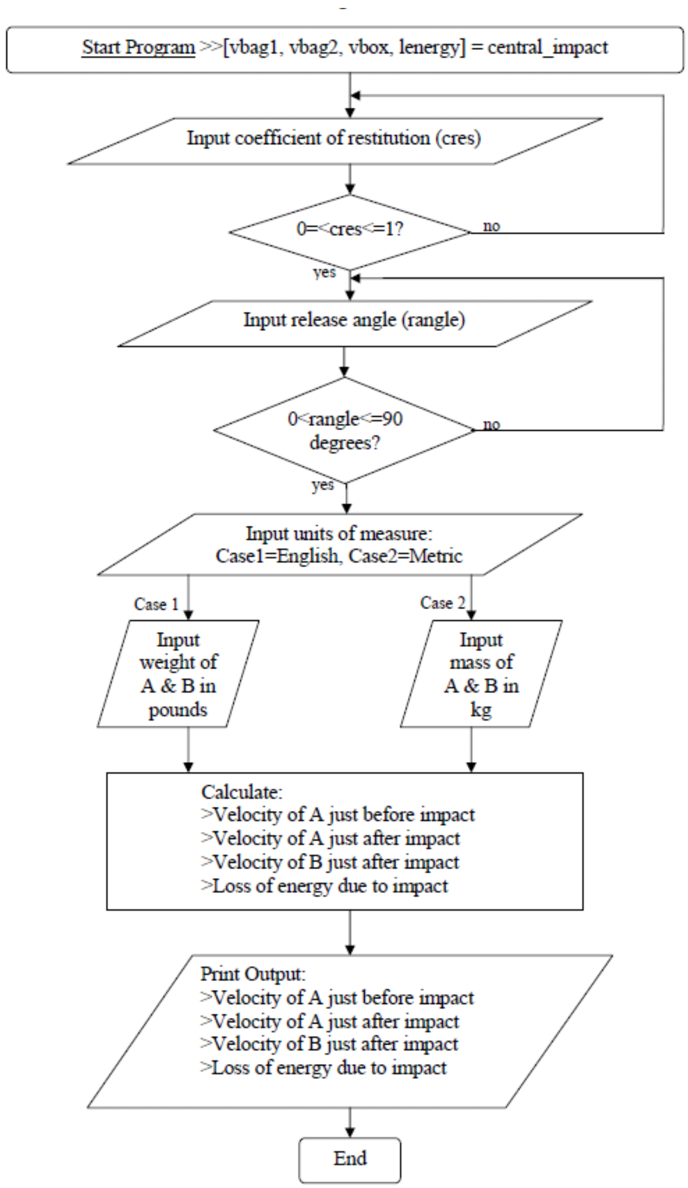

Figure 1-3: Flowchart of the program central_impact.

MATLAB Code:

```
function [vsack1, vsack2, vchest, lenergy] = central_impact()
%This function is for a central impact problem. Input
%parameters are: the angle of release of a sack tied by a
%3 foot rope %above a chest, the weight of the sack and
%the chest it strikes, and the coefficient of restitution,
%it calculates the speed of the sack just before and just
%after impact, the speed of the chest after impact, and
%the energy lost due to the impact. Written Gregg Zappone,
%modified by Simin Nasseri

clc
help central_impact
```

```
format compact

fprintf('Refer to the given figure. \n\n')
imshow('central_impact.jpg','InitialMagnification',80)
cres=input('Input the coefficient of restitution (MUST BE >=0 and <=1): ');
while cres<0 || cres>1
    cres=input('Input the coefficient of restitution (MUST BE >=0 and <=1):
');
end
rangle=input('Input the release angle in degrees (MUST BE >0 AND <=90
DEGREES): ');
while  rangle<=0 || rangle>90
    rangle=input('Input the release angle in degrees (MUST BE >0 AND <=90
DEGREES): ');
end

casenum=input('If entering data in English units, enter "1", if in Metric
enter "2": ');
while casenum~=1 && casenum~=2
    casenum=input('Invalid entry. Enter "1" or "2": ');
end

if casenum==1
    fprintf('\n')
    mA=input('Input the weight of the sack IN POUNDS: ');
    while mA<=0
        mA=input('Invalid entry. Enter a value >0: ');
    end
    mB=input('Input the weight of the chest IN POUNDS: ');
    while mB<=0
        mB=input('Invalid entry. Enter a value >0: ');
    end
    L=input('Please enter the length of the rope in FEET:  ') ;
    %velocity of sack just before impact
    vsack1=sqrt(mA*(L-L*cosd(rangle))/(.5*mA/32.2));

    %solving 2 equations with 2 unknowns
    A=[mA/32.2, mB/32.2; -1, 1]; %Matrix of coefficients
    B=[mA*vsack1/32.2; cres*vsack1];  %Vector of constants
    X=A\B;
    vsack2=X(1); %velocity of sack after impact
    vchest =X(2);  %velocity of chest after impact

    %Calculate loss of energy during collision
    lenergy=.5*mB/32.2*vchest^2+.5*mA/32.2*vsack2^2-.5*mA/32.2*vsack1^2;

    fprintf('The velocity of the sack just before impact is %.2f f/s. \n',
vsack1)
    fprintf('The velocity of the sack just after impact is %.2f ft/s. \n',
vsack2)
    fprintf('The velocity of the chest just after impact is %.2f ft/s. \n',
vchest)
    fprintf('The loss of energy during collision is %.2f ft.lb. \n', lenergy)

else
    mA=input('\nInput the mass of the sack IN KILOGRAMS: ');
    while mA<=0
```

```
        mA=input('Invalid entry. Enter a value >0: ');
    end
    mB=input('Input the mass of the chest IN KILOGRAMS: ');
    while mB<=0
        mB=input('Invalid entry. Enter a value >0: ');
    end
    L=input('Please enter the length of the rope in METER') ;
    %velocity of sack just before impact
    vsack1=sqrt((mA*9.81)*(L-L*cosd(rangle))/(.5*mA));

    %solving 2 equations with 2 unknowns
    A=[ mA, mB; -1, 1]; %Matrix of coefficients
    B=[mA*vsack1; cres*vsack1];  %Vector of constants
    X=A\B;
    vsack2=X(1); %velocity of sack after impact
    vchest =X(2);  %velocity of chest after impact

    %Calculate loss of energy during collision
    lenergy=.5*mB*vchest^2+.5*mA*vsack2^2-.5*mA*vsack1^2;

    fprintf('\n')
    fprintf('The velocity of the sack just before impact is %4.2f m/s.\n',
vsack1)
    fprintf('The velocity of the sack just after impact is %4.2f m/s.\n',
vsack2)
    fprintf('The velocity of the chest just after impact is %4.2f m/s.\n',
vchest)
    fprintf('The loss of energy during collision is %4.2f Nm.\n', lenergy)
end
```

Let's run the function and consider the mass of the sack as 5 lb, mass of the chest as 20 lb, the release angle as 90 degrees, the length of the rope as 3 feet and the coefficient of restitution as 0.5.

After running the function, we notice that the velocities of the sack before and after impact are 13.90 f/s, and -2.78 ft/s respectively. The negative sign indicates that the direction of motion will be changed after the impact which is correct considering the weight of the chest. The velocity of the chest right after the impact is 4.17 ft/s and the loss of energy during collision is -9.00 ft.lb. Negative sign indicates the loss of energy which is due to deformation during the collision.

Command Window

```
>>[vsack1, vsack2, vchest, lenergy] = central_impact()

This function is for a central impact problem. Input
parameters are: the angle of release of a sack tied by a
3 foot rope %above a chest, the weight of the sack and
the chest it strikes, and the coefficient of restitution,
it calculates the speed of the sack just before and just
after impact, the speed of the chest after impact, and
the energy lost due to the impact. Written Gregg Zappone,
modified by Simin Nasseri

Refer to the given figure.

Input the coefficient of restitution (MUST BE >=0 and
<=1): .5
Input the release angle in degrees (MUST BE >0 AND <=90
DEGREES): 90
If entering data in English units, enter "1", if in Metric
enter "2": 1

Input the weight of the sack IN POUNDS: 5
Input the weight of the chest IN POUNDS: 20
Please enter the length of the rope in FEET:  3
The velocity of the sack just before impact is 13.90 f/s.
The velocity of the sack just after impact is -2.78 ft/s.
The velocity of the chest just after impact is 4.17 ft/s.
The loss of energy during collision is -9.00 ft.lb.
```

Appendix

References Used:
1. Engineering Mechanics Dynamics, 12th Edition, by R.C. Hibbeler, Prentice Hall, ISBN-10: 0136077919.
2. Solving Mechanical Engineering Problems with MATLAB; Dr. Simin Nasseri, ISBN 10: 1-60797-524-6, Linus Publishing Co. 2015.