MSC.ADAMS

MSC.ADAMS Basic Full Simulation Package Training Guide

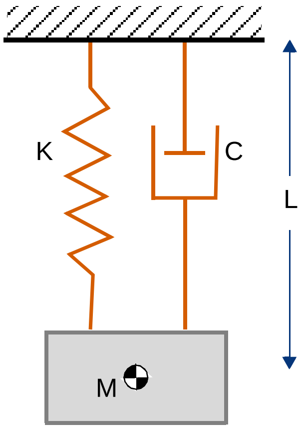
Release 2005

ADAM*V2005*Z*FSP*Z*SM-ADM701-TG

Visit us at: www.mscsoftware.com

N-15 NONLINEAR SPRING

Investigate the differences between a linear spring and a nonlinear spring using a spline function.



M: 187.224 Kg

K: 5.0 N/mm

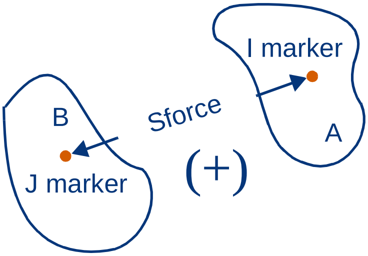
C: 0.05 N-sec/mm

L_0: 400mm

F_0: 0

What's in this module:

Characteristics of action-reaction single-component forces (Sforces)



The characteristic:	**Defines:**
Bodies	Two (A, B)
Points of application	Two (I and J markers)
Vector components	One
Orientation	Acts along the line of sight (between the I and J markers) ◆ **Positive** force **repels** the two parts ◆ **Negative** force **attracts** the two parts
Magnitude	User-defined

See also: Characteristics of an action-reaction S-force, on page 378

Note: MSC.ADAMS applies action and reaction forces to the I and J markers that it automatically creates.

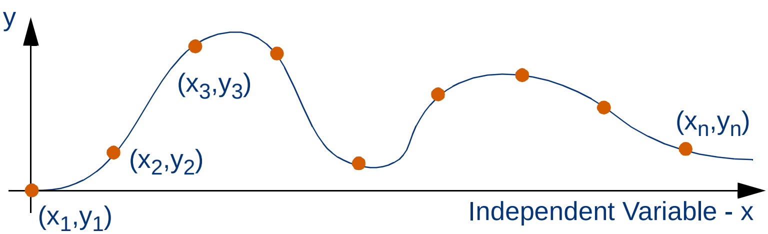
Spline Functions

Test data that can be incorporated into a simulation includes

- Empirical data from suppliers or standard tables for:
  - Nonlinear compliances (force versus velocity).
  - Curves for torque versus motor speed (torque versus angular velocity).
- Data taken from physical prototype simulations for:
  - Accelerometer data (acceleration versus time).
  - Tire lateral force as a function of normal force and slip angle.

To incorporate data into a simulation

- **First**, create a spline from either:
  - Data points entered manually into the Spline Editor.
  - Imported test data from a file.



- **Then**, reference the spline through a spline function used in a motion or force. Several interpolation methods are available (using the function type):
  - Cubic-fitting method (CUBSPL)
  - Akima-fitting method (AKISPL)
  - B-spline method (CURVE)

AKISPL Function

Syntax for AKISPL function

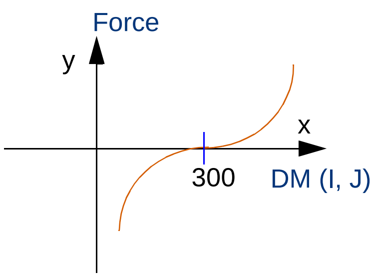
AKISPL (x, z, spline, iord)

- **x -** Independent variable specifying the value along the x-axis.
- **z -** Optionally, a second independent variable specifying the value along the z-axis of the surface being interpolated.
- **spline -** Spline used to map the one-to-one correspondence of the dependent variables (y) against independent variable values (x or z).
- **iord -** An integer variable that specifies the order of the interpolated point (usually 0, but can be 1 or 2).

Example of an AKISPL function

AKISPL (DM(I, J), 0, spline_1, 0)

DM (I, J)	Force
x	y
150	-1000
200	-200
250	-50
300	0
350	50
400	200
450	100

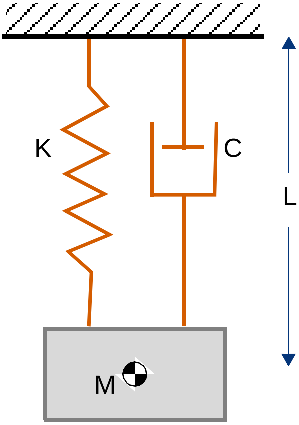


Note: You can create the CUBSPL and CURVE functions exactly as you create the AKISPL function.

Workshop 15—Nonlinear Spring

Problem statement

Investigate the differences between a linear spring and a nonlinear spring using a spline function.



M: 187.224 Kg

K: 5.0 N/mm

C: .05 N-sec/mm

L_0: 400mm

F_0: 0

Start the workshop

Start by importing the model you created in the last workshop. Note that this file is not in the current working directory.

To start the workshop:

1 Start ADAMS/View from the directory ***exercise_dir*/mod_15_spring**.

2 From the directory ***exercise_dir*/mod_14_spring_damper**, import the model that you created in the previous module.

If you need a fresh copy of the model, import the command file spring_damper_completed.cmd from the directory *exercise_dir*/mod_14_spring_damper/completed.

Replace the predefined spring damper

Now you will replace the spring damper, that is already in the model, with a user-defined linear spring damper.

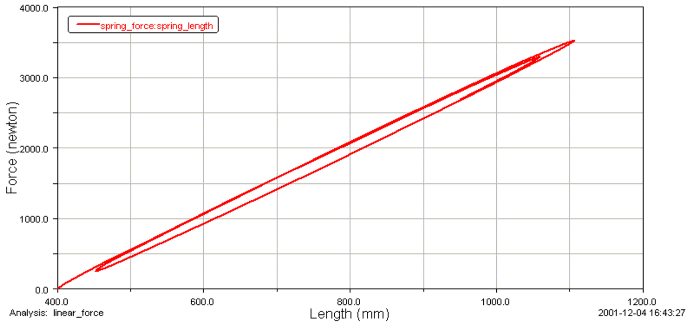
To replace the spring damper:

1. Delete the predefined spring damper.
2. Create a single-component, action-reaction force ()
   - **Run-time Direction:** **Two Bodies**
   - **Characteristic:** **K and C**
   - **K:** **5.0**
   - **C:** **0.05**
3. Right-click **SFORCE_1**, and then select **Info**. Note that the syntax of the function matches that of a spring damper, introduced in Magnitude based on stiffness and damping coefficients, on page 214.

   Note: You could have entered the linear spring-damper function manually in the Function Builder.
4. Right-click **SFORCE_1** and select **Measure**.
5. Create a measure, named **spring_force**:
   - **Characteristic**: **Force**
   - **Component: mag**
6. To see oscillations, run a **2-second**, **50-step** simulation.

7 Plot **spring_force** versus **Spring_Length**.

Note that the slope at the beginning of this plot is 5 (K). The time-dependency of damping (C=.05) results in a deviation from a slope of 5 (slope ~ stiffness +/- damping).



8 Save the simulation results () as **linear_force**.

Change the linear spring to a nonlinear spring

In this section, you change the spring damper you just created to a nonlinear spring. You'll import spring stiffness data to define the spring properties.

To change the spring:

1 To import the spring stiffness data, from the **File** menu, select **Import**.

2 Set the following parameters, and then select **OK**.

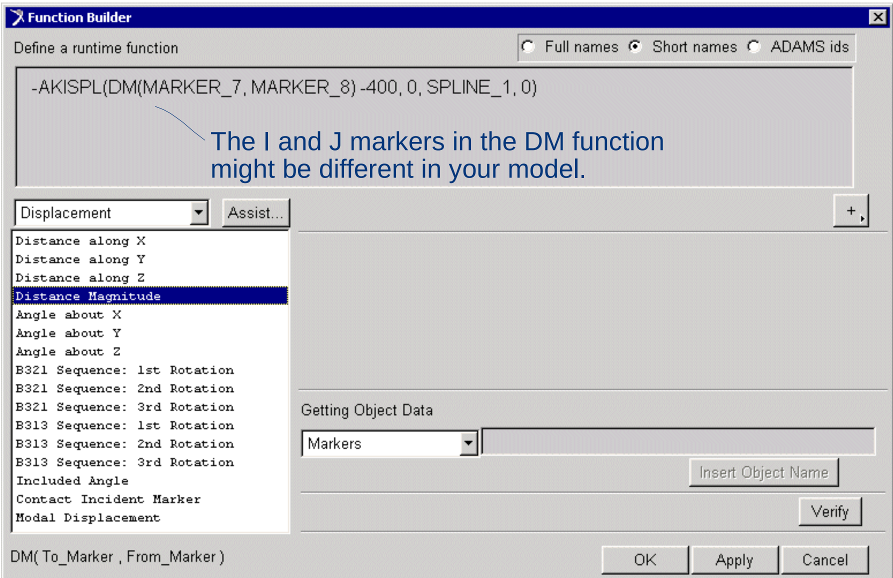
- **File Type: Test Data**
- **Create Splines**
- **File to Read: *exercise_dir*/mod_15_spring/spring_data.txt**
- **Independent Column Index: 1** (Because the first column is the independent column.)
- **Units: Force**
- **Model Name: .spring_mass**

3 To open **SPLINE_1** in the Spline Editor, from the **Build** menu, point to **Data Elements**, point to **Spline**, and then select **Modify**.

4 View the plot to understand the relationship between the deformation (x-axis) and stiffness force (y-axis).

Tip: In the upper right corner, set **View as** to **Plot**.

5 In the model, right-click **SFORCE_1**, and then select **Modify** to replace the force function describing the single-component force with an Akima spline function, as shown next.



Compare the linear and nonlinear forces

To compare the forces:

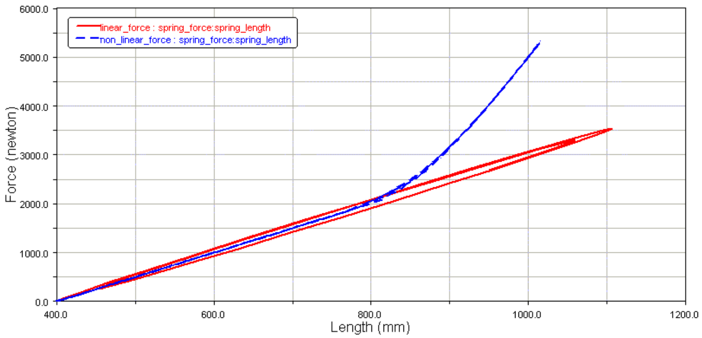
1 Verify that the nonlinear spring is working properly by running a **2-second**, **50-step** dynamic simulation.

2 Save the simulation results as **non_linear_force**.

3 Overlay the two plots:

- **spring_force vs. Spring_Length** for the **linear_force simulation**
- **spring_force vs. Spring_Length** for the **non_linear_force simulation**

Note that in the nonlinear case, the curve changes slope as Spring_Length increases.



Save your work

To save your work:

1 Save only the model.

If you want to further explore the model, as suggested in the next section, leave the model open. Otherwise, proceed with the next step.

2 Exit ADAMS/View.

Optional tasks

1 Create a spline in the Spline Editor to represent a nonlinear damping force versus velocity: From the **Build** menu, point to **Data Elements**, point to **Spline**, and then select **New**.

2 Add the spline function representing a damping force to the single-component force function so you have a nonlinear spring-damper in between the mass and ground.

Tip: You will need to change the Akima spline function so that the first independent variable tracks velocity (VR function).

Module review

1 What are the four inputs for a spline function?

____

____

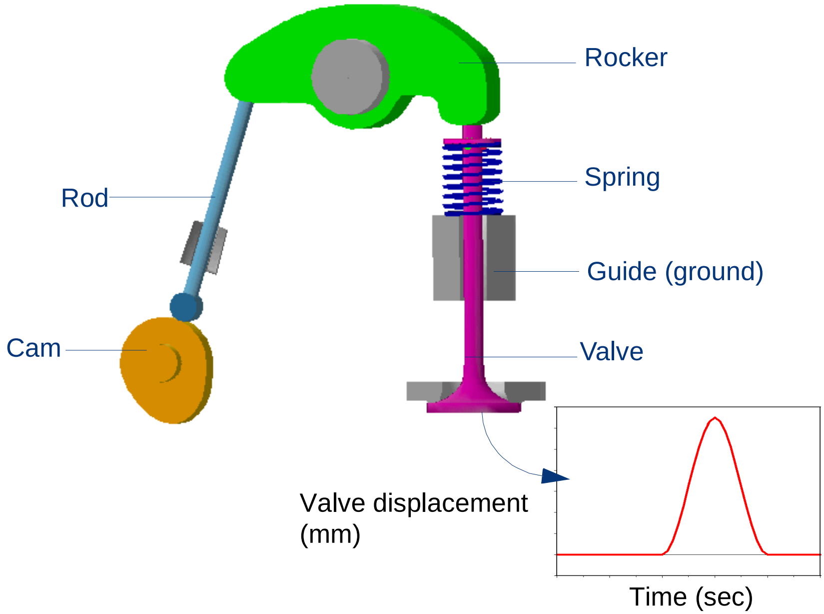
2 If you are not sure what inputs are required for an ADAMS/Solver function, where in the online help would you look for assistance?

____

____

C-21 Cam-Rocker-Valve

Design a cam profile based on desired valve displacement, and ensure that there is no follower liftoff when the cam is rotated at 3000 rpm.



What's in this module:

Splines from Traces

Definition of spline from a trace

- A point trace tracks a location of a marker or circle over time with respect to another part.
- ADAMS/View can create a two- or three-dimensional spline from a trace.
- Creating a spline from a trace is used to back-calculate (reverse engineer) the shape of an existing part based on its motion (cam synthesis).

Notes:

- When you trace an object and create a spline from it, the point or circle should move in a smooth, even path.
- If the path is closed, you should simulate for one cycle only.

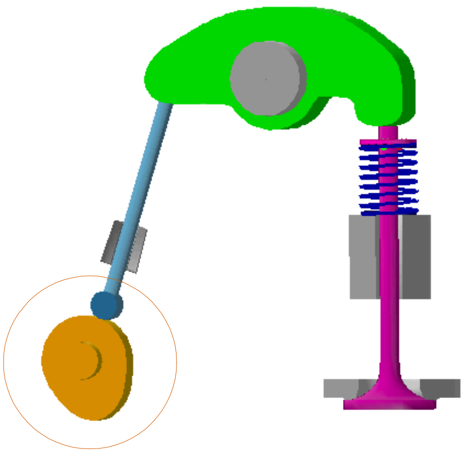
Curve Constraints

Types of curve constraints in MSC.ADAMS

- Point-on-curve
- Curve-on-curve

Curve-on-curve constraints

- Used where a curved edge on one part always follows a curved edge on a different part.
- Remove two DOF.
- Modeling of curve-on-curve constraints requires:
  - Two parts
  - Two curves that will always remain in contact
- Typical applications include general cam-to-cam systems.



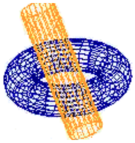
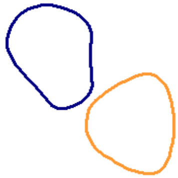
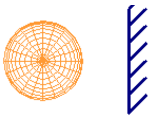
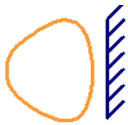
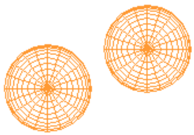
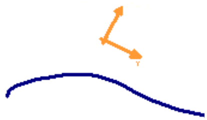
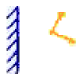
Note: Curve-on-curve constraints do not allow lift off.

See also: DOF removed by curve constraints, on page 376

Contact forces

- Are special forces acting on parts that are activated when part geometries come in contact with each other.
- Have values that are determined by a set of contact parameters identical to those in the IMPACT function.
- Multiple contact forces can be combined to create more complex contacts.

Contact pairs in MSC.ADAMS

solid-to-solid		curve-to-curve	
sphere-to-plane		curve-to-plane	
sphere-to-sphere		point-to-curve	
		point-to-plane	

Automated Contact Forces...

Things to note while creating automated contact forces

- Point-to-curve
- Curve-to-curve

The xy planes of the two reference markers must be parallel.

- Sphere-to-plane
- Curve-to-plane
- Point-to-plane

The z-axis of the reference marker of the plane (the plane's normal vector) must point away from the plane and at the circle or sphere.

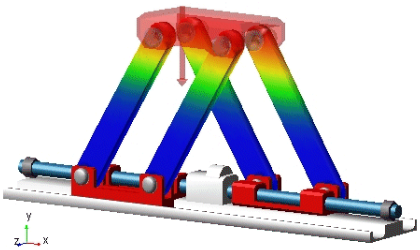
Flexible Parts—ADAMS/AutoFlex

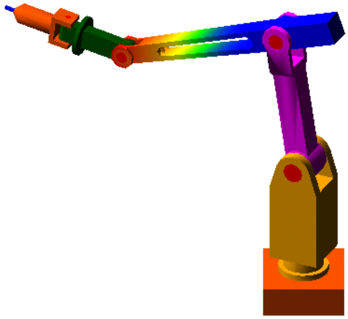
Better loading predictions for durability analyses

- The flexible component is the focus of your attention.
- Basically asking the question: "What is the system doing to my flexible component?"

Improved system performance

- The model fidelity is the focus of your attention. Component flexibility is just another parameter of the system design.
- Basically asking the question: "What is the flexible component doing to my system?"





Allows you to create flexible bodies in the MSC.ADAMS environment

Allows for easy substitutions of flexible bodies for rigid bodies in your MSC.ADAMS models

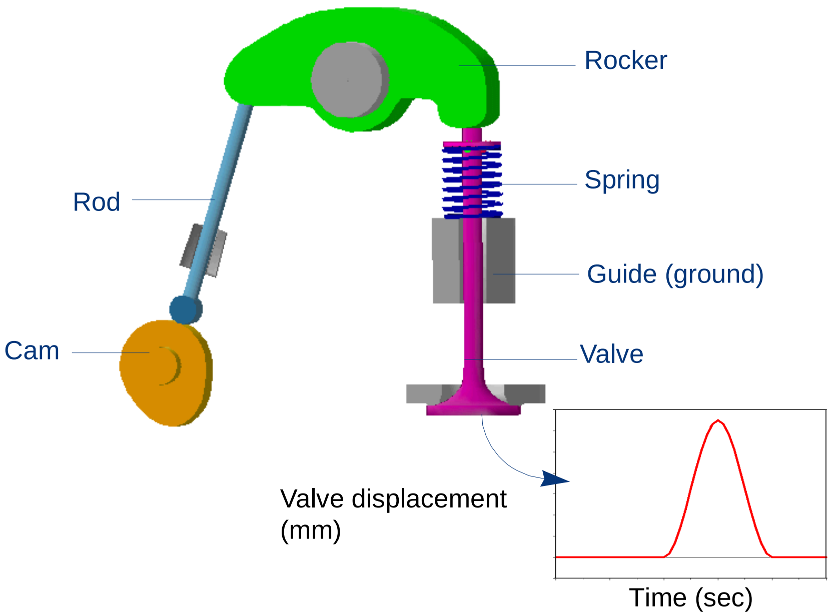
Can perform quick modifications on the flexible bodies to perform multiple iterations of the flexible body model

F1 To run through a workshop, see the ADAMS/AutoFlex Training Guide. For more information, see the ADAMS/AutoFlex online help.

Workshop 21—Cam-Rocker-Valve

Problem statement

Design a cam profile based on desired valve displacement, and ensure that there is no follower liftoff when the cam is rotated at 3000 rpm.



Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a velocity of 1 rotation per second.
- The rocker pivots about a pin attached to the engine block (ground).
- The valve displaces up and down as the rocker moves.
- When the valve moves, it lets small amounts of air in the chamber below it (not modeled here).

Note: At the location of the translational joint, between the valve and ground, the model includes a spherical dummy part. You will use this dummy part when you make the valve a flexible part. This dummy part will not affect the rigid body dynamics.

Start the workshop

To start the workshop:

1. Open ADAMS/View from the directory ***exercise_dir*/mod_21_camrocker.**
2. From the directory ***exercise_dir*/mod_21_camrocker**, import the model command file **valve_train_start.cmd**.

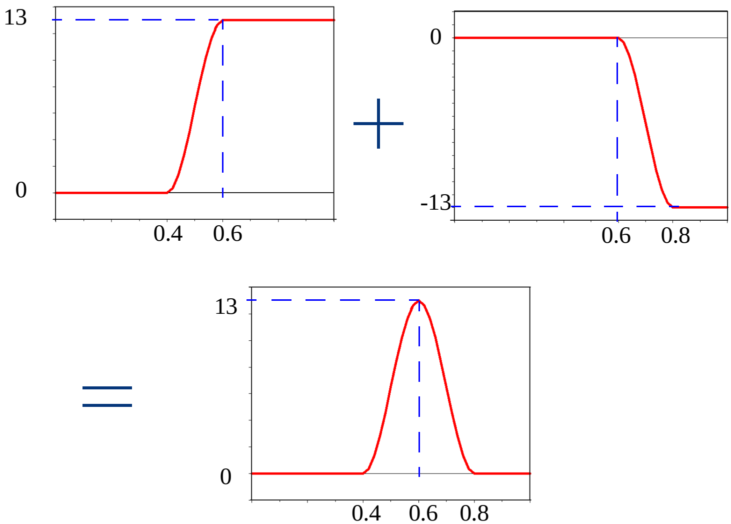
   The file contains a model named **valve_train**.

Apply motion

To apply motion:

1. Use the **Translational Joint Motion** tool  to add a motion to the joint, **Valve_Ground_Jt**, such that its displacement appears as shown next:

   Add two STEP functions.



   Tip: The functions should look as follows: STEP(time, .4, 0,.6,13)+ STEP(time,.6,0,.8,-13).

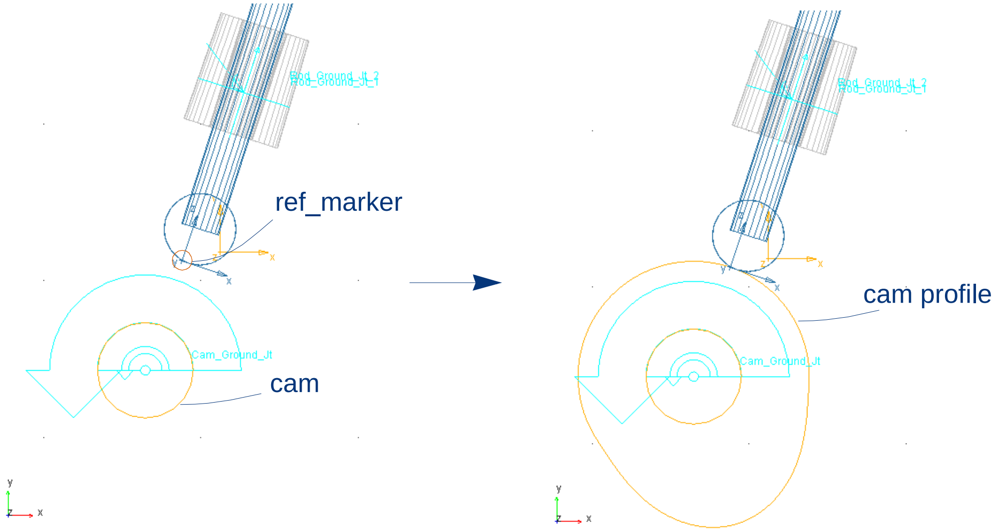
2. Run a **1-second**, **100-step** simulation to verify that the valve displaces as a result of the joint motion.

Create a cam profile

Use a point trace to create a cam profile.

To use a point trace:

1. From the **Review** menu, select **Create Trace Spline.**
2. Select the circle on the rod (**rod.CIRCLE_1**) and then the part named **cam**.
3. Verify that you now have a spline representing the cam profile.



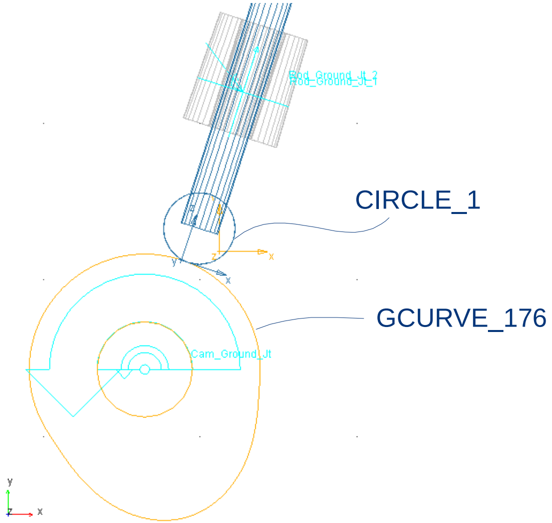
4. Run a simulation to verify that the **Rod** appears to move along the surface of the **Cam**.

Constrain the rod to the cam

To constrain the rod:

1 Delete the joint motion on the joint, **Valve_Ground_Jt**.

2 Use the **Curve-Curve Constraint** tool  to create a curve-on-curve constraint between the circle on the **Rod** and the cam profile on the **Cam**.



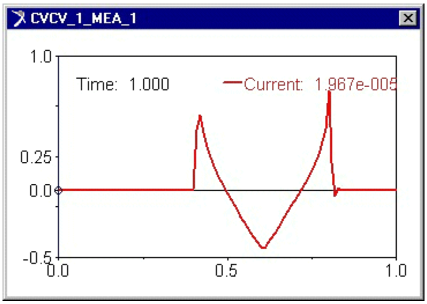
3 Run a simulation to verify that the new constraint works.

Measure the force in the curve-on-curve constraint

To measure the force:

- Create a force measure for the curve-on-curve constraint (right-click the constraint and then select **Measure**). Measure the force along the z-axis of **ref_marker**, which belongs to the rod:
  - **Characteristic:** **Force**
  - **Component:** **Z**
  - **Represent coordinates in:** **ref_marker**

  The curve-on-curve constraint applies a negative force that keeps the rod follower on the cam, avoiding any liftoff.



Make the cam-to-rod contact more realistic

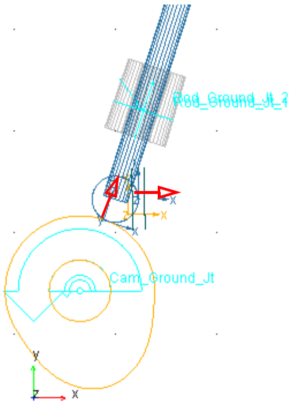
Now you'll replace the curve-on-curve constraint with a curve-to-curve contact force.

To replace the curve-on-curve constraint:

1. Deactivate the curve-on-curve constraint you created in Step 2 on page 308.
2. From the Main Toolbox, right-click the **Create Forces** tool stack, and then select the **Contact** tool .

3 Use the following contact parameters:

- **Contact Name**: **rod_cam_contact**
- **Contact Type**: **Curve to Curve**
- **I Curve:** **CIRCLE_1**
- **J Curve:** **GCURVE_176**
- Use the **Change Direction** tool  to make sure that the normal arrows point outward from the curves, as shown next:



- **Normal Force**: **Impact**
- **Stiffness (K)**: **1e6 (N/mm)**
- **Force Exponent (e)**: **1.5**
- **Damping (C)**: **10 (N-sec/mm)**
- **Penetration Depth (d)**: **1e-3 mm**
- **Friction Force:** **Coulomb**
- **Coulomb Friction:** **On**
- **Static Coefficient** (μ_s): **0.08**

- **Dynamic Coefficient** (μ_d): **0.05**
- **Stiction Transition Vel.** (v_s): **1 (mm/sec)**
- **Friction Transition Vel.** (v_t): **2 (mm/sec)**
- Run a simulation to check if liftoff occurs.

Prevent liftoff using a spring damper

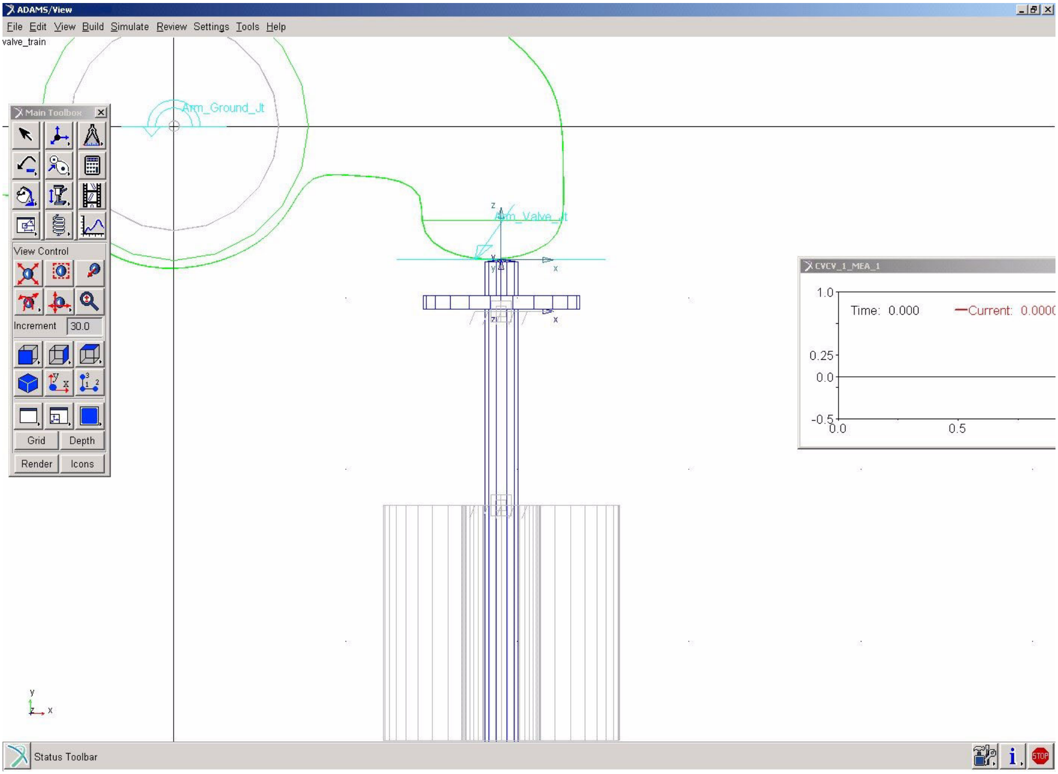
To prevent liftoff:

1 Add a marker on the valve at the location, **Valve_Point**:

- **Add to Part**
- From the screen, select **valve** and the location **Valve_Point**.

2 Add a spring damper between the marker you just created and the point, **Ground_Point** (which is a point on ground, at the top of the guide) using the following parameters:

- **Stiffness (K)**: **20 (N/mm)**
- **Damping (C)**: **0.002 (N-sec/mm)**
- **Preload**: **100 N**



3 Find the static equilibrium of the model ().

Do not reset the model before going on to the next step.

Note: You perform the static equilibrium to eliminate the transient effect that results from the time-dependent damping characteristic of the spring damper. In addition, positioning the model in static equilibrium establishes initial contact between the roller and the cam.

4 Run a dynamic simulation to view the effects of the spring starting from static equilibrium.

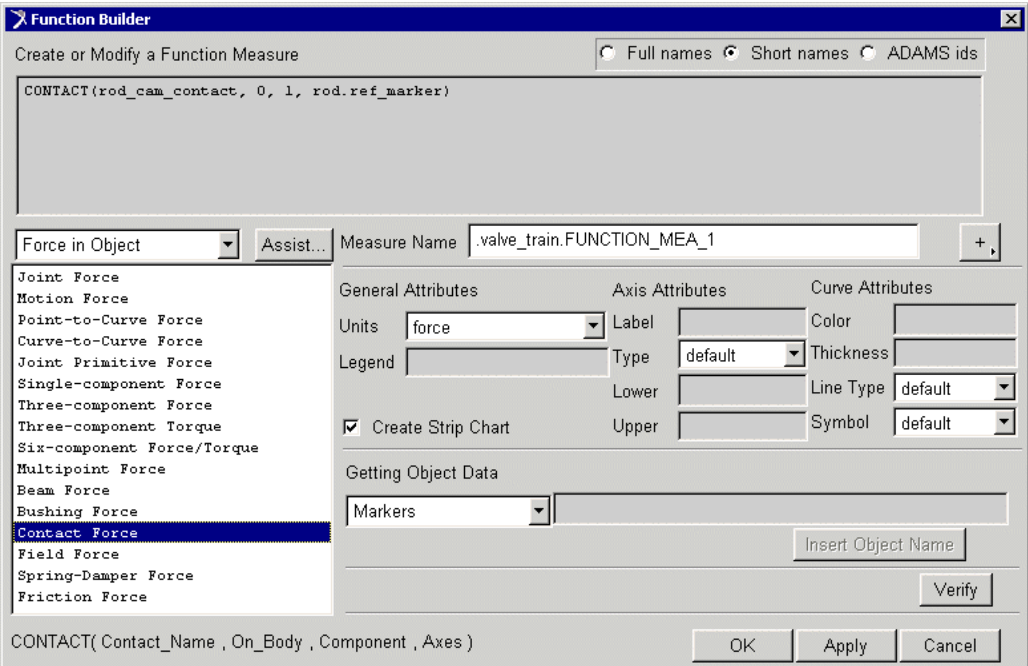
5 Modify the rotational motion on the cam to a speed of **3000 rpm**. Enter the function as follows: **-50*360d*time**.

6 To view only one rotation of the cam, run a static equilibrium followed by a dynamic simulation for **end=1/50 seconds**, **steps=100**. An easy way to run this simulation sequence is to create a simulation script.

7 Measure the contact force (**Build** → **Measure** → **Function** → **New**).

- **Category: Force in Object**

Note: Make sure the function looks as shown next:



8 Rerun the simulation to populate the new measure stripchart.

9 Modify the spring-damper characteristics (stiffness, damping, and preload) to prevent liftoff based on the new rotational speed of the cam.

Note: Experiment with different values until the no-lift criteria is met.

10 Save the model.

Create and substitute the flexible part

You will define the mesh properties and then use the Automatic Replace Part option to substitute this flexible body automatically into your model. The Automatic Replace Part option does the following:

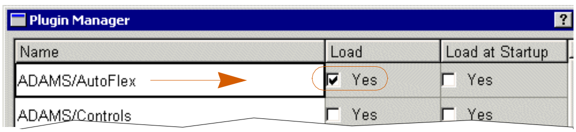
- Finds attachments based on the location of joints and forces that are applied to the rigid body
- Finds the closest four slave nodes of the temporary mesh to each attachment point and connects them through rigid bars
- Creates the flexible body
- Applies the joints and forces at the attachment points of the flexible body
- Deactivates the rigid body

To load the ADAMS/AutoFlex plugin:

1 From the **Tools** menu, point to **Plugin Manager**.

2 To the right of **autoflex**, set **Load** to **Yes**.

3 Select **OK**.

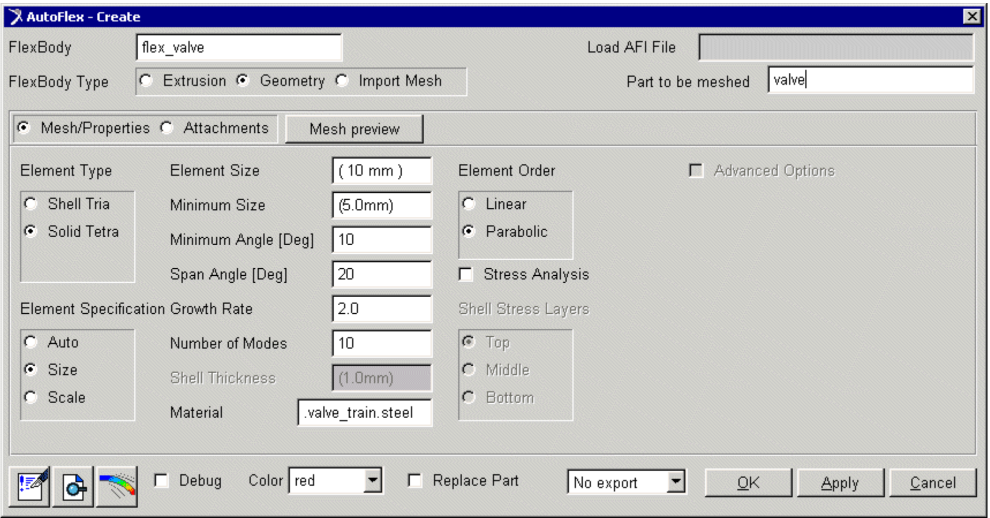


To create the mesh:

1 From the **Build** menu, point to **Flexible Bodies**, and then select **ADAMS/AutoFlex**.

- **FlexBody Type:** **Geometry**
- **Part to be meshed:** **valve**
- **FlexBody Name:** **flex_valve**
- **Element Type:** **Solid Tetra**
- **Element Specification:** **Size**
- **Element Size:** **10 mm**
- **Number of Modes:** **10**
- **Element Order:** **Parabolic**

- Leave all other options at their default settings.
- Select **Mesh preview.**



To substitute the flexible body into the model:

1. At the bottom of the ADAMS/AutoFlex dialog box, select **Replace Part**.
2. Select **OK**.

   ADAMS/AutoFlex automatically substitutes the flexible part into your model to replace the rigid part. It also writes the MNF and AFI files to your current working directory. This step may take a couple of minutes to complete.

To verify that the rigid part is a free body:

1. Display the Database Navigator.
2. Set the pull-down menu at the top to **Graphical Topology**.
3. Double-click **valve_train.**
4. Select **valve.**
5. The valve should not be attached to any other part. If the valve is attached to any parts, take corrective measures.

Run a simulation

To run a simulation:

1 To view only one rotation of the cam, run a static equilibrium followed by a dynamic simulation for **end=1/50 seconds**, **steps=100**.

2 Use ADAMS/PostProcessor to investigate how the flexible body affects the model.

Does liftoff occur in the model now? ________________

Save your work

To save your work:

1 Save the model.

If you want to further explore the model, as suggested in the next section, leave the model open. Otherwise, proceed with the next step.

2 Exit ADAMS/View.

Optional tasks

Change the shape of the follower:

1 Delete the circle-to-curve contact force between the **Rod** circle and the **Cam** curve.

2 Add a spline to the **Rod** that is a shape other than a circle and use that as the follower.

Change the grid spacing to **5mm** in the **x** and **y** directions and draw the new follower geometry on the grid.

3 Recreate the contact force between the **Rod** and the **Cam** using the new spline as the follower.

Use a curve-to-curve contact force.

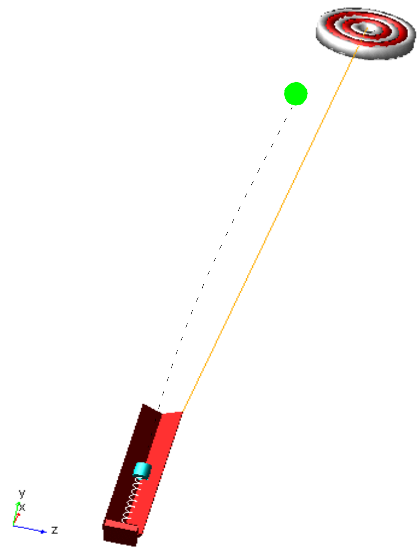
Module review

1 How many DOF are removed by adding a curve-on-curve constraint?

2 How many DOF are removed by a curve-to-curve force?

T-22 Target Practice

Complete the construction of a parametric gun and find the launch-spring stiffness that will allow the marbles to hit the target.



What's in this module:

Multi-Component Forces

Types of multi-component forces

- Vector force (three translational components)
- Vector torque (three rotational components)
- General force vector (three translational, three rotational components)

Characteristics of vector force

The characteristic:	**Defines:**
Bodies	Two
Points of application	Two (action force at *I marker* and reaction force at floating *J marker*)
Vector components	Three translational
Orientations	Based on reference marker (*R marker*)
Magnitudes (F_x, F_y, F_z)	User-defined

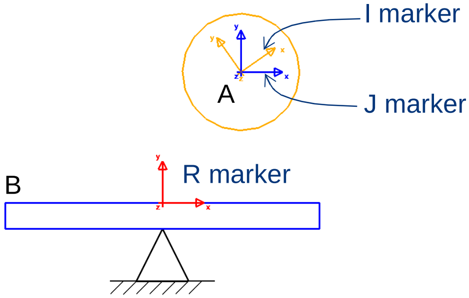
Notes:

- The floating J marker always maintains the same location as the I marker.
- The characteristics of other multi-component forces conceptually work the same way.

Multi-Component Forces...

Example of a force vector

- A vector force representing a contact between a ball and a cantilever:



I marker belongs to part A

J marker belongs to part B but floats its location with the I marker

R marker belongs to part B

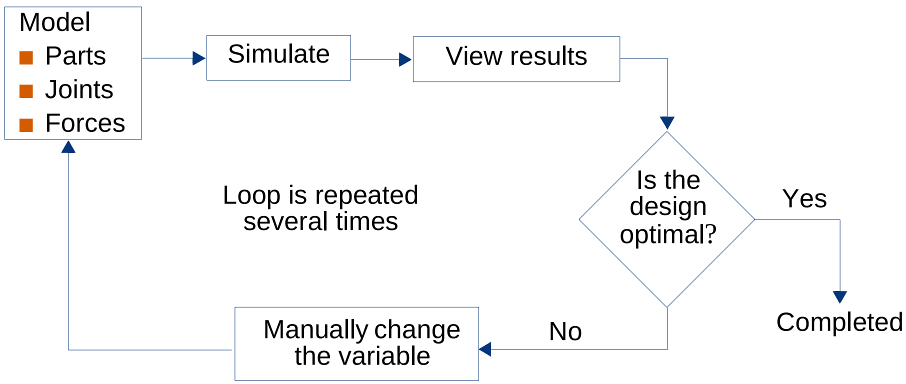
- Because the J marker belongs to part B, the force acts on part B when the bodies collide.
- Because the J marker moves with the I marker, part B knows where to apply the reaction force.

Note: In the example, the J and R markers must belong to the same part. However, the R marker can belong to *any* part.

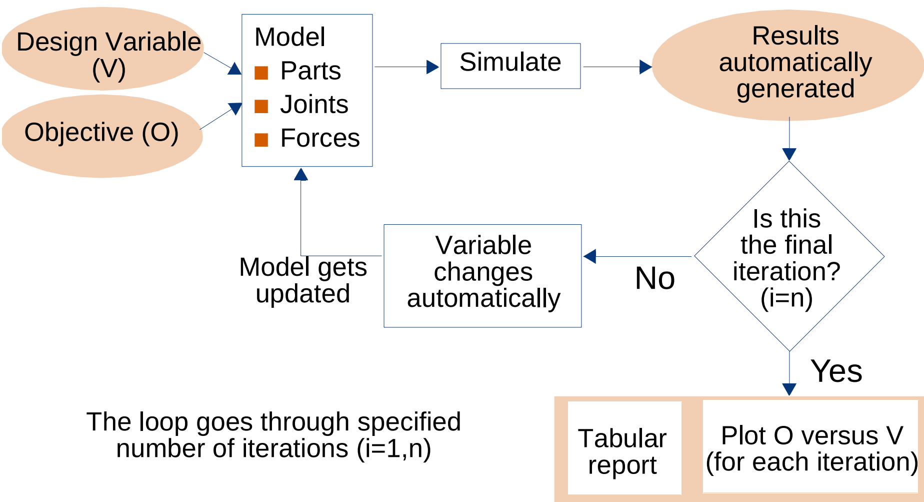
See also: Characteristics of a multi-component force, on page 375

Design Studies

Trial and error method (manual iterations)



Design study method (automated iterations)



Design Studies...

Definition of a design study

- Varies a single design variable (V) across a range of values.
- Runs a simulation at each value.
- Reports the performance measure for each simulation.

From the results generated, you can determine:

- The best value for V among the values simulated.
- The approximate design sensitivity of V (rate of change of performance measure with respect to V).

Sensitivity, S, at iteration, *i*

$$S_i = \frac{1}{2}\left(\frac{O_{i+1} - O_i}{V_{i+1} - V_i} + \frac{O_i - O_{i-1}}{V_i - V_{i-1}}\right)$$

- Looking at Trial 4 (i = 4):

$$S_4 = \frac{1}{2}\left(\frac{(-0.62784) - 0.017103}{10.7 - 10.6} + \frac{(-0.017103) - 0.58166}{10.6 - 10.5}\right)$$

S_4 = -6.0475

- S_4 is the approximate slope at Trial 4 (tip_y_loc=10.6) in the plot.

Design Study Summary

Model Name : stamp
Date Run : 15:48:55 23-Dec-98

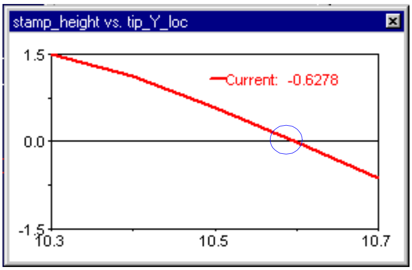
Objectives

O1) Minimum of stamp_height
Units : inch
Maximum Value: 1.48945 (trial 1)
Minimum Value: -0.627838 (trial 5)

Design Variables

V1) tip_Y_loc
Units : inch

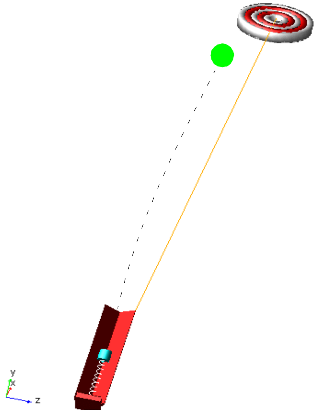
Trial	stamp_height	tip_Y_loc	Sensitivity
1	1.4894	10.300	-3.6131
2	1.1281	10.400	-4.5389
3	0.58166	10.500	-5.7262
4	-0.017103	10.600	-6.0475
5	-0.62784	10.700	-6.1073



Workshop 22—Target Practice

Problem statement

Complete the construction of a parametric gun and find the launch-spring stiffness that will allow the marbles to hit the target.



Model description

- Currently, the model has all the geometry, parts, and constraints it needs.
- There is a single-component force representing the spring force between the hammer and the launch pad.
- The single-component force is designed so that it changes characteristics depending on the type of simulation being run (static versus dynamic):
    - If a static simulation is run, the spring has a free length of 40 mm.
    - If a dynamic simulation is run, the spring has a free length of 100 mm.
- Therefore, initially run a static simulation, so the marble falls on the hammer and compresses the spring a little. Then when you run a dynamic simulation, the spring thinks it is compressed a great deal, and shoots the marble.
- The model is already parameterized with variables describing the elevation angle of the launch pad and the stiffness and damping of the spring.
- You will only modify the stiffness of the spring.
- Initially, the spring stiffness is 20 N/mm.

Workshop 22—Target Practice...

Start the workshop

Import the file to build the model target_practice.

To start the workshop:

1 Start ADAMS/View from the directory ***exercise_dir*/mod_22_target_practice**.

2 Import the model command file **target_practice_start.cmd**.

Create a contact between the hammer and the marble

To create a contact:

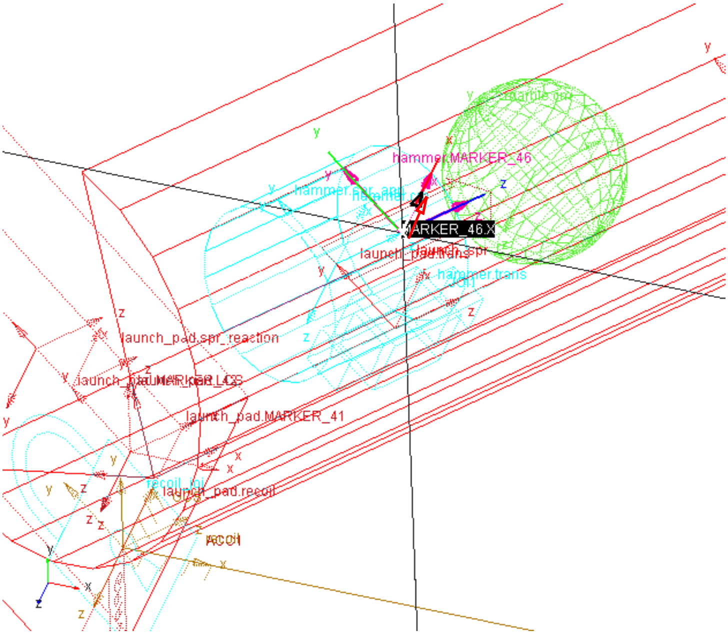
1 Run a simulation to see the forces that affect the model in its current configuration.

2 Create a marker on the hammer to represent the plane in the sphere-to-plane contact force that you will create next. Orient the z-axis of this marker so it points toward the center of mass of the ball.

- **Add to Part: hammer**
- **Orientation: Z-Axis** (orient the z-axis along the x-axis of the **cm marker of the marble**)
- In your model, right-click at the face of the hammer, and then select **hammer.CYL10.E1 (center)**. Note that hammer.CYL10.E1 (center) is a place-holder that represents the geometry, and is only accessible when prompted for a location.

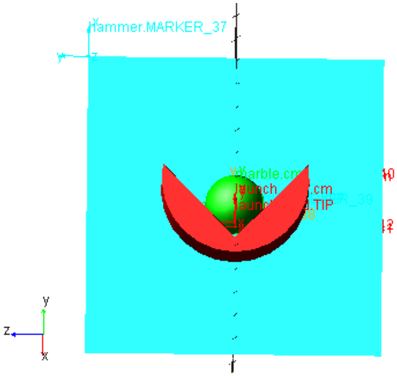
3 Orient the working grid along the XY-plane of the new marker, aligned with the face of the hammer (**Settings → Working Grid**)

- **Set Location: Pick** (from your model, select the marker you just created, **MARKER_46**)
- **Set Orientation: X-Y-Axes** (first select the x- and then the y-axis of **MARKER_46**, as prompted in the Status bar)



4 Adjust the view until the new working grid's xy-plane encompasses the window, as shown next.

Tip: Set the model to a Right view, then rotate the model down, and turn shading on.



5 Use the **Plane** tool  to create a plane to be used in the contact definition:

- **Add to Part: hammer** (right-click the sphere image and select hammer)
- Left-click on any grid point outside of the hammer face and drag across, covering the hammer face.

6 Reset the working grid to the default position:

- **Set Location: Global Origin**
- **Set Orientation: Global XY**

Verify ADAMS/Solver equilibrium settings

To verify the settings:

1 From the **Settings** menu, point to **Solver**, and then select **Equilibrium**.

2 Verify that **Error** is set to **1.0E-002**.

3 Select **Close**.

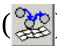
Create contact between marble and hammer

To create the contact:

- Create a sphere-to-plane contact force () that represents the contact between the marble and the hammer:
  - **Contact type:** **Sphere-to-Plane**
  - **Sphere:** **ELL3**
  - **Plane:** **PLANE_36**
  - **Stiffness:** **100 (N/mm)**
  - **Force Exponent:** **1.01**
  - **Damping:** **1.0 (N-sec/mm)**
  - **Penetration Depth:** **0.2 mm**

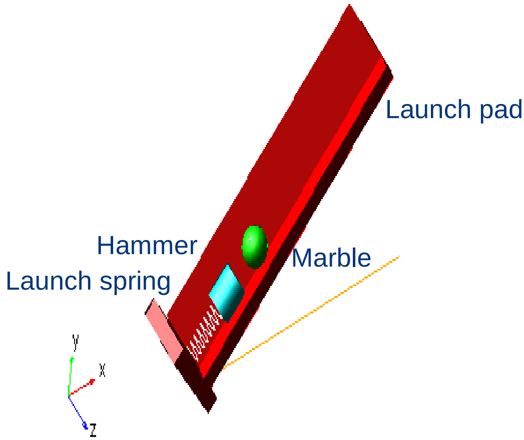
Create contact between marble and launch pad

To create the contact:

- Create a solid-to-solid contact force () that represents the contact between the marble and the launch pad:
  - **Contact type:** **Solid-to-Solid**
  - **I Solid:** **ELL3**
  - **J Solid:** **CSG_35**
  - **Stiffness:** **100 (N/mm)**
  - **Force Exponent:** **1.01**
  - **Damping:** **1.0 (N-sec/mm)**
  - **Penetration Depth:** **0.2 mm**
  - **Friction Force:** **Coulomb**
  - **Static Coefficient:** **0.3**
  - **Dynamic Coefficient:** **0.1**
  - **Stiction Transition Vel.:** **50**
  - **Friction Transition Vel.:** **200**

Simulating the model

The marble is not initially in contact with the hammer; therefore, you must precede each dynamic simulation that you submit with a static equilibrium simulation.



To simulate the model:

1 Using ADAMS/Solver commands, create a script that performs a static equilibrium simulation followed by a dynamic simulation:

```
SIMULATE/STATIC
SIMULATE/DYNAMIC, END=1.0, STEPS=100
```

The script will make running simulations easier and will be needed for the design study later.

2 Create a measure, named **x_displacement**, of the marble’s global x displacement (right-click the marble and select **Measure**):

- **Characteristic: CM position**
- **Component: X**
- **From/At: ground**

3 Run the scripted simulation you created in step 1.

4 Save the simulation results () as **no_aero**.

Workshop 22—Target Practice...

Include aerodynamic drag force

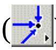
Use the following parameters to describe the global x and y components of the drag force:

- $F_x = -1/2 * \rho * V_x * V_m * C_d * A$
- $F_y = -1/2 * \rho * V_y * V_m * C_d * A$

where:

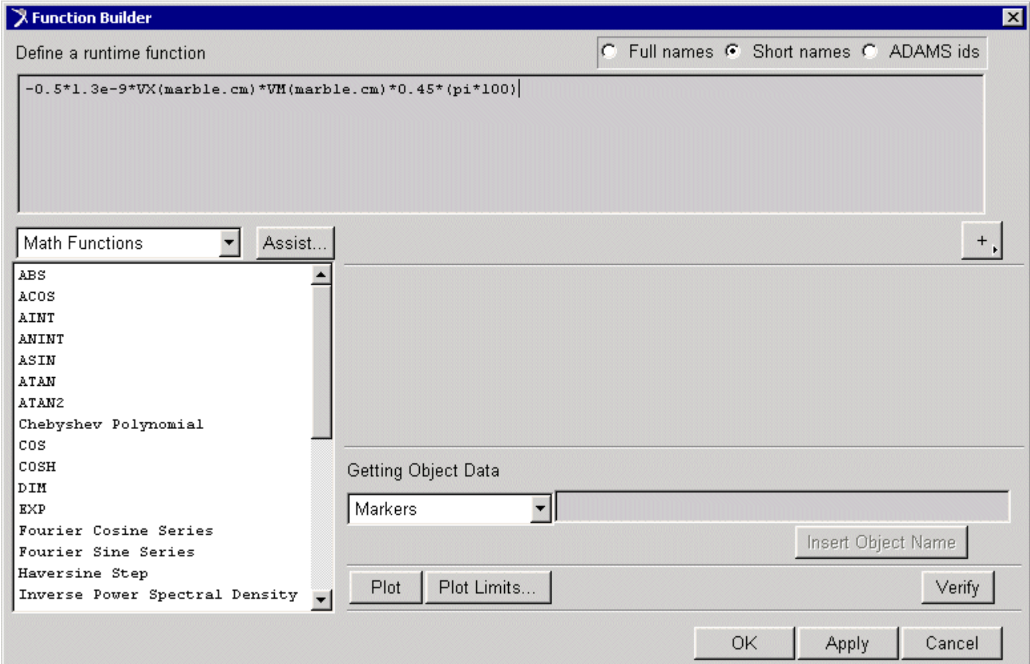
- ρ = 1.3e-9 kg/mm3 = density of air
- V_x = global x component of the marble's velocity
- V_y = global y component of the marble's velocity
- V_m = magnitude of the marble's velocity
- C_d = 0.45 = drag coefficient
- $A = \pi r^2$ = two-dimensional area of the marble face

To include aerodynamic force:

1. Create a multi-component force () at the marble center of mass, between the **marble** and **ground**, whose directions are aligned with the global coordinate system (***Hint***: Set **Characteristic** to **Custom**).

2. Select the **Function Builder** tool ... to the right of **X Force**.

3 Enter the aerodynamic function as shown next:

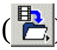


4 Select **Verify**, and then select **OK**.

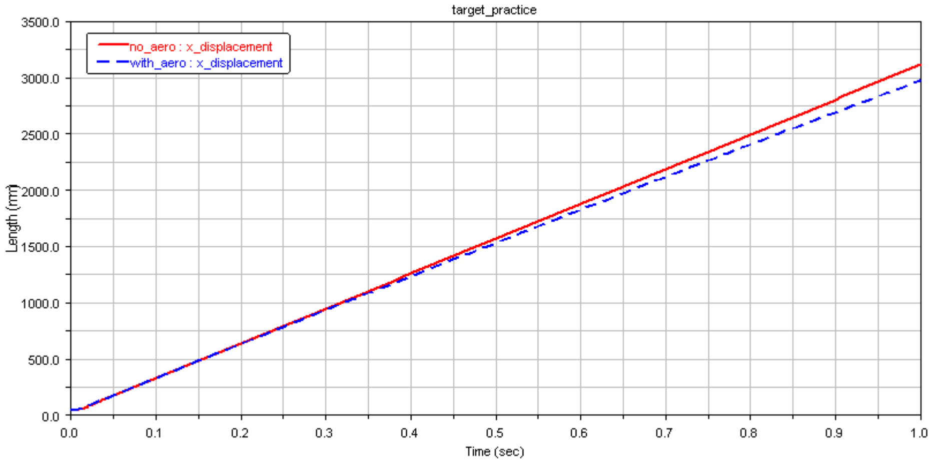
5 Select the **Function Builder** tool  to the right of **Y Force**.

6 Enter the function similar to the one above, changing only **VX** to **VY**, and then verify the function.

7 Run a simulation using the script you created in Step 1, on page 330.

8 Save the simulation results () as **with_aero**.

9 In ADAMS/PostProcessor, compare the x displacement of the marble for each set of simulation results (with and without aerodynamic forces).



Track when the simulation is complete

To track:

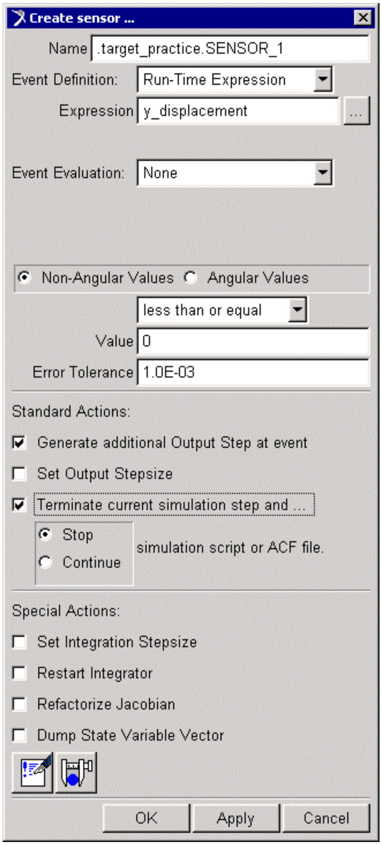
1 Create a measure, named **y_displacement**, that tracks the global y displacement of the center of the marble.

- **Characteristic:** **CM position**
- **Component:** **Y**
- **From/At:** **ground**

2 Create a sensor to determine when the ball passes the global xz plane.

Have the sensor monitor the global y displacement measure created earlier, and when it is less than or equal to 0.0 mm, direct the sensor to:

- Generate an additional output step.
- Terminate the current simulation step and stop the simulation script.



Set up a design study

To set up a design study:

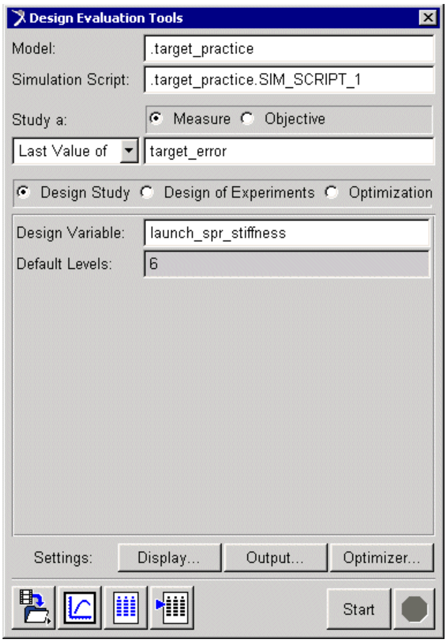
1 Create a point-to-point measure of the global x distance from the target center (**.ground.target**) to the marble center (**.marble.cm**) and name it **target_error**.

2 From the **Settings** menu, point to **Solver**, and then select **Display**.

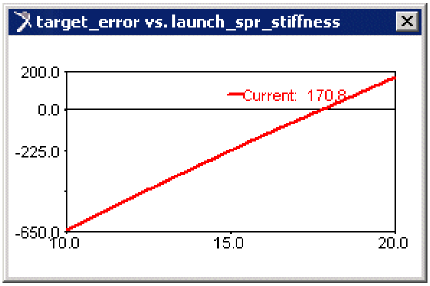
3 Set **Update Graphics** to **Never**.

4 Run a design study that gives the last value of **target_error** over six default levels of the existing design variable, **launch_spr_stiffness**.

- From the **Simulate** menu, select **Design Evaluation**.
- Fill in the dialog box as follows:



5 Select **Start**.



6 Note the approximate value of stiffness at which the marble hits the target and use it to answer Question 2 in Module review, on page 336.

Save your work

- Save your model and then exit ADAMS/View.

Optional tasks

1 Multiply the aerodynamic forces by STEP functions to eliminate any aerodynamic effects that might occur when the marble moves to the static position.

Tip: The independent variable of the STEP function will be DM(.marble.cm, .TIP).

2 Tighten the minimum and maximum values of the design variable of the launch-pad stiffness and increase the number levels to 10 to achieve a more precise solution.

3 Run additional design studies for other design variables.

Module review

1 What defines the force directions (F_x, F_y and F_z) in a force vector?

____

____

2 What is the approximate value of stiffness at which the marble hits the target?

____

____