

# 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](http://www.mscsoftware.com)

The information in this document is furnished for informational use only, may be revised from time to time, and should not be construed as a commitment by MSC.Software Corporation. MSC.Software Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

#### **Copyright Information**

This document contains proprietary and copyrighted information. MSC.Software Corporation permits licensees of MSC.ADAMS<sup>®</sup> software products to print out or copy this document or portions thereof solely for internal use in connection with the licensed software. No part of this document may be copied for any other purpose or distributed or translated into any other language without the prior written permission of MSC.Software Corporation.

Copyright © 2005 MSC.Software Corporation. All rights reserved. Printed in the United States of America.

#### **Trademarks**

ADAMS, EASY5, MSC, MSC., MSC.ADAMS, MSC.EASY5, and all product names in the MSC.ADAMS Product Line are trademarks or registered trademarks of MSC.Software Corporation and/or its subsidiaries.

NASTRAN is a registered trademark of the National Aeronautics Space Administration. MSC.Nastran is an enhanced proprietary version developed and maintained by MSC.Software Corporation. All other trademarks are the property of their respective owners.

#### **Government Use**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software) and DFARS 227.7202 (Commercial Computer Software and Commercial Computer Software Documentation), as applicable.



# CONTENTS

---

## Contents 3

### Welcome to MSC.ADAMS Basic Training 9

- A Brief History of MSC.ADAMS 10
- About MSC.Software 11
- Content of Course 12
- Getting Help 13

### Stamping Mechanism 17

- Virtual Prototyping Process 18
- Workshop 1—Stamping Mechanism 19

### ADAMS/View Interface Overview 27

- Model Hierarchy 28
- Renaming Objects 29
- ADAMS/View Interface 30
- Simple Simulations 31
- Saving Your Work 32
- Workshop 2—ADAMS/View Interface Overview 34

### ADAMS/PostProcessor Interface Overview 45

- PostProcessing Interface Overview 46
- Animating 47
- Plotting 48
- Reporting 49
- Workshop 3—ADAMS/PostProcessor Overview 50

### Falling Stone 61

- Coordinate Systems 62
- Part Coordinate System 63
- Coordinate System Marker 64
- Differences Between Parts and Geometry 65
- Parts, Geometry, and Markers 66
- Types of Parts in ADAMS/View 67
- Part Mass and Inertia 68
- Measures 69
- Workshop 4—Falling Stone 70

# Contents...

---

## Projectile Motion 79

- Part Initial Conditions 80
- Initial Velocities 81
- Point Trace 82
- Workshop 5—Projectile Motion 83

## One DOF Pendulum 93

- Constraints 94
- Use of Markers in Constraints 95
- Degrees of Freedom (DOF) 96
- Joint Initial Conditions (ICs) 97
- General Constraints (GCON) 98
- Merging Geometry 99
- Angle Measures 100
- Workshop 6—One DOF Pendulum 101

## Inclined Plane 117

- Euler Angles (Rotation Sequence) 118
- Precise Positioning: Rotate 119
- Modeling Friction 120
- Measures in LCS 123
- Workshop 7—Inclined Plane 124

## Lift Mechanism I 139

- Building Geometry 140
- Construction Geometry Properties 142
- Solid Geometry 144
- Precise Positioning: Move 145
- Workshop 8—Lift Mechanism I 146

## Lift Mechanism II 155

- Applying Motion 156
- Joint Motion 157
- Functions in MSC.ADAMS 158
- Workshop 9—Lift Mechanism II 159

## Lift Mechanism III 165

- Types of Joint Primitives 166
- Perpendicular Joint Primitive 167
- Workshop 10—Lift Mechanism III 169

# Contents...

---

<b>Suspension System I</b>	<b>173</b>
Applying Point Motions	174
System-Level Design	175
Workshop 11—Suspension System I	176
<b>Suspension System II</b>	<b>183</b>
Taking Measurements	184
Displacement Functions	185
Importing CAD-Based Geometry	186
Workshop 12—Suspension System II	187
<b>Suspension-Steering System</b>	<b>195</b>
Add-On Constraints	196
Couplers	197
Assembling Subsystem Models	198
Workshop 13—Suspension-Steering System	199
<b>Spring Damper</b>	<b>207</b>
Assemble Simulation	208
Simulation Hierarchy	209
Types of Simulations	210
Forces in MSC.ADAMS	212
Spring Dampers in MSC.ADAMS	213
Magnitude of Spring Dampers	214
Workshop 14—Spring Damper	215
<b>Nonlinear Spring</b>	<b>221</b>
Single-Component Forces: Action-Reaction	222
Spline Functions	223
AKISPL Function	224
Workshop 15—Nonlinear Spring	225
<b>Suspension-Steering System II</b>	<b>231</b>
Bushings	232
Workshop 16—Suspension-Steering System II	233
<b>Hatchback I</b>	<b>239</b>
Impact Functions	240
Velocity Functions	242
Workshop 17—Hatchback I	243

# Contents...

---

## Hatchback II 251

- STEP Function 252
- Scripted Simulations 253
- ADAMS/Solver Commands 254
- Workshop 18—Hatchback II 255

## Hatchback III 263

- ADAMS/Solver Overview 264
- Files in ADAMS/Solver 265
- Example of an ADAMS/Solver Dataset (.adm) File 266
- Stand-Alone ADAMS/Solver 267
- Solver Compatibility 268
- Example: 2D Pendulum 269
- Formulation of the Equations of Motion 270
- Phases of Solution 271
- Debug/Eprint (dynamics) 276
- Workshop 19—Hatchback III 278

## Hatchback IV 287

- Sensors 288
- Design Variables 289
- Workshop 20—Hatchback IV 290

## Cam-Rocker-Valve 299

- Splines from Traces 300
- Curve Constraints 301
- Automated Contact Forces 302
- Flexible Parts—ADAMS/AutoFlex 304
- Workshop 21—Cam-Rocker-Valve 305

## Target Practice 319

- Multi-Component Forces 320
- Design Studies 322
- Workshop 22—Target Practice 325

## Recommended Practices 337

- General Approach to Modeling 338
- Modeling Practices: Parts 339
- Modeling Practices: Constraints 340
- Modeling Practices: Compliant Connections 341
- Modeling Practices: Run-time Functions 342
- Debugging Tips 344

# Contents...

---

Switch Mechanism Workshop	349
Tables	373
Constraints Tables (Incomplete)	374
Forces Tables (Incomplete)	375
Constraint Tables (Completed)	376
Forces Tables (Completed)	378
Answer Key	379
Index	387





# WELCOME TO MSC.ADAMS BASIC TRAINING

---

MSC.ADAMS Full Simulation Package is a powerful modeling and simulating environment that lets you build, simulate, refine, and ultimately optimize any mechanical system, from automobiles and trains to VCRs and backhoes.

The MSC.ADAMS Basic Full Simulation Package training guide teaches you how to build, simulate, and refine a mechanical system using MSC.Software's MSC.ADAMS Full Simulation Package.

## What's in this section:

- A Brief History of MSC.ADAMS, 10
- About MSC.Software, 11
- Content of Course, 12
- Getting Help, 13

# A Brief History of MSC.ADAMS

---

ADAMS: Automatic Dynamic Analysis of Mechanical Systems.

Technology was implemented about 25 years ago.

Mechanical Dynamics Incorporated (MDI) formed by researchers who developed the base ADAMS code at University of Michigan, Ann Arbor, MI, USA. MDI has been part of MSC.Software Corporation since 2002.

Large displacement code.

Systems-based analysis.

Original product was ADAMS/Solver, an application that solves nonlinear numerical equations. You build models in text format and then submit them to ADAMS/Solver.

In the early 90's, ADAMS/View was released, which allowed users to build, simulate, and examine results in a single environment.

Today, industry-specific products are being produced, such as ADAMS/Car, ADAMS/Rail, and ADAMS/Engine.

# About MSC.Software

---

Find a list of MSC.Software products at:

- <http://www.mscsoftware.com/products/products.cfm>

Find a list of MSC.ADAMS products at:

- [http://www.mscsoftware.com/products/products\\_detail.cfm?PI=413](http://www.mscsoftware.com/products/products_detail.cfm?PI=413)

Find additional training at:

- <http://www.engineering-e.com/training/>
- Or your local support center

Run through verification problems at:

- <http://support.adams.com/kb/faq.asp?ID=kb9587.dasp>

# Content of Course

---

## After taking this course you will be able to:

- Build ADAMS/View models of moderate complexity.
- Understand MSC.ADAMS product nomenclature and terminology.
- Understand basic modeling principles and extend your proficiency by creating progressively more complex models.
- Use the crawl-walk-run approach to virtual prototyping.
- Debug your models for the most common modeling challenges (for example, redundant constraints, zero masses, and so on).
- Use and be informed about all methods of MSC.ADAMS product support.
- Use the product documentation optimally.

## Organization of guide

This guide is organized into modules that get progressively more complex. Each module focuses on solving an engineering-based problem and covers mechanical system simulation (MSS) concepts that will help you use MSC.ADAMS most optimally. The earlier workshops provide you with more step-by-step procedures and guidance, while the later ones provide you with less.

Each module is divided into the following sections:

- 1 Problem statement
- 2 Concepts
- 3 Workshop
- 4 Optional tasks
- 5 Module review

# Getting Help

## Online help

To access the online help, do either of the following:

- From the **Help** menu, select **ADAMS/View Help** to display the home page for the ADAMS/View online help.
- While working in any ADAMS/View dialog box, press **F1** to display online help specific to that dialog box.

Once the online help is displayed, you can browse through the table of contents or the index, or search for any terms.

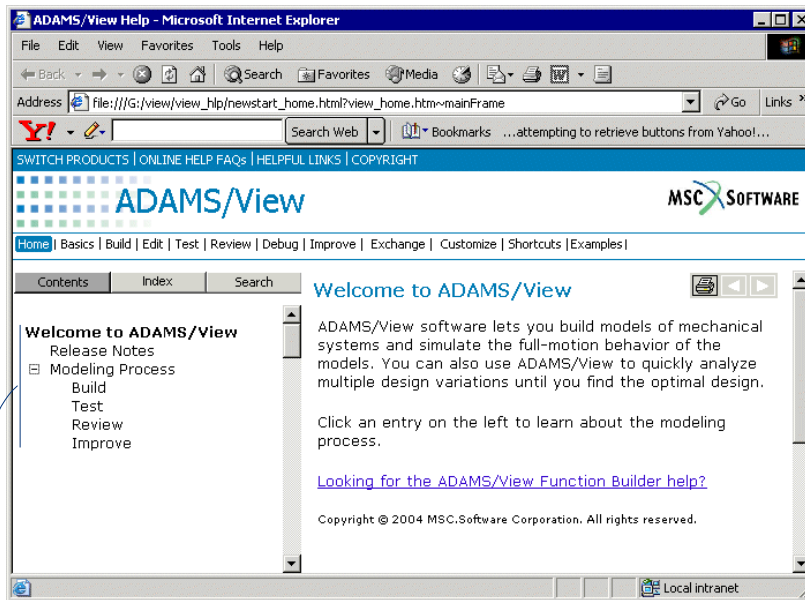


Table of contents for selected tab

# Getting Help...

---

## Technical support

To find your support center, go to <http://www.mscsoftware.com/support/contacts/index.cfm>

To read the Service Level Agreement, go to [http://www.mscsoftware.com/support/prod\\_support/adams/ADM\\_02ZZZLT\\_T\\_SERL\\_HJ\\_R6.pdf](http://www.mscsoftware.com/support/prod_support/adams/ADM_02ZZZLT_T_SERL_HJ_R6.pdf)

## Knowledge base

Go to <http://support.adams.com/kb>

For a quick tour, go to <http://www.adams.com/news/newsletter/vol3/kbtour.htm>

## Consulting services:

<http://www.mscsoftware.com/services/esg/>

## MSC Virtual Product Development Community

- To join the community of MSC.Software users, go to:  
<http://forums.mscsoftware.com>.
- Select **MSC.ADAMS** to view the MSC.ADAMS discussions.
- Select **MSC News** to view product alerts and company news and events.

The screenshot shows the MSC.ADAMS Discussion Community forum page. The browser title is "MSC.ADAMS Discussion Community: Viewing list of forums - Microsoft Internet Explorer". The address bar shows "http://forums.mscsoftware.com/adams/ubbthreads.php". The page features a navigation menu with "MSC.ADAMS" highlighted. Below the menu is a table of forum threads.

General	Threads	Posts	Last post
<b>Full Simulation Package</b> Solver, View, PostProcessor, Linear, and other general questions. Moderator: jslat	1505 (76 new)	6215 (323 new)	Re: Is it possible to get ... (alain77) - 07/23/04 08:46 AM
<b>University</b> New users, university students and faculty, and Formula SAE/Mini-Baja questions. Moderator: jjanevic	1028 (7 new)	4166 (12 new)	Re: faceting tolerance - c... (asult) - 07/22/04 07:27 PM
<b>Suggestions/Feedback</b> Have a suggestion about how this web based forum tool should work? Have a feature request for the system? This is the place for you... Moderator: Jeffrey_Graff	65 (55 new)	297 (260 new)	Re: Favourite Thread &... (jkypa) - 07/20/04 06:46 AM
<b>Webinar Questions</b> Questions and Answers from recent MSC.ADAMS Webinar presentations. Moderator: michael_brewster	26 (26 new)	48 (48 new)	Re: apply constrain or joi... (ijake) - 04/11/04 03:10 AM

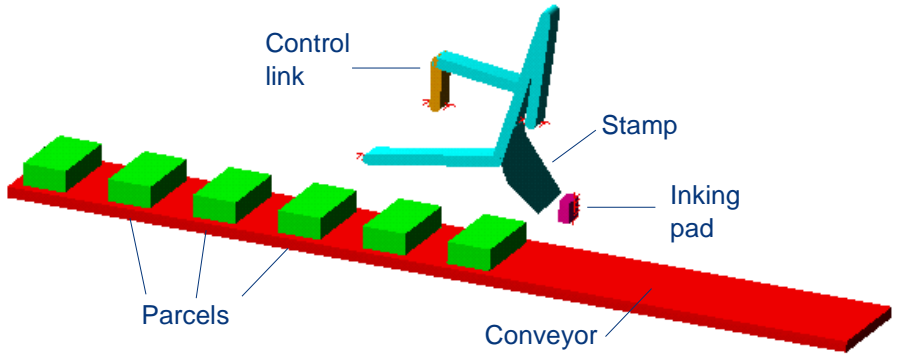
Industry-Specific Products	Threads	Posts	Last post
<b>Car</b> Discussions about ADAMS/Car, including use with ADAMS/Driver. Moderator: apert	646	2813 (2 new)	Re: SDI and ADAMS Four Pos... (brian_edwards) - 07/23/04 10:32 AM
<b>Pre, Chassis</b> Discussions about ADAMS/Pre and ADAMS/Chassis, including use with ADAMS/Insight and ADAMS/Driver. Moderator: jkypa	186 (179 new)	856 (839 new)	Re: information after runn... (Dan_Leo) - 07/23/04 11:59 AM
<b>Engine</b> Discussions about ADAMS/Engine. Moderator: hskov	131 (12 new)	501 (51 new)	Re: harmonic series file (stuecklschwaiger) - 07/23/04 03:53 AM



# 1

## STAMPING MECHANISM

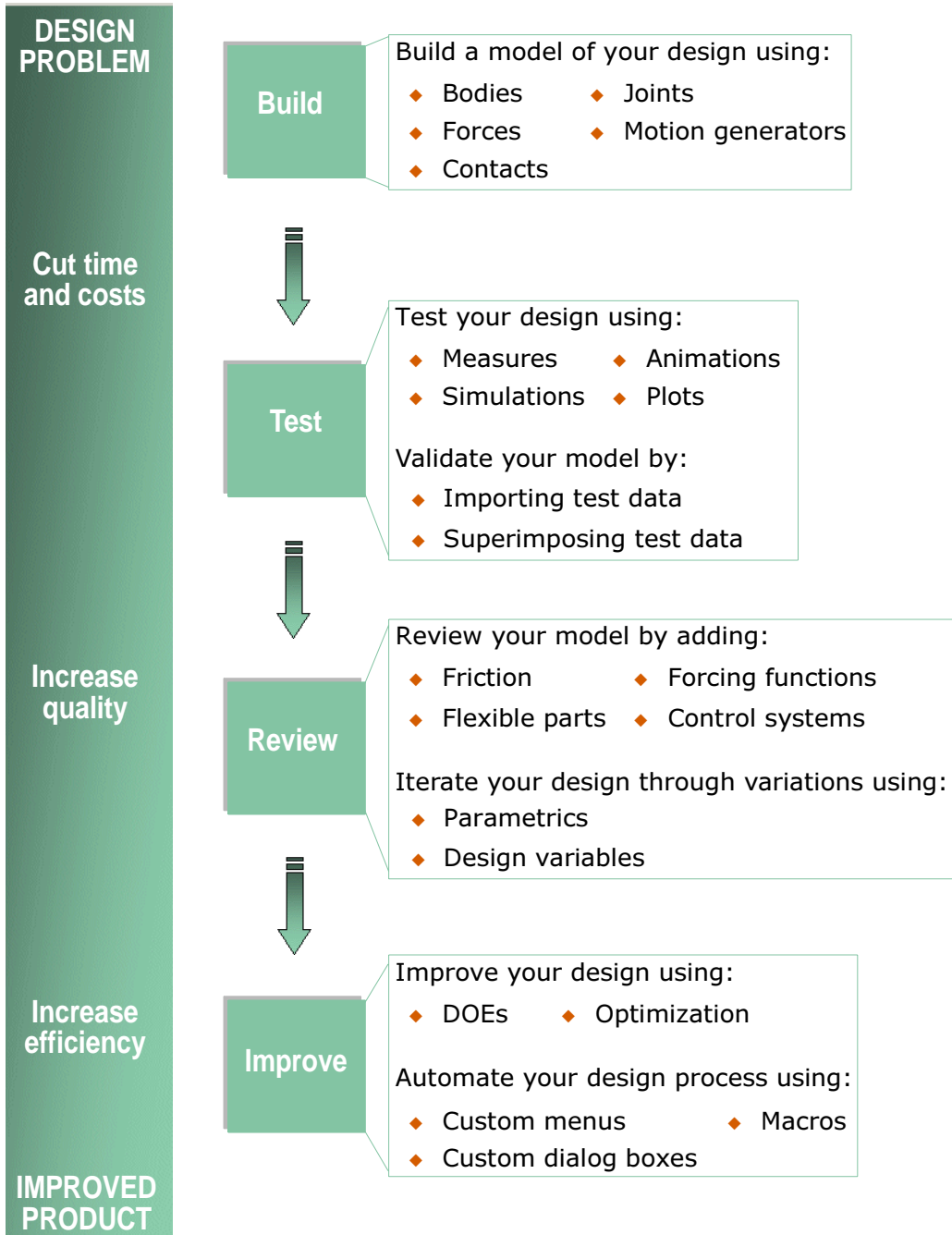
Understand the virtual prototyping process by improving the design of the stamping mechanism shown next:



### What's in this module:

- Virtual Prototyping Process, 18
- Workshop 1—Stamping Mechanism, 19
  - ◆ Module review, 25

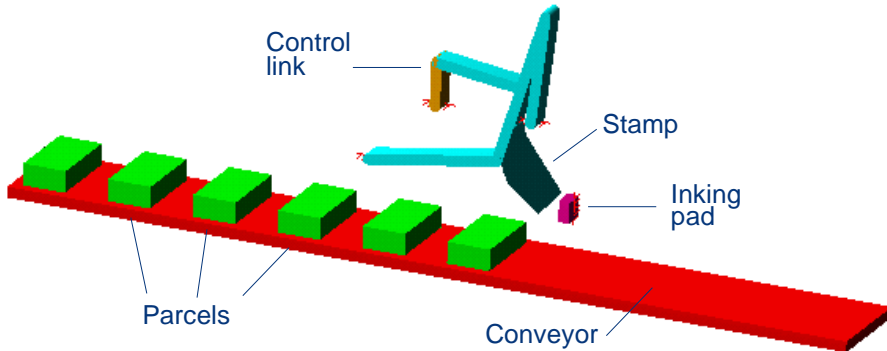
# Virtual Prototyping Process



# Workshop 1—Stamping Mechanism

## Problem statement

Understand the virtual prototyping process by improving the design of the stamping mechanism shown next:



## Model description

- This model represents a mechanism for stamping parcels that are moving along a conveyor belt.
- During the work cycle, the stamp does not contact the parcels that it is supposed to label.
- To fix this design flaw, modify the length of the control link.

# Workshop 1—Stamping Mechanism...

---

## Start the workshop

Your first step will be to start ADAMS/View from the directory *exercise\_dir/mod\_01\_stamper*. When you start ADAMS/View from that directory, it automatically builds the model stamp and a fully customized version of ADAMS/View.

We provide separate instructions for starting ADAMS/View in UNIX and Windows. Follow the set of instructions below depending on the platform you are on.

### If on Windows, to start ADAMS/View:

- 1 From the **Start** menu, point to **Programs**, point to **MSC.Software**, point to **MSC.ADAMS 2005**, point to **AView**, and then select **ADAMS - View**.

ADAMS/View starts and the Welcome dialog box appears.

- 2 From the Welcome dialog box, select **Import a file**, and then click the file folder.

The Select Directory dialog box appears.

- 3 Find and select the directory **mod\_01\_stamper** (*exercise\_dir/mod\_01\_stamper*).

- 4 Select **OK**.

The File Import dialog box appears.

- 5 Set **File Type** to **ADAMS/View Command File (\*.cmd)**.

- 6 Right-click the **File to read** text box, and then select **Browse**.

The Select File dialog box appears.

- 7 Select the file, **aview.cmd**, and then select **Open**.

- 8 Select **OK**.

ADAMS/View imports the commands to build:

- Model named stamp.
- Fully customized version of ADAMS/View.

After importing the commands, an Information window appears.


- 9 Read the information describing the model, and then, in the upper right corner, select **Close**.

# Workshop 1—Stamping Mechanism...

---

If on UNIX, to start ADAMS/View:

**1** At the command prompt, enter the command to start the MSC.ADAMS Toolbar, and then press **Enter**. The standard command that MSC.Software provides is `adamsx`, where `x` is the version number, for example `adams05`, which represents MSC.ADAMS 2005.

**2** From the MSC.ADAMS toolbar, right-click the **ADAMS/View** tool .

**3** Select **Change Settings for A/View**.

The Change Settings for A/View dialog box appears.

**4** Select **Working directory**.

**5** Right-click the **Working directory** text box, and then select **Select a Directory**.

The Select a Directory dialog box appears.

**6** Select the directory `mod_01_stamper` (*exercise\_dir/mod\_01\_stamper*).

**7** Select **OK**.

**8** From the **Change Settings for A/View** dialog box, select **OK**.

**9** From the MSC.ADAMS toolbar, select the **ADAMS/View** tool.

ADAMS/View starts and automatically imports the commands to build:

- Model named stamp.

- Fully customized version of ADAMS/View.

After importing the commands, the Information window appears.

**10** Read the information describing the model, and then, in the upper right corner, select **Close**.

# Workshop 1—Stamping Mechanism...

---

## Change the model

In this section, you see how you can change the length of the control link (control\_link).

To change the model:

- 1 From the **Stamper** menu, select **Setting Up Model**.  
The Stamper\_Setup dialog box appears.
- 2 Use the arrow buttons to modify the length of the control\_link.
  - The buttons shift the location of the top of the control\_link upward and downward 3 mm at a time.
  - The parts connected to the control\_link are parameterized in such a way as to move the appropriate amount automatically whenever you adjust the length of control\_link.
- 3 Watch the model change as you press these buttons.
- 4 To reset your model to the original configuration, select **Reset**.  
Leave the Stamper\_Setup dialog box open and continue with the next step.

## Simulate the model

Now, you'll simulate the model to see how it behaves.

To simulate the model:

- 1 From the **Stamper** menu, select **Simulate**.  
The Stamper\_Simulate dialog box appears.
- 2 To simulate the current design variation, ensure that **Single** is selected.
- 3 To solve the equations of motion for the current design, select **Apply**.  
**Note:** You selected to display the model at every output step. If you were to change **Model Update** from **At Every Output Step** to **Never**, the model would not update on the screen but would solve faster.  
When a single simulation is completed, ADAMS/View tells you what the penetration was during the simulation. A positive number indicates penetration.
- 4 To continue, select **OK**.
- 5 Leave the Stamper\_Simulate dialog box open and continue with the next step.

# Workshop 1—Stamping Mechanism...

## Investigate results

Now you'll look at the results of the simulation as an animation and a plot.

To investigate results:

- 1 From the **Stamper** menu, select **Investigate Results**.

The `Stamper_Investigate` dialog box appears.

- 2 To see the motion resulting from your last simulation, select **Animate Results**.

If necessary, use the stop sign in the lower right corner of the window to stop an animation before it has completed.

- 3 To plot the vertical travel of the stamper with respect to the parcel tops versus time, as calculated from your last simulation, select **Measure Stamp Height above Parcels**.

A stripchart appears, which shows a plot the height of the stamp above the parcels.

- 4 To save an existing curve so that the next simulation curve will not overwrite the existing curve, but will be superimposed on the saved curve, select **Save Curve**.

## Manually find the correct height

Now change the model again to find the correct height at which the stamp makes minimal contact with the parcels.

To find the correct height:

- Repeat the steps on the previous page until you can identify the `control_link` length at which the stamp makes contact with the parcels, using 3 mm increments. Use this value to answer Question 1 in [Module review](#), on page 25.

If `stamp_height > 0`, stamper does not make contact with parcels.

If `stamp_height < 0`, stamper makes contact with parcels.

# Workshop 1—Stamping Mechanism...

## Perform a design study

Now you'll perform a design study. The design study automatically analyzes the model using the specified upper and lower limits for control\_link length, and the specified number of runs. Default values are given, but you can modify them if desired.

To perform a design study:

- 1 On the Stamper\_Simulate dialog box, select **Design Study**.
- 2 To speed up the simulation, set **Model Update** to **Never**.
- 3 Select **Apply** to submit the design study.

The design study automatically analyzes the model and a stripchart and Information window appears when the study is complete.

- 4 From the Information window, identify the range of the control\_link length values within which the stamp makes contact with the parcels. Use this range to answer Question 2 in [Module review](#), on page 25.
- 5 Close the Information window.

## Perform an optimization study

Now, you'll perform an optimization study. During an optimization study, ADAMS/View systematically varies the control\_link length and runs a number of simulations until the specified penetration is achieved to within a set tolerance.

To perform an optimization study:

- 1 On the Stamper\_Simulate dialog box, select **Optimization**.
- 2 Set the **Desired Penetration** to **4 mm**.

**Note:** ADAMS/View wraps the 4 mm in parentheses () to denote an expression. If you did not enter units, ADAMS/View uses the default units set for the model.

- 3 Set **Model Update** to **Never**.
- 4 Select **Apply** to submit the optimization study.

The Information window appears, displaying the control\_link length for a maximum penetration of 4.00.

- 5 From the displayed value of the control link length, note the maximum penetration. Use this value to answer Question 3 in [Module review](#), on page 25.

# Workshop 1—Stamping Mechanism...

---

6 Select OK.

The value on the Stamper\_Setup dialog box also updates to the optimized value.

7 Exit ADAMS/View:

- From the **File** menu, select **Exit**.
- From the dialog box that appears, select **Exit, don't save**.

## Module review

1 Using 3 mm increments, at what control link length do you first notice penetration?

---

---

2 From the design study, what control link length results in penetration? How does this compare with your previous results?

---

---

3 If you specify a maximum desired penetration of 4 mm, what is the optimal length of the control link? How close is the maximum actual penetration from the maximum desired penetration?

---

---

4 How many moveable parts does the model consist of?

---

---

5 How many joints does the model consist of?

---

---

6 What would happen if you deleted the conveyor belt?

---

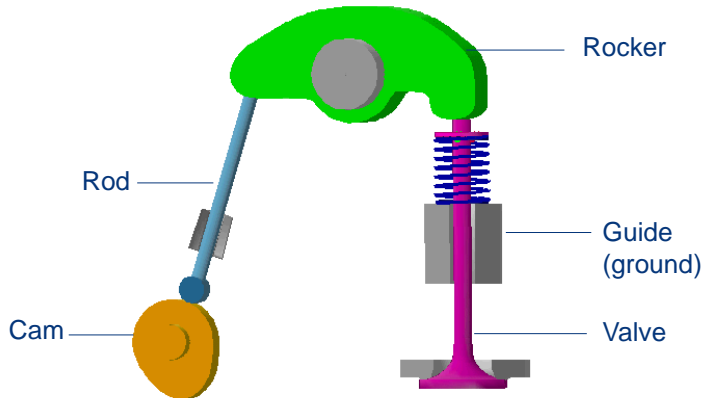
---



# 2

## ADAMS/VIEW INTERFACE OVERVIEW

Use the ADAMS/View graphical-user interface (GUI) to manipulate, simulate, review, and refine the model shown next:



For more information, see the ADAMS/View online help.

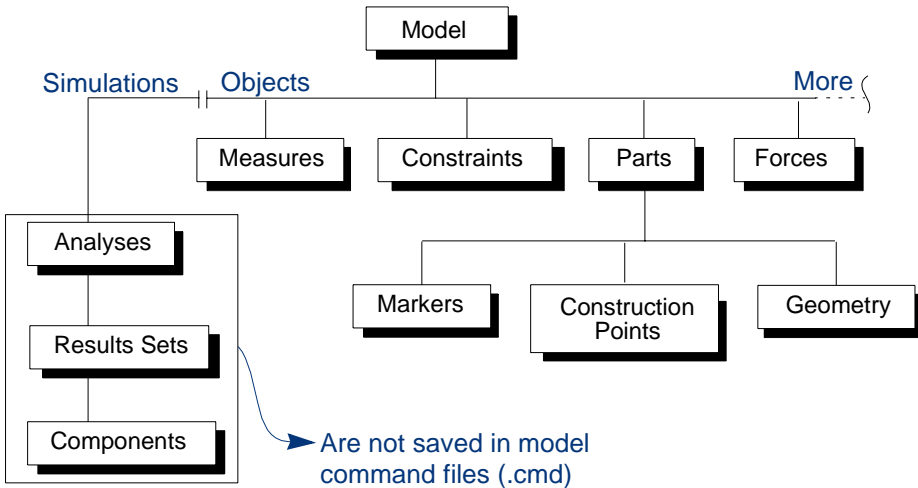
### What's in this module:

- Model Hierarchy, 28
- Renaming Objects, 29
- ADAMS/View Interface, 30
- Simple Simulations, 31
- Saving Your Work, 32
- Workshop 2—ADAMS/View Interface Overview, 34
  - ◆ Module review, 43

# Model Hierarchy

## ADAMS/View modeling hierarchy

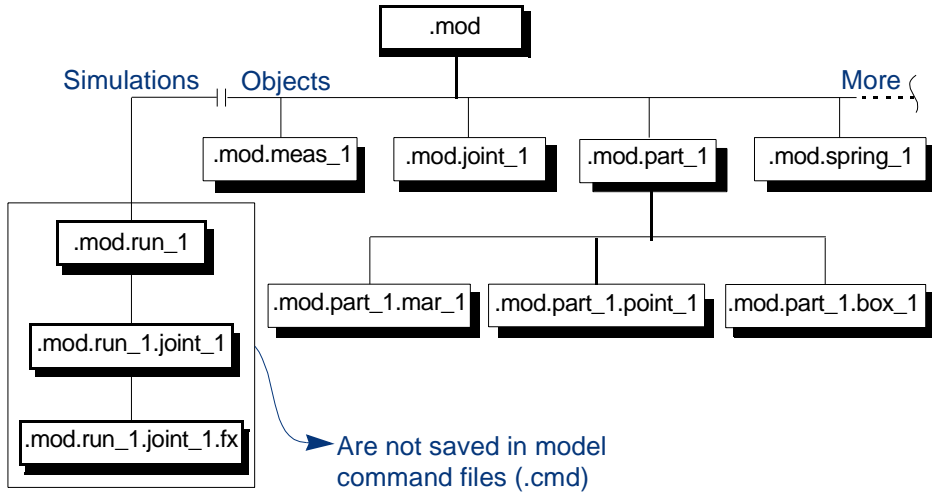
- ADAMS/View names objects based on this model hierarchy. For example, ADAMS/View names geometry as .model\_name.part\_name.geometry\_name.
- To change the parent for an object, rename the object.



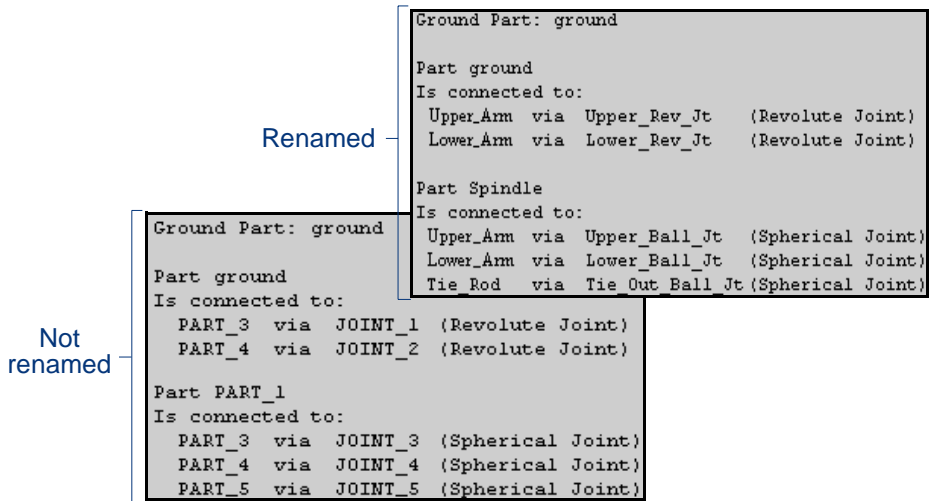
**See also:** [Assembling Subsystem Models](#), on page 198

# Renaming Objects

## ADAMS/View naming conventions

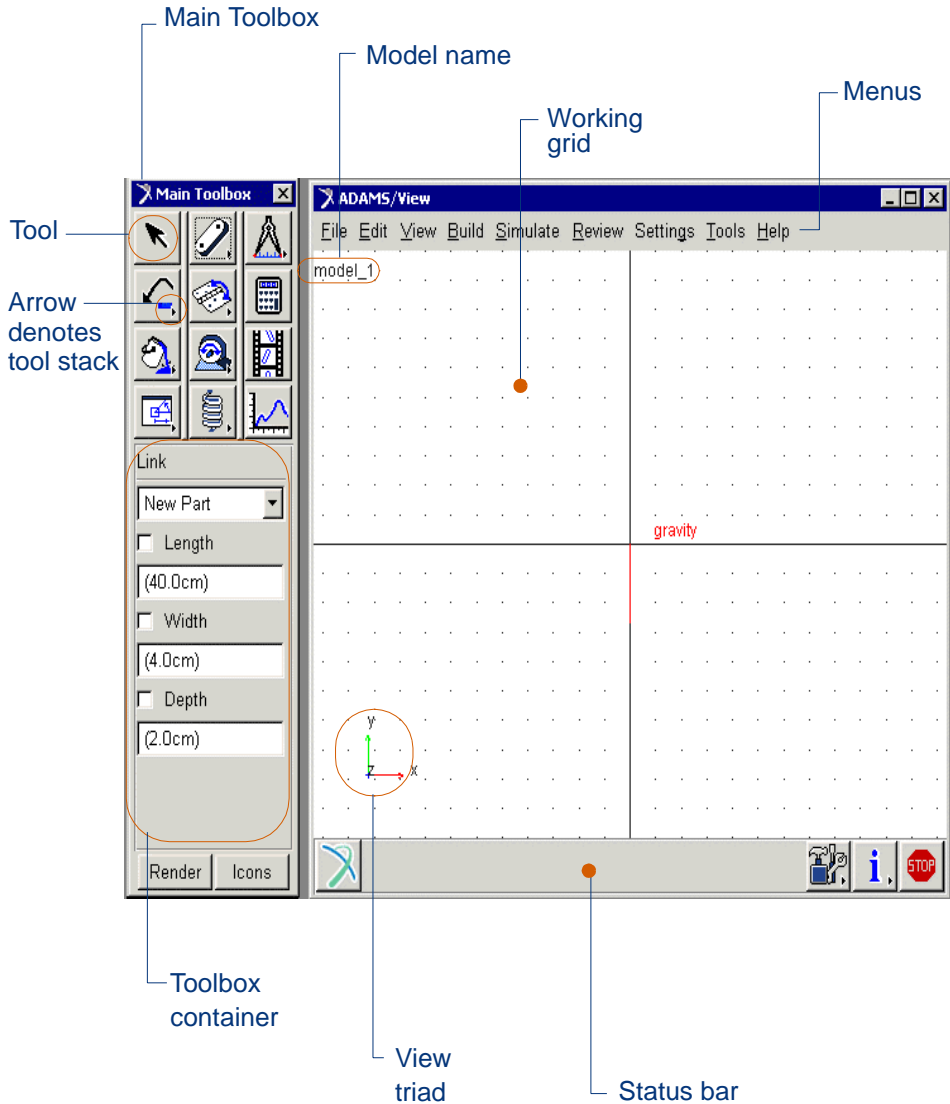


## Renaming objects clarifies model topology as follows



# ADAMS/View Interface


## ADAMS/View main window



# Simple Simulations

## Simulation versus animation

- Simulations are solutions to equations of motion describing a mechanical system.
- Animations display a graphical playback of previously completed simulations.



The image shows a software interface titled "Main Toolbox" with a grid of icons. The "Simulation" section is highlighted, showing playback controls, a "Default" dropdown, an "End Time" field with the value "5.0", a "Steps" field with the value "50", a "No Debug" dropdown, and "Render" and "Icons" buttons. Annotations with lines point to the "Simulation tool" icon, the "Animation tool" icon, the "End Time" field, and the "Steps" field.

Simulation time interval

**End time:** absolute point in time to stop simulation

**Duration:** relative amount of time to simulate over

Simulation tool

Animation tool

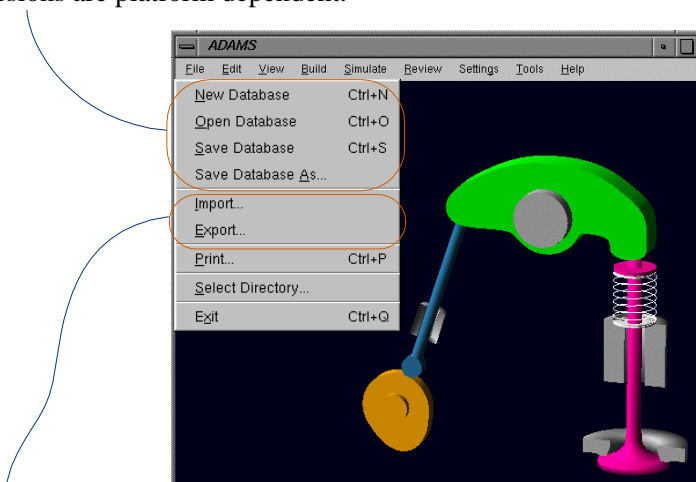
Simulation output

**Step size:** amount of time between steps

**Steps:** total number of steps in a specified amount of time

## Most common formats in which you can save ADAMS/View models

- ADAMS/View database files (.bin)
  - ◆ Include the entire modeling session including models, simulation results, plots, and so on.
  - ◆ Are typically very large.
  - ◆ Are platform independent in MSC.ADAMS, as of version 11.0, but all other versions are platform dependent.



- ADAMS/View command files (.cmd)
  - ◆ Include only model elements and their attributes.
  - ◆ Are relatively small, editable text files.
  - ◆ Are platform independent.

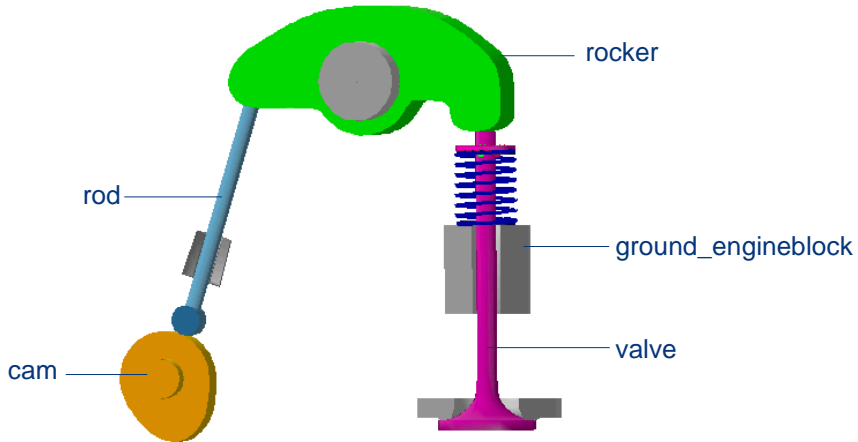
## Other formats in which you can import and export data

- ADAMS/Solver input files (.adm)
- Geometry files (STEP, IGES, DXF, DWG, Wavefront, Stereolithography)
- Test and spreadsheet data files
- Simulation results files (.msg, .req, .out, .gra, .res).

# Workshop 2—ADAMS/View Interface Overview

## Problem statement

Use ADAMS/View to manipulate, simulate, review, and refine the following model:



## Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a given velocity.
- The rod (follower) moves translationally based on its constraint to the cam.
- The rocker pivots about a pin attached to the engine block.
- The spring is always in compression to try and keep the rod in contact with the cam.
- The valve moves vertically as the rocker rotates.
- When the valve moves, it lets small amounts of air into the chamber below it (not modeled here).

## Tips before you start

While working on this exercise, notice:

- The use of the right mouse button.
- The function of single-clicks and double-clicks.
- The messages on the Status bar.
- The animation options.


## Start the workshop

Start ADAMS/View from the directory *exercise\_dir/mod\_02\_aview\_interface* and import the model command file *valve.cmd*. It contains commands to build a model named *valve*.

To start ADAMS/View in Windows:

- On the **Start** menu, point to **Programs**, point to **MSC.Software**, point to **MSC.ADAMS 2005**, point to **AView**, and then select **ADAMS - View**.

To start ADAMS/View in UNIX:

- From the MSC.ADAMS Toolbar, select the **ADAMS/View** tool .

To load the workshop files:

- 1 From the Welcome dialog box, select **Import a file**.
- 2 Click the file folder.  
The Find Directory dialog box appears.
- 3 Find and select the directory **mod\_02\_aview\_interface** (*exercise\_dir/mod\_02\_aview\_interface*).
- 4 Select **OK**.  
The File Import dialog box appears.
- 5 Set **File Type** to **ADAMS/View Command File (\*.cmd)**.
- 6 Right-click the **File to read** text box, and then select **Browse**.  
The Select File dialog box appears.

## Workshop 2—ADAMS/View Interface Overview...

- 7 Find and select the file, `valve.cmd` and then select **Open**.
- 8 Select **OK**.

### View the model

Now you'll learn how you can view models from different angles using the keyboard shortcuts for zooming, translating, and rotating.

To view the model from different angles:

- 1 To view a list of keyboard shortcuts, move the cursor away from the model, and then right-click in the ADAMS/View window.  
A menu appears, listing the keyboard shortcuts. To close the menu, left-click away from the menu.
- 2 In the space below, write the shortcut keys for performing the following view operations.

Rotate:

---

Translate:

---

Zoom in and out:

---

Zoom into a specific area:

---

Fit:

---

Front view:

---

- 3 Press the key representing the desired view operation, and follow the instructions in the Status bar.

# Workshop 2—ADAMS/View Interface Overview...

## Rename parts

Now you'll rename the parts to match the names given in the figure in the problem statement on page 34.

As you go through these instructions, notice that right-clicking always gives you a list of choices, while left-clicking selects an object.


### To rename parts:

- 1 Move the cursor over a part and right-click. (For example, move the cursor over the rocker part.)
- 2 Point to **Part:PART\_<x>**, and then select **Rename**.  
The Rename Object dialog box appears.
- 3 In the **New Name** text box, enter **.valve.<part name>**, and then select **OK**. (For example, for the rocker, you would enter: **.valve.rocker**.) See the problem statement on page 34 for a listing of part names.
- 4 Continue renaming parts.

## Inspect the model

Now inspect the model to determine the number and type of constraints in the model and check if the model verified correctly. Use the values to answer Question 1 in [Module review](#), on page 43.

### To determine the number and type of constraints:

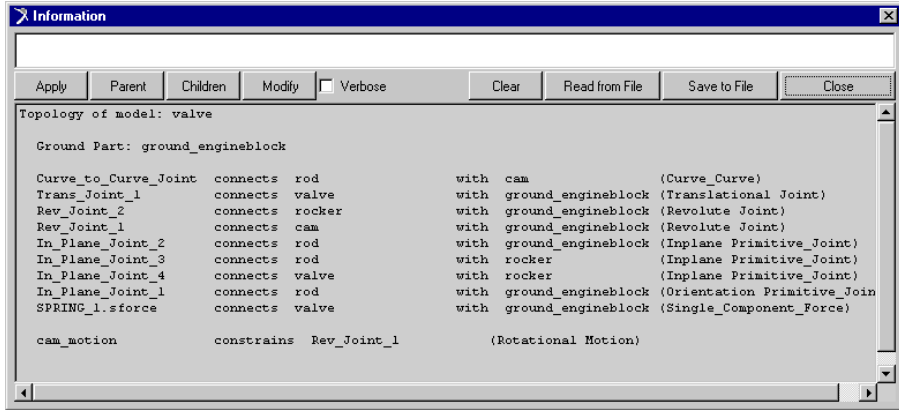
- 1 Right-click the **Information** tool stack  on the right side of the **Status** bar, and then select the **Model topology by constraints** tool.



Model topology  
by constraints tool

# Workshop 2—ADAMS/View Interface Overview...

The Information window appears as shown next:

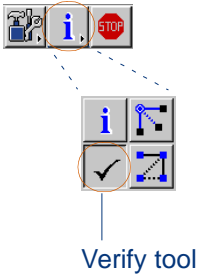


- 2 Note the number and type of constraints and use them to answer Question 1 in [Module review](#), on page 43.
- 3 Select Close.

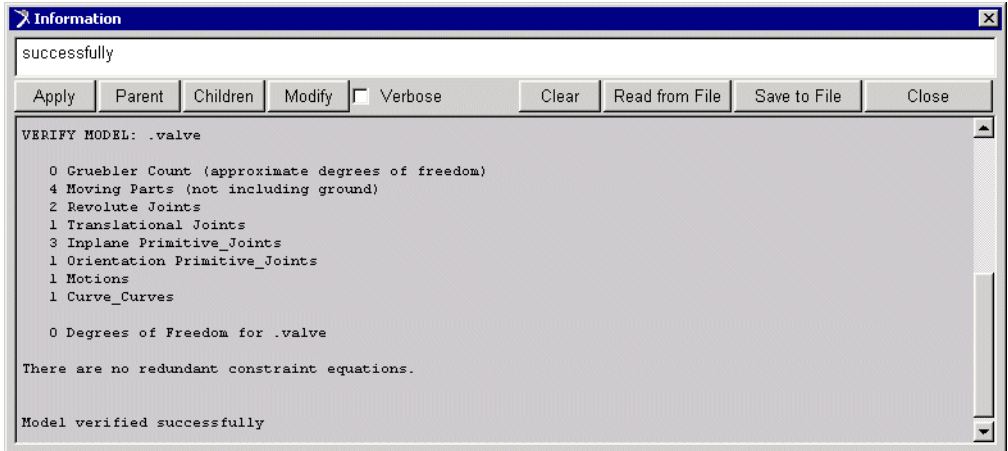
# Workshop 2—ADAMS/View Interface Overview...

To check if the model verified successfully:

- 1 Right-click the **Information** tool stack , again, and then select the **Verify** tool.



The Information window appears as shown next:




Note that the text **Model verified successfully** appears in the Information window.

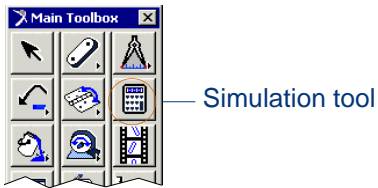
- 2 Select **Close**.

## Simulate the model

In this section, you'll run a simulation for 2 seconds with 100 steps, and save the simulation results.



To run a simulation:

- 1 From the Main Toolbox, select the **Simulation** tool .



- 2 In the container that appears in the lower portion of the Main Toolbox:


- Select **Default**.
- Select **End Time**, and in the text box, below **End Time**, enter **2.0**.
- In the text box below **Steps**, enter **100**.

- 3 Select the **Play** tool .
- 4 When the simulation is complete, select the **Reset** tool .

To save the simulation results:

- 1 From the **Simulate** menu, select **Interactive Controls**.

The Simulation Control dialog box appears.

- 2 To save the last simulation results to the database under a new name, select the **Save Simulation** tool .


The Save Run Results dialog box appears.

- 3 In the **Name** text box, enter a name for the simulation results, such as **first\_results**.
- 4 Select **OK**.
- 5 Close the Simulation Control dialog box.

## Animate the results

In this section, you'll review the results of the simulation as an animation, which is a graphical playback of a simulation. You'll use the built-in ADAMS/View tools to run the animation. Optionally, you could use the ADAMS/PostProcessor tools to run animations.

### Animate the model with icons turned off (default):

- 1 From the Main Toolbox, select the **Animation** tool .
- 2 Select the **Play** tool.
- 3 When the animation is complete, select the **Reset** tool.

### Animate the model with icons turned on:

- 1 From the **Review** menu, select **Animation Controls**.  
The Animation Controls dialog box appears.
- 2 At the bottom of the Animation Controls dialog box, select **Icons**.
- 3 Select the **Play** tool.
- 4 When the animation is complete, select the **Reset** tool.
- 5 Close the Animation Controls dialog box.

## Save your work

Now you'll save your work so the saved file contains only the model information.

To save your work:

- 1 From the **File** menu, select **Export**.
- 2 Set **File Type** to **ADAMS/View Command File**.
- 3 In the **File Name** text box, enter **valve1**.
- 4 In the **Model Name** text box, enter **valve**.
- 5 Select **OK**.
- 6 If you want to further explore the model, as suggested in the next section, leave the model open. Otherwise, proceed with the next step.
- 7 From the **File** menu, select **Exit**.
- 8 From the dialog box that appears, select **Exit, don't save**.

## Optional tasks

Have fun with the model:

This exercise introduces you to the ADAMS/View interface. Manipulate the model and experiment with it as much as you want.

# Workshop 2—ADAMS/View Interface Overview...

## Module review

**1** How many constraints are there in this system? What type of constraints are they?

---

---

**2** Is it possible to have more than one model in a database?

---

---

**3** Is geometry a direct child of a model? If not, what is geometry a child of?

---

---

**4** If you are in the middle of an operation and you are not sure what input ADAMS/View wants next, where should you look?

---

---

**5** If you are working with our technical support staff and you want them to look at one of your files, what file format would you send them, a .cmd or .bin? Why?

---

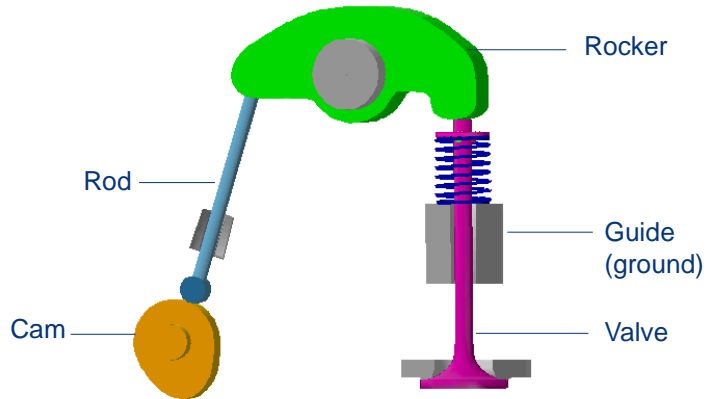
---



# 3

## ADAMS/PostProcessor INTERFACE OVERVIEW

Use the ADAMS/PostProcessor interface to simulate, review, and refine the model shown next:



For more information, see the ADAMS/PostProcessor online help.

### What's in this module:

- PostProcessing Interface Overview, 46
- Animating, 47
- Plotting, 48
- Reporting, 49
- Workshop 3—ADAMS/PostProcessor Overview, 50
  - ◆ Module review, 60

# PostProcessing Interface Overview

---

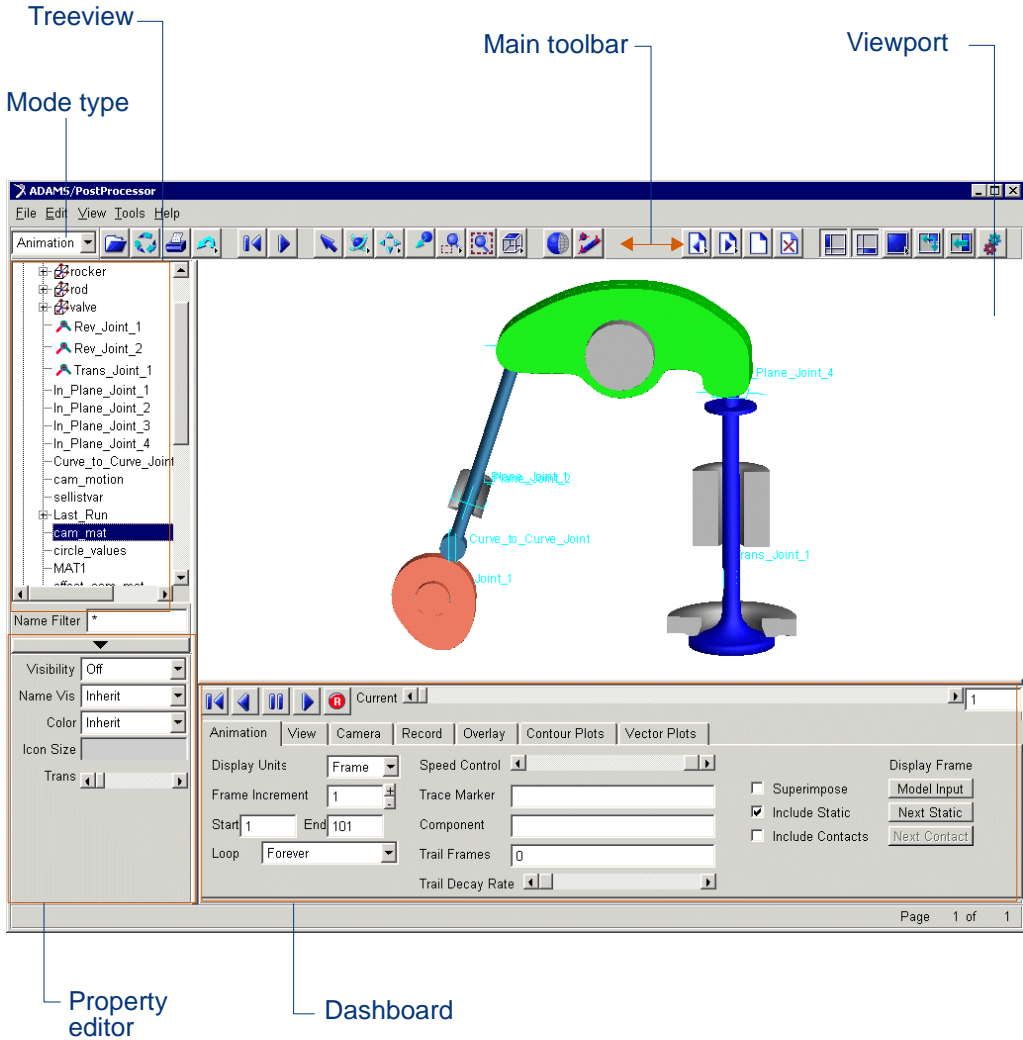
ADAMS/PostProcessor has three modes:

- Animation
- Plotting
- Report
- Plot 3D (Available only for ADAMS/Vibration analyses)

## Example:

The tools in the Main toolbar change if you switch between the modes, as shown on the next few pages.

# Animating



For more information, see the Animate tab in the ADAMS/PostProcessor online help.

# Plotting

Treeview

Mode type

Main toolbar

Viewport

ADAMS/PostProcessor

File Edit View Plot Tools Help

Plotting

page\_1

- plot\_1
  - date
  - title
  - curve\_1
  - curve\_2
  - haxis
  - vaxis
  - legend\_object
- page\_2

Name Filter \*

Legend Curve\_to\_Curve\_...

Color Red

Line Style Solid

Line Weight 2.0

Symbol None

Symbol Inc 1

Hotpoints No

Name Filter

Force (pound\_force)

Time (sec)

2003-08-08 15:02:53

valve

Curve\_to\_Curve\_Joint.FX

Curve\_to\_Curve\_Joint.FY

Simulation	Filter	Result Set	Component	Surf
Last_Run (2003-08-08)		cam_motion	A1	<input type="checkbox"/>
		cam_XFORM	A2	<input type="checkbox"/>
		Curve_to_Curve_Joint	FX	<input type="checkbox"/>
		Displ_RodTip_along_Global_Yax:	FY	<input type="checkbox"/>
		Displ_RodTip_along_RodAxis	FZ	<input type="checkbox"/>
		FMA87_XFORM	TX	<input type="checkbox"/>
		FMA88_XFORM	TY	<input type="checkbox"/>
		Force_in_Spring	TZ	<input type="checkbox"/>
		In_Plane_Joint_1		<input type="checkbox"/>

Name Filter

Page 1 of 2

Property editor

Dashboard



For more information, see the Plot tab in the ADAMS/PostProcessor online help.

# Reporting

Treeview  
Mode type  
Main toolbar  
Viewport

ADAMS/PostProcessor  
File Edit View Tools Help

Report

page\_1  
├── plot\_1  
│ ├── date  
│ ├── title  
│ ├── curve\_1  
│ ├── curve\_2  
│ ├── haxis  
│ ├── vaxis  
│ └── legend\_object  
├── page\_2  
├── page\_3  
└── report\_2

FILE TYPE AND SOURCE 6  
ADAMS RESULTS FILE FTLBS\_PFRACA 0.321740493603057E+02  
ADAMS VERSION - 2003.0.0 A.RES  
MSC.Software  
Unknown  
2003-08-08 15:13:37  
!  
DATA SET TITLE 3  
!  
RESULTS TITLE 3  
!  
NUMBER OF ANALYSIS BLOCKS 2  
102 205  
!  
RIGID PARTS MAP 7  
SYESYESYES

1	5	10	15
2	6	11	16
3	7	12	17
4	8	13	18
5	9	14	19

Name Filter \*

File Name Adams.res

Font Size 10

Page 3 of 3

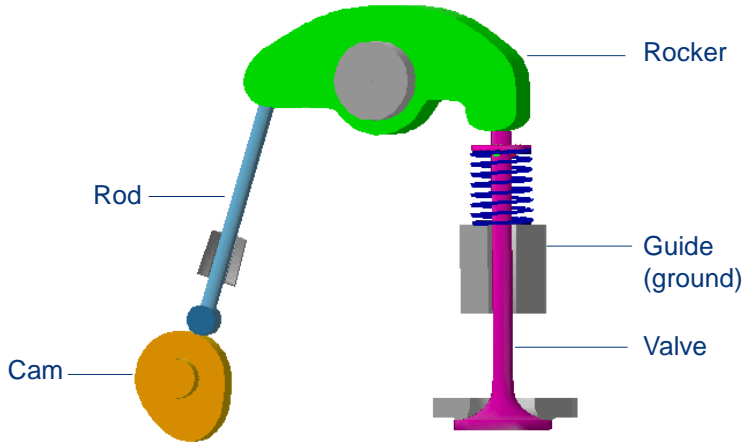


For more information, see the Report tab in the ADAMS/PostProcessor online help.

# Workshop 3—ADAMS/PostProcessor Overview

## Problem statement

Use ADAMS/PostProcessor to manipulate, review, and refine the results of the valvetrain model you simulated in the previous module.



## Model description

- The model represents a valvetrain mechanism.
- The cam is being rotated at a given velocity.
- The rod (follower) moves translationally based on its constraint to the cam.
- The rocker pivots about a pin attached to the engine block.
- The spring is always in compression to try and keep the rod in contact with the cam.
- The valve moves vertically as the rocker rotates.
- When the valve moves, it lets small amounts of air into the chamber below it (not modeled here).

## Start the workshop

Start ADAMS/View from the directory *exercise\_dir/mod\_03\_ppt\_interface* and import the model command file *valve1.cmd*. This is the command file you created in the previous workshop. The command file contains commands to build a model named *valve*.

To start the workshop:

- 1 Start ADAMS/View.
- 2 From the Welcome dialog box, select **Import a file**.
- 3 Click the file folder.

The Find Directory dialog box appears.

- 4 Find and select the directory *mod\_03\_ppt\_interface* (*exercise\_dir/mod\_03\_ppt\_interface*).
- 5 Select **OK**.

The File Import dialog box appears.

- 6 Set **File Type** to **ADAMS/View Command File (\*.cmd)**.
- 7 Right-click the **File to read** text box, and then select **Browse**.

The Select File dialog box appears.

- 8 Find and select the file, *valve1.cmd*, which you created in the previous workshop, and then select **Open**.

Note that the model file is not in the current working directory. It is in the directory *exercise\_dir/mod\_02\_aview\_interface*.

If you need a fresh copy of the model, import the command file *valve1\_completed.cmd* from the directory *exercise\_dir/mod\_02\_aview\_interface/completed*.


- 9 Select **OK**.

## Simulate the model

Run a simulation for 2 seconds with 100 steps, and then save the simulation results.

To run a simulation:

- 1 From the Main Toolbox, select the **Simulation** tool.

**Tip:** If you closed the Main Toolbox, you can display it again by clicking the Main Toolbox tool  on the Status bar.

- 2 From the container in the Main Toolbox:
  - Select **Default**.
  - Select **End Time**, and in the text box, below **End Time**, enter **2.0**.
  - In the **Steps** text box, enter **100**.
- 3 Select the **Play** tool.
- 4 When the simulation is complete, select the **Reset** tool.
- 5 Save the simulation results, with the name **second\_results**, just as you did on page 40 of [Workshop 2—ADAMS/View Interface Overview](#).

## Plot the results

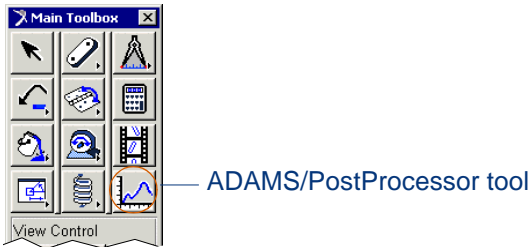
Now you'll plot the results using ADAMS/PostProcessor. You will plot:

- Displacement of the valve versus time. ADAMS/View tracks this data through a measure called `valve_displacement`.
- Force in the spring versus time. ADAMS/View tracks this data through a measure called `force_in_spring`.


# Workshop 3—ADAMS/PostProcessor Overview...

To plot the results:

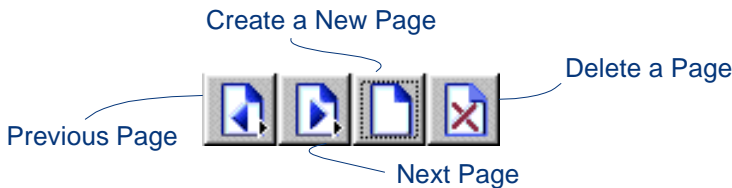
- 1 From the Main Toolbox, select the **ADAMS/PostProcessor** tool  or press F8.




ADAMS/PostProcessor starts.

- 2 At the bottom of the window in the dashboard, from the **Simulation** list, select the name of the results set you saved in the previous section.
- 3 Set **Source** to **Measures**.
- 4 From the **Measure** list, select **Valve\_Displacement**.
- 5 In the right corner of the dashboard, select **Add Curves**.
- 6 From the toolbar, select the **Create a New Page** tool .

The following figure shows the Create a New Page tool and other page tools.



In the treeview, shown on the left side of the ADAMS/PostProcessor window, you now have two pages.

- 7 From the **Measure** list, select **Force\_in\_Spring**.
- 8 Select **Add Curves**.
- 9 To return to ADAMS/View, in the upper right corner of the main toolbar, select the **ADAMS/View** tool  or press F8.

**Note:** If you resized the ADAMS/PostProcessor window, the ADAMS/View tool is no longer visible. To display the tool, stretch the window.

## Manipulate model characteristics

You'll first find the spring stiffness coefficient, and then you'll modify the spring stiffness to 200 lbf/foot.

To find the spring stiffness coefficient:

- 1 Zoom in on the spring by typing a lowercase **w**, and then drawing a window around the spring.
- 2 Right-click the spring, point to **Spring:SPRING\_1**, and then select **Info**.  
The Information window appears.
- 3 Note the value of the damping coefficient.
- 4 Use the value to answer [Question 2](#) in [Module review](#), on page 60.
- 5 Select **Close**.

To modify the spring stiffness to 200 lbf/foot:

- 1 Right-click the spring, point to **Spring:SPRING\_1**, and then select **Modify**.  
The Modify a Spring-Damper Force dialog box appears.
- 2 In the **Stiffness Coefficient** text box, enter **200 (lbf/foot)**.  
**Note:** In the value you entered, the parentheses ( ) are necessary because you enter compound fractional units.
- 3 Select **OK**.
- 4 Fit the model on the screen by typing a lowercase **f**.

## Simulate the model


Run a simulation for 2 seconds with 100 steps, and then save the simulation results (as `third_results`), just as you did in [Simulate the model](#), on page 52.

# Workshop 3—ADAMS/PostProcessor Overview...

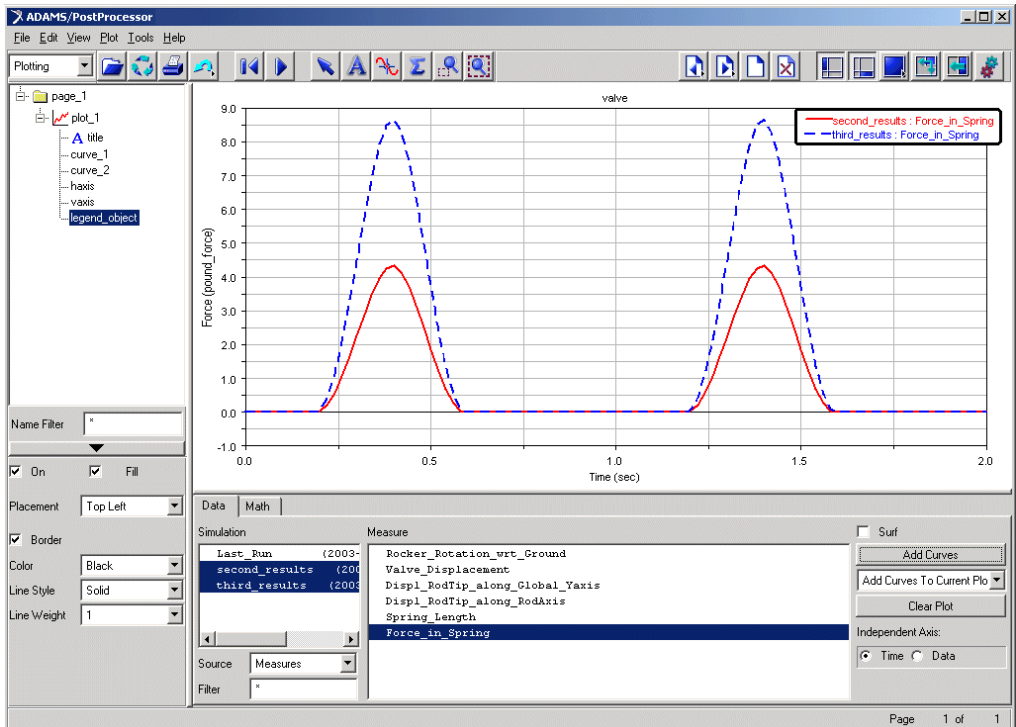
## Overlay plots of force in spring for both simulations

Now you are going to overlay the results of both of your simulations to see the differences between the spring forces.

To overlay plots:

- 1 From the Main Toolbox, select the **ADAMS/PostProcessor** tool .
- 2 From the **Simulation** list, select the new simulation in your session.
- 3 Set **Source** to **Measures**.
- 4 From the **Measure** list, select **Force\_in\_Spring**.
- 5 Below the heading **Independent Axis:**, ensure that **Time** is selected.
- 6 Select **Add Curves**.

Notice the dashboard settings in the next figure.



## Get plot statistics

Now you'll use the online help to find out how to get plot statistics and then find the plot statistics for the `force_in_spring` value.

To use the online help to help you get plot statistics:

- 1 From ADAMS/PostProcessor's **Help** menu, select **ADAMS/PostProcessor Help**.
- 2 From the **Plot** tab, select the topic **Displaying Plot Statistics**.
- 3 Use the Plot Statistics toolbar to find the maximum `Force_in_Spring` value in the second simulation.
- 4 Once you find the `Force_in_Spring` value, use it to answer Question 3 in [Module review](#), on page 60.

## Modify the plot graphics

Now you'll modify the graphics of the plot to make the information in it more readable.

To give the plot a title:

- 1 In the treeview, expand `page_2` by clicking the + sign.
- 2 Expand `plot_2`.
- 3 Select `title`.
- 4 In the property editor below the treeview, clear the current title `valve`, and then enter the new title **Spring Force vs. Time**.
- 5 Select **Enter**.

To label the vertical axis as **Spring Force (lbf)**:

- 1 In the treeview, select `vaxis`.
- 2 In the property editor, select the **Labels** tab.
- 3 Change the label to **Spring Force (lbf)**.

# Workshop 3—ADAMS/PostProcessor Overview...

To modify the legend text and its placement:

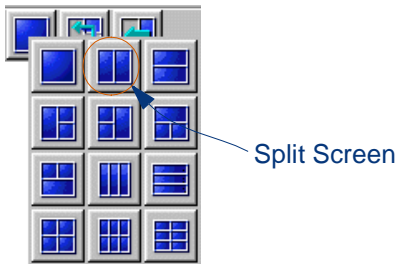
- 1 In the treeview, select **curve\_1**.
- 2 In the property editor below, change the **Legend** text box to **k=100(lbf/foot)**.
- 3 Change the legend for **curve\_2** to **k=200**.
- 4 In the treeview, select **legend\_object**.
- 5 In the property editor, set **Placement** to **Top Right**.

## Add an animation

ADAMS/PostProcessor lets you display animations and plots at the same time. In this section, you'll add an animation next to your plot. You can also run the animation and watch the results appear in the plot.

To add an animation next to your plot:

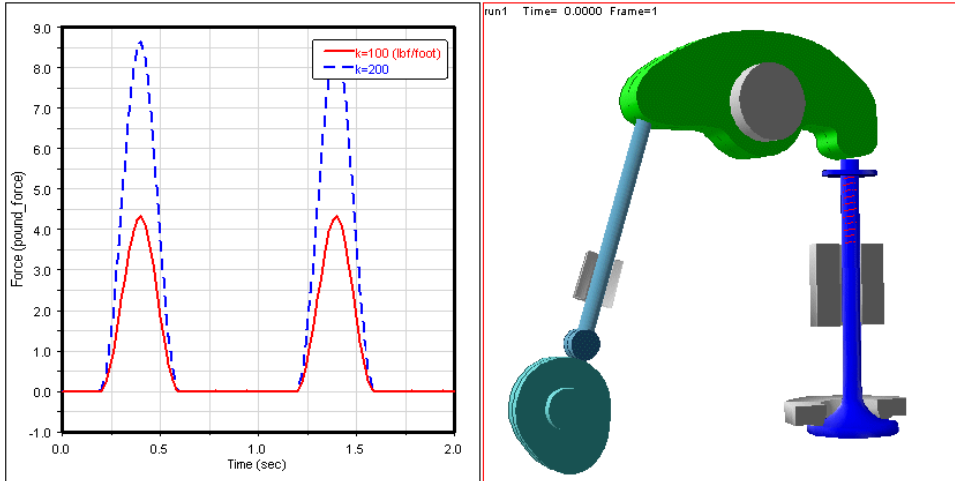
- 1 In the treeview, select **page\_2**.
- 2 Split the screen by right-clicking on the **Page Layout** tool stack in the main toolbar, and selecting the **Split Screen** tool.



- 3 Set the new viewport to **Animation** by right-clicking in the viewport and choosing **Load Animation** from the pop-up menu.

# Workshop 3—ADAMS/PostProcessor Overview...

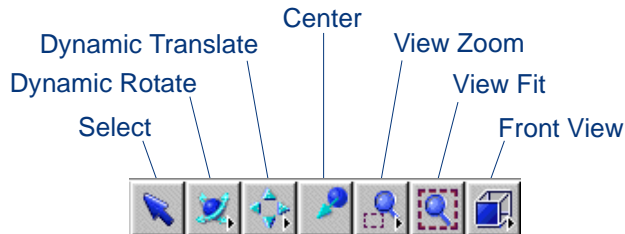
- 4 From the Database Navigator, select one of the simulation results that you want to animate.
- 5 Select OK.



## Viewing results

To view an animation of the results:

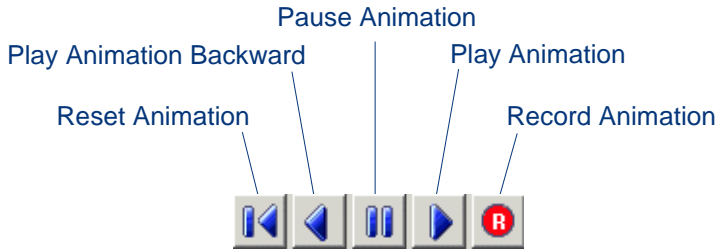
- Adjust your view of the model on your screen using the tools in the main toolbar. The figure below highlights some of the tools that are available. Try experimenting with the rotate, zoom, and translate tools.



# Workshop 3—ADAMS/PostProcessor Overview...

To play an animation of the results:

- Play an animation of your model using the tools that are located above the viewport and in the dashboard. Experiment with the play and pause tools.



## Modifying the graphics of your animation

To modify the graphics settings of your animation:

- 1 From the dashboard, select the **View** tab.
- 2 Experiment with the available options.

To change the color of the cam:


- 1 From the treeview, expand the model by clicking on the + sign.
- 2 Select **cam**.
- 3 In the property editor, set **Color** to **Coral**.

To enlarge the graphics that illustrate force:

- 1 From the **Edit** menu, select **Preferences**.  
The PPT Preferences dialog box appears.
- 2 In the **Force Scale** text box, enter a value that is greater than 50, and then select **Save**.  
**Note:** Make sure that you save your changes in this dialog box before you close it. If you do not save your changes, they will not be made.
- 3 Select **Close**.

# Workshop 3—ADAMS/PostProcessor Overview...

To change the view from shaded to wireframe:

- On the main toolbar, select the **Wireframe/shaded** tool .

You can now animate the model and view the position and direction of the reaction force.

## Save your ADAMS/PostProcessor session

To save your session:

- 1 Return to ADAMS/View.
- 2 Save your work and then exit ADAMS/View.

## Module review

- 1 What is the mass of the valve? What is this mass currently based on?

---

---

- 2 What was the damping coefficient of the spring when you first opened the model?

---

---

- 3 What was the maximum spring force when the spring coefficient was 200 lbf/foot?

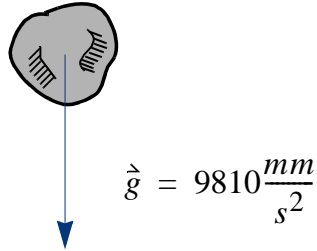
---

---

# 4 FALLING STONE

---

Find the displacement, velocity, and acceleration of a stone after one second, when the stone, with zero initial velocity, falls under the influence of gravity.



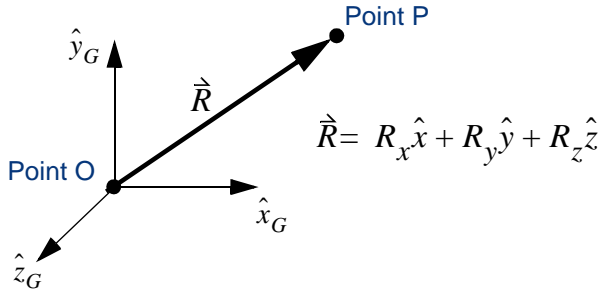
## What's in this module:

- Coordinate Systems, 62
- Part Coordinate System, 63
- Coordinate System Marker, 64
- Differences Between Parts and Geometry, 65
- Parts, Geometry, and Markers, 66
- Types of Parts in ADAMS/View, 67
- Part Mass and Inertia, 68
- Measures, 69
- Workshop 4—Falling Stone, 70
  - ◆ Module review, 77

# Coordinate Systems

## Definition of a coordinate system (CS)

- A coordinate system is essentially a measuring stick to define kinematic and dynamic quantities.



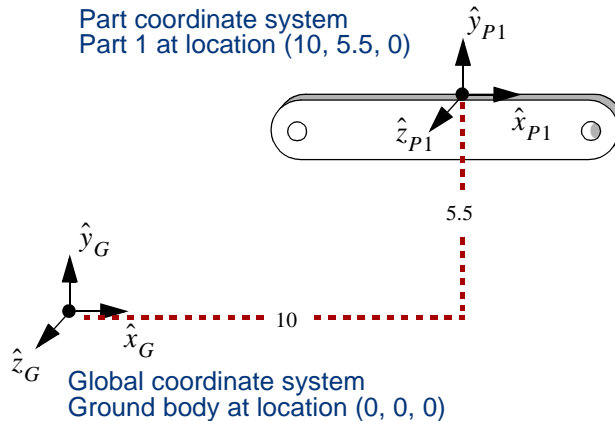
## Types of coordinate systems

- Global coordinate system (GCS):
  - ◆ Rigidly attaches to the ground part.
  - ◆ Defines the absolute point (0,0,0) of your model and provides a set of axes that is referenced when creating local coordinate systems.
- Local coordinate systems (LCS):
  - ◆ Part coordinate systems (PCS)
  - ◆ Markers

# Part Coordinate System

## Definition of part coordinate systems (PCS)

- They are created automatically for every part.
- Only one exists per part.
- Location and orientation is specified by providing its location and orientation with respect to the GCS.

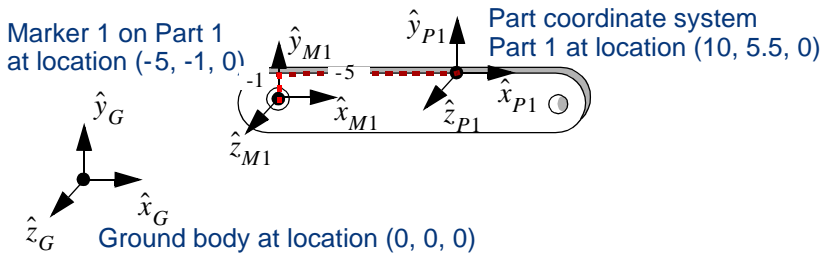


- When created, each part's PCS has the same location and orientation as the GCS.

# Coordinate System Marker

## Definition of a marker

- It attaches to a part and moves with the part.
- Several can exist per part.
- Its location and orientation can be specified by providing its location and orientation with respect to GCS or PCS.



- It is used wherever a **unique location** needs to be defined. For example:
  - ◆ The location of a part's center of mass.
  - ◆ The reference point for defining where graphical entities are anchored.
- It is used wherever a **unique direction** needs to be defined. For example:
  - ◆ The axes about which part mass moments of inertia are specified.
  - ◆ Directions for constraints.
  - ◆ Directions for force application.
- By default, in ADAMS/View, all marker locations and orientations are expressed in GCS.

# Differences Between Parts and Geometry

## Parts

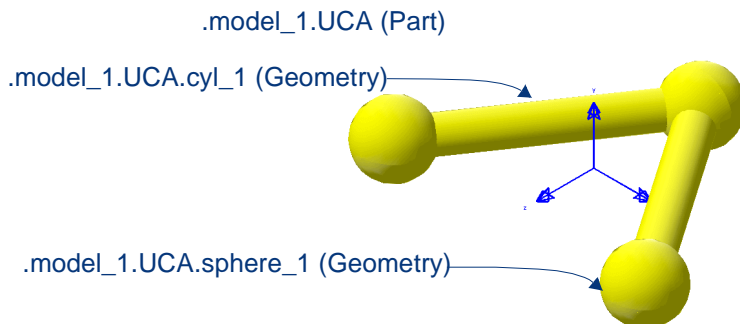
Define bodies (rigid or flexible) that can move relative to other bodies and have the following properties:

- Mass
- Inertia
- Initial location and orientation (PCS)
- Initial velocities

## Geometry

- Is used to add graphics to enhance the visualization of a part using properties such as:
  - ◆ Length
  - ◆ Radius
  - ◆ Width
- Is not necessary for most simulations.

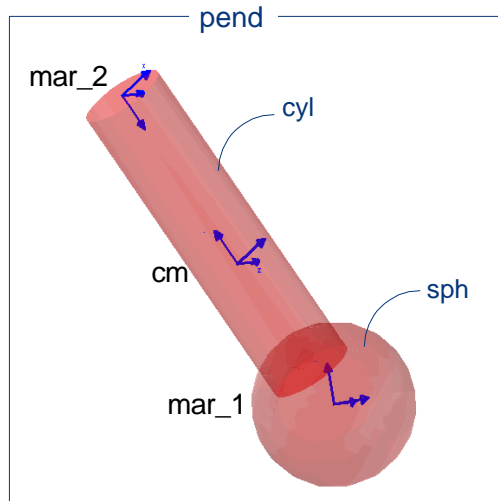
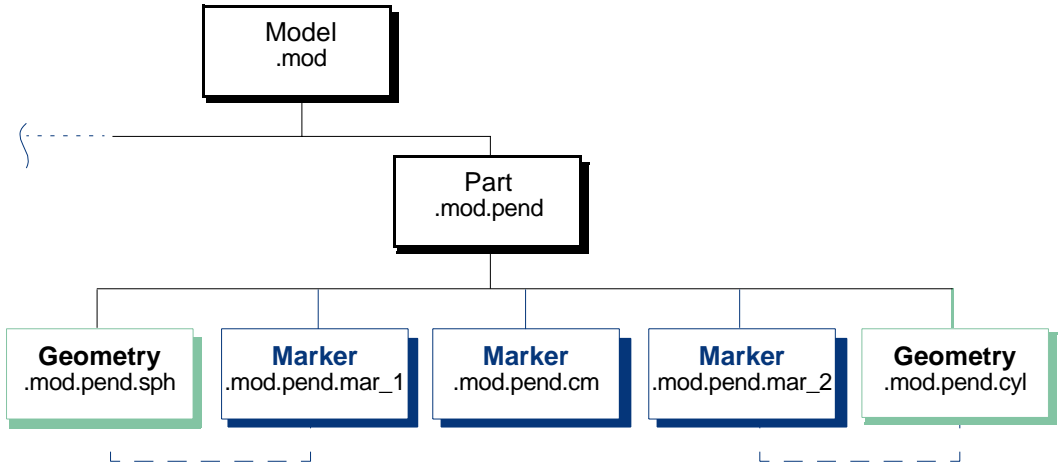
**Note:** Simulations that involve contacts do require the part geometry to define when the contact force will turn on or off. We will discuss contact forces in [Hatchback IV](#), on page 287.



# Parts, Geometry, and Markers

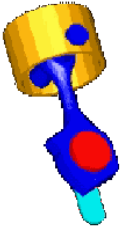
## Dependencies in MSC.ADAMS

To understand the relationship between **parts**, **geometry**, and **markers** in ADAMS/View, it is necessary to understand the dependencies shown next:



# Types of Parts in ADAMS/View

## Rigid bodies



- Are movable parts.
- Possess mass and inertia properties.
- Cannot deform.

## Flexible bodies



- Are movable parts.
- Possess mass and inertia properties.
- Can bend when forces are applied to them.

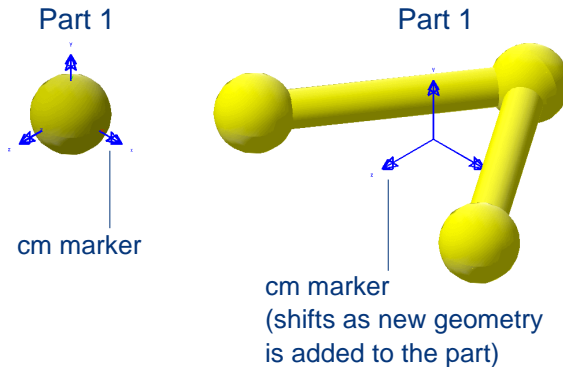
## Ground part

- Must exist in every model and is automatically created when a model is created in ADAMS/View.
- Defines the GCS and the global origin and, therefore, remains stationary at all times.
- Acts as the inertial reference frame for calculating velocities and acceleration.

# Part Mass and Inertia

## Mass and inertia properties

- ADAMS/View automatically calculates mass and inertial properties **only** for three-dimensional rigid bodies.
- ADAMS/View calculates the total mass and inertia of a part based on the part's density and the volume of its geometry.
- You can change these properties manually.
- ADAMS/View assigns mass and inertial properties to a marker that represents the part's center of mass (cm) and principal axes.
- You can change the position and orientation of the part's cm marker.





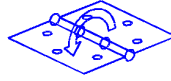
- The orientation of the cm marker also defines the orientation of inertial properties  $I_{xx}$ ,  $I_{yy}$ ,  $I_{zz}$ .

## Measures in MSC.ADAMS

- Represent data that you would like to quantify during a simulation, such as:
  - ◆ Displacement, velocity, or acceleration of a point on a part
  - ◆ Forces in a joint
  - ◆ Angle between two bodies
  - ◆ Other data resulting from a user-defined function
- Capture values of measured data at different points in time over the course of the simulation.

## Definition of object measures

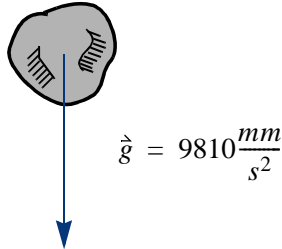
Measure pre-defined measurable characteristics of parts, forces, and constraints in a model.

 <p><b>Part measure characteristics:</b></p> <ul style="list-style-type: none"><li>■ CM position</li><li>■ CM velocity</li><li>■ Kinetic energy</li><li>■ Others</li></ul>	 <p><b>Spring measure characteristics:</b></p> <ul style="list-style-type: none"><li>■ Deformation</li><li>■ Force</li></ul>	 <p><b>Joint measure characteristics:</b></p> <ul style="list-style-type: none"><li>■ Relative velocity</li><li>■ Force</li><li>■ Torque</li><li>■ Others</li></ul>
---	---	--

# Workshop 4—Falling Stone

## Problem statement

Find the displacement, velocity, and acceleration of a stone after one second, when the stone with zero initial velocity, falls under the influence of gravity.



## Start the workshop


First, start ADAMS/View and create a model in the directory *exercise\_dir/mod\_04\_falling\_stone*. Executing ADAMS/View in that directory ensures that all saved data gets stored there.

To start the workshop:


- 1 Start ADAMS/View.
- 2 In the Welcome dialog box:
  - Under the heading, **How would you like to proceed**, select **Create a new model**.
  - Set the directory to *exercise\_dir/mod\_04\_falling\_stone*.
  - Name the model **projectile**.
  - Verify that **Gravity** is set to **Earth Normal (-Global Y)**.
  - Verify that **Units** are set to **MMKS - mm, Kg, N, s, deg**.
- 3 Select **OK**.

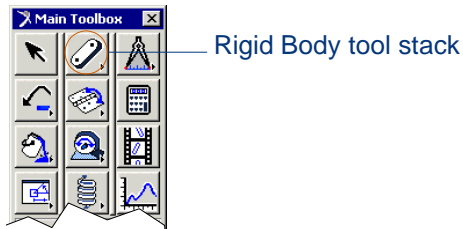
# Workshop 4—Falling Stone...

## Build the stone

Use the **Sphere** tool  to create a stone part with a 50 mm radius and its center at the global origin. You'll also rename the part and set its mass to 1 kg.

### To build the stone:

- 1 To view the coordinates as you create the sphere so you know its size, from the **View** menu, select **Coordinate Window**.
- 2 From the Main Toolbox, right-click the **Rigid Body** tool stack, and then select the **Sphere** tool .



- 3 Follow the Status bar instructions and pick the center of the sphere at the global origin, then drag the cursor until you create a sphere with a 50 mm radius.

### To rename the stone:

- 1 Right-click the sphere, point to **Part:PART\_2**, and then select **Rename**.
- 2 In the **New Name** text box, enter **.projectile.Stone**, and then select **OK**.

### To set the mass to 1 kg:

- 1 Right-click the sphere, point to **Part:Stone**, and then select **Modify**.
- 2 In the **Define Mass by** text box, select **User Input**.
- 3 If an alert box opens, select **Close**.
- 4 In the **Mass** text box, enter **1.0**.
- 5 Select **OK**.

## Workshop 4—Falling Stone...

### Create measures for the falling stone

To calculate the vertical displacement, velocity, and acceleration of the stone's cm marker in the  $\hat{y}_g$  direction, you'll create three object (part) measures. You'll set Y as the component to measure.

To calculate the displacement of the stone in the  $\hat{y}_g$  direction:

- 1 Right-click the stone, point to **Part:Stone**, and then select **Measure**.
- 2 In the **Measure Name** text box, enter **displacement**.
- 3 Set **Characteristic** to **CM position**.
- 4 Set **Component** to **Y**.
- 5 Set **From/At** to **.projectile.Stone.cm**.
- 6 Select **Create Strip Chart**.
- 7 Select **OK**.

A measure stripchart appears. It is empty because you need to run a simulation before ADAMS/View has the necessary information for the stripchart.

To calculate the velocity of the stone in the  $\hat{y}_g$  direction:

- 1 Right-click the stone, and select **Measure**.
- 2 In the **Measure Name** text box, enter **velocity**.
- 3 Set **Characteristic** to **CM velocity**.
- 4 Set **Component** to **Y**.
- 5 Set **From/At** to **.projectile.Stone.cm**.
- 6 Select **Create strip chart**.
- 7 Select **OK**.

## Workshop 4—Falling Stone...



To calculate the acceleration of the stone in the  $\hat{y}_g$  direction:

- Follow the instructions above but set **Measure Name** to **acceleration**, and **Characteristic** to **CM acceleration**.

### Verify the model

Now you'll verify the model. When you verify the model, ADAMS/View checks for error conditions, such as misaligned joints, unconstrained parts, or massless parts in dynamic systems and alerts you to other possible problems in the model.




To verify the model:

- 1 In the right corner of the Status bar, right-click the **Information** tool stack , and then select the **Verify** tool .
- 2 In the Information window, check that the model has verified successfully.
- 3 Close the Information window.

### Set up and run a simulation


Now you'll zoom out the display so that the falling stone is clearly visible while it simulates. You'll then simulate it for 1 second with 50 steps.

To zoom out:

- 1 Select the **Select** tool  to display the view control options in the toolbox.
- 2 Select the **Zoom** tool , and then click and drag the mouse to zoom out until the entire working grid is visible.
- 3 Select the **Translate** tool , and then drag the working grid to the top of the screen.

## Workshop 4—Falling Stone...

To run a simulation for 1 second with 50 steps:

- 1 In the Main Toolbox, select the **Simulation** tool .
- 2 In the **End Time** text box, enter 1.0 and in the **Steps** text box, enter 50.
- 3 Select the **Play** tool.

As the stone falls, ADAMS/View plots the corresponding data on the displacement, velocity, and acceleration graphs.

- 4 When the simulation ends, reset the model to the input, or design configuration by selecting the **Reset** tool.
- 5 Animate the simulation to replay the simulation without simulating again.

### Find the values of displacement, velocity, and acceleration


Now you'll use ADAMS/PostProcessor to find the stone's displacement, velocity, and acceleration after 1 second.

To run ADAMS/PostProcessor:

- Right-click the blank area inside the stripchart `.projectile.displacement`, point to **Plot:scht1**, and then select **Transfer to Full Plot**.

ADAMS/PostProcessor replaces the ADAMS/View window.

To find the value of the stone's displacement:

- 1 In ADAMS/PostProcessor, from the main toolbar, select the **Plot Tracking** tool .
- 2 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.
- 3 In the area below the menu bar, the value of X is displayed as 1. Note the value of Y; this is your answer.
- 4 Compare this value of Y to the results given in the closed-form solution on page 78.
- 5 Use the value to answer Question 1 in [Module review](#), on page 77.

## Workshop 4—Falling Stone...

---

To find the value of the stone's velocity after 1 second:

1 Select **Surf**.

This lets you view a selected measure without using the Add Curves button.

2 Set **Source** to **Measures**.

3 From the **Measure** list, select **velocity**.

4 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.

5 In the area below the menu bar, the value of X appears. It is 1. Note the value of Y; this is your answer.

6 Compare this value of Y to the results given in the [Closed-form solution](#), on page 78.

7 Use the value to answer Question 2 in [Module review](#), on page 77.

To find the value of stone's acceleration after 1 second:

1 Set **Source** to **Measures**.

1 From the **Measure** list, select **acceleration**.

2 To display the acceleration plot, select **Surf**.

3 Because you want to know the final conditions after 1 second, move the cursor over the end point of the plot.

4 In the area below the menu bar, the value of X will be displayed as 1. Note the value of Y; this is your answer.

5 Compare this value of Y to the results given in the [Closed-form solution](#), on page 78.

6 Use the value to answer Question 3 in [Module review](#), on page 77.

7 To return to ADAM/View and close all three plots, select the **ADAMS/View** tool.

# Workshop 4—Falling Stone...

---

## Save your work

Now save your work such that the file contains only the model information. You will use this model in the next module.

**Tip:** Save the model as a command file.

To save your work:

- 1 From the **File** menu, select **Export**, and then select **OK**.
- 2 If you want to further explore the model, as suggested in the next section, leave the model open. Otherwise, proceed with the next step.
- 3 Exit ADAMS/View.

## Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks because you will use this model in the next module. If you must save the model after performing these tasks, give the model a different name.

To inspect the behavior of the stone after changing its mass:

- 1 Change the mass of the stone to 2 kg.
- 2 Simulate the model.
- 3 Compare the results of this simulation with the results of the simulation where the mass of the stone was 1 kg.
- 4 Does changing the mass affect the displacement, velocity, or acceleration?
- 5 Measure the kinetic energy of the stone. Do these results make sense?

$$\text{K.E.} = (1/2)m*v^2$$

- 6 Exit ADAMS/View.

# Workshop 4—Falling Stone...

---

## Module review

1 What is the displacement of the stone after one second?

---

---

2 What is the velocity of the stone after one second?

---

---

3 What is the acceleration of the stone after one second?

---

---

4 What are the most basic building blocks in MSC.ADAMS, which are used in parts, constraints, forces, and measures?

---

---

5 Why is the ground part automatically created?

---

---

6 Can ADAMS/View automatically calculate mass properties for two-dimensional geometry?

---

---

# Workshop 4—Falling Stone...

## MSC.ADAMS results

- Displacement after 1 sec = -4903.3 mm
- Velocity after 1 sec = -9806.6 mm/sec
- Acceleration after 1 sec = -9806.6 mm/sec<sup>2</sup>

## Closed-form solution

Analytical solution:

$$s = \frac{1}{2} (at^2) = 4903.325 \text{ mm}$$

$$v = at = 9806.65 \text{ mm/sec}$$

$$a = g = 9806.65 \text{ mm/sec}^2$$

$$KE = (1/2) * 1 \text{ kg} * (9806.65 \text{ mm/sec})^2 = 4.8085 \text{ e}7 (\text{kg} * \text{mm}^2 / \text{s}^2)$$

$$\text{conversion to N: } 1 \text{ N} = 1 (\text{kg} * \text{m} / \text{s}^2)$$

$$KE = 4.80852 \text{ e}7 [(\text{kg} * \text{mm} / \text{s}^2) (1 \text{ m} / 100 \text{ mm})] * \text{mm} = 48085.2 \text{ N} * \text{mm}$$

Where:

$$s = \text{Distance (mm)}$$

$$a = \text{Acceleration (mm/sec}^2)$$

$$t = \text{Time (sec)}$$

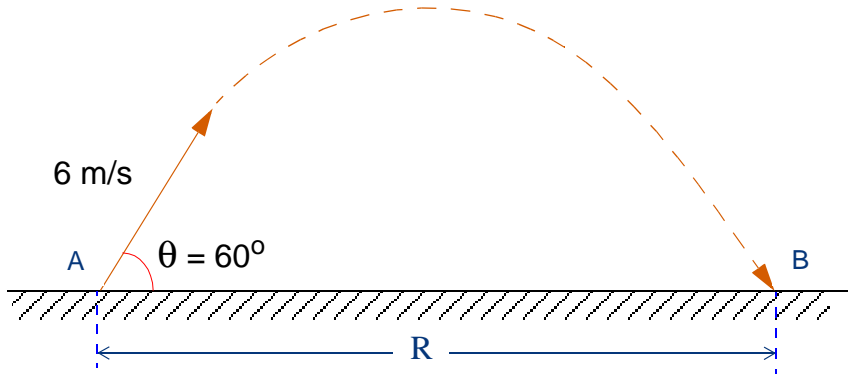
$$v = \text{Velocity (mm/sec)}$$

$$m = \text{mass (kg)}$$

# 5

## PROJECTILE MOTION

Compute the range,  $R$ , when a stone is launched as a projectile with an initial speed of  $6 \text{ m/s}$  at an angle of  $60^\circ$ , as shown next.



### What's in this module:

- Part Initial Conditions, 80
- Initial Velocities, 81
- Point Trace, 82
- Workshop 5—Projectile Motion, 83
  - ◆ Module review, 90

# Part Initial Conditions

---

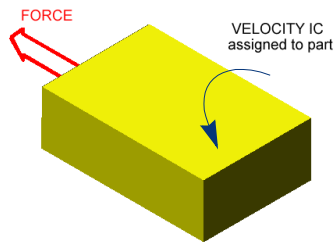
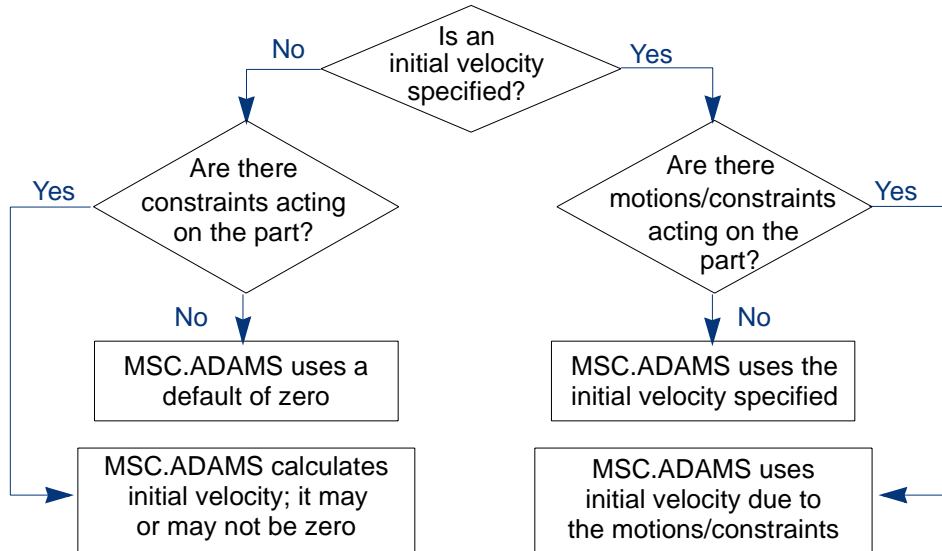
## Initial location and orientation

- The design configuration of all the parts (their part coordinate systems) in a model defines their initial locations and orientations.
- You can fix a part's location and orientation so it can be used during the assemble simulation procedure (covered later).

# Initial Velocities

## Initial velocities

In MSC.ADAMS, a part initially moves (at  $t = 0$ ) as follows:



# Point Trace

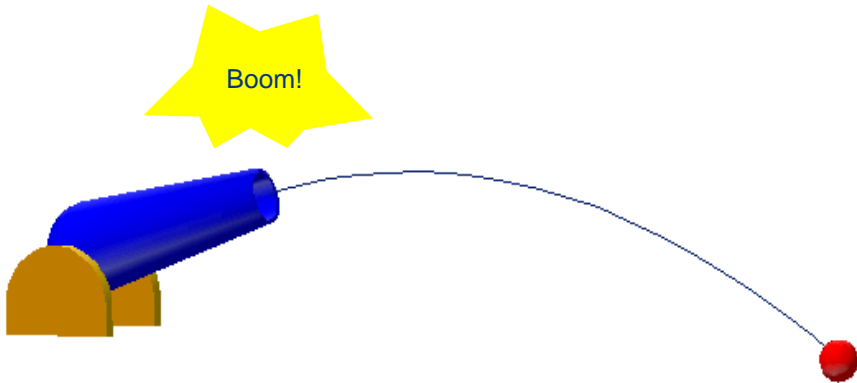
---

## Definition of a point trace

- Tracks the location of a marker during an animation.
- Can be used to visualize the clearance between two bodies during a simulation.

## Example of a point trace

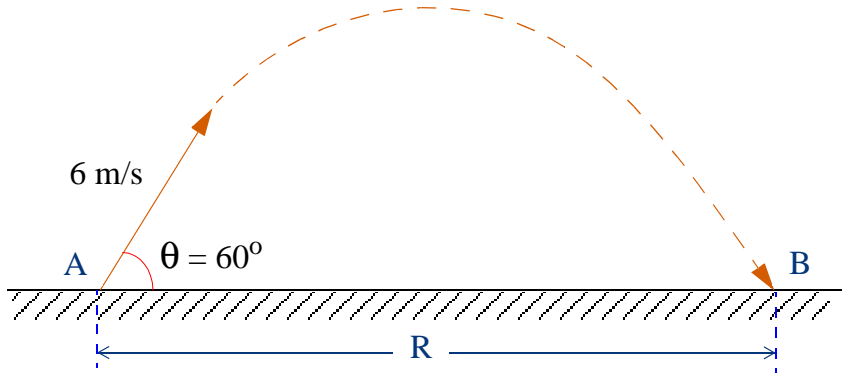
- Trajectory of a ball.



# Workshop 5—Projectile Motion

## Problem statement

Compute the range,  $R$ , when a stone is launched as a projectile with an initial speed of  $6 \text{ m/s}$  at an angle of  $60^\circ$ , as shown next.



## Model description

In this workshop, you use the model you built in [Workshop 4—Falling Stone](#), on page 70.

# Workshop 5—Projectile Motion...

## Start the workshop

To start the workshop, import the model that you created in the previous module.

To start the workshop:


- 1 In the Welcome dialog box, under the heading, **How would you like to proceed**, select **Import a file**.
- 2 Set the directory to *exercise\_dir/mod\_05\_projectile*. Executing ADAMS/View in this directory ensures that all saved data gets stored here.
- 3 Select **OK**.
- 4 Find and select the model file, **projectile.cmd**, which you completed in the previous workshop.

Note that the model file is not in the current working directory. It is in the directory *exercise\_dir/mod\_04\_falling\_stone*.

If you need a fresh copy of the model, import the command file *stone\_completed.cmd* from the directory *exercise\_dir/mod\_04\_falling\_stone/stone\_completed*.

- 5 Select **OK**.

## Build the plane

In this section, you'll build a plane using the Box tool . The plane will have the following dimensions:

- Length: 3500 mm
- Height: 100 mm
- On ground


Before building the plane, you'll set up the display by resetting the working grid to 4000 mm x 3000 mm with spacing of 50 mm, and zooming out.

# Workshop 5—Projectile Motion...

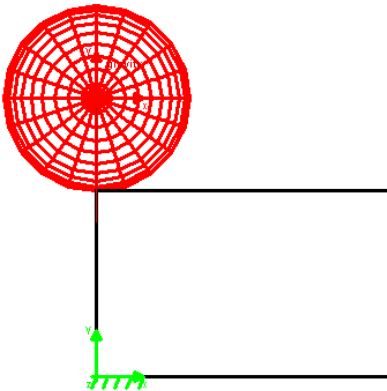
To set the display:

- 1 From the **Settings** menu, select **Working Grid**.
- 2 In the **Size: X** text box, enter **4000**.
- 3 In the **Size: Y** text box, enter **3000**.
- 4 In the **Spacing** text boxes, enter **50**, and then select **OK**.
- 5 Zoom out by typing a lowercase **z**, and then clicking and dragging the mouse to zoom out and view the entire working grid.

To build the plane:

- 1 Turn on the coordinate window (from the **View** menu, select **Coordinate Window**).
- 2 From the Main Toolbox, right-click the **Rigid Body** tool stack, and then select the **Box** tool .
- 3 In the toolbox container:
  - Select **On Ground**.
  - Select **Length**, and then in the **Length** text box, enter **3500 mm**.
  - Select the **Height** and **Depth** options, and then in the **Height** and **Depth** text boxes, enter **100 mm**.
- 4 Use the mouse to select the corner of the box at **0, -150, 0**.

The stone should appear to be balanced at the upper left corner of the plane in a front view.




# Workshop 5—Projectile Motion...

## Set initial conditions

Now set initial velocity conditions for the stone as follows:

- $V_{x_o} = 6000 \cdot \cos(60^\circ) = 3000 \text{ mm/sec}$
- $V_{y_o} = 6000 \cdot \sin(60^\circ) = 5196 \text{ mm/sec}$

To set initial conditions:

- 1 Reset the Main Toolbox by selecting the **Select** tool .
- 2 Right-click the stone, point to **Part:Stone**, and then select **Modify**.
- 3 Set **Category** to **Velocity Initial Conditions**.
- 4 Under **Translational velocity along**, select **X axis**, and in the **X axis** text box, enter  $(6 \cdot \cos(60d)(\text{m/sec}))$  or  $(3000(\text{mm/sec}))$ .
- 5 Under **Translational velocity along**, select **Y axis**, and in the **Y axis** text box, enter  $(6 \cdot \sin(60d)(\text{m/sec}))$  or  $(5196(\text{mm/sec}))$ .
- 6 Select **OK**.

## Create measures for projectile motion

Next, create an object (part) measure to calculate the horizontal displacement,  $\hat{x}_g$ , of the stone's center of mass (cm) marker when it is projected.

To create a measure:

- 1 Right-click the stone, point to **Part:Stone**, and then select **Measure**.
- 2 In the **Measure Name** text box, enter **R\_displacement**.
- 3 Set **Characteristic** to **CM position**.
- 4 Set **Component** to **X**.
- 5 Set **From/At** to **.projectile.Stone.cm**.
- 6 Select **Create Strip Chart**.
- 7 Select **OK**.

# Workshop 5—Projectile Motion...

## Run a simulation

Run a simulation for 1.5 seconds, using a sampling rate of .02 seconds.

To run a simulation:

- 1 From the Main Toolbox, select the **Simulation** tool.
- 2 In the **End Time** text box, enter 1.5.
- 3 In the **Step Size** text box enter 0.02.
- 4 Select the **Play** tool.






ADAMS/View runs the simulation and plots the corresponding data in a stripchart.

- 5 When the simulation ends, select the **Reset** tool.

## Find the range, R

Using animation tools, determine the time at which the stone encounters the plane. Use the time value to answer Question 1 in [Module review](#), on page 90.

To find the range:

- 1 From the Main Toolbox, select the **Animation** tool .
- 2 Select the **Play** tool.
- 3 When the stone makes contact with the plane, select the **Stop** tool .
- 4 Use the **Step Forward**  +1 and **Step Backward**  -1 tools to obtain the exact point at which the stone makes contact with the plane.
- 5 Note the time at which the stone makes contact with the plane in the plot. (The time is displayed in the upper-left corner of the ADAMS/View window.)
- 6 Select the **Select** tool .

# Workshop 5—Projectile Motion...

## Create a point trace

Create a point trace to view the trajectory of the projectile during an animation.


To create a point trace:

- 1 From the **Review** menu, select **Animation Controls**.  
The Animation Controls dialog box appears. The remaining steps used to create the point trace will be done in this dialog box.
- 2 Select **Icons**.
- 3 Set **No Trace** to **Trace Marker**.
- 4 Right-click the empty text box that appears, select **Marker**, and then select **Browse**.
- 5 From the Database Navigator, select **Stone.cm**.
- 6 Note that the marker name is entered into the text box, and then select the **Play** tool.
- 7 Close the Animation Controls dialog box.

## Find horizontal displacement

In ADAMS/PostProcessor, use the Plot Tracking tool to return the horizontal displacement corresponding to the time step determined earlier. Use the value to answer Question 2 in [Module review](#), on page 90.

To find the horizontal displacement:

- 1 Right-click a blank area inside the **R\_displacement** stripchart, point to **Plot: scht1**, and then select **Transfer to Full Plot**.  
ADAMS/PostProcessor replaces ADAMS/View.
- 2 Select the **Plot Tracking** tool .
- 3 Because you want to know the displacement when the stone makes contact with the plane, move the cursor over the plot until the value of X is equal to the time at which contact was made.
- 4 Note the value of displacement, Y. This is your answer for Question 2 in [Module review](#), on page 90.
- 5 Compare this value of Y to the results given in the closed-form solution on page 91.
- 6 Return to ADAMS/View.

# Workshop 5—Projectile Motion...

---

## Save your work

Save your work such that the file contains not only the model information, but also the results and plots.

To save your work:

- 1 From the **File** menu, select **Save Database As**.
- 2 In the **File Name** text box, enter **projectile**, and then select **OK**.  
ADAMS/View creates a binary file that contains not only the model information but also the results and plots.
- 3 If you want to further explore the model, as suggested in the next section, leave the model open. Otherwise, proceed with the next step.
- 4 Exit ADAMS/View.

## Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To follow the stone during an animation:

- 1 Zoom in on the stone.
- 2 From the **Review** menu, select **Animation Controls**.  
Now change the reference frame while animating.
- 3 On the Animation Controls dialog box, change **Fixed Base** to **Base Part**. Select the part to which you want to fix the camera.
- 4 Go to the ADAMS/View online help and look up the Animation Controls dialog box to read about the other functionality available.

# Workshop 5—Projectile Motion...

---

## Module review

**1** At what time does the stone encounter the plane?

---

---

**2** What is the range,  $R$ , as defined in the problem statement?

---

---

**3** If a part's initial velocity conflicts with a system constraint, which will take precedence during a simulation?

---

---

**4** What modifications would be necessary to convert the stone into a pendulum?

---

---

# Workshop 5—Projectile Motion...

## MSC.ADAMS results

R = 3180 mm (Can vary slightly depending on several factors, most likely the sampling rate.)

## Closed-form solution

Analytical solution:

The analytical solution for R, the range covered by the projectile, is as follows:

$$x_o = 0 \quad x_f = R$$

$$y_o = 0 \quad y_f = 0$$

$$V_{x_o} = 6000 \times \cos 60^\circ = 3000 \text{ mm/sec}$$

$$V_{y_o} = 6000 \times \sin 60^\circ = 5196 \text{ mm/sec}$$

$$y_f = y_o + V_{y_o} t - \frac{1}{2} g t^2$$

$$0 = 0 + 5196t - 0.5 \times 9806 \times t^2$$

$$0 = (5196 - 4905t)t$$

$$t = 1.06 \text{ sec}$$

$$x_f = x_o + V_{x_o} t$$

$$R = 0 + 3000 \times 1.06$$

$$R = 3180 \text{ mm}$$

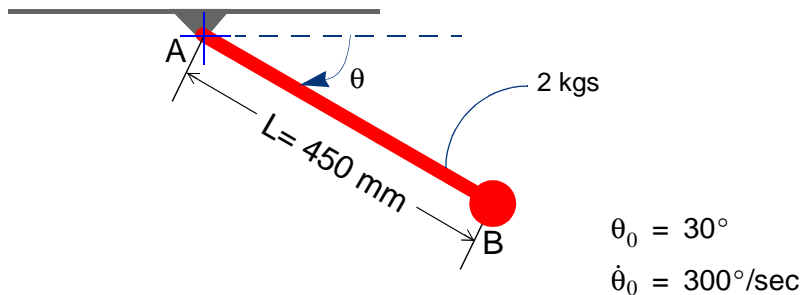
## Workshop 5—Projectile Motion...

---

# 6

## ONE DOF PENDULUM

Find the initial force supported by a pin at A for a bar that swings in a vertical plane, given the initial angular displacement ( $\theta_0$ ) and initial angular velocity ( $\dot{\theta}_0$ ). Also, find the pendulum frequency.



### What's in this module:

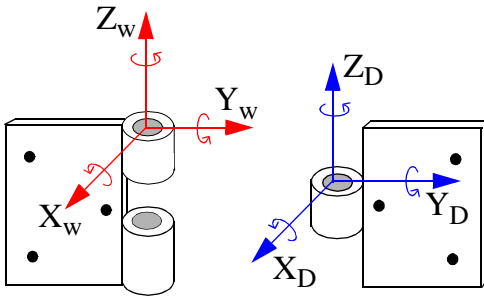
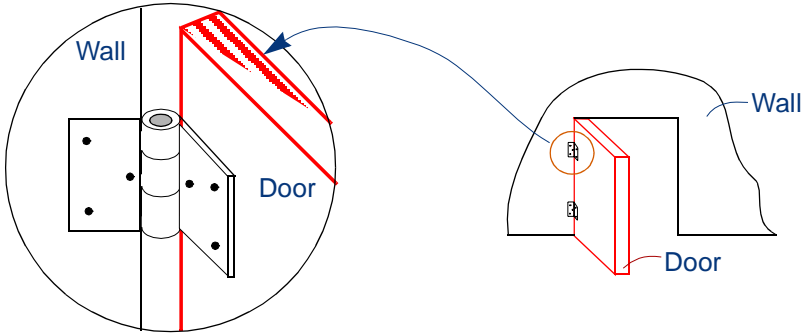
- Constraints, 94
- Use of Markers in Constraints, 95
- Degrees of Freedom (DOF), 96
- Joint Initial Conditions (ICs), 97
- A ball is rolling down an inclined plane (no slip). Which constraints will you use?, 98
- Angle Measures, 100
- Workshop 6—One DOF Pendulum, 101
  - ◆ Module review, 114

# Constraints

## Definition of a constraint

- Restricts relative movement between parts.
- Represents idealized connections.
- Removes rotational and/or translational DOF from a system.

## Example



### Translational constraints of the hinge

$$X_D - X_W = 0$$

$$Y_D - Y_W = 0$$

$$Z_D - Z_W = 0$$

### Rotational constraints of the hinge

$$\Phi_D - \Phi_W = 0 \text{ (about x-axis)}$$

$$\theta_D - \theta_W = 0 \text{ (about y-axis)}$$

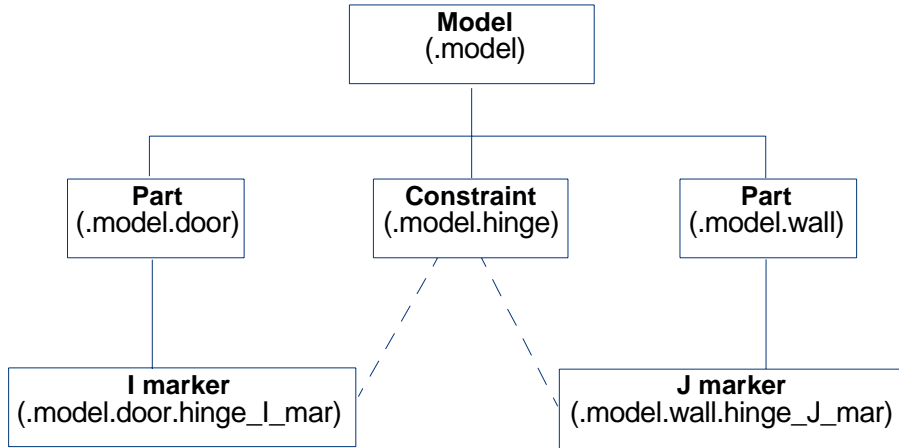
Therefore,  $\phi_D$  and  $\phi_W$  are free

# Use of Markers in Constraints

## Constraint equations in MSC.ADAMS

- Constraints are represented as algebraic equations in ADAMS/Solver.
- These equations describe the relationship between two markers.
- Joint parameters, referred to as I and J markers, define the location, orientation, and the connecting parts:
  - ◆ First marker, **I**, is fixed to the first part.
  - ◆ Second marker, **J**, is fixed to the second part.

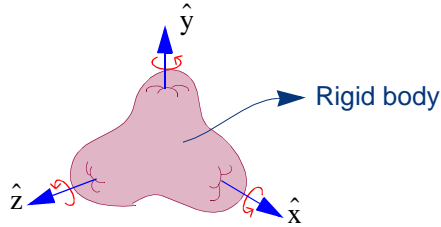
## Anatomy of a constraint in MSC.ADAMS



# Degrees of Freedom (DOF)

## Constraints and DOF

- Each DOF in mechanical system simulation (MSS) corresponds to at least one equation of motion.
- A freely floating rigid body in three-dimensional space is said to have six DOF.
- A constraint removes one or more DOF from a system, depending on its type.



## Determining the number of system DOF

- ADAMS/View provides an **estimated** number of system DOF by using the Gruebler's Count:

$$\text{System DOF} = (\text{number of movable parts} \cdot 6 \text{ DOF/part}) - \sum_{i = \text{type}} [\# \text{ Constraints} \cdot \# \text{ DOF (Constraint)}]$$

- ADAMS/View also provides the **actual** number of system DOF, as it checks to see if:
  - ◆ Appropriate parts are connected by each constraint.
  - ◆ Correct directions are specified for each constraint.
  - ◆ Correct type of DOF (translational versus rotational) are removed by each constraint.
  - ◆ There are any redundant constraints in the system.

**See also:** [DOF removed by a revolute joint](#), on page 376

# Joint Initial Conditions (ICs)

---

## Characteristics of joint initial conditions

- You can specify displacement and velocity initial conditions for revolute, translational, and cylindrical joints.
- ADAMS/View uses the specified initial conditions of the joint while performing a simulation, regardless of any other forces acting on the joint.
- If you do not specify joint ICs, ADAMS/Solver calculates the conditions of the connecting parts while performing a simulation depending on the other forces acting on the joint.

**Question:** What would happen if the joint initial conditions in a system were different from the part initial conditions?

# General Constraints (GCON)

---

User-defined constraints available only with C++ Solver.

Define an arbitrary constraint specific to a model.

Use C++ solver run-time functions to define the constraint.

Allow you to specify both holonomic and non-holonomic constraints.

Constraints are guaranteed to be handled correctly as long as the FUNCTION is in Pfaffian form. (Functions should be linear in velocity.)

Constraints are enforced as  $FUNC = 0$ . You can use any of the available ADAMS/Solver functions to define FUNC.

## Example

- The spherical joint, JOINT/1, SPHERICAL, I=1,J=3, could be replaced by the following set of three GCONs:

GCON/1, I=2, FUNC=DX(2,3)

GCON/2, I=2, FUNC=DY(2,3)

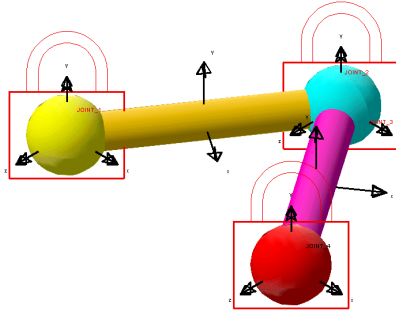
GCON/3, I=2, FUNC=DZ(2,3)

- I marker is used to specify the marker on which you want to measure the reaction forces.
- A ball is rolling down an inclined plane (no slip). Which constraints will you use?

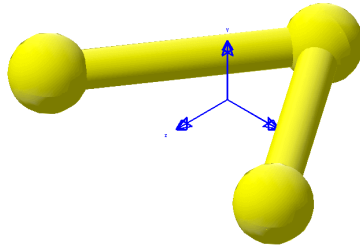
# Merging Geometry

## Methods of attaching multiple geometry to a part

- Using **fixed joint** to constrain geometric objects.



- Adding new geometry to an existing part.



**Note:** ADAMS/Solver handles simulations better if you merge geometry on a rigid part as opposed to constraining multiple parts.

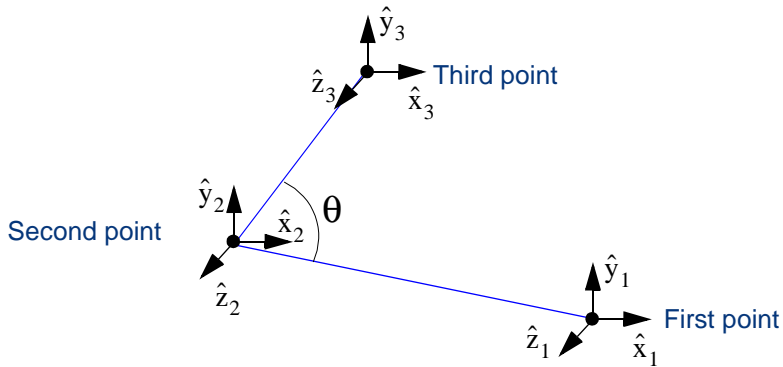
**Question:** When you merge geometry is the overlapping volume accounted for?

# Angle Measures

## Definition of angle measures

They are used to measure the included angle,  $\theta$ :

- Between two vectors
- Defined by three markers
- Defined throughout a simulation



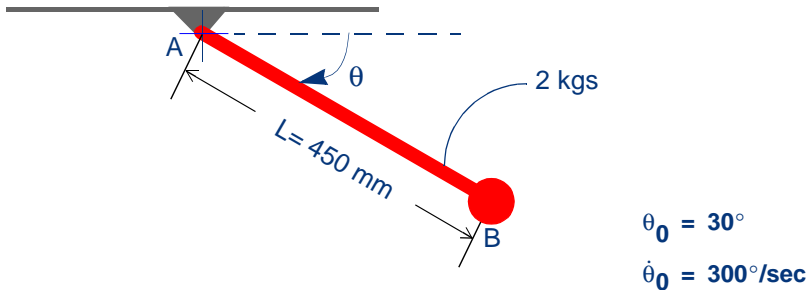
### Notes:

- The units used for angle measures are in current ADAMS/View angle units (degrees or radians).
- The sign convention (+/-) is defined such that the first nonzero value is positive.

# Workshop 6—One DOF Pendulum

## Problem statement

Find the initial force supported by the pin at A for a bar that swings in a vertical plane, given the initial angular displacement ( $\theta_0$ ) and initial angular velocity ( $\dot{\theta}_0$ ). Also, find the pendulum frequency.



## Start the workshop

First, you'll start ADAMS/View from the directory *exercise\_dir/mod\_06\_pendulum* and then you'll create a new model. Executing ADAMS/View in this directory ensures that all saved data gets stored here.

To start ADAMS/View and create a model:

- Start ADAMS/View:
  - ◆ Set the directory to *exercise\_dir/mod\_06\_pendulum*.
  - ◆ Create a new model named **pendulum**, with Gravity set to **Earth Normal (-Global Y)**, and Units set to **MMKS - mm, Kg, N, s, deg**.


# Workshop 6—One DOF Pendulum...

## Build the pendulum link

Now, build the link section of the pendulum using the following parameters:

- Width: 20 mm
- Depth: 27.5mm
- Endpoints: (0, 0, 0) and (450, 0, 0)

### To build the link:

- 1 Turn on the coordinate window.
- 2 From the Main Toolbox, right-click the **Rigid Body** tool stack, and then select the **Link tool** .
- 3 In the container:
  - Select **New Part**.
  - Select **Length**, and in the **Length** text box, enter **450 mm**, and then press **Enter**.
  - Select **Width**, and in the **Width** text box, enter **20 mm**, and then press **Enter**.
  - Select **Depth**, and in the **Depth** text box, enter **27.5 mm**, and then press **Enter**.
- 4 Using the mouse, select **0, 0, 0** and **450,0,0** as the endpoint locations.

**Tip:** Use the Location Event (right-click away from the model) to help select the endpoints. When you right-click, the Location Event appears in the lower left corner of the ADAMS/View window. Enter the coordinates for the link in the upper text box and then press **Enter**.


# Workshop 6—One DOF Pendulum...

## Build the sphere section

Next, build the sphere section of the pendulum using the following parameters:

- Add to Part
- Radius: 25 mm
- Centerpoint: 450, 0, 0

To build the sphere section:

- 1 From the Main Toolbox, right-click the **Rigid Body** tool stack, and then select the **Sphere** tool .
- 2 In the container:
  - Select **Add to part**.
  - Select **Radius**, and in the **Radius** text box, enter **25 mm**, and then press **Enter**.
- 3 Using the mouse, select **PART\_2**, which is the link, as the part to add to.
- 4 Using the mouse, select **450,0,0** as the location.

## Rename the pendulum

Now you'll rename the pendulum from PART\_2 to Pendulum.

To rename the pendulum:

- 1 Right-click the **link**, point to **Part:PART\_2**, and then select **Rename**.  
The Rename Object dialog box appears.
- 2 In the **New Name** text box, enter **.pendulum.pendulum**, and then select **OK**.

# Workshop 6—One DOF Pendulum...

## Set the mass of the pendulum

Now, set the mass of the pendulum to 2 kg, set all three inertias ( $I_{xx}$ ,  $I_{yy}$ ,  $I_{zz}$ ) to 0, and change the location of the center of mass.

To set the mass of the pendulum:

- 1 Right-click the **pendulum**, point to **Part: pendulum**, and then select **Modify**.
- 2 Set **Define Mass by** to **User Input**.
- 3 In the **Mass** text box, enter **2.0**.
- 4 In the **Inertia** text boxes ( $I_{xx}$ ,  $I_{yy}$ ,  $I_{zz}$ ), enter **0**.
- 5 Right-click the **Center of Mass Marker** text box, point to **pendulum.pendulum.cm**, and then select **Modify**.
- 6 In the **Location** text box, enter **450, 0, 0**.
- 7 Select **OK** in both dialog boxes.

You will receive a warning in the Message Window concerning the change in position of your center of mass marker.

- 8 Select **Close** to close the Message Window.

Your model should look like this (with shading turned on):




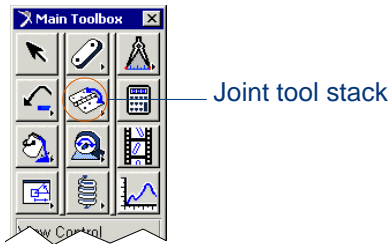
# Workshop 6—One DOF Pendulum...

## Build the pivot

In this section, you'll build the pivot by creating a revolute joint between ground and the pendulum at location A, as shown in the figure on the page 94, and rename it Pivot.

To build the pivot:

- 1 From the Main Toolbox, right-click the **Joint** tool stack, and then select the **Revolute joint** tool .



- 2 In the container, select 2 **Bod-1 Loc** and **Normal to Grid**.
- 3 Select the **pendulum** as the **first body**.
- 4 Select the **ground** as the **second body**.
- 5 Select **0, 0, 0** as the **location**.

To rename the joint:

- 1 Right-click the revolute joint, point to **Joint:JOINT\_1**, and then select **Rename**.
- 2 In the **New Name** text box, enter **.pendulum.pivot**, and then select **OK**.

# Workshop 6—One DOF Pendulum...

## Create measures

Create two object (joint) measures to track the force supported by the pin, resolved in the  $\hat{x}_g$  and  $\hat{y}_g$  directions.

To create object measures:

- 1 Right-click the **pivot** joint, point to **Joint:pivot**, and then select **Measure**.
- 2 In the dialog box:
  - In the **Measure Name** text box, enter **pivot\_force\_x**.
  - Set **Characteristic** to **Force**, and select **X** as the **Component**.
  - Be sure **.pendulum.MARKER\_4** and **Create Strip Chart** are selected.
  - Select **Apply**.

A stripchart displays the force during simulation and animation.

- 3 In the dialog box:
  - In the **Measure Name** text box, enter **pivot\_force\_y**.
  - Set **Characteristic** to **Force**, select **Y** as the **Component**.
  - Be sure **.pendulum.MARKER\_4** and **Create Strip Chart** are selected.
  - Select **OK**.


A stripchart displays the force during simulation and animation.

# Workshop 6—One DOF Pendulum...

## Create a reference marker

Create a marker on ground to use as a reference location for the angle measure you will create in the next section. Instead of right-clicking on the marker to change its name, you'll use the **Edit** menu.

To create a reference marker:

- 1 On the Main Toolbox, right-click the **Rigid Body** tool stack, and then select the **Marker** tool .
- 2 In the container, be sure that **Add to Ground** and **Global XY** are selected.
- 3 Using the mouse, select **450, 0, 0** as the location.
- 4 With the marker still selected, from the **Edit** menu, select **Rename**.
- 5 In the **New Name** text box, enter `.pendulum.ground.angle_ref`, and then select **OK**.

## Create angle measure

Now, create the angle measure to track the angular displacement of the pendulum,  $\theta$ .

To create an angle measure:

- 1 From the **Build** menu, point to **Measure**, point to **Angle**, and then select **New**.
- 2 In the **Measure Name** text box, enter `pend_angle`.
- 3 Right-click the **First Marker** text box, point to **Marker**, and then select **Pick**.
- 4 On the screen, pick a marker that is on the pendulum and at its end (for example, select the cm marker).  
*Tip:* Right-click the end of the pendulum to select the cm marker.
- 5 Right-click the **Middle Marker** text box, point to **Marker**, and then select **Pick**.
- 6 Pick a marker that is at the pivot location.
- 7 Right-click the **Last Marker** text box, point to **Marker**, and then select **Pick**.

## Workshop 6—One DOF Pendulum...

---

- 8 Pick the marker that is on the ground and at the end of the pendulum (this is the marker that you created in the previous section, `.pendulum.ground.angle_ref`).

**Note:** By aligning the marker `.pendulum.ground.angle_ref` with the cm marker, the initial value of the measure will be zero.

- 9 Select **OK**.

### Specify initial conditions

In this section, you'll specify the following joint initial conditions:

- Displacement initial condition of  $\theta_0 = 30^\circ$
- Initial velocity condition of  $\dot{\theta}_0 = 300^\circ/\text{sec}$

To specify the initial conditions:


- 1 Right-click the **pivot** joint, point to **Joint:pivot**, and then select **Modify**.
- 2 Select **Initial Conditions**.
- 3 In the Joint Initial Conditions dialog box:
  - Select **Rot. Displ** and, in the **Rot Displ.** text box, enter **-30**.
  - Select **Rot. Velo.** and, in the **Rot Velo.** text box, enter **-300**.
- 4 Select **OK** in both dialog boxes.

# Workshop 6—One DOF Pendulum...

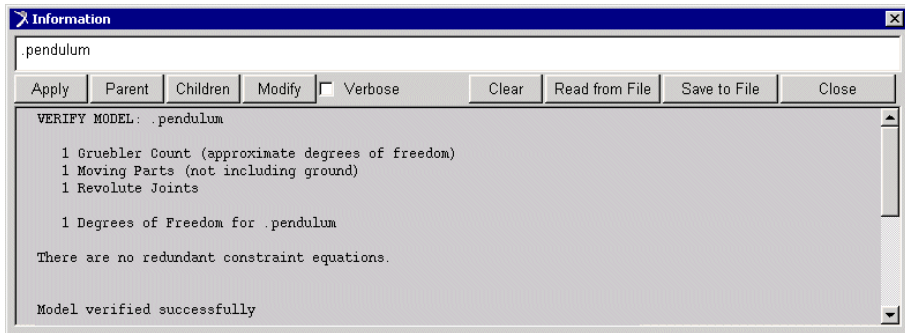
## Verify your model

Before simulating your model, verify it.

To verify your model:

- 1 Select the **Verify** tool  (from the **Status** bar, right-click the **Information** tool stack ,).

The Information window appears as shown next:



You also receive a warning that the initial conditions for the joint position does not match the design configuration. This is what we expect.

- 2 Close the Information window.

## Simulate your model

Run a simulation for 2 seconds.

To simulate your model:

- Run a simulation for 2 seconds with 100 steps, just as you did in [Simulate the model](#), on page 52.

# Workshop 6—One DOF Pendulum...


## Determine global components

Now, determine the global components (x, y) of the initial force supported by the pivot. Use the value to answer Question 1 in [Module review](#), on page 114.

To determine global components:

- 1 Right-click the blank area inside the `pend_angle` stripchart, point to **Plot: scht1**, and then select **Transfer to Full Plot**.

ADAMS/PostProcessor replaces ADAMS/View.

- 2 Select the **Plot Tracking** tool .
- 3 Move the cursor over the plot at  $t=0$ .
- 4 In the area below the main toolbar, note the value of Y.
- 5 In the dashboard, select **Clear Plot**.
- 6 Set **Source** to **Measures**.
- 7 From the **Measure** list, select `pivot_force_x`.
- 8 Select **Surf**.
- 9 Move the cursor over the plot at  $t=0$ .
- 10 In the area below the main toolbar, note the value of Y.
- 11 From the **Measure** list, select `pivot_force_y`.
- 12 Move the cursor over the plot at  $t=0$ .
- 13 In the area below the main toolbar, note the value of Y.

# Workshop 6—One DOF Pendulum...

## Determine the frequency of the pendulum

Estimate the frequency by determining the period (seconds) and then inverting that value to obtain Hertz. This is the answer to Question 2 in [Module review](#), on page 114.

To determine frequency:

- 1 From the Measure list, select **pend\_angle**.
- 2 Estimate the period of the curve.
- 3 Invert the period to find Hertz.
- 4 Return to ADAMS/View.
- 5 Save results as **Joint\_res**.
- 6 Save the curves on all three strip charts by right-clicking each curve, and then selecting **Save curve**.



## Use GCON

To modify the part:

- 1 Deactivate **Joint:pivot** and the corresponding measures: **pivot\_force\_x** and **pivot\_force\_y**.
- 2 Modify the pendulum to specify the initial conditions: right-click **Part:pendulum**, and then select **Modify**.
- 3 Set **Category** to **Name and Position**.
- 4 In the **Orientation** text box, enter **-30d,0,0**, and then select **Apply**.
- 5 Set **Category** to **Velocity Initial Conditions**.
- 6 Set **Angular velocity about** to **Marker**.
- 7 Right-click the corresponding text box, and then select **pendulum.Marker\_1**.
- 8 Select **Z axis**.
- 9 In the corresponding text box, enter **-300d**.
- 10 Select **OK**.

# Workshop 6—One DOF Pendulum...

To create general constraints:

- 1 From the Main Toolbox, right-click the **Joint** tool stack , and then select the **General Constraints** tool .
- 2 In the **General Constraint Name** text box, enter **trans\_X**.
- 3 In **f(q)=** text box, enter **DX(Marker\_1)**.
- 4 In the **Report forces on Marker** text box, enter **Marker\_1**.
- 5 Create four more general constraints with following options:
  - trans\_y      DY(Marker\_1)
  - trans\_z      DZ(Marker\_1)
  - rot\_x        AX(Marker\_1)
  - rot\_y        AY(Marker\_1)
- 6 Simulate the model.
- 7 Compare the results from the saved result set **joint\_res** and the new result set obtained by using **GCON** in place of the revolute joint.
- 8 Compare reaction forces of the constraints **trans\_x** and **trans\_y** with **pivot\_force\_x** and **pivot\_force\_y**.

## Save your work

To save your work:

- 1 Use the **Save As** option to save your modeling session as a binary file. This file will contain not only the model information, but also the results and plots.  
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

Save your work before performing these tasks. Do not save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To find the frequency of the pendulum automatically by performing a Fast Fourier Transformation (FFT) on the plot of theta versus time:

- 1 Run a simulation using the following settings:
  - ◆ End time = 1.65 (approximate time of one period)
  - ◆ Steps = 127
- 2 In ADAMS/PostProcessor, from the **Plot** menu, select **FFT**.  
The FFT dialog box appears.
- 3 When preparing for an FFT operation, we recommend that:
  - The number of points be an even power of two (for example, 128, 256, 512, and so on). By solving the equation and asking for 127 steps, you will get 128 data points; 127 + 1 for the initial conditions.
  - You set **Window Type** to **Rectangle**.
  - You select **Detrend Input Data**.
- 4 To perform the FFT, select **Apply**. To learn more about these values, press **F1**.  
You should get approximately the same frequency as you did by calculating it manually.  
The peak value of the resultant curve is at the natural frequency.
- 5 Return to ADAMS/View.
- 6 Exit ADAMS/View.

# Workshop 6—One DOF Pendulum...

## Module review

- 1 What are the global components of the initial force supported by the pivot?  

---

---
- 2 What is the frequency of the pendulum using the initial conditions in the problem statement?  

---

---
- 3 If the initial velocity of a part can be set through a connecting joint *and* the part itself, which will ADAMS/View use if they are both set?  

---

---
- 4 If a model (human\_hip) had two parts (femur and hip\_bone) constrained by a joint, I and J markers would be created by MSC.ADAMS. If one marker were named MAR\_1, and the other MAR\_2, what would the complete name of the I and J markers be?  

---

---
- 5 Can the I and J markers for a joint belong to the same part? Why?  

---

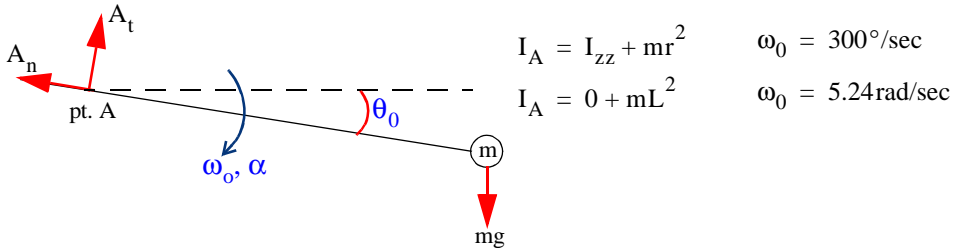
---

# Workshop 6—One DOF Pendulum...

## MSC.ADAMS results

- Horizontal force supported by the pivot at A =  $-A_n \cos 30$ .
- Vertical force supported by the pivot at A =  $A_n \sin 30$ .

## Closed-form solution



The analytical solution for the force supported by the pivot at A when  $\theta_0 = 30^\circ$  and  $\omega_0 = 300$  degrees/sec:

$$\Sigma M_A = I_A \alpha \quad -mg(L \cos 30) = (mL^2)\alpha$$

$$g \cos 30 = L\alpha$$

$$\alpha = -\frac{g}{L} \cos 30$$

$$\alpha = -18.88\text{rad}/\text{sec}^2$$

$$\Sigma F_t = m r \alpha \quad mg \cos 30 - A_t = mL\alpha$$

$$A_t = m(g \cos 30 - L\alpha)$$

$$A_t = 0\text{N}$$

$$\Sigma F_n = m r \omega^2 \quad A_n - mg \sin 30 = mL\omega^2$$

$$A_n = m(g \sin 30 + L\omega^2)$$

$$A_n = 34.53\text{N}$$



# 7

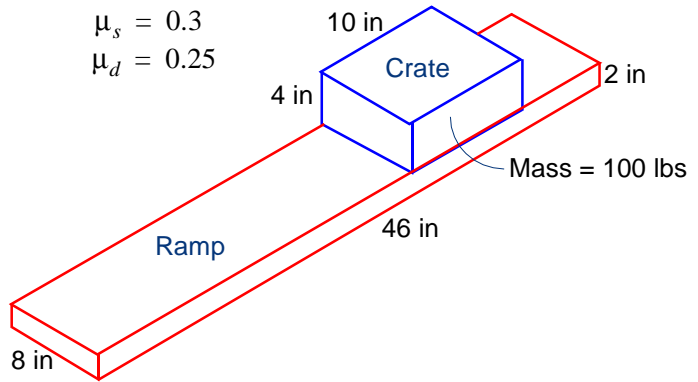
## INCLINED PLANE

Find the minimum inclination that will ensure that a crate slides off an inclined plane, using the properties shown next:

$$\vec{g} = 32.2 \text{ ft/sec}^2$$

$$\mu_s = 0.3$$

$$\mu_d = 0.25$$



### What's in this module:

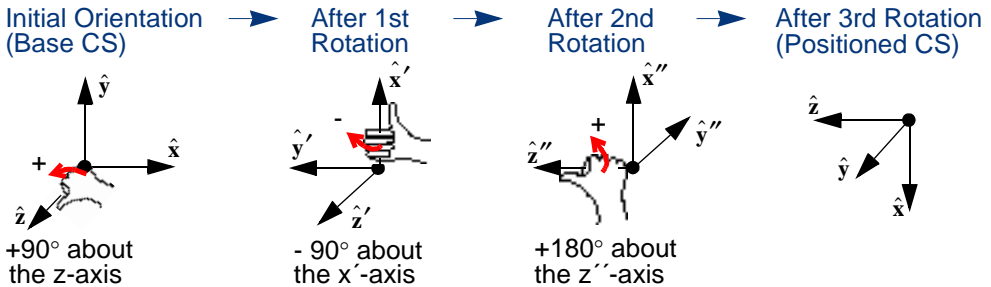
- Euler Angles (Rotation Sequence), 118
- Precise Positioning: Rotate, 119
- Modeling Friction, 120
- Measures in LCS, 123
- Workshop 7—Inclined Plane, 124
  - ◆ Module review, 135

# Euler Angles (Rotation Sequence)

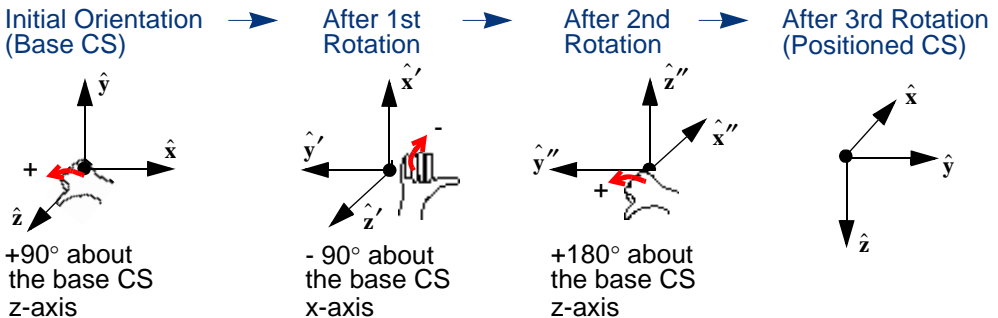
## Definition of Euler angles

- ADAMS/View uses three angles to perform three rotations about the axes of a coordinate system.
  - These rotations can be **space-fixed** or **body-fixed** and are represented as Body [3 1 3], Space [1 2 3], and so on, where:
    - ◆ 1 = x axis
    - ◆ 2 = y axis
    - ◆ 3 = z axis
- } For rotation about these axes, use the right-hand rule
- Default in MSC.ADAMS is Body [3 1 3].

## Example of body [3 1 3]: $[90^\circ, -90^\circ, 180^\circ]$



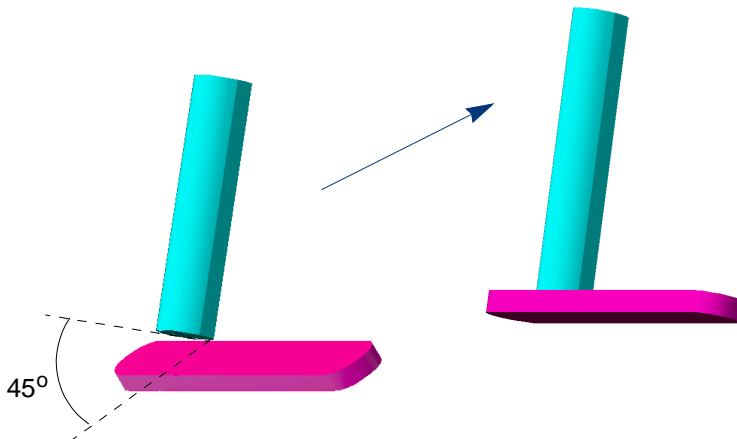
## Example of space [3 1 3]: $[90^\circ, -90^\circ, 180^\circ]$ :



# Precise Positioning: Rotate

To rotate objects about an axis in ADAMS/View, specify:

- The objects to rotate.
- The axis about which the objects are rotated.
- The angle through which the objects are rotated.



**Note:** Be careful with the sign of the angle. ADAMS/View uses the right-hand rule. You can rotate several objects at once about the same axis.

# Modeling Friction

---

Joint friction can be applied to:

- Translational joints (Translational Joint, DOF Removed by, see page 376)
- Revolute joints
- Cylindrical joints
- Hooke/Universal joints
- Spherical joints

Friction forces ( $F_f$ )

- Are independent of the contact area between two bodies.
- Act in a direction opposite to that of the relative velocity between the two bodies.
- Are proportional to the normal force (N) between the two bodies by a constant ( $\mu$ ).

$$F_f = \mu N$$

Phases that define friction forces

- Stiction
- Transition
- Dynamic

# Modeling Friction...

## Idealized case

■ Stiction:  $|V_{rel}| = 0$

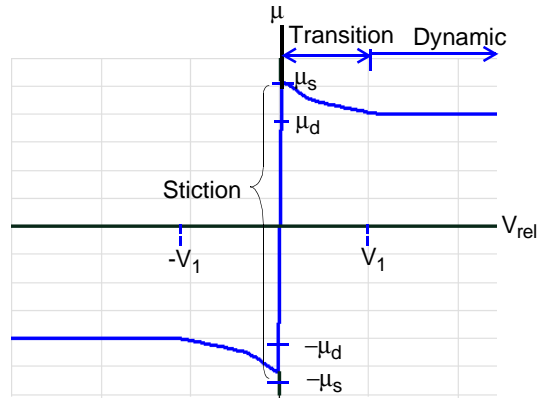
$$0 < \mu < \mu_s$$

■ Transition:  $0 < |V_{rel}| = V_1$

$$\mu_d < \mu < \mu_s$$

■ Dynamic:  $V_1 < |V_{rel}|$

$$\mu = \mu_d$$



## ADAMS/Solver case

■ Stiction:  $|V_{rel}| < \Delta V_s$

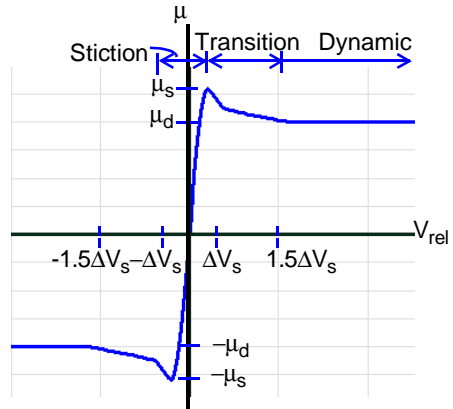
$$0 < \mu < \mu_s$$

■ Transition:  $\Delta V_s < |V_{rel}| < 1.5\Delta V_s$

$$\mu_d < \mu < \mu_s$$

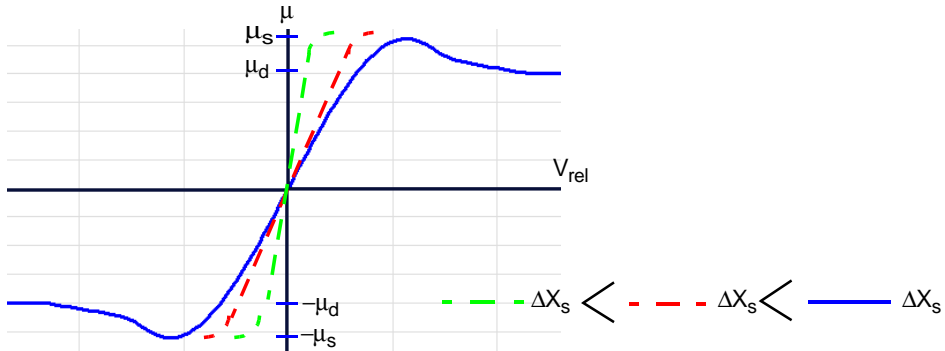
■ Dynamic:  $1.5\Delta V_s < |V_{rel}|$

$$\mu = \mu_d$$



# Modeling Friction...

## Effect of maximum deformation on friction



## Input forces to friction

- Always include preload and reaction force.
- Bending and torsional moment are possible (however, advanced uses of joint friction are beyond the scope of this course).

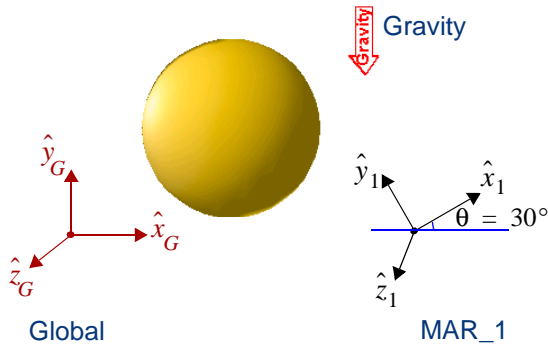
# Measures in LCS

Measures can be represented in:

- Global coordinate system (GCS) (default)
- A marker's local coordinate system (LCS)

## Example

- When a ball falls due to gravity:



- Acceleration due to gravity in the GCS using  $\hat{x}_g, \hat{y}_g, \hat{z}_g$  symbols to represent the global x, y, and z components is:

$$\vec{g} = (0\hat{x}_g - 9.81\hat{y}_g + 0\hat{z}_g) \frac{m}{s^2}$$

- Acceleration due to gravity in MAR\_1's coordinate system is:

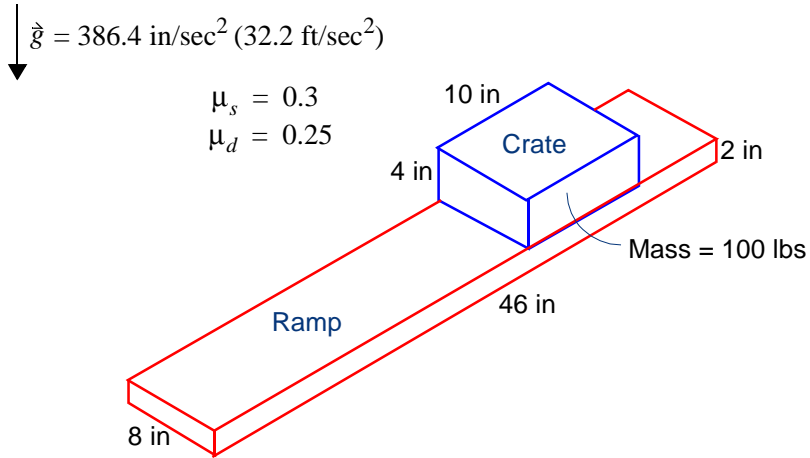
$$\vec{g} = ((-9.81 \sin 30^\circ)\hat{x}_1 - (9.81 \cos 30^\circ)\hat{y}_1 + 0\hat{z}_1) \frac{m}{s^2}$$

$$\vec{g} = (-4.91\hat{x}_1 - 8.50\hat{y}_1 + 0\hat{z}_1) \frac{m}{s^2}$$

# Workshop 7—Inclined Plane

## Problem statement

Find the minimum inclination that will ensure that a crate slides off an inclined plane, using the properties shown next:



## Start the workshop

First, you'll start ADAMS/View from the directory `exercise_dir/mod_07_inclined_plane` and then create a new model.

To start the workshop:

- Start ADAMS/View:
  - ◆ Set the directory to `exercise_dir/mod_07_inclined_plane`.
  - ◆ Create a new model named `inclined_plane`, with Gravity set to **Earth Normal** (-Global Y), and Units set to **IPS - inch, lbm, lbf, s, deg**.

# Workshop 7—Inclined Plane...

## Adjust the working grid

Now adjust the spacing and orientation of the working grid.

To adjust the spacing and orientation of the working grid:

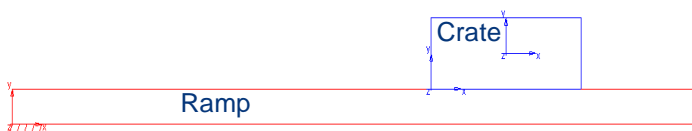
- 1 From the **Settings** menu, select **Working Grid**.
- 2 Set **Spacing** to **1 in** in the **x** and **y** direction.
- 3 Make sure that the working grid is oriented along the global **XY** direction (default setting when you open ADAMS/View). The **Set Orientation** pull-down menu allows you to choose **Global XY**, **YZ**, **XZ**, or custom orientation.
- 4 Select **OK**.

## Build the parts

When creating parts, use an inclination angle of  $0^\circ$ . You will rotate the parts to the desired inclination angle later in the exercise. Be sure to set the ramp geometry to be on ground.

To build the parts:

- 1 Build the ramp geometry using the following parameters:
  - ◆ **On Ground**
  - ◆ **Length:** 46 in
  - ◆ **Height:** 2 in
  - ◆ **Depth:** 8 in
- 2 Build the crate geometry using the following parameters:
  - ◆ **New Part**
  - ◆ **Length:** 10 in
  - ◆ **Height:** 4 in
  - ◆ **Depth:** 8 in
- 3 Position the crate near the end of the ramp, as shown next:



# Workshop 7—Inclined Plane...

---

To modify the parts:

- 1 Rename the crate and the ramp geometry as shown in the figure on page 124.  
**Note:** Make sure that you rename just the geometry and not the ground part.
  - Right-click on the large box, point to **Block: BOX\_1**, and then select **Rename**.
  - Enter **Ramp**.
  - Select **OK**.
  - Repeat this step for **PART\_2**, naming it **Crate**.
- 2 Set the **Mass** of the crate to **100 lbm**:
  - Right-click the crate, point to **Part:Crate**, and then select **Modify**.
  - Set **Define Mass by** to **User Input**.
  - In the **Mass** text box, enter **100 lbm**.
  - Select **OK**.

## Set the model's inclination angle

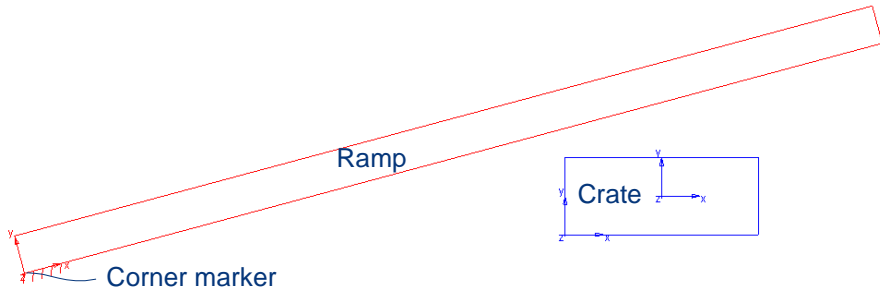
Now you are going to rotate the model  $15^\circ$ . Because the ramp is on ground and you cannot rotate ground, to rotate the ramp, you are going to change the orientation of the ramp's corner marker to 15, 0, 0. The orientation of this marker sets the orientation for the ramp. You'll use the **Rotate** tool to rotate the crate since it is not on ground. You'll rotate the crate about the same axis that you rotated the ramp about.

# Workshop 7—Inclined Plane...


To rotate the ramp to  $\theta = 15^\circ$ :

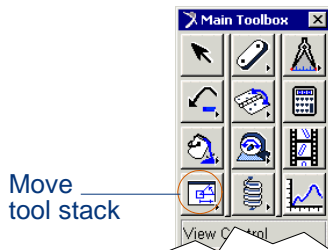
- 1 Right-click the ramp's corner marker, point to **MARKER\_1**, and then select **Modify**.
- 2 In the **Orientation** text box, change 0,0,0 to 15,0,0.

Figure 1. Model of Inclined Plane



To rotate the crate 15°:

- 1 In the Main Toolbox, from the **Move** tool stack, select the **Align & Rotate** tool .

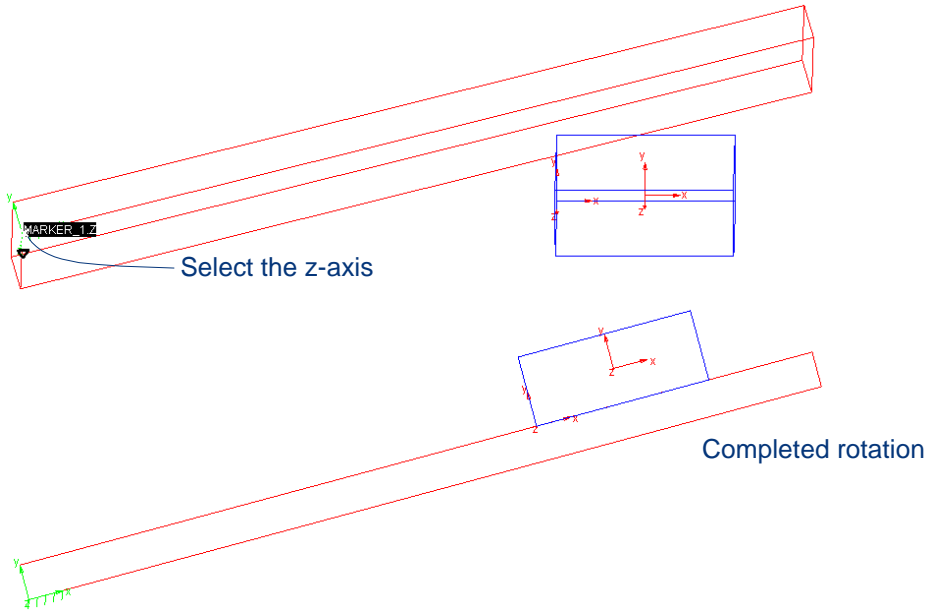


- 2 In the container, in the **Angle** text box, enter 15 as the increment by which to rotate the crate. After you enter this value, if your cursor does not sweep the text box (that is, move from inside the text box to outside the text box), press Enter to ensure that the text box registered the value you entered.
- 3 Select the crate as the object to rotate.

# Workshop 7—Inclined Plane...

- 4 Select the z-axis of **MARKER\_1** (on the ramp) as the axis about which to rotate.


**Tip:** To easily select the z-axis, it helps if you slightly rotate the view around the x-axis.



## Constrain the model

Now you'll create a translational joint between the ramp and the crate.

To constrain the model:

- From the Main Toolbox, right-click the **Joint** tool stack, and then select the **Translational Joint** tool .
- ◆ Use the options **2 Bod-1 Loc** and **Pick Feature**.
- ◆ Set the location of the translational joint at the crate geometry's base marker, **MARKER\_2**.
- ◆ Set the vector so it points up the ramp: select the x-axis of **MARKER\_2**.

# Workshop 7—Inclined Plane...

---

## Take measurements

To create a measure:

- Create an object (part) measure for the crate's acceleration along the ramp as you did in [To create object measures](#);, on page 106:
  - ◆ Characteristic: **CM acceleration**
  - ◆ Component: **X**
  - ◆ Represent coordinates in: **MARKER\_1** (for the location of the corner marker, see Figure 1 on page 127).

## Verify the mechanism (crawl-walk-run)

To verify the mechanism:

- 1 Simulate the model for **1 second** and **50 steps**.
- 2 Find the value of the crate's (constant) acceleration.

To verify this value, see [Without friction](#) in the [Closed-form solution](#), on page 136.


If the values do not match, check the units in the closed-form solution and in the model.

## Refine the model

In this section, you'll add joint friction to the translational joint using the  $\mu_s$ ,  $\mu_d$  values from the problem statement on page 124. You'll then simulate the model to see if the crate slides off the ramp.

**Tip:** Be sure that the only friction forces to consider are those resulting from reaction forces.

To add friction and simulate:

- 1 Display the joint's modify dialog box (right-click the translational joint, point to **Joint:JOINT\_1**, and then select **Modify**).
- 2 In the lower right corner of the Modify dialog box, select the **Friction** tool .
- 3 Fill in the coefficients of friction provided on page 124.
- 4 Leave the remaining friction parameters at their default values.
- 5 In the **Input Forces to Friction** section, clear the selection of **Bending Moment** and **Torsional Moment**.
- 6 Simulate the model and note if the crate slides off the ramp.
- 7 Right-click on the curve in the stripchart, and then select **Save Curve**.

# Workshop 7—Inclined Plane...

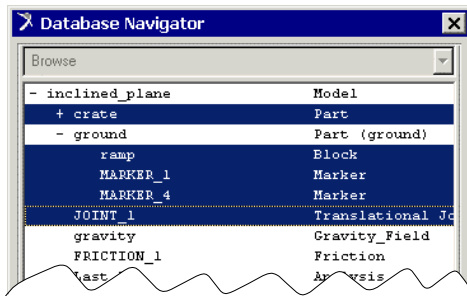
## Rotate the ramp and crate to $\theta = 20^\circ$

To rotate the ramp and crate, you'll create a group consisting of the crate part, joints, and geometry making up the ramp. You'll then select that group and rotate it.

To create a group:


- 1 From the **Build** menu, select **Group**.
- 2 Make a group, named **rotated\_objects**, containing:
  - The crate part.
  - The joint.
  - All of the geometry (including markers) on the ramp, but not the ground part itself, because, remember, you cannot rotate ground.

**Tip:** Right-click the **Objects in Group** text box, and browse for the objects you need. You can select multiple objects by holding down the **Ctrl** key. The next figure shows the objects you should select:




# Workshop 7—Inclined Plane...

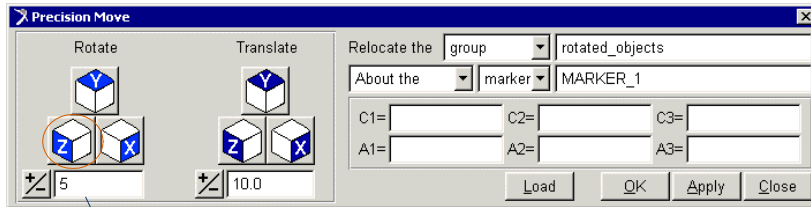
To rotate the group:

- 1 In the Main Toolbox, from the **Move** tool stack, select the **Precision Move** tool .
- 2 Set **Relocate the** to **group**.
- 3 In the text box to the right of **Relocate the**, enter the group name, **rotated\_objects**.
- 4 Set the menus in the second row to **About the** and **marker**.
- 5 In the text box to the right of these menus, enter **MARKER\_1**.

The Precision Move tool rotates objects in increments about a specified axis of the marker you just selected.

- 6 In the  text box, enter 5.
- 7 Select the **Z**-axis box. Note that you can select the axis box (either X, Y, or Z) to rotate a group to the desired orientation.

The following shows the Precision Move dialog box after you've completed this procedure.



Because you've already rotated the ramp to 15 degrees, and now you want to rotate it to 20, enter 5 as the angle.

- 8 Do **not** select **OK** or **Apply**, just select **Close**.

# Workshop 7—Inclined Plane...

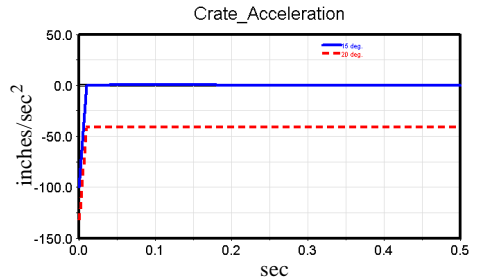
## Find inclination angles

To find the inclination angles between which the crate starts to slide:

- 1 Simulate the model and note if the crate slides off the ramp.

For an end time of 0.5 seconds, verify that the crate acceleration versus time stripchart matches the adjoining figure.

The initial spike is due to the acceleration (due to gravity) present at  $t=0$ .



- 2 Through trial and error, find the approximate angle (within  $0.5^\circ$ ) at which the crate starts to slide off the ramp. Use it to answer Question 1 in [Module review](#), on page 135.



## Save your work

Save your model and, unless you want to further explore the model as suggested in the next section, exit ADAMS/View.

## Optional tasks

Save your work before performing these tasks. **Do not** save your work after performing these tasks. If you must save the model after performing these tasks, give the model a different name.

To view an animation and its corresponding plot simultaneously:

- 1 Open ADAMS/PostProcessor.
- 2 Create two views by right-clicking the **Window layout** tool  on the toolbar and selecting the **2 Views, side by side** tool .
- 3 Select the left view.
- 4 Plot crate acceleration versus time.
- 5 Select the right view, and then change the **Plotting** menu to **Animation**.
- 6 Right-click the right view, and then select **Load Animation**.
- 7 Run the animation.

# Workshop 7—Inclined Plane...

---

## Module review

- 1 What is the approximate angle (within  $0.5^\circ$ ) at which the crate starts to slide off the ramp?

---

---

- 2 How can you tell if you can automatically add friction to a joint type?

---

---

- 3 What are the I and J markers?

---

---

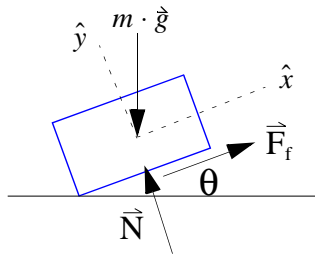
- 4 If a joint with friction enabled crosses its stiction threshold velocity ( $\Delta V_s$ ), how does the maximum stiction displacement ( $\Delta X_s$ ) affect the system?

---

---

# Workshop 7—Inclined Plane...

## Closed-form solution



Without friction:

$$\Sigma F_x = ma_x : -mg \cdot \sin\theta = ma_x$$

$$a_x = -g \sin\theta$$

$$\text{For } \theta = 15^\circ, a_x = -32.2 \sin(15^\circ)$$

$$a_x = -99.96 \text{ in/sec}^2 \text{ } (-8.33 \text{ ft/sec}^2)$$

With friction:

$$\Sigma F_y = 0 : -mg \cdot \cos\theta + N = 0$$

$$N = mg \cdot \cos\theta$$

Maximum angle ( $\theta_{max}$ ) at which the crate will not slide:

$$\Sigma F_x = 0 : F_f - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s \cdot N - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s \cdot mg \cdot \cos\theta_{max} - mg \cdot \sin\theta_{max} = 0$$

$$\mu_s - \tan\theta_{max} = 0$$

$$\theta_{max} = \text{atan}(\mu_s) = \text{atan}(0.30) = 16.7^\circ$$

## Workshop 7—Inclined Plane...

Once the crate starts sliding,

$$\Sigma F_x = ma_x: F_f - mg \cdot \sin \theta = ma_x$$

$$\mu_k \cdot N - mg \cdot \sin \theta = ma_x$$

$$\mu_k \cdot mg \cdot \cos \theta - mg \cdot \sin \theta = ma_x$$

$$\mu_k \cdot \cos \theta - \sin \theta = \frac{a_x}{g}$$

$$a_x = (\mu_k \cos \theta - \sin \theta) \cdot g$$

For  $\theta = 20^\circ$ ,  $a_x = (0.25 \cdot \cos 20^\circ - \sin 20^\circ) \cdot 32.2 \text{ ft/sec}^2$

$$a_x = -40.3 \text{ in/sec}^2 \text{ } (-3.45 \text{ ft/sec}^2)$$

### MSC.ADAMS results

- At angle  $\theta = 15^\circ$ ,  $a = 6.63e^{-5} \approx 0$
- At angle  $\theta = 20^\circ$ , the crate accelerates down the inclined plane at:  
 $a = -41.35 \text{ in/sec}^2 \text{ } (-3.45 \text{ ft/sec}^2)$
- Based on the angular increments of  $0.5^\circ$ ,  
 $(16.5^\circ < \theta_{\max} < 17.0^\circ)$

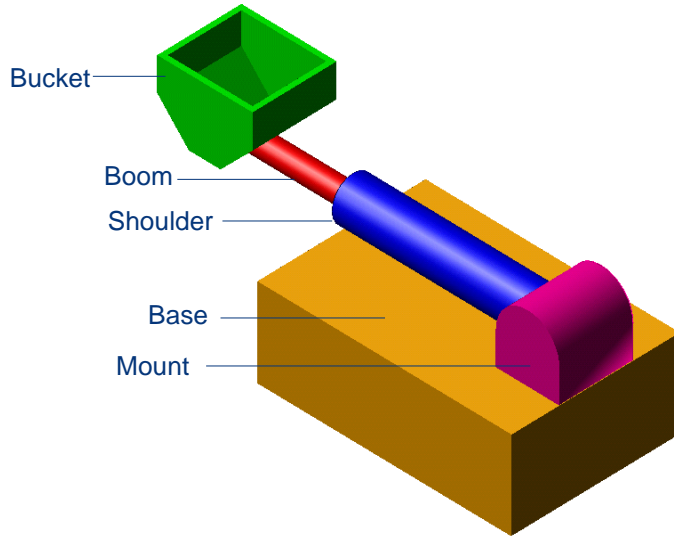
## Workshop 7—Inclined Plane...

---

# 8

## LIFT MECHANISM I

Use ADAMS/View to create each moving part of the lift mechanism shown next:



### What's in this module:

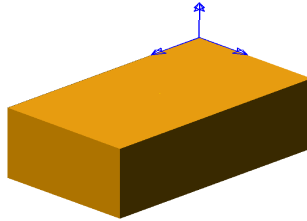
- Building Geometry, 140
- Construction Geometry Properties, 142
- Solid Geometry, 144
- Precise Positioning: Move, 145
- Workshop 8—Lift Mechanism I, 146
  - ◆ Module review, 153

# Building Geometry

---

## Properties of geometry

- It must belong to a part and moves with the part.
- It is used to add graphics to enhance the visualization of a part.
- It is not necessary for performing simulations.
- Locations and orientations are defined indirectly by parts using anchor markers.



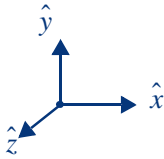
**Note:** If you move an anchor marker, all associated geometry moves with it.  
Conversely, anchor markers move when you move the associated geometry.

## Types of geometry in ADAMS/View

- Construction geometry
  - ◆ Includes objects that have no mass (spline, arc, and so on).
  - ◆ Is used to define other geometry.
- Solid geometry
  - ◆ Includes objects with mass (box, link, and so on).
  - ◆ Can be based on construction geometry.
  - ◆ Is used to automatically calculate mass properties for the parent part.

# Construction Geometry Properties

## Marker geometry



Has:

- Anchor marker, which is itself
- Parent: part
- Orientation and location

## Point geometry



Has:

- No anchor marker
- Parent: part
- Location

# Construction Geometry Properties...

## Polyline geometry



Has:

- No anchor marker
- Parent: part
- One line or multiple lines
- Open or closed
- Length, vertex points, and angle

## Arc geometry



Has:

- Anchor marker
- Parent: part
- Start and end angle, radius

## Spline geometry



Has:

- Anchor marker
- Parent: part
- Segment count, open/closed, points

# Solid Geometry

## Block geometry



Has:

- Anchor marker, which is the corner marker
- Parent: part
- Length (x), height (y), depth (z) with respect to corner marker

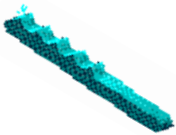
## Torus geometry



Has:

- Anchor marker, which is the center marker
- Parent: part
- Radius of ring (xy plane), radius of circular cross section ( $\perp$  to xy plane)

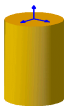
## Extrusion geometry



Has:

- Anchor marker, which is the reference marker
- Parent: part
- Open/closed profile, depth, forward/backwards

## Cylinder geometry



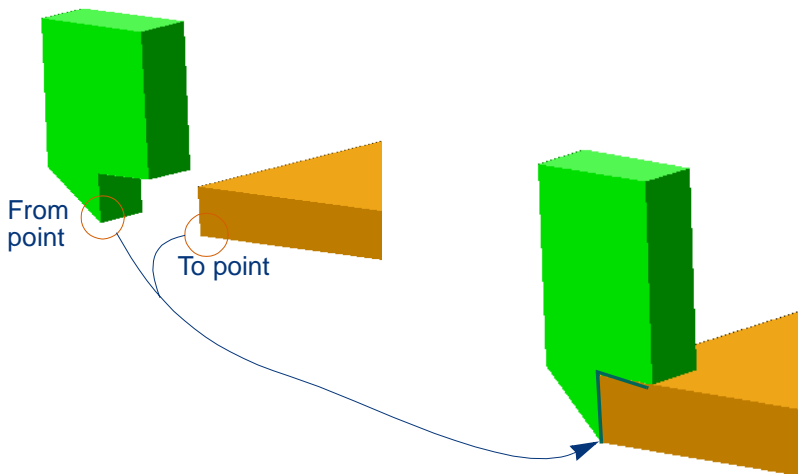
Has:

- Anchor marker, which is the center marker (placed at first end)
- Parent: part
- Length (z), radius

# Precise Positioning: Move

To move objects in ADAMS/View, specify:

- The object being moved (or copied).
- And:
  - ◆ Either, a point on the object, and the location to which the selected point will be moved.
  - ◆ Or, a vector and a distance along the vector.

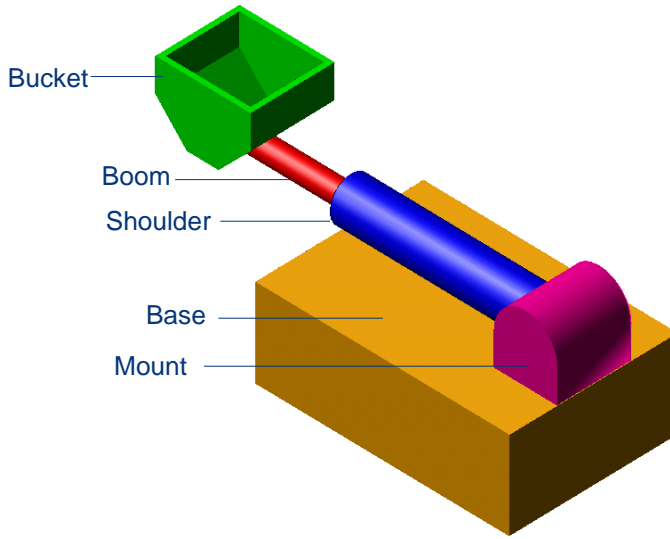


The moved object maintains its orientation.

# Workshop 8—Lift Mechanism I

## Problem statement

Use ADAMS/View to create each moving part of the lift mechanism shown next:



## Workshop challenge

If you'd like a challenge, build this model without going through the detailed steps presented on the following pages. Use the dimensions shown on page 148 to:

- Create the base
- Create the mount
- Create the shoulder
- Create the boom
- Fillet the mount to round off the edges
- Create the bucket
- Chamfer the bottom edges of the bucket
- Hollow out the bucket

Otherwise, continue by following the detailed instructions.

# Workshop 8—Lift Mechanism I...

## Background mechanism information

- The following diagrams provide the dimensions for building the lift mechanism.
- All units are in **meters**.

## Tips before you start

- Check the three-dimensional view of the model at regular intervals to verify that the parts are being placed in the right location.
- Rename parts as soon as you build them.
- You should save your model periodically throughout the modeling process. This becomes more important as you start to build complex models (recall the crawl-walk-run approach introduced in [Workshop 5—Projectile Motion](#), on page 83).

## Start the workshop

First, you'll start ADAMS/View from the directory *exercise\_dir/mod\_08\_lift\_mech\_1* and create a new model.

To start the workshop:

- Start ADAMS/View:
  - ◆ Set the directory to *exercise\_dir/mod\_08\_lift\_mech\_1*.
  - ◆ Create a new model named **lift\_mech**, with **Gravity** set to **Earth Normal (-Global Y)**, and **Units** set to **MKS - m, kg, N, s, deg**.

## Set up the working environment

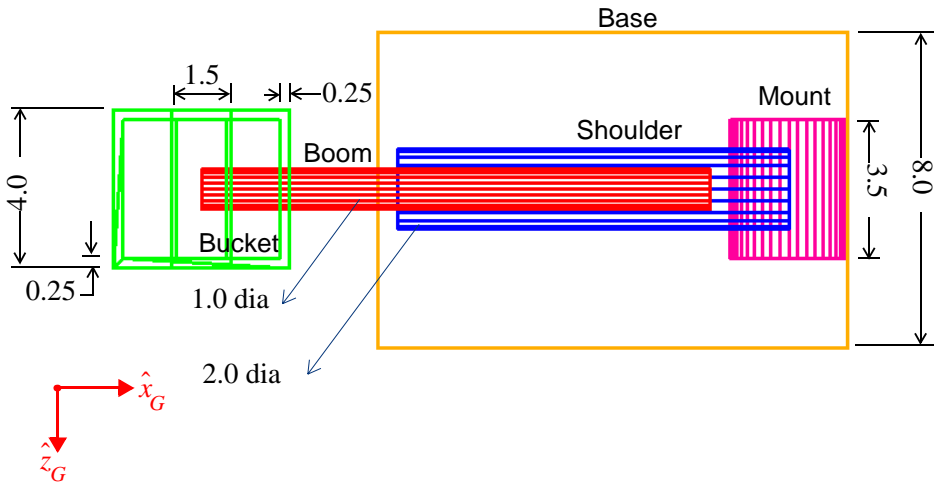
Now you'll set up the MSC.ADAMS environment to make it easier to build the model.

To set up the working environment:

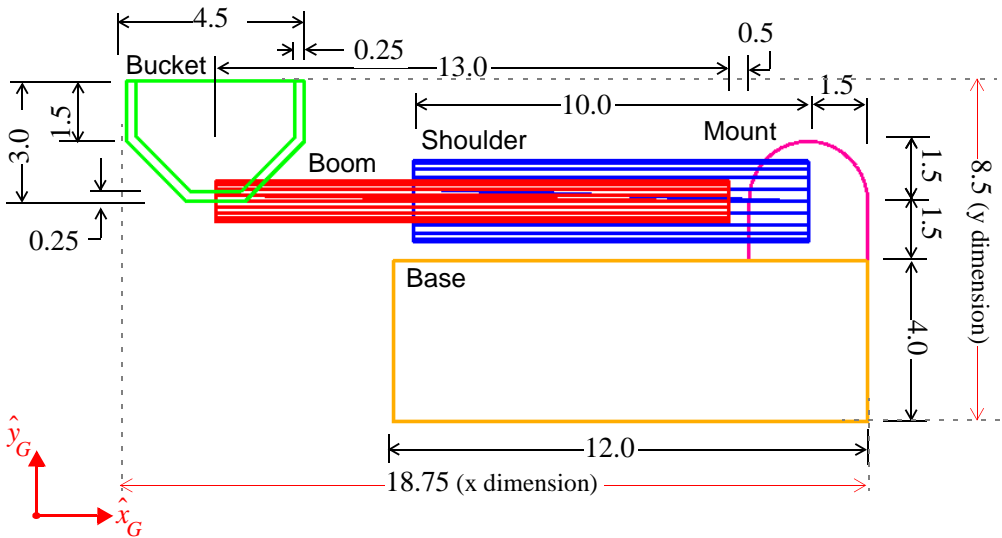
- 1 Adjust the grid based on the measurements given in the diagrams on page 148. The grid must be slightly larger than the maximum height and width of the model. (A 20 m x 20 m grid, with 1 m spacing in each direction should be good.)
- 2 Because the grid spacing is much greater than the default, you'll have to zoom out to see the grid on your screen.

# Workshop 8—Lift Mechanism I...

Top View of Lift Mechanism



Front View of Lift Mechanism



## Build all parts except for the bucket

In this section, you'll create all the parts except the bucket. For information on how to build the parts, refer to the diagrams on page 148.

To build the parts:


1 Build the **base** part.

**Tip:** Note the orientation of the block with respect to the xy plane.

To get the marker coordinates, right-click on the location where you're starting drawing the base, point to the name of the marker, and then select Info.

2 Be default, the screen icons are set for models in millimeters. Because your model is in meters, you should adjust the icon sizes so you can see the icons. To adjust the icons, from the **Settings** menu, select **Icons**, and then set **New Size** to 1.

3 Build the main feature of the **mount** part by creating a block.

4 Inspect your model. Note that the mount must be centered on the base. If necessary, use the vector option of the **Point-to-Point** tool  (also known as the Position: Move-Translate tool) on the **Move** tool stack to slide the mount along the base, in the global z direction, by 2.25 m.

5 Before building the shoulder, set the working grid to cut through the center of the block representing the mount part: **Settings** → **Working Grid** → **Set Location** → **Pick**, and then select the cm marker of the block.


6 Change the spacing of the working grid to .5 m. If you do not change the spacing, you will notice that when you try to create the shoulder part, ADAMS/View snaps to the nearest grid point, thus building the shoulder in a position that is not parallel to the base part.

7 Use the **Cylinder** tool  to build the **shoulder** part.

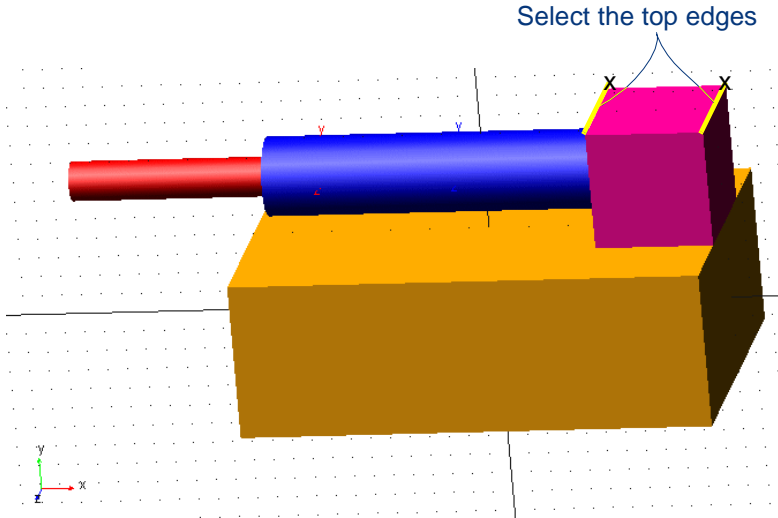
8 Build the **boom** part.

9 Use the Location Event, as you did in [Build the pendulum link](#), on page 102, to start the cylinder 2 meters over from the center-of-mass (cm) marker of the mount part.

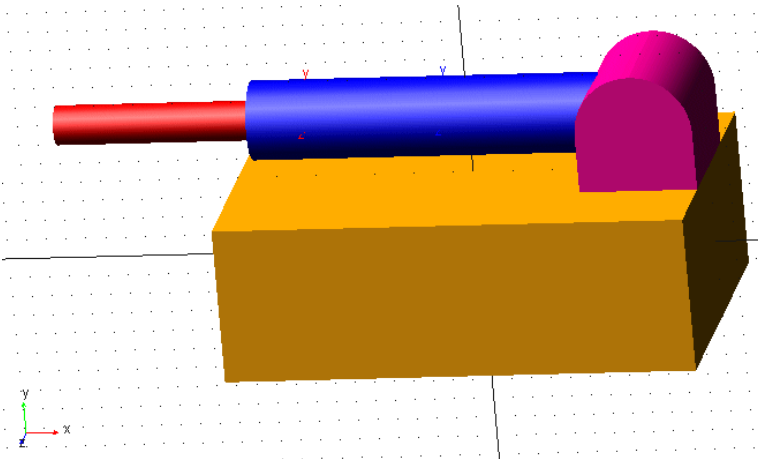
# Workshop 8—Lift Mechanism I...

10 Apply fillets to the mount part using the **Fillet** tool 

- In both the **Radius** and **End Radius** text boxes, enter 1.5 m.
- Left-click each edge, and then right-click to create. Refer to the next figure to see the edges you should select.



The filleted mount part should look as shown next:




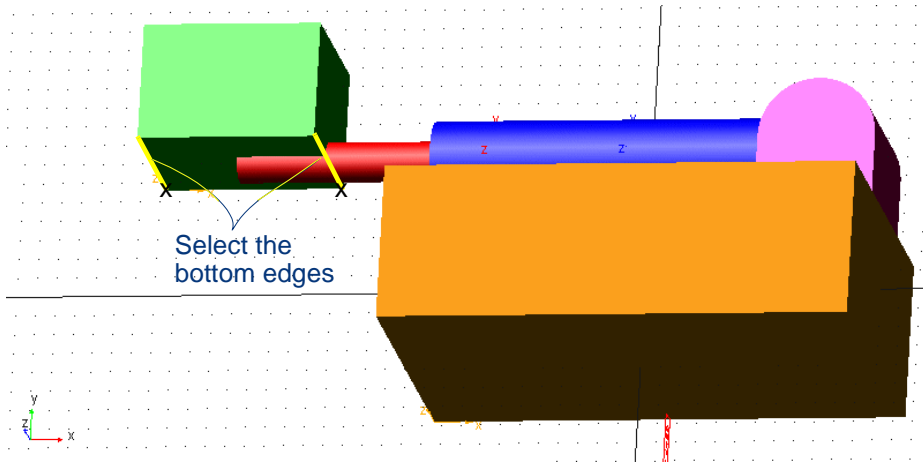
# Workshop 8—Lift Mechanism I...

## Build the bucket

Now build the bucket.

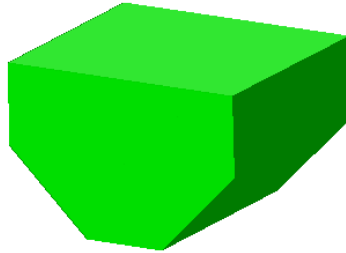
To build the bucket:


- 1 Build a block with the largest dimensions of the bucket:
  - Length: 4.5 m
  - Height: 3.0 m
  - Depth: 4.0 m
- 2 Chamfer the front and back, bottom corners of the block using the Chamfer tool 
  - In the **Width** text box, enter 1.5 m.
  - Left-click each edge, and then right-click to create. Refer to the next figure to see the edges you should select.



## Workshop 8—Lift Mechanism I...

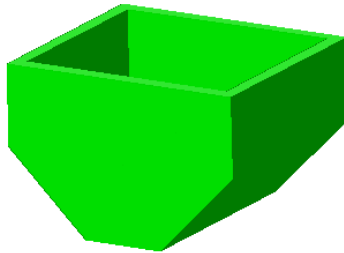
The chamfered bucket should look as shown next:



3 Hollow out the solid bucket using the **Hollow** tool 



- In the **Thickness** text box, enter **0.25 m**.
- Pierce the top face of the bucket.

The hollowed bucket should look as shown next:



### Check model topology by parts

To check model topology:

- Check model topology by parts (from the **Status** bar, right-click the **Information** tool stack , and then select the **Model topology by parts** tool ) to ensure that there are no floating parts that are not accounted for.

There should be six parts, including ground.

### Save your work

Save your work such that the saved file contains only the model topology and not the results (**File** → **Export**).

# Workshop 8—Lift Mechanism I...

---

## Optional tasks

To refine the geometry of the lift mechanism:

- 1 Using the **Torus** tool, add tires to the lift mechanism.
- 2 Using the **Fillet** tool, round the edges of the base.

## Module review

- 1 What is the basic difference between construction geometry and solid geometry?

---

---

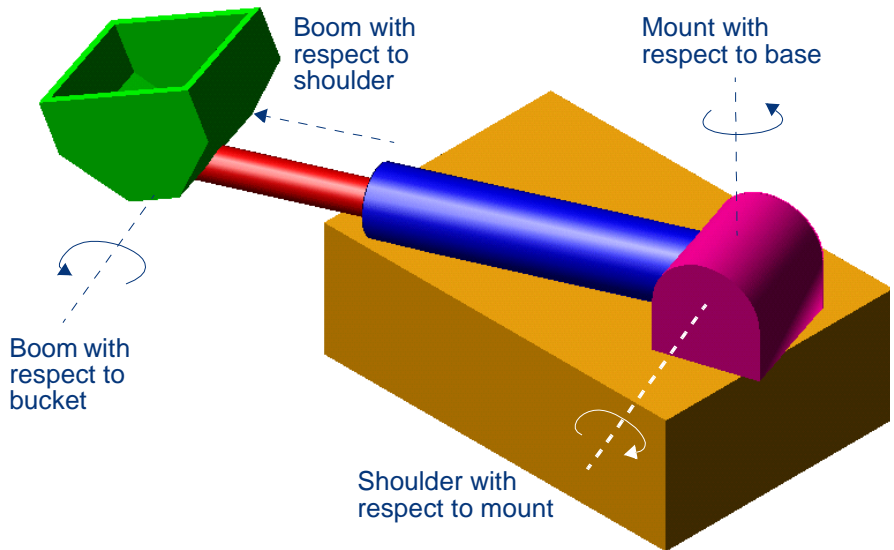
- 2 In this workshop were instructions for changing the position of parts. Name three of the four methods introduced.

---

---



Constrain the lift mechanism model by adding joints and motions as shown next:



### What's in this module:

- Fixed Joint, DOF Removed by, 376
- Applying Motion, 156
- Joint Motion, 157
- Functions in MSC.ADAMS, 158
- Workshop 9—Lift Mechanism II, 159
  - ◆ Module review, 164

# Applying Motion

---

## ADAMS/View provides two types of motions

- Joint motion
- Point motion

### Joint motion

- There are two types:
  - ◆ Translational: applied to **translational** or **cylindrical** joints (removes 1 DOF).
  - ◆ Rotational: applied to **revolute** or **cylindrical** joints (removes 1 DOF).
- You define the joint to which motion is applied.
- MSC.ADAMS automatically uses the joint's I and J markers, bodies, and single DOF.
- You define function for magnitude.

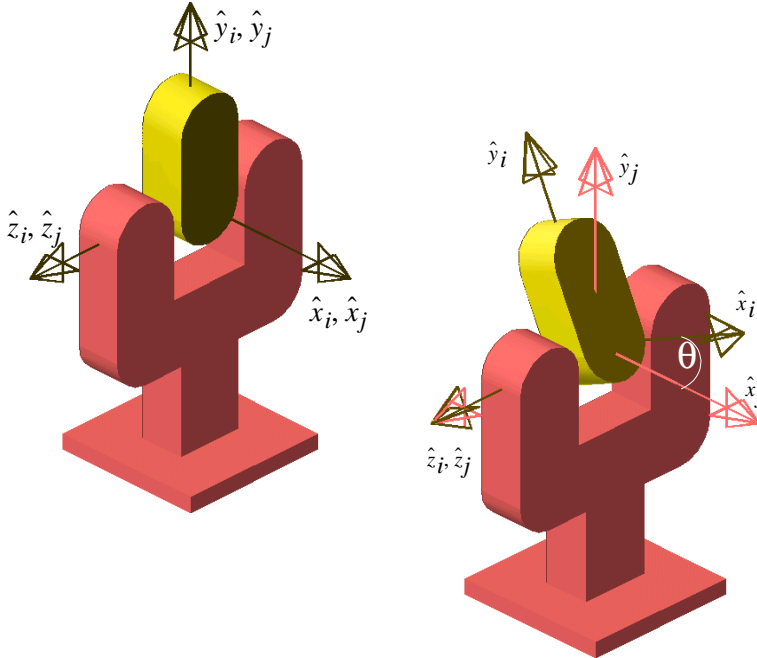
**Questions:** How does a motion remove DOF?

Does this mean that a motion is considered a constraint?

# Joint Motion

## Marker usage in joint motions

- The I and J markers (and, therefore, the parts to which they belong) referenced in the joint move with respect to each other as follows:



- The I and J markers overlap when motion  $\theta_t = 0$ .
- During simulation, the z-axes of both markers are aligned.
- You can define motion magnitude as a:
  - ◆ Displacement
  - ◆ Velocity
  - ◆ Acceleration function of time

## Definition of functions in MSC.ADAMS

- You use functions to define magnitudes of input vectors used in:
  - ◆ Motion drivers
  - ◆ Applied forces
- Functions can depend on time or other system states, such as displacement, velocity, and reaction forces.
- Every function evaluates to a single value at each particular point in time.
- Motion drivers can only be a function of time:
$$M = f(\text{time})$$
- Functions defining motion driver magnitudes can be:
  - ◆ Displacement (time)
  - ◆ Velocity (time)
  - ◆ Acceleration (time)

**Note:** You can use the Function Builder to create and verify functions in ADAMS/View. To access the Function Builder, right-click any text box that expects a function.

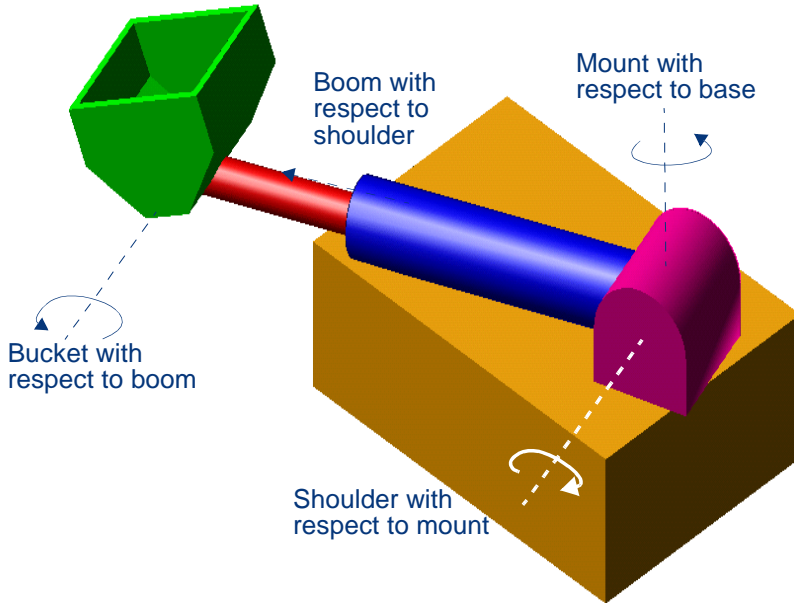


Display the Function Builder and press F1 to learn about creating functions.

# Workshop 9—Lift Mechanism II

## Problem statement

Constrain the lift mechanism model by adding joints and motions as shown next:



## Workshop challenge

If you'd like a challenge, add the following joints and motions without going through the detailed steps presented on the following pages:

- Rotational motion to the mount-to-base joint.  $D(t) = 360d \cdot \text{time}$
- Rotational motion to the shoulder-to-mount joint.  $D(t) = \text{STEP}(\text{time}, 0, 0, 0.10, 30d)$
- Translational motion to the boom-to-shoulder joint.  $D(t) = \text{STEP}(\text{time}, 0.8, 0, 1, 5)$
- Rotational motion to the bucket-to-boom joint.  $D(t) = 45d \cdot (1 - \cos(360d \cdot \text{time}))$

Otherwise, continue by following the detailed instructions.

# Workshop 9—Lift Mechanism II...

## Model description

In this workshop, you use the model you built in [Workshop 8—Lift Mechanism I](#), on page 146.

## Tips before you start

- Use the figure on page 159 to find out what type of constraints you need.
- Simulate the model at regular intervals to check the constraints.
- Rename joints as soon as you create them.
- Adjust icon sizes whenever necessary (see [Set up the working environment](#), on page 147 of [Workshop 8—Lift Mechanism I](#).)

## Start the workshop

Note that the file for this exercise is not in the current working directory.

To start the workshop:

- Start ADAMS/View:
  - ◆ Set the directory to *exercise\_dir/mod\_09\_lift\_mech\_2*.
  - ◆ From the directory *exercise\_dir/mod\_08\_lift\_mech\_1*, import the model that you created in the previous module.






If you need a fresh copy of the model, import the command file *lift\_mech\_1\_completed.cmd* from the directory *exercise\_dir/mod\_08\_lift\_mech\_1/completed*.

# Workshop 9—Lift Mechanism II...

## Constrain the parts

In this section, you'll constrain the parts that you created in the previous workshop. The figure in the [Problem statement](#), on page 159, shows how you should constrain the parts.

To constrain the parts:



- 1 Use the **Fixed joint** tool  to fix the **base** to **ground**.
- 2 Constrain **mount** to **base** ():
  - Use the options **2 Bod-1 Loc** and **Pick Feature**.
  - Place the joint at the mount's cm marker.
  - Select the + y-axis as the axis for the rotation.
- 3 Constrain **shoulder** to **mount** ():
  - Use the option **Normal To Grid**.
  - Right-click to select the cylinder's anchor marker.
- 4 Constrain the **boom** to the **shoulder** ():
  - Use the option **Pick Feature**.
  - Select the x-axis as the axis for the translation.
- 5 Constrain the **bucket** to the **boom** ():
  - Use the option **Normal To Grid**.
  - Select the end point of the cylinder.

# Workshop 9—Lift Mechanism II...

## Verify the model (crawl-walk-run)

Before continuing, check your work by checking model topology and by performing a simulation.

To verify your model:

- 1 Check model topology by constraints (from the **Status** bar, right-click the **Information** tool stack , and then select the **Model topology by constraints** tool ) to ensure that all the parts are constrained as expected.
- 2 Perform a simulation.


Are the visual results of the simulation (the animation), what you expected?

## Add joint motions to your model

When adding motions, follow the instructions in the Status bar. The Status bar instructs you to select a legitimate joint for the particular type of motion (for example, a revolute joint for rotational motion). When your cursor hovers over a potential joint, its name appears in the working window. To select a joint, simply left-click the joint, once its name appears.

Build the joint motions using the default expressions in the Main Toolbox container and then modify the expressions using the Rotational Joint Motion Modify dialog box (right-click the joint, point to the joint name, and then select Modify).

To add joint motions:

- 1 Use the **Rotational Joint Motion** tool  to add a motion to the mount-to-base joint such that:  
$$D(t) = 360d \cdot \text{time}$$
- 2 Add a motion to the shoulder-to-mount joint such that:  
$$D(t) = -\text{STEP}(\text{time}, 0, 0, 0.10, 30d)$$

**Note:** By using Normal to Grid, the motion will be opposite the illustration in [Problem statement](#), on page 159 (by the right-hand rule). When a motion is opposite of what you expect, add a negative sign in front of the expression, in the Modify Motion dialog box.

We will discuss the specifics of the STEP function in the next module, [Lift Mechanism III](#), on page 165.

## Workshop 9—Lift Mechanism II...

---

3 Add a translational motion to the boom-to-shoulder joint such that:

$$D(t) = -\text{STEP}(\text{time}, 0.8, 0, 1, 5)$$

4 Add a motion to the bucket-to-boom joint such that:

$$D(t) = 45d*(1-\cos(360d*\text{time}))$$

### Run a simulation

To run a simulation:

Run a simulation such that the mount achieves one full rotation.

### Save your work

To save your work:

- Save the model such that the saved file contains only the model topology and not the results (**File** → **Export**).

### Optional tasks

If you did not already do so as explained in the [Optional tasks](#), on page 153, for Lift Mechanism I:

- Add tires to your model using the **Torus** tool.
- Constrain the tires to the base using revolute joints.

# Workshop 9—Lift Mechanism II...

---

## Module review

1 What are the markers that a joint refers to called?

---

---

2 When motion is applied to a joint, what dictates its direction (positive versus negative)?

---

---

3 Are motions considered a constraint? Why?

---

---

4 Is it possible to determine the torque required to achieve a prescribed motion imposed on a revolute joint? How?

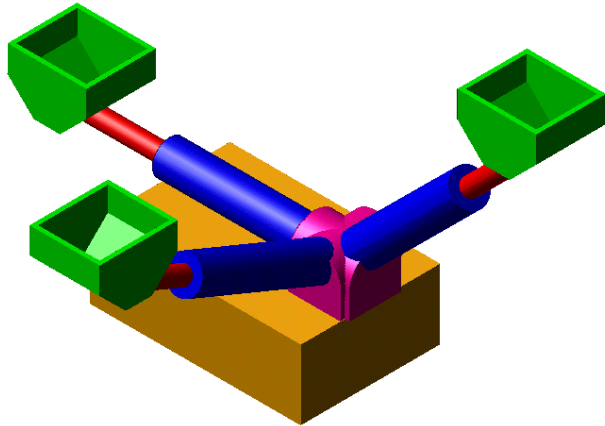
---

---

# 10 LIFT MECHANISM III

---


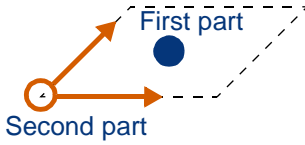
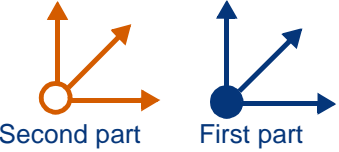
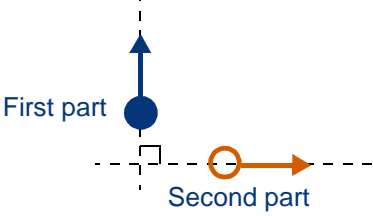
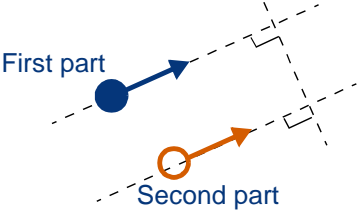
Constrain the bucket such that the base of the bucket always maintains its horizontal orientation (therefore, keeping the bucket-passenger safe) as shown next:



## What's in this module:

- Types of Joint Primitives, 166
- Perpendicular Joint Primitive, 167
- Workshop 10—Lift Mechanism III, 169
  - ◆ Module review, 172

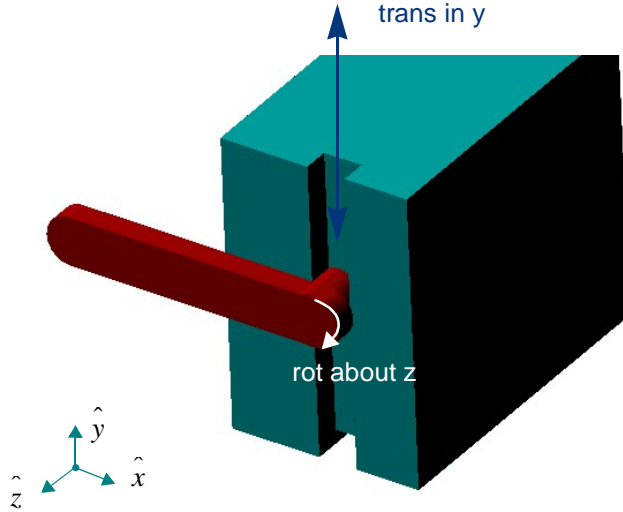
# Types of Joint Primitives

Description:	DOF removed:	Illustration:
<p><b>Inline</b> - One point can only move along a straight line</p>	<p>Two translational</p>	
<p><b>Inplane</b> - One point can only move in a particular plane</p>	<p>One translational</p>	
<p><b>Orientation</b> - One coordinate system cannot rotate with respect to another</p>	<p>Three rotational</p>	
<p><b>Perpendicular</b> - One coordinate system can rotate about two axes</p>	<p>One rotational</p>	
<p><b>Parallel axis</b> - One coordinate system can rotate about one axis</p>	<p>Two rotational</p>	

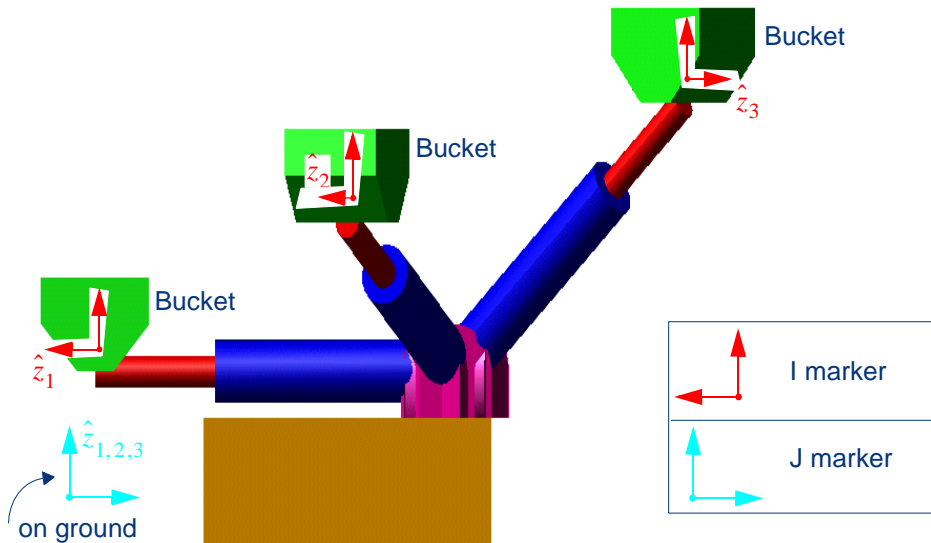
**See also:** DOF removed by joint primitives, on page 374

# Perpendicular Joint Primitive

Example using inline and parallel primitives



Example of I and J markers in a perpendicular joint primitive



## Perpendicular Joint Primitive...

---

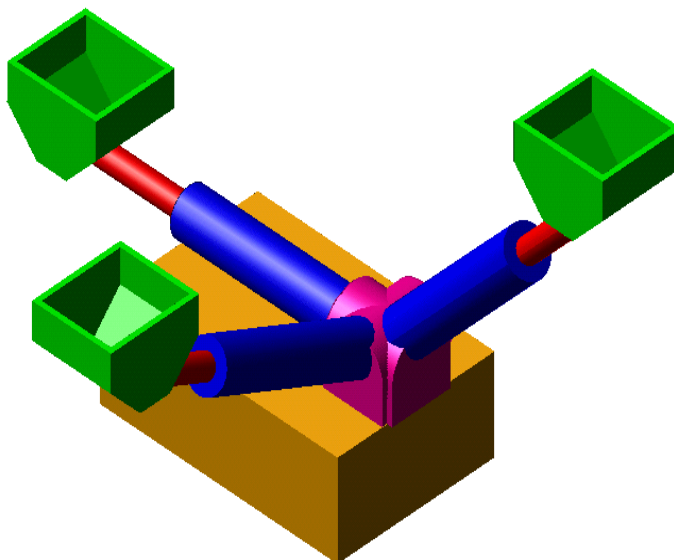
- I marker:
  - ◆ Parent part: Bucket
  - ◆ Its yz-plane is coplanar to the ground plane.
- J marker:
  - ◆ Parent part: ground
  - ◆ Its z-axis is perpendicular to the z-axis of the I marker.
- When constrained, the z-axes of the I and J markers are **always perpendicular** during simulation.
- Use the construction method 2 Bod-2 Loc.

**Question:** Would the lift mechanism behave any differently if the J marker's parent part was Base?

# Workshop 10—Lift Mechanism III

## Problem statement

Constrain the bucket such that the base of the bucket always maintains its horizontal orientation (thus keeping the bucket passenger safe) as shown next:



## Model description

In this workshop, you use the model you saved in [Workshop 9—Lift Mechanism II](#), on page 159.

# Workshop 10—Lift Mechanism III...

## Start the workshop

Note that the file for this exercise is not in the current working directory.

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_10\_lift\_mech\_3*.
- 2 From the directory *exercise\_dir/mod\_09\_lift\_mech\_2*, import the model that you created in the previous module.

If you need a fresh copy of the model, import the command file *lift\_mech\_II\_completed.cmd* from the directory *exercise\_dir/mod\_09\_lift\_mech\_2/completed*.

## Constrain the bucket

Now you will constrain the bucket to ground using the appropriate joint primitive.

To constrain the bucket:

- 1 Delete the motion on the bucket-to-boom joint.
- 2 Verify the model. The model should have only one degree of freedom.
- 3 From the **Build** menu, select **Joints**.
- 4 Select the appropriate joint primitive and use it to constrain the bucket:
  - Use the construction method **2 Bod-2 Loc** and see [Example of I and J markers in a perpendicular joint primitive](#), on page 167 for assistance.
  - Select the bucket and then select the ground.
  - When selecting the markers, note that I marker can be any marker on the bucket part and the J marker can be any ground marker (note that there is one at the fixed joint location).
  - When selecting the direction, select the first direction as the x direction of the I marker, and the second direction as the y direction of the J marker.

# Workshop 10—Lift Mechanism III...

## Verifying the orientation

Now you will confirm that you created your joint primitive correctly by verifying the orientation of its I and J markers.

To verify the orientation:

- 1 Get information on the perpendicular joint primitive that you just created by right-clicking on the joint icon and selecting **Info**.
- 2 Note the names of the I and J marker, and then select **Close**.  
I marker name: \_\_\_\_\_  
J marker name: \_\_\_\_\_
- 3 Select the I marker (on the bucket) and verify that the z-axis of the marker on the bucket is pointing in the (positive or negative) global x direction.
- 4 Select the J marker (on the ground part) and verify that the z-axis of the marker on ground is pointing in the (positive or negative) global y direction.

If the I and J markers for your joint primitive do not match the orientations in steps 3 and 4, then you may have made a mistake when you created this joint.

## Verify the model and then run a simulation

In this section, you will perform a simulation with icons on.

To verify the model and run a simulation:

- 1 Verify the model. The model should have zero degrees of freedom.
- 2 From the **Settings** menu, point to **Solver**, and then select **Display**.
- 3 Set **Icons** to **On**.
- 4 Simulate the model.

# Workshop 10—Lift Mechanism III...

## Save your work

To save your work:

- 1 Save the model such that the saved file contains only the model topology and not the results (**File** → **Export**).
- 2 Unless you want to further experiment with the model, as instructed in the next section, exit ADAMS/View.

## Optional tasks

To complete extra tasks from previous module:

If you did not already do so as mentioned in the [Optional tasks](#), on page 153, for Lift Mechanism 1:

- Add tires to your model using the **Torus** tool.
- Constrain the tires to the base using revolute joints.

To make the bucket transparent:

- 1 From the **View** menu, select **Render Mode**, and then select **Shaded**.
- 2 Right-click the bucket, point to **Part: bucket**, and then select **Appearance**.
- 3 To set the transparency of the part, use the slider bar:  
0%: fully visible, 100%: completely invisible.

To check graphical topology of the model:

- 1 From the **Tools** menu, select **Database Navigator**.
- 2 Set the menu at the top of the Database Navigator to **Graphical Topology**.
- 3 Browse to the bucket and notice how many constraints act on the bucket.

## Module review

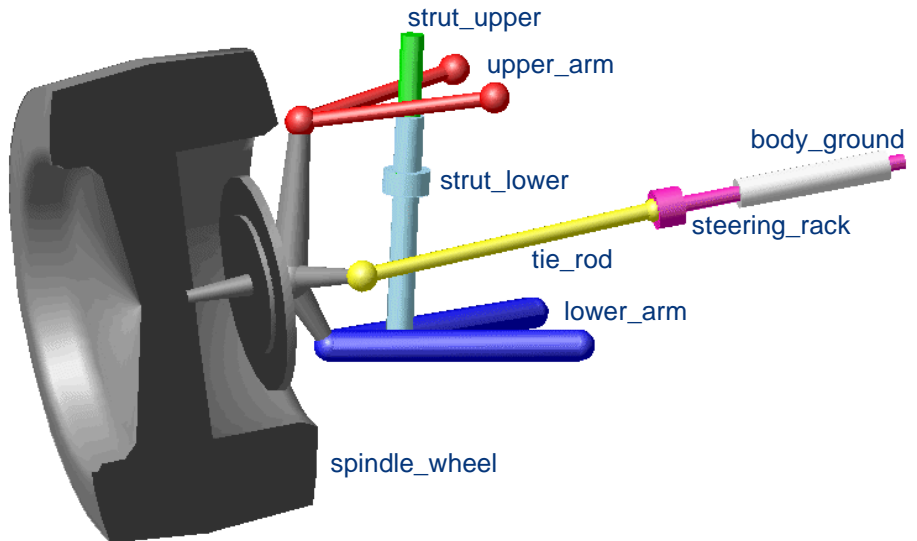
- 1 When you use the construction method of 2 Bod-2 Loc, how does the order in which you select parts affect the order in which you select the locations and orientations?

---

---

# 11 SUSPENSION SYSTEM I

Set up the suspension such that it moves 80 mm in jounce and rebound.



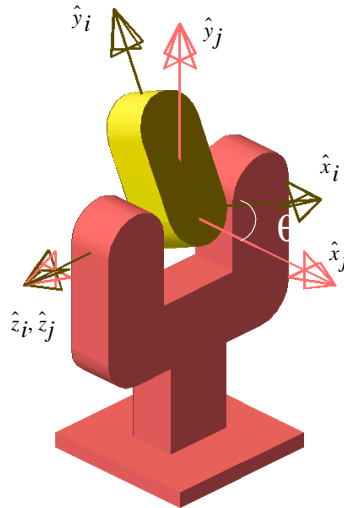
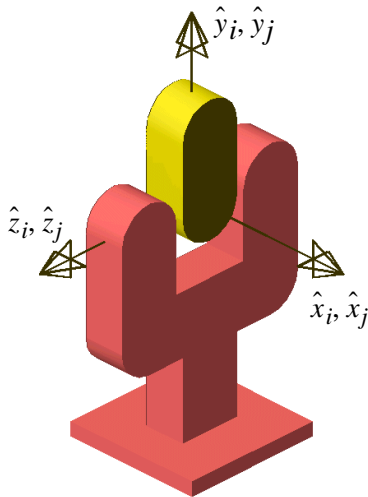
## What's in this module:

- Applying Point Motions, 174
- Workshop 11—Suspension System I, 176
  - ◆ Module review, 182

# Applying Point Motions

## Point motions

- There are two types:
  - ◆ Single-point motion (removes 1 DOF)
  - ◆ General-point motion (removes 1 to 6 DOF)
- You define:
  - ◆ I and J markers to which motion is applied (via two bodies, location and orientation).
  - ◆ Constraint nature of the motion (between 1 and 6 DOF).
  - ◆ Functions for magnitudes of motion.

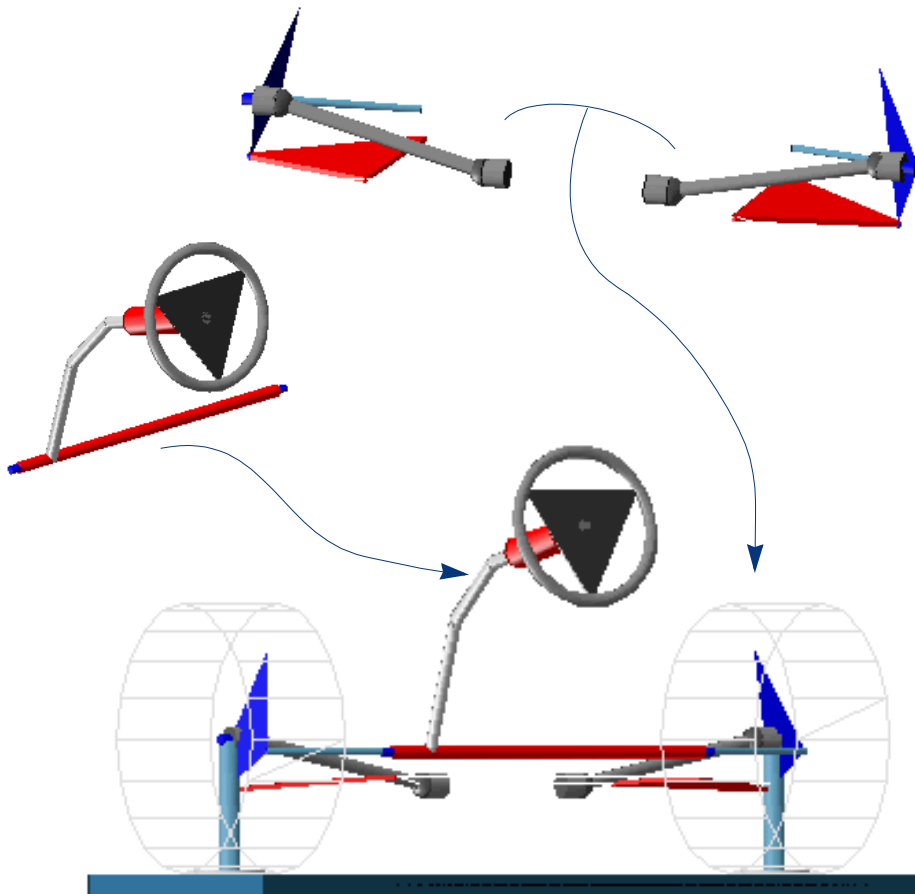


# System-Level Design

## The crawl-walk-run approach

- Do not build the entire mechanism at once.
- As you add a new component, make sure that it works correctly.
- Check your model at regular intervals.

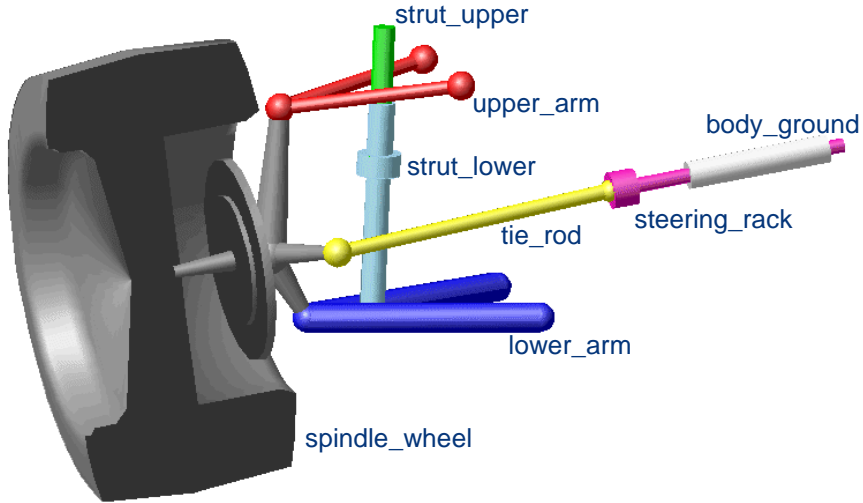
Avoid the need for complex debugging by following the crawl-walk-run approach



# Workshop 11—Suspension System I

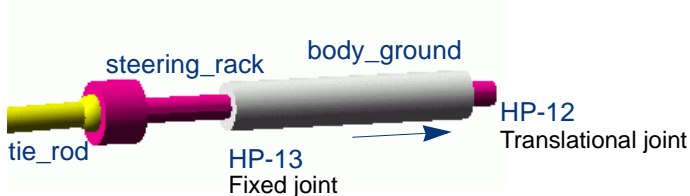
## Problem statement

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.



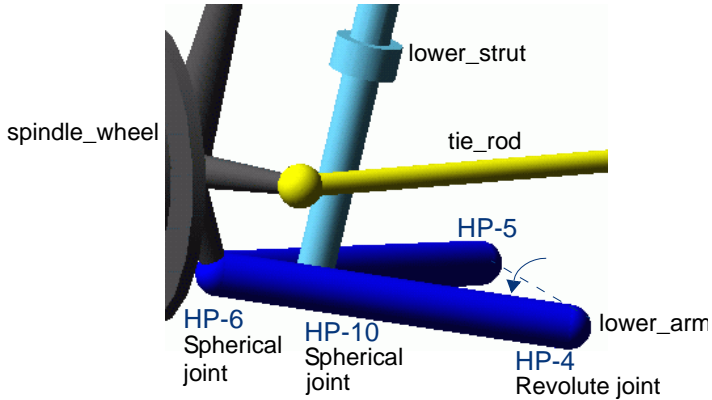
## Model description

- The given model is a geometric representation of a short-long arm (SLA) suspension subsystem.
- The steering\_rack and body\_ground are constrained as shown in the following figure:
  - ◆ A translational joint connects the steering\_rack to the body\_ground.
  - ◆ A fixed joint connects the body\_ground to ground.

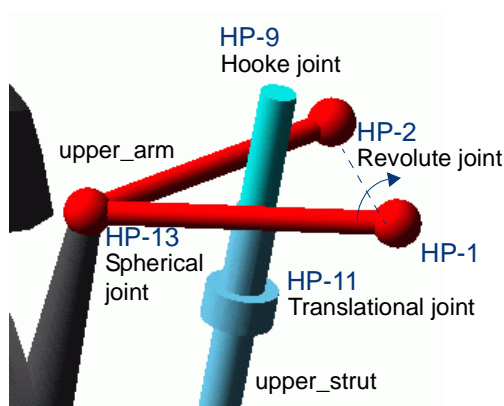


# Workshop 11—Suspension System I...

- The lower\_arm and lower\_strut are constrained as shown next:
  - ◆ A spherical joint connects the lower\_strut to the lower\_control\_arm.
  - ◆ A revolute joint connects the lower\_arm to the body\_ground.



- The upper\_arm and upper\_strut are constrained as shown next:
  - ◆ A revolute joint connects the upper\_arm to the body\_ground.
  - ◆ A hooke joint connects the upper\_strut to the body\_ground.



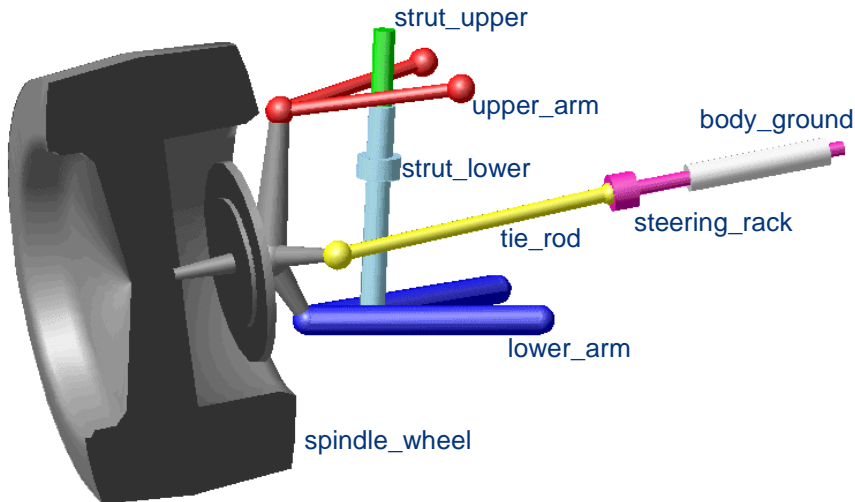
# Workshop 11—Suspension System I...

## Start the workshop

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_11\_suspension\_1*.
- 2 Import the model command file *suspension\_parts\_start.cmd*.

This file contains commands to build a model named *suspension* and the following parts with geometric representation:



# Workshop 11—Suspension System I...

## Inspect the model

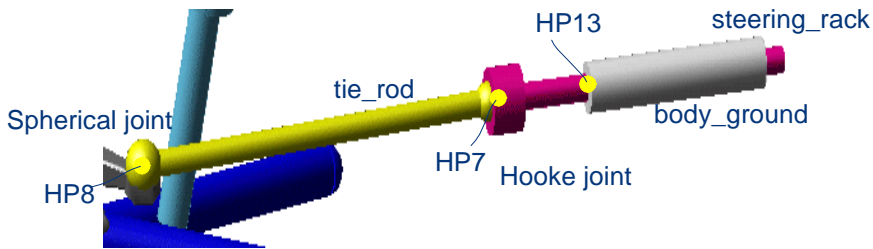
In this section, you'll investigate the model to note its movement and topology, especially that of the part `tie_rod`.

To inspect the model:

- 1 Simulate the model, noting the movement of `tie_rod`.
- 2 From the **Tools** menu, select **Database Navigator**.
- 3 Set the pull-down menu at the top of the Database Navigator to **Graphical Topology**.
- 4 Double-click **suspension**, and then select **tie\_rod**.

## Constrain the suspension subsystem model


Constrain the tie rod as shown next:



When constraining the `tie_rod`, use the ADAMS/View hardpoints provided with the model.

# Workshop 11—Suspension System I...

To constrain the model:

1 Create a spherical joint :

- Select the **2 Bod-1 Loc** option, **Normal to Grid**.
- **First body:** Spindle\_Wheel
- **Second body:** tie\_rod
- Set the location to **HP8**.

**Note:** HP8 belongs to ground. We are simply referencing its location for the creation of the new markers (I and J) that represent the joint.

2 Use the **Hooke joint** tool  to create a hooke joint:

- Select the **2 Bod-1 Loc** option, **Pick Feature**.
- **First body:** tie\_rod
- **Second body:** steering\_rack
- Set the location to **HP7**.
- Set the first direction from vector **HP7** to **HP8**.
- Set the second direction from vector **HP7** to **HP13**.


**Tip:** When setting the direction, move your cursor in the direction of the ending vector (HP8 and HP13, respectively), until its name appears on the screen. When the name appears, left-click to select it.

- 3 Inspect the tie rod again, and notice the connection representations to the steering\_rack and spindle\_wheel.
- 4 Simulate the model.

# Workshop 11—Suspension System I...

## Apply motions

To apply motions:

- 1 At the marker `.Spindle_Wheel.Center`, apply a point motion  in the y direction of the `Spindle_Wheel.Center`.

**Tip:** Make sure **Construction** is set to **1 Location, Pick Feature**.

- 2 Modify the motion to use the function,  $\text{Displacement}(\text{time}) = 80 * \sin(360d * \text{time})$ .
- 3 Modify the translational joint, `rck_body_joint`, between the `steering_rack` and the `body_ground` to be a fixed joint, so that the `steering_rack` is unable to translate during a simulation.



## Verify and simulate the model

Now, to see the model's full range of motion, simulate it.

To verify and simulate the model:

- 1 Verify the model.
- 2 Run a one-second, 50-step simulation.

## Save your work

To save your work:

- 1 Save your model as `suspension_parts.cmd`.

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.

# Workshop 11—Suspension System I...

---

## Optional tasks

To modify hardpoint locations:

- 1 From the **Tools** menu, select **Table Editor**.
- 2 From the options along the bottom of the Table Editor, select **Points**.
- 3 Change the **Loc Y** value of **HP3** from **351.05** to **400**.

As you make this change, note how the upper arm's connection to the spindle changes.

## Module review

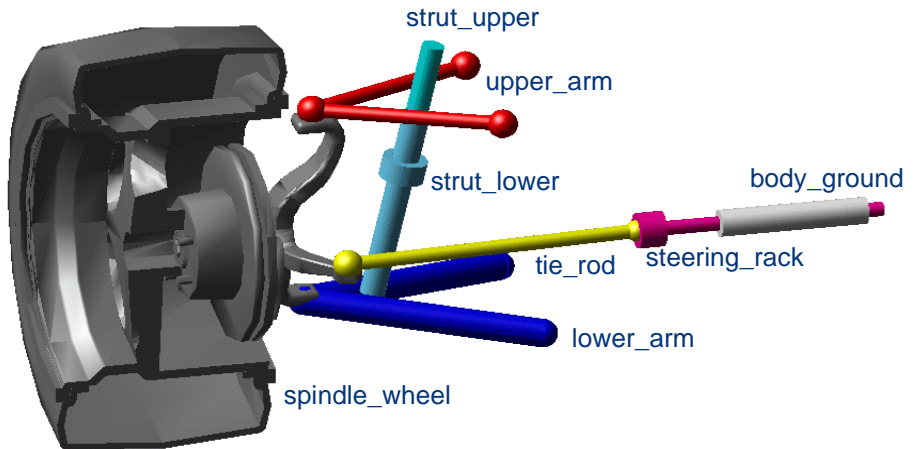
- 1 What is the difference between a point motion and a joint motion?

---

---

# 12 SUSPENSION SYSTEM II

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.



## What's in this module:

- Taking Measurements, 184
- Displacement Functions, 185
- Importing CAD-Based Geometry, 186
- Workshop 12—Suspension System II, 187
  - ◆ Module review, 194

# Taking Measurements

---

## Point-to-point measures

- Measure kinematic characteristics of one point relative to another point, such as the relative velocity or acceleration.
- To define them, you specify:
  - ◆ Characteristic (displacement, velocity, or acceleration)
  - ◆ To-point marker location (I marker)
  - ◆ From-point marker location (J marker, default is global origin)
  - ◆ Represent coordinates in marker coordinate system (R marker, default is GCS)
  - ◆ Component to return (x, y, z, or magnitude)
- ADAMS/View uses displacement, velocity, or acceleration functions.

## Function measures

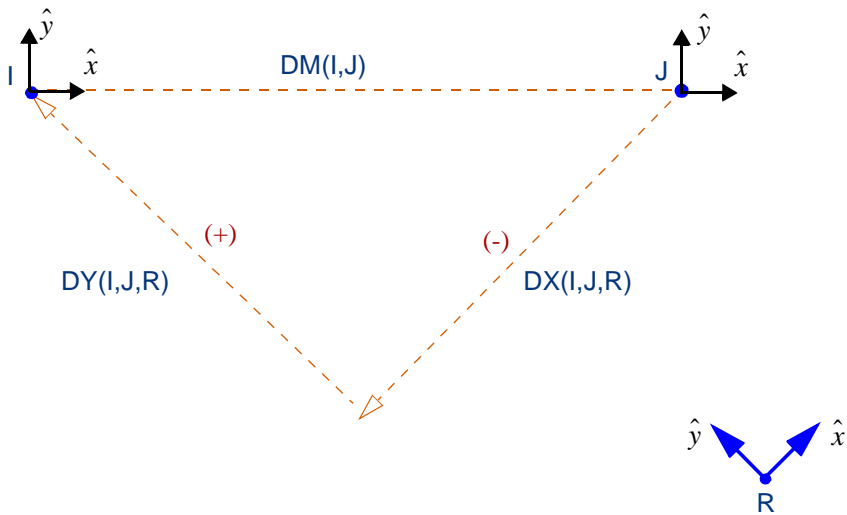
- Let you evaluate arbitrary, user-defined expressions of interest during solution runtime, such as:
  - ◆ Flow rate
  - ◆ Aerodynamic pressure
  - ◆ Stress
- You can create them in the Function Builder.
- Unlike other measures, function measures let you specify plotting attributes.

# Displacement Functions

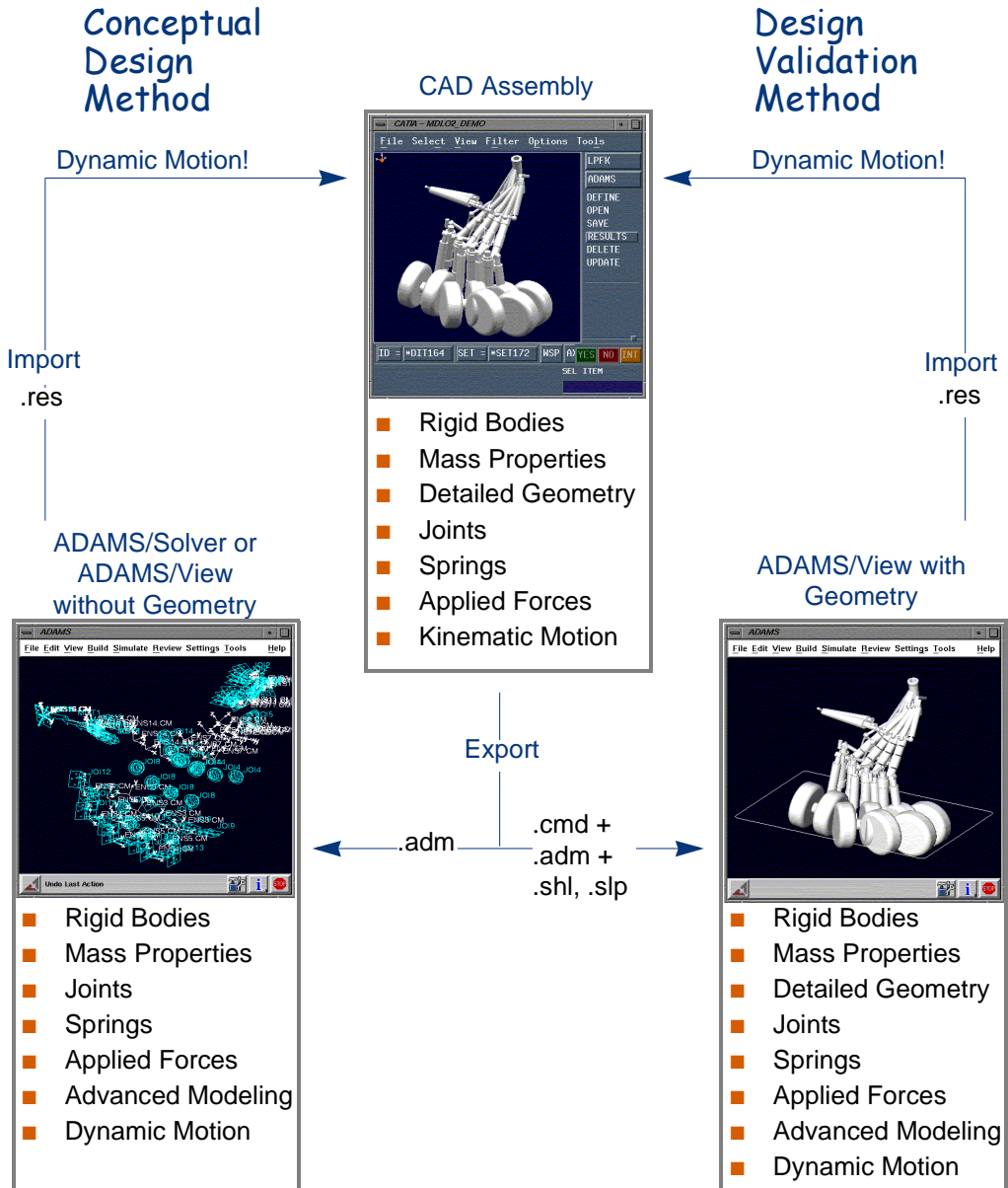
## Displacement functions

- For translational displacement, return scalar portions of vector components (measurements are taken to I from J, resolved in R's CS), as shown below.
- For rotational displacement, return angles associated with a particular rotation sequence.

## Example



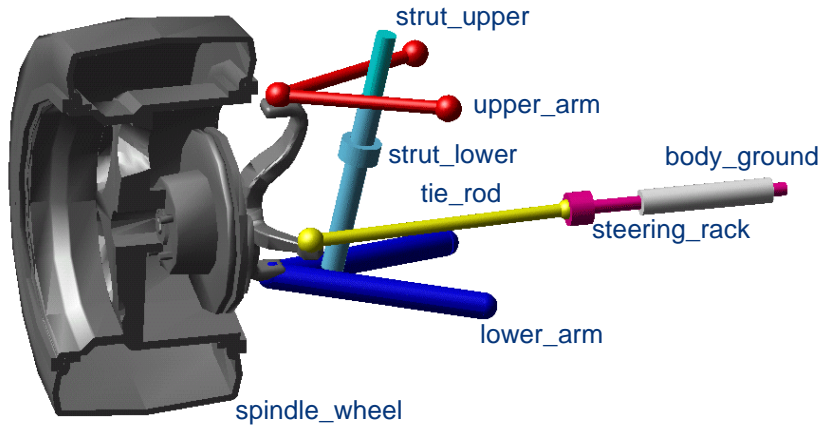
# Importing CAD-Based Geometry



# Workshop 12—Suspension System II

## Problem statement

Inspect the toe angle that the wheel exhibits throughout its vertical travel of 80 mm in jounce and rebound.



## Model description

In this workshop, you use the model you built in [Workshop 11—Suspension System I](#), on page 176.

# Workshop 12—Suspension System II...

## Start the workshop

Note that the file for this workshop is not in the current working directory.

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_12\_suspension\_2*.
- 2 From the directory *exercise\_dir/mod\_11\_suspension\_1*, import the model that you created in the previous workshop.

If you need a fresh copy of the model, import the command file *suspension\_1\_completed.cmd* from the directory *exercise\_dir/mod\_11\_suspension\_1/completed*.

- 3 Simulate the model to verify the motion.

## Create measures

To create measures:

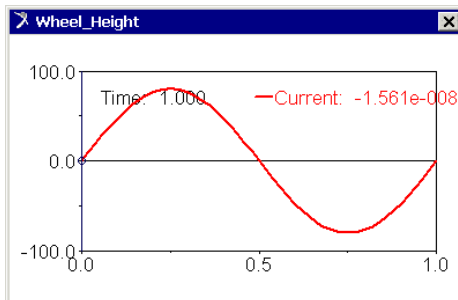
- 1 Create a point-to-point measure, named *.suspension.Wheel\_Height*, for the relative wheel displacement in the  $\hat{y}_G$  direction:

- To Point: *Spindle\_Wheel.Center*
- From Point: *ground.WH\_ref*

**Tip:** From the **Build** menu, point to **Measure**, point to **Point-to-Point**, and then select **New**.

- 2 Run a one-second, 50-step simulation.

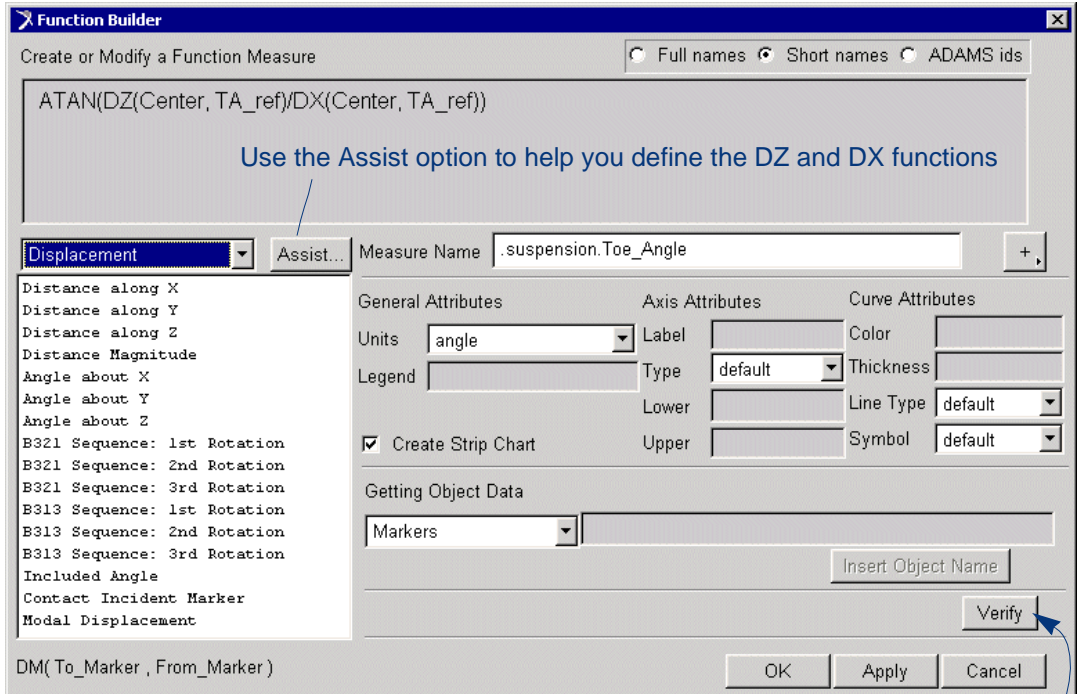
ADAMS/View displays the relative wheel displacement as shown next.



## Workshop 12—Suspension System II...

- Using an ADAMS/Solver function measure, create a toe angle measure using the markers **Spindle\_Wheel.Center** and **Spindle\_Wheel.TA\_ref** (see Figure 2 on page 190):
  - From the **Build** menu, point to **Measure**, point to **Function**, and then select **New**.
  - Fill in the Function Builder as shown next.

To get help while working in the Function Builder, press **F1**

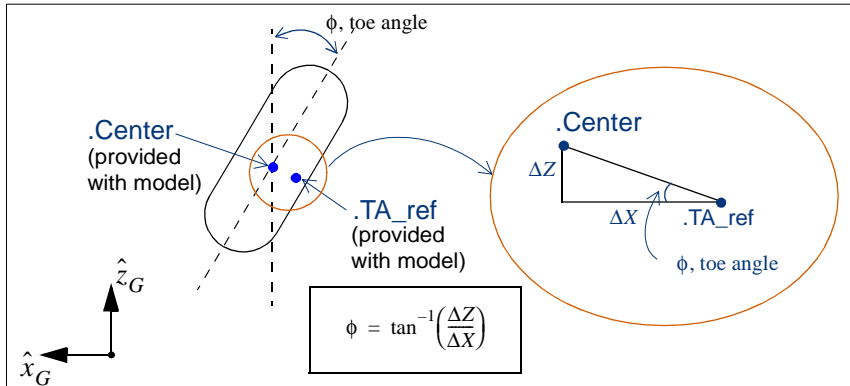


Before selecting **OK**, select **Verify** to check the syntax of your function

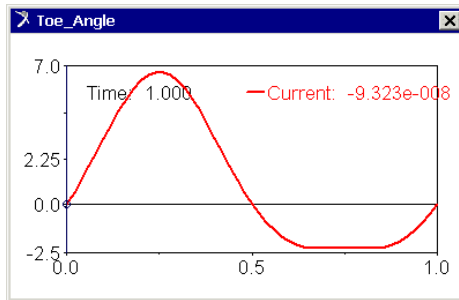
**Note:** You must run a simulation after creating the function to view its plot.

# Workshop 12—Suspension System II...

Figure 2. Markers for Toe Angle Measure



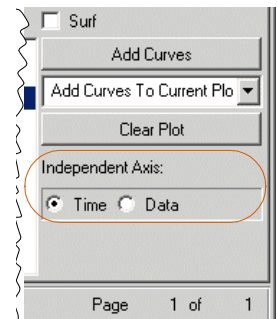
ADAMS/View updates the measure stripchart.



4 In ADAMS/PostProcessor, plot toe angle versus wheel height.

- Set Source to Measures.
- Set Independent Axis to Data.
- Select Wheel Height.
- Select OK.
- From the Measure list, select Toe\_Angle.
- Select Add Curves.

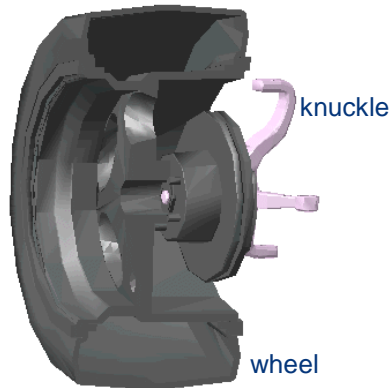
5 Return to ADAMS/View.



# Workshop 12—Suspension System II...

## Import CAD-based geometry

Now, you'll import more realistic, CAD-based spindle/wheel geometry, as shown next.



The two geometry files that make up the spindle/wheel are:

- wheel.slp
- knuckle.slp

They are render files, which have an extension of .slp. They were created in Pro/ENGINEER. By default, when you import the files, ADAMS/View names the geometry based on the Pro/ENGINEER assembly from which they came and not based on their file names. In this case, the CAD geometry came from a model named suspensn. Therefore, ADAMS/View names the geometry suspensn and suspensn\_2.

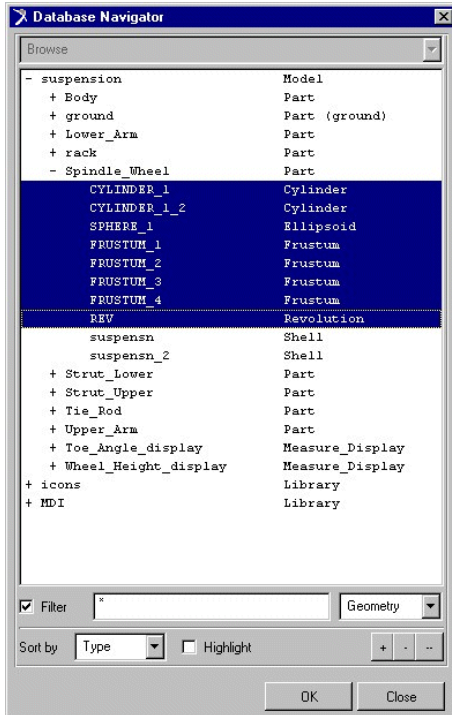
When you export your model, ADAMS/View exports one .cmd file (suspension.cmd) and one .shl file for *each* CAD geometry (suspensn.shl and suspensn\_2.shl).

### To import the geometry:

- 1 Import the geometry files located in *exercise\_dir/mod\_12\_suspension\_2/suspension\_cad*:
  - From the **File** menu, select **Import**.
  - Set **File Type** to **Render**.
  - Attach the geometry to the part named **Spindle\_Wheel**.
  - Import one file, select **Apply**, and then import the other.

# Workshop 12—Suspension System II...

- Turn off the appearance of ADAMS/View spindle geometry so that only the CAD geometry is visible:
  - From the **Edit** menu, select **Appearance**.
  - Highlight the following:



- Select **OK**.
- Change the **Visibility** setting to **Off**.
- Select **OK**.

# Workshop 12—Suspension System II...

## Save your work

To save your work:

- 1 Save your model as `suspension_parts.cmd`.

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.

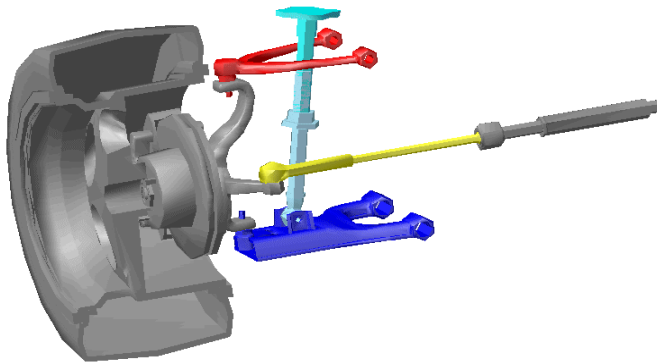
## Optional tasks

Replace other ADAMS/View geometry with CAD geometry:

- 1 From the directory `exercise_dir/mod_12_suspension/suspension_cad/more_susp_cad`, import the rest of the CAD-based suspension component geometry.

**Tips:** Remember to associate each CAD geometry with the appropriate part in the ADAMS/View model. Not all the CAD geometry is associated with the Spindle\_Wheel part.

These geometry files are called render files, which have an extension of `.slp`. There is one file for each ADAMS/View part.



- 2 Turn off the appearance of ADAMS/View geometry so that only the CAD geometry is visible.

# Workshop 12—Suspension System II...

---

## Module review

- 1 Is there any difference between a point-to-point measure and a function measure using a displacement function (for example,  $DX(I, J, R)$ )?

---

---

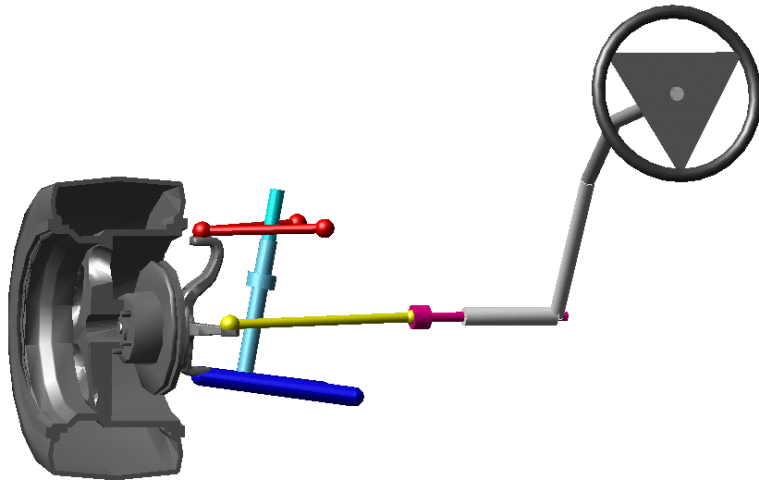
- 2 Where does a CAD file fall in the model hierarchy? What is the CAD file a child of?

---

---

# 13 SUSPENSION-STEERING SYSTEM

Assemble a suspension-steering system and inspect the toe angle that the wheel exhibits at steering wheel angles of  $45^\circ$ ,  $0^\circ$ , and  $-45^\circ$ .



## What's in this module:

- Add-On Constraints, 196
- Couplers, 197
- Assembling Subsystem Models, 198
- Workshop 13—Suspension-Steering System, 199
  - ◆ Module review, 206

# Add-On Constraints

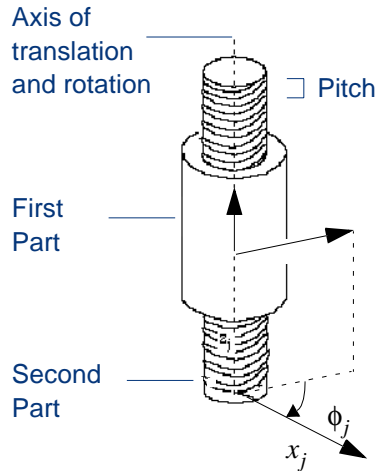
## Add-on (complex) constraints

- Set up relationships between existing constraints in a system.
- Connect parts directly and indirectly.

## Types of add-on constraints

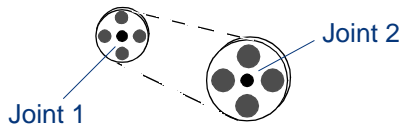
---

### Screw Joints



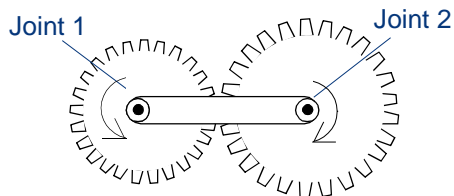
---

### Couplers



---

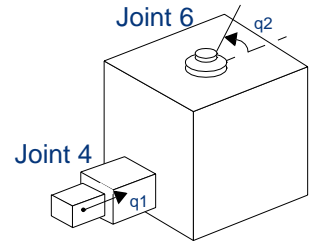
### Gears



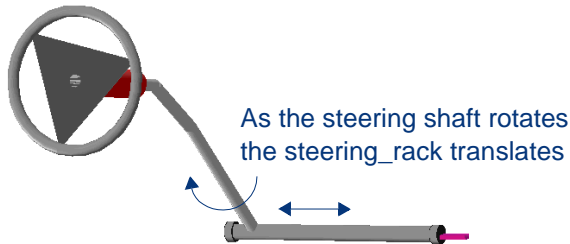
# Couplers

## Definition of couplers

- Couplers connect multiple parts indirectly by coupling 2 joints.
- Couplers remove 1 DOF.
- Couplers can be defined:
  - ◆ By displacements
  - ◆ By scales
  - ◆ User defined
- Modeling of couplers requires two joints (applicable types are translation, revolute, and cylindrical)



## Example of a coupler

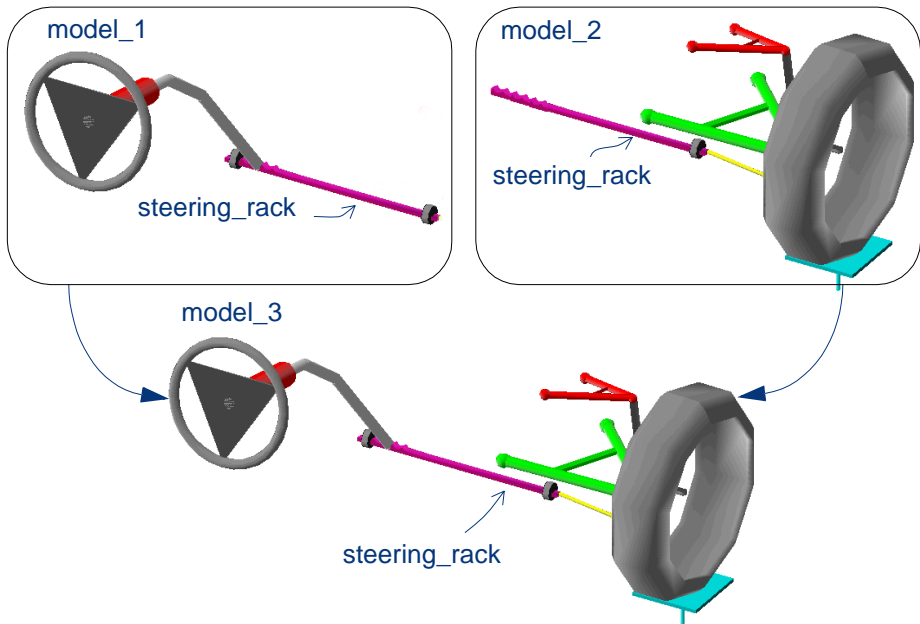


For help on defining By Displacement and User Defined, press F1.

# Assembling Subsystem Models

## When you assemble models

- Any number of models can be assembled.
- Assembling models will create a new model.
- All assembled models (model1, model2) will continue to exist in the database along with the new model (model3).



## Parts in assembled models

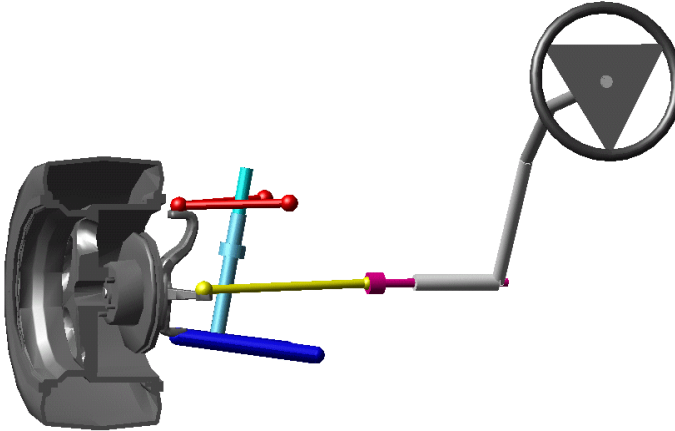
- They maintain their global location and orientation, unless otherwise specified.
- If parts have the same name in different merged models, ADAMS/View will either:
  - ◆ Merge them into one part.
  - ◆ Rename the parts.

**See also:** [Model Hierarchy](#), on page 28

# Workshop 13—Suspension-Steering System

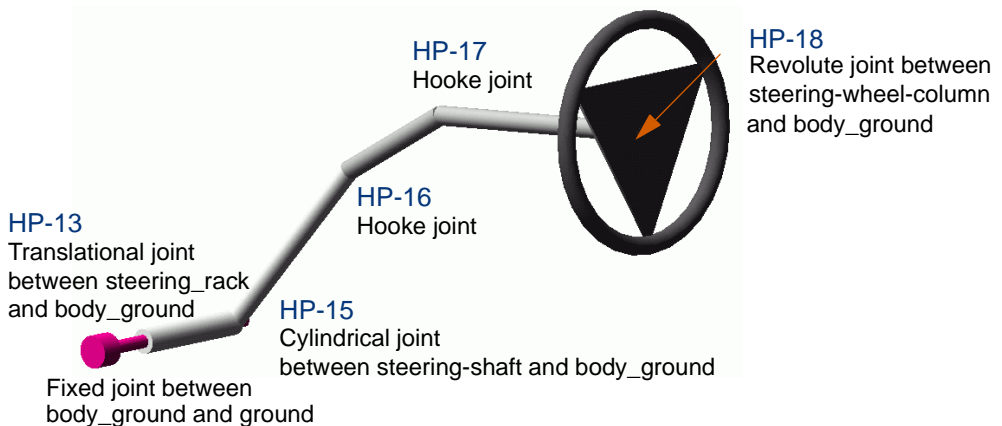
## Problem statement

Assemble a suspension-steering system and inspect the toe angle that the wheel exhibits at steering wheel angles of  $45^\circ$ ,  $0^\circ$ , and  $-45^\circ$ .



## Model description

- You will use the following two models in this workshop:
  - ◆ A geometric representation of a short-long arm (SLA) suspension subsystem.
  - ◆ A geometric representation of a rack-and-pinion steering system.
- The rack-and-pinion steering model is constrained as shown next:



# Workshop 13—Suspension-Steering System...

---

## Start the workshop

Note that the file for this workshop is not in the current working directory.

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_12\_suspension\_2*.
- 2 From this directory, import the model that you created in the previous module.

If you need a fresh copy of the model, change your working directory to *exercise\_dir/mod\_12\_suspension\_2/completed*, then import the command file *suspension\_2\_completed.cmd*.

## Change working directory

Change the directory to *exercise\_dir/mod\_13\_susp\_steer*. Running ADAMS/View in this directory ensures that all saved data gets stored there.

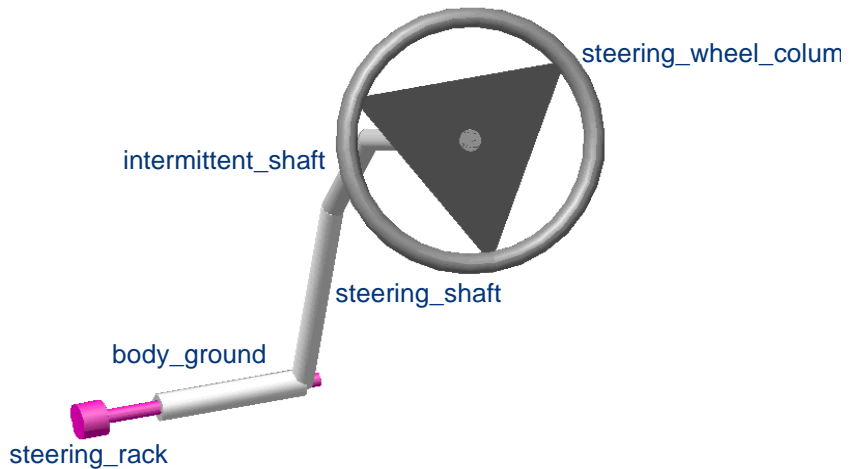
To change the working directory:

- 1 From the **File** menu, select **Select Directory**.
- 2 Change to *exercise\_dir/mod\_13\_susp\_steer*.

# Workshop 13—Suspension-Steering System...

## Import the steering model

Import the steering model, shown below. It is in the command file `steering_parts_start.cmd`. The file contains a model named `rack_and_pinion_steering`.



To import the model:


- Import the ADAMS/View model command file `steering_parts_start.cmd`.

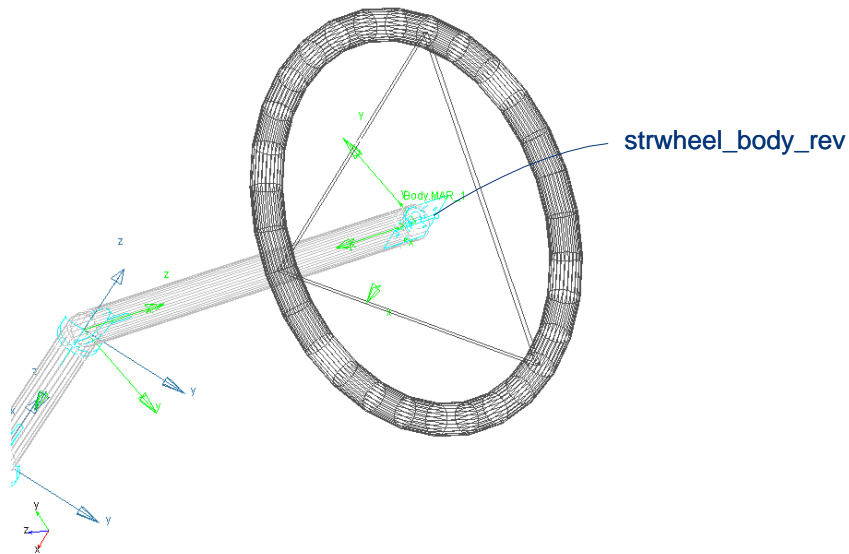
# Workshop 13—Suspension-Steering System...


## Constrain the steering model

Now you'll constrain the steering model. Each time you add a modeling element, you'll simulate the model to verify its movement.

To constrain the steering model:

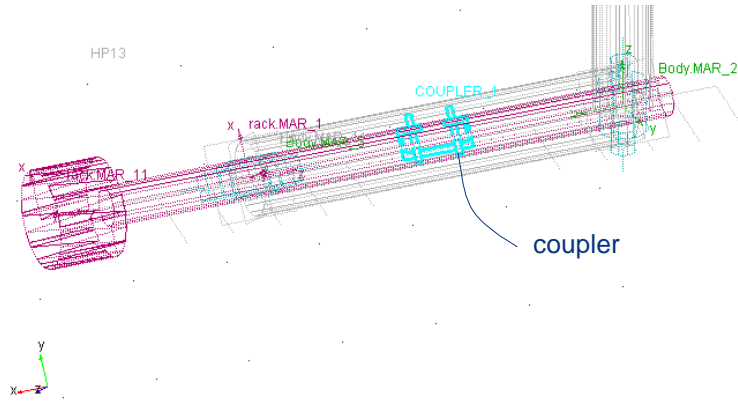
- 1 Apply a rotational joint motion (  ) to the revolute joint, **strwheel\_body\_rev**, on the **steering\_wheel\_column**.



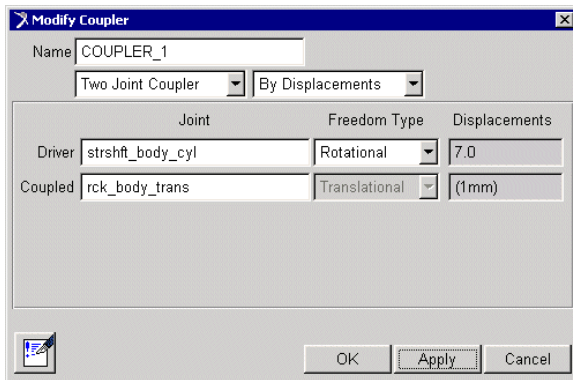
- 2 Right-click the motion icon, point to **Motion: MOTION\_1**, and then select **Modify**.
- 3 In the **Function (time)** text box, enter the following function:  $45d*\sin(360d*time)$  (be sure that **Type** is set to **Displacement**).
- 4 Run a one-second, 50-step simulation.
- 5 Use the **Coupler** tool (  ) to couple the rotation (**strshft\_body\_cyl**) of the **steering\_shaft** with the translation (**rck\_body\_trans**) of the **steering\_rack**.

# Workshop 13—Suspension-Steering System...

- 6 Right-click the coupler icon, point to **Coupler: COUPLER\_1**, and then select **Modify**.



- 7 Modify the coupler so that for every  $7^\circ$  of rotation the **steering\_rack** travels 1 mm:



- 8 To verify that the **steering\_rack** travels as expected, simulate the model.

## Assemble the suspension and steering models

To assemble the models:

- 1 Assemble the `rack_and_pinion_steering` model with the `suspension` model:
  - From the **Tools** menu, select **Command Navigator**.
  - In the Command Navigator, double-click `model`, and then double-click `assemble`.
  - Name the model `steering_suspension`.
  - Right-click the **Model Names** text box, point to **Guesses**, and then select `*`. Selecting `*` prompts ADAMS/View to include in the text box every item that is displayed under the **Guesses** option.
- 2 To display the assembled model, from the **View** menu, select **Model**.
- 3 Double-click `steering_suspension`.

## Verify the model

Verify the model to ensure it is not overconstrained. To ensure that the model is not overconstrained, look for duplicate joints as explained next.

To verify the model:

- 1 From the **Tools** menu, select **Database Navigator**.
- 2 Set the top pull-down menu in the Database Navigator to **Graphical Topology**.
- 3 Double-click the new model name, `steering_suspension`, and then select `body_ground`.
- 4 Delete the unwanted joints by right-clicking them and selecting **Delete** (make sure that **Highlight** is not selected):
  - Delete one of the fixed joints between `body_ground` and `ground`.
  - Delete the fixed joint between `body_ground` and `steering_rack`.

**Note:** Leave both the rotational motion and the revolute joint between `body_ground` and `steering_wheel_column` because both are needed.

# Workshop 13—Suspension-Steering System...

- 5 To redisplay the measures you created in [Create measures](#), on page 188, from the **Build** menu, point to **Measure**, and then select **Display**.
- 6 Select both measures, **Wheel\_Height** and **Toe\_Angle**.

## Run and compare a series of simulations

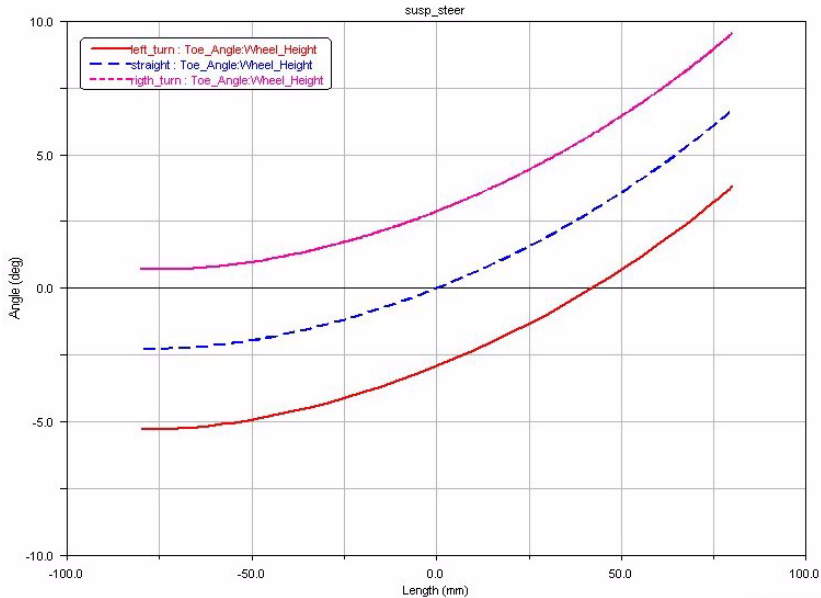
In this section, you'll run three simulations, each with different steering wheel angles. You'll then compare the results of the simulations.

To run and compare a series of simulations:

- 1 Modify the motion you created in Step 3 on page 202 to be a constant  $45^{\circ}$  steering wheel angle (Displacement(time) = 45d).
- 2 Simulate the model.
- 3 Save the simulation results as **right\_turn**. Save the results just as you did in [To save the simulation results](#); on page 40 of [Workshop 2—ADAMS/View Interface Overview](#).
- 4 Run a simulation with a  $0^{\circ}$  steering wheel angle (Displacement(time) = 0d).
- 5 Save the simulation results as **straight**.
- 6 Run a simulation with a  $-45^{\circ}$  steering wheel angle (Displacement(time) = -45d).
- 7 Save the simulation results as **left\_turn**.
- 8 Launch ADAMS/PostProcessor.

# Workshop 13—Suspension-Steering System...

- 9 Plot `toe_angle` versus `wheel height` for all three simulations, on the same plot, just as you did on page 190.



## Save your work

To save your work:

- 1 Save the database so you retain the analyses.
- 2 Exit ADAMS/View.

## Module review

- 1 What information do you need to provide ADAMS/View to create a coupler?

---

---

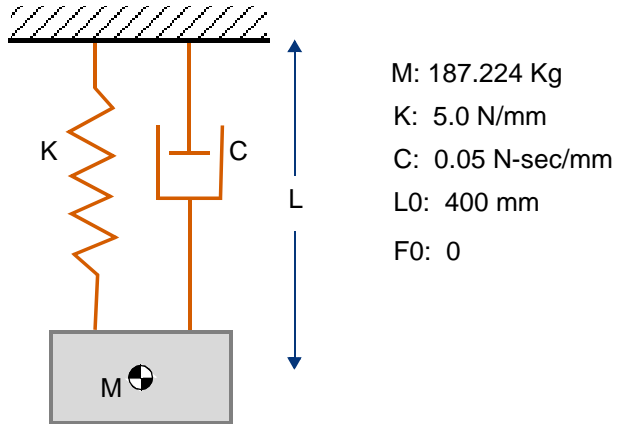
- 2 What is the default name that ADAMS/View assigns to simulation results?

---

---

# 14 SPRING DAMPER

Create and investigate the linear spring-damper system shown in the following figure, using different types of simulations in ADAMS/View.



## What's in this module:

- Assemble Simulation, 208
- Simulation Hierarchy, 209
- Types of Simulations, 210
- Forces in MSC.ADAMS, 212
- Spring Dampers in MSC.ADAMS, 213
- Workshop 14—Spring Damper, 215
  - ◆ Module review, 219

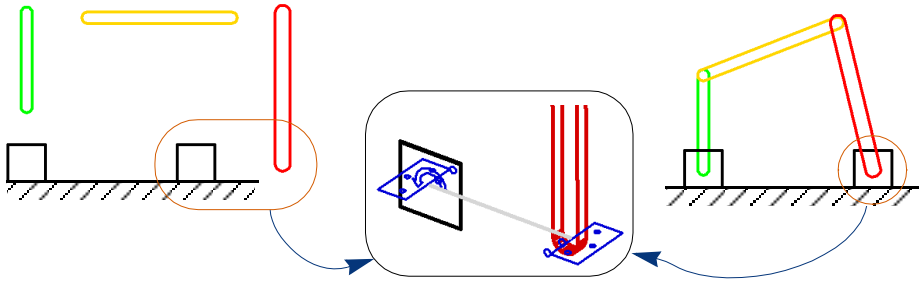
# Assemble Simulation

## Definition of assemble simulation

- Attempts to resolve any conflicts in the initial conditions specified for the entities in the model (for example, broken joints).
- Is also known as an **initial conditions simulation**.

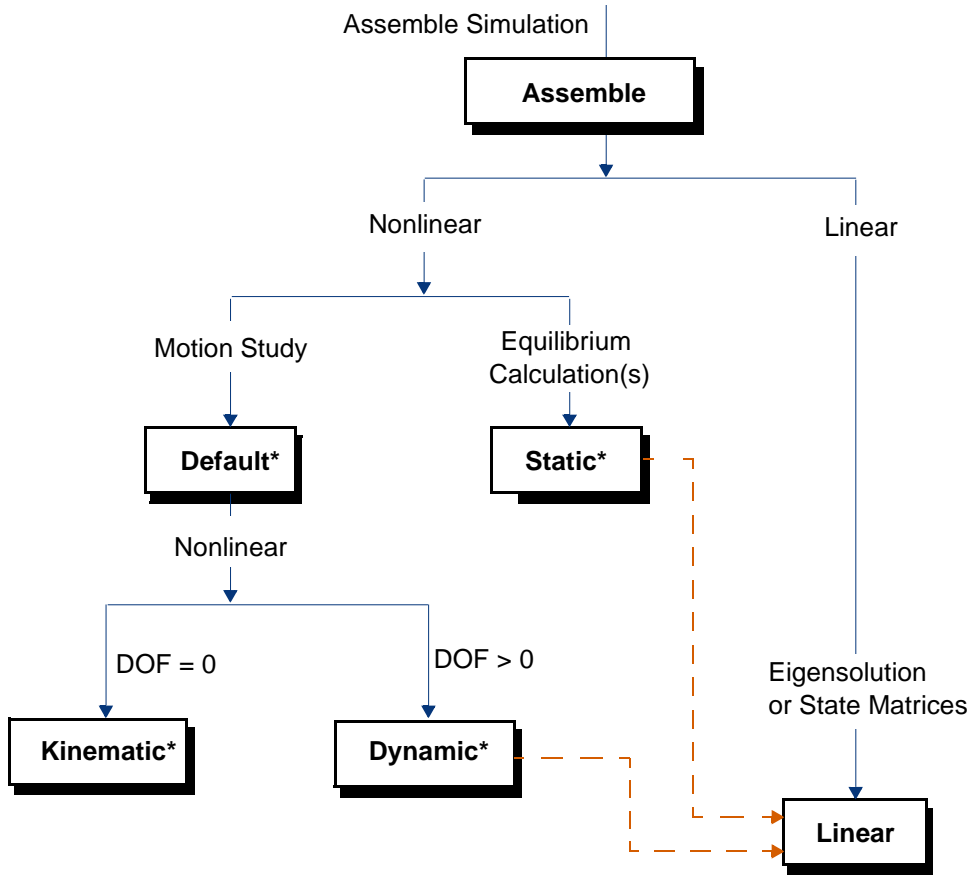
## Initial location and orientation of parts

- You specify the initial position and orientation for a part when you create it.
- For a part to be held fixed during the assemble simulation, you can specify up to three positions ( $\hat{x}_G, \hat{y}_G, \hat{z}_G$ ) and up to three orientations (psi, theta, phi).



**Note:** Use initial positions sparingly. If you fix the initial positions of too many parts, the assemble simulation can fail.

# Simulation Hierarchy



\* Automatically performs an assemble simulation

**Note:** Often a linear simulation is used after a static equilibrium or dynamic simulation.

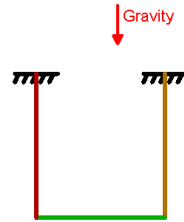


While working in any ADAMS/View dialog box, press **F1** to display online help specific to that dialog box.

# Types of Simulations

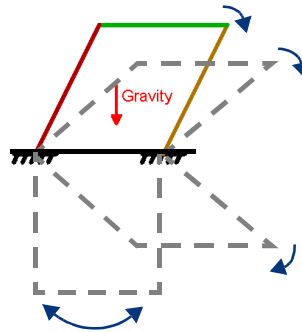
## Static

- System  $\text{DOF} > 0$ .
- All system velocities and accelerations are set to zero.
- Can fail if the static solution is a long way from the initial condition.



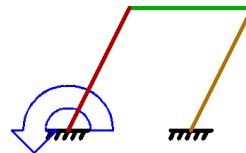
## Dynamic

- System  $\text{DOF} > 0$ .
- Driven by a set of external forces and excitations.
- Nonlinear differential and algebraic equations (DAEs) are solved.



## Kinematic

- System  $\text{DOF} = 0$ .
- Driven by constraints (motions).
- Only constraint (algebraic) equations are being solved.
- Calculate (measure) reaction forces in constraints.

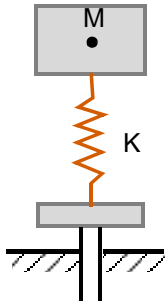


# Types of Simulations...

## Linear

- ADAMS/Solver can linearize the system of nonlinear equations of motion about a particular operating point.
- From the linear set of equations, you can ask for an eigen-simulation to obtain eigenvalues and eigenvectors for the linearized system to:
  - ◆ Visualize the natural frequencies and mode shapes of your system.
  - ◆ Compare with test data or results data from FEA.

## Example of linear simulation



- Must linearize about an operating point (often the equilibrium).
- Extraction of natural frequency.
- Natural frequency =  $\sqrt{\frac{K}{M}}$ .

## Definition of forces

- Try to make parts move in certain ways.
- Do not perfectly connect parts together the way constraints do.
- Do not absolutely prescribe movement the way motion drivers do.
- Neither add nor remove DOF from a system.

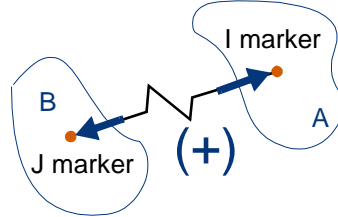
## Characteristics of forces

<b>The characteristic:</b>	<b>Defines:</b>
Bodies	Which parts are affected
Points of application	Where the parts are affected
Vector components	How many vector components there are
Orientation	How the force is oriented
Magnitude	If the force is pre-defined or user-defined

# Spring Dampers in MSC.ADAMS

## Definition of spring dampers

- They are pre-defined forces.
- They represent compliance:
  - ◆ Between two bodies.
  - ◆ Acting over a distance.
  - ◆ Along or about one particular direction.



## Characteristics of spring dampers

The characteristic:	Defines:
Bodies	Two (A, B)
Points of application	Two (I and J marker)
Vector components	One
Orientation (only for translational)	Acts along the line of sight between the I and J markers: <ul style="list-style-type: none"><li>■ <b>Positive</b> force <b>repels</b> the two parts</li><li>■ <b>Negative</b> force <b>attracts</b> the two parts</li></ul>
Magnitude	Pre-defined equation based on either: <ul style="list-style-type: none"><li>■ Stiffness and damping coefficients (linear)</li><li>■ Splines based on test data (nonlinear)</li></ul>

**See also:** Characteristics of a spring damper, on page 378

# Magnitude of Spring Dampers

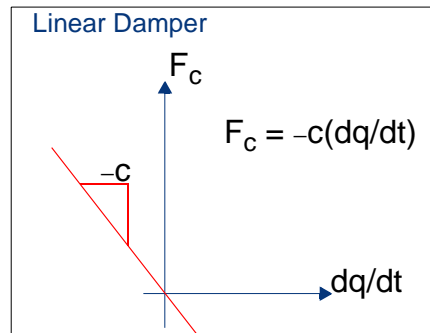
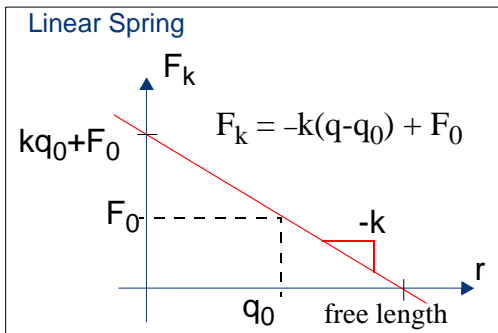
## Magnitude based on stiffness and damping coefficients

- Linear spring-damping relationship can be written as:

$$\text{Force}_{\text{SPDP}} = -k(q - q_0) - c\dot{q} + F_0$$

where:

- $q$  - Distance between the two locations that define the spring damper
  - $\dot{q}$  - Relative speed of the locations along the line-of-sight between them
  - $k$  - Spring stiffness coefficient (always  $> 0$ )
  - $c$  - Viscous damping coefficient (always  $> 0$ )
  - $F_0$  - Reference force of the spring (preload)
  - $q_0$  - Reference length (at preload, always  $> 0$ )
  - $t$  - Time
- In ADAMS/Solver, the user-defined equation is:  
 $-k*(DM(I, J) - q_0) - c*VR(I, J) + F_0$

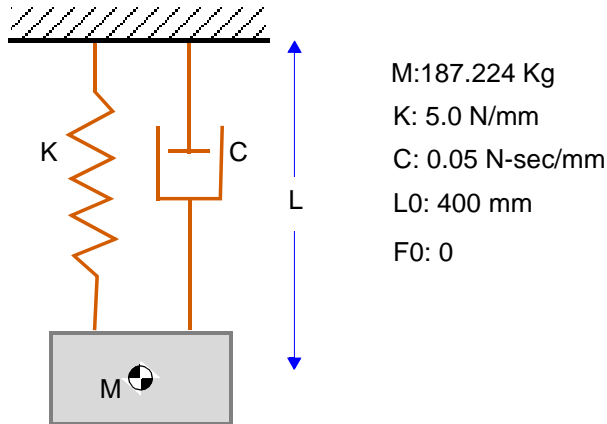


- Spring-damper forces become ill-defined if endpoints become coincident because of undefined direction.

# Workshop 14—Spring Damper

## Problem statement

Create and investigate the linear spring-damper system shown next, using different types of simulations in MSC.ADAMS.



## Start the workshop

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_14\_spring\_damper*.
- 2 Create a new model named **spring\_mass**.

## Build and constrain the model


To build and constrain the model:

- 1 Build the block with the given mass.  
**Tip:** Right-click the part and select **Modify**. Set **Define Mass By** to **User Input**.
- 2 Constrain the block to move only in the  $y_G$  direction.  
**Tip:** Add a translational joint.
- 3 To verify the expected behavior, simulate the model.

# Workshop 14—Spring Damper...


## Add the pre-defined spring damper

To add a pre-defined spring damper:

- 1 To ensure that the spring damper is aligned with the  $\hat{y}_G$  direction, move the working grid to the cm of the block (see [Build all parts except for the bucket](#), on page 149).
- 2 Use the **Translational spring damper** tool , to create the spring damper, aligned along the  $\hat{y}_G$  direction, between the CM marker of the block and a point on ground 400 mm above it (make sure to use the K and C values from [Problem statement](#), on page 215). To set the second location, use the Location Event (see [Build the pendulum link](#), on page 102):
  - Right-click anywhere on the ground to display the Location Event.
  - Change from **Rel. to Origin** to **Rel. to Grid**.
  - Enter **0, 400, 0**.
  - Select **Apply**.
- 3 To ensure that the spring damper has a free length ( $q_0$ ) of 400 mm with a preload of 0, from the **Tools** menu, select **Measure Distance** to measure the spring.

## Find the force in spring damper at static equilibrium

To find the force at static equilibrium:

- 1 Use the **Static Equilibrium** tool  to run a static equilibrium simulation.
- 2 Note the value of the force graphic.

**Tip:** To display the force value, from the **Settings** menu, select **Force Graphics**. Select **Display Numeric Values**.
- 3 Zoom out until you can see the force value.

The block's mass is 187.224 kg. Therefore, to balance the force of gravity, the spring damper must generate:

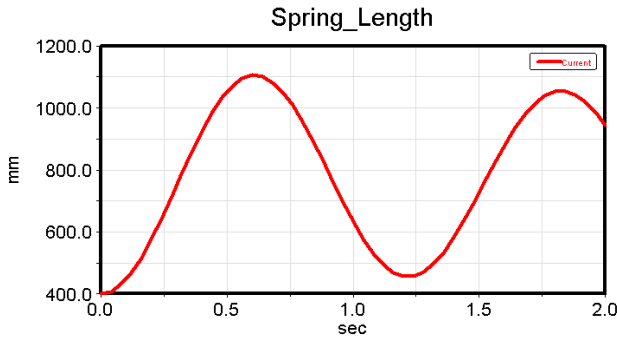
$$187.224\text{kg} \cdot 9.80665\text{m/s}^2 (=1836.04\text{ N})$$

# Workshop 14—Spring Damper...

## Run a simulation and create a measure

To create a measure and run a simulation:

- 1 Create a point-to-point measure, named **Spring\_Length**, to measure the spring's length. Measure to the upper spring-damper attachment point from the block's cm marker.





The measured value at  $t=0$  should be 400 mm.

- 2 To see the oscillation, run a 2-second, 50-step dynamic simulation.

## Find the natural frequency

To find the natural frequency:

- 1 Run another static equilibrium simulation ()
- 2 Select **Close** but do not reset the model.
- 3 From the **Simulate** menu, select **Interactive Controls**.
- 4 Select the **Compute Linear Modes** tool ()
- 5 View the results.
- 6 Note the natural frequency, and compare this value with that given in [Closed-form solution](#), on page 219.

# Workshop 14—Spring Damper...

---

## 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

Add a DOF to the model:

- 1 Modify the translational joint to be a cylindrical joint.

- 2 Linearize about the static equilibrium position.

Do the resulting modes make sense?

- 3 Add a torsional spring damper that resists the rotation of the cylindrical joint.

- 4 Linearize about the static equilibrium position.

Are the results different from those above (no torsional spring damper)?

- 5 Do not save your work.

## MSC.ADAMS results

$$\omega_n = 0.8222 \text{ Hz}$$

$$\omega_n = (0.8222 \text{ Hz})(2 \cdot \pi \text{ rad}) = 5.168 \text{ rad/sec}$$

# Workshop 14—Spring Damper...

## Module review

- 1 At design configuration, do the  $\hat{z}$  directions of markers referenced in a revolute joint have to be aligned? Does this information get reported when verifying a model?

---

---

## Closed-form solution

Checking the natural frequency of the system:

At equilibrium:

$$m\ddot{x} + c\dot{x} + kx = 0$$

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = 0$$

Laplace Transform is:

$$s^2 + \frac{c}{m}s + \frac{k}{m} = 0 \Leftrightarrow s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

Therefore:

$$\omega_n^2 = \frac{k}{m}$$

$$\omega_n = \sqrt{\frac{k}{m}}$$

$$k = 5 \text{ N/mm} = 5000 \text{ N/m}$$

$$m = 187.224 \text{ kg}$$

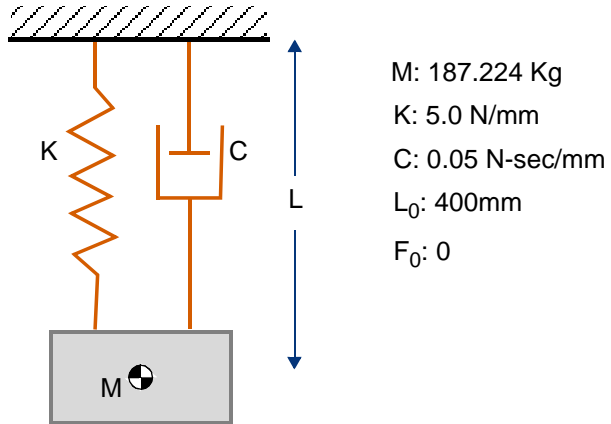
$$\omega_n = \sqrt{\frac{5000}{187.224}} \text{ rad/sec}$$

$$\omega_n = 5.168 \text{ rad/sec}$$



# 15 NONLINEAR SPRING

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

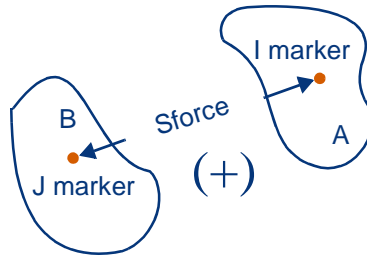


## What's in this module:

- Single-Component Forces: Action-Reaction, 222
- Spline Functions, 223
- AKISPL Function, 224
- Workshop 15—Nonlinear Spring, 225
  - ◆ Module review, 230

# Single-Component Forces: Action-Reaction

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) <ul style="list-style-type: none"><li>◆ <b>Positive</b> force <b>repels</b> the two parts</li><li>◆ <b>Negative</b> force <b>attracts</b> the two parts</li></ul>
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.

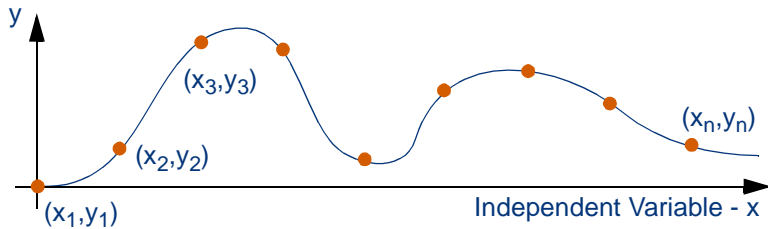
# Spline Functions

## 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

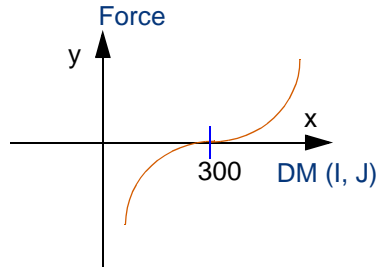
AKISPL (x, z, spline, iord)

- **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

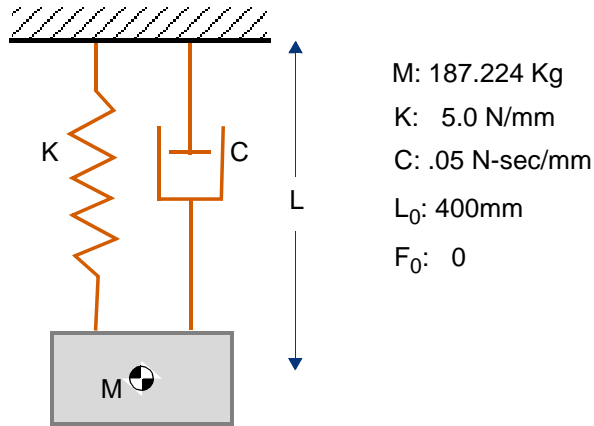


**Note:** You can create the CUBSPL and CURVE functions exactly as you create the AKISPL function.

# Workshop 15—Nonlinear Spring

## Problem statement

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



## 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:

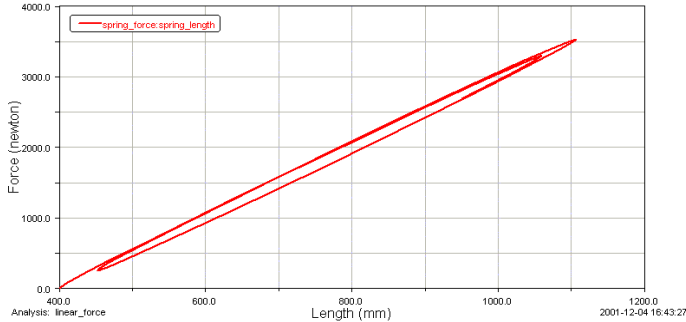
- 1 Delete the predefined spring damper.
- 2 Create a single-component, action-reaction force ()
  - Run-time Direction: Two Bodies
  - Characteristic: K and C
  - K: 5.0
  - C: 0.05
- 3 Right-click **SFORCE\_1**, and then select **Info**. Note that the syntax of the function matches that of a spring 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.

# Workshop 15—Nonlinear Spring...

## 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  $\sim$  stiffness  $\pm$  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:

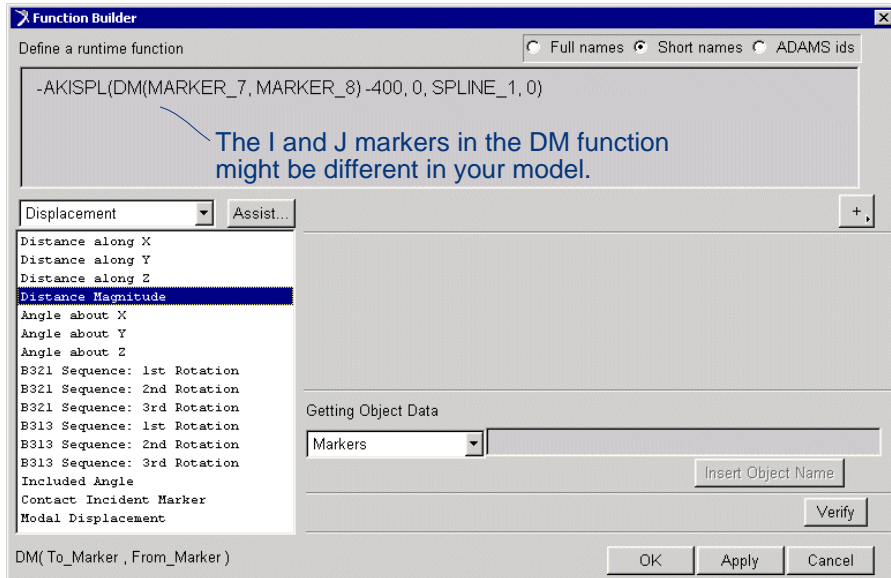
- 1 To import the spring stiffness data, from the **File** menu, select **Import**.
- 2 Set the following parameters, and then select **OK**.
  - **File Type:** Test Data
  - **Create Splines**
  - **File to Read:** `exercise_dir/mod_15_spring/spring_data.txt`
  - **Independent Column Index:** 1 (Because 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**.

# Workshop 15—Nonlinear Spring...

- 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.



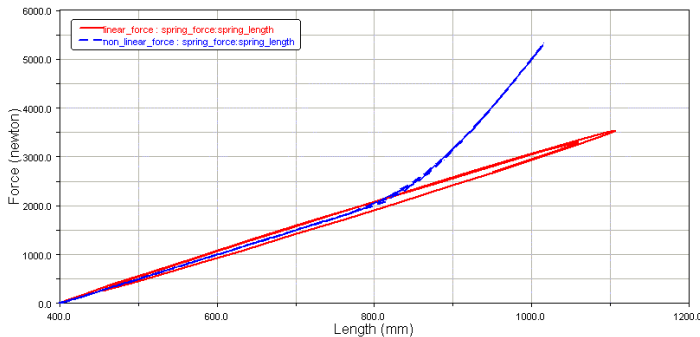
# Workshop 15—Nonlinear Spring...

## 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.

# Workshop 15—Nonlinear Spring...

---

## 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?

---

---

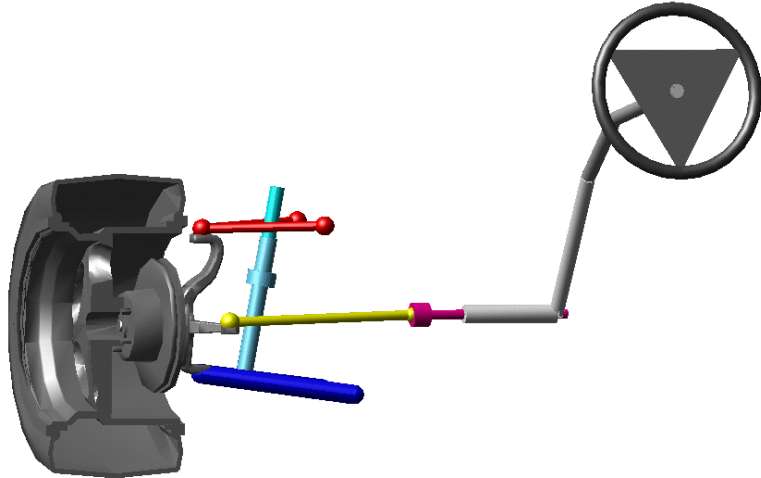
- 2 If you are not sure what inputs are required for an ADAMS/Solver function, where in the online help would you look for assistance?

---

---

# 16 SUSPENSION-STEERING SYSTEM II

Investigate the effect on toe angle when you replace the idealized constraint between the lower control arm and ground with bushings, while the steering wheel is held at an angle of  $0^\circ$ .



## What's in this module:

- Bushings, 232
- Workshop 16—Suspension-Steering System II, 233
  - ◆ Module review, 238

## Definition of a bushing

- Pre-defined force.
- Represents compliance:
  - ◆ Between two bodies.
  - ◆ Along or about three vectors.

## Characteristics of a bushing

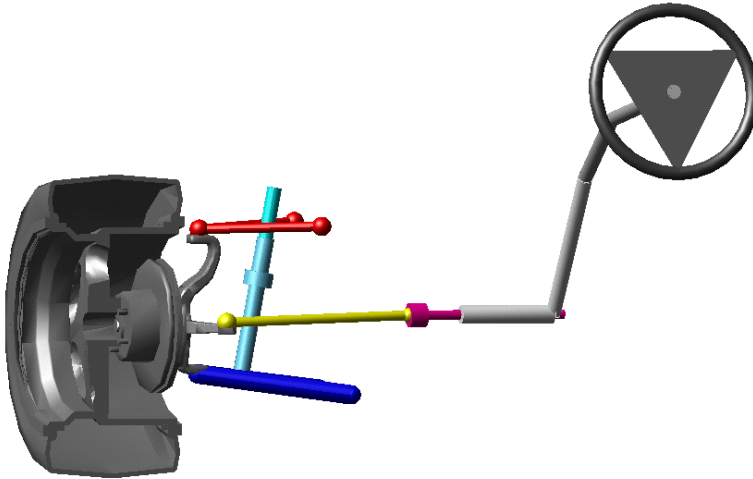
Characteristic:	Description:
Bodies	Two
Points of application	Two (action force at I marker and reaction force at J marker)
Vector components	Three translational and three rotational
Orientations	Based on the J marker
Magnitudes ( $F_x, F_y, F_z + T_x, T_y, T_z$ )	Pre-defined equation based on: <ul style="list-style-type: none"><li>■ Stiffness matrix, <math>[K]</math></li><li>■ Damping matrix, <math>[C]</math></li></ul>

**See also:** Forces Tables (Incomplete), on page 375

# Workshop 16—Suspension-Steering System II

## Problem statement

Investigate the effect on toe angle when you replace the idealized constraint between the lower control arm and ground with bushings, while the steering wheel is held at an angle of  $0^\circ$ .



## Model description

- The model is the short-long-arm front suspension model combined with a steering model that you created in the previous workshop.
- A spring damper has been added to represent the force input of a coil-over shock.
- Currently, a revolute joint connects the lower control arm to the frame of the vehicle.
- You are going to replace the revolute joint with two bushings and investigate the differences.

# Workshop 16—Suspension-Steering System II...

## Start the workshop


To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_16\_susp\_steer\_2*.
- 2 Import the command file *susp\_steer\_2\_start.cmd*.

## Run a baseline simulation

You'll start by running a simulation with the model as it currently is to see how it performs with a revolute joint.

To run a baseline simulation:

- 1 Verify that the steering wheel angle is a constant  $0^\circ$  (Displacement(time) = 0d).
- 2 Run a simulation for **1 second** with **50** output steps.
- 3 Save the simulation results () as **with\_joint**.

## Deactivate the revolute joint

Now, instead of removing the revolute joint, you'll just deactivate it so it is not used in simulations.

To deactivate the revolute joint:


- 1 Right-click the *lowerarm\_grnd\_rev* revolute joint that currently exists between **Lower\_Arm** and ground.
- 2 Select **(De)activate**.
- 3 Clear the selection of **Object Active**.

# Workshop 16—Suspension-Steering System II...

## Create bushings between Lower\_Arm and ground

You will need to create two bushings because there are two connection points between Lower\_Arm and ground.

To create bushings:

- 1 From the Main Toolbox, point to the **Create Forces** tool stack, and select the **Bushing**  tool.
- 2 Create the rear bushing with the following properties, using the options **2 Bod-1 Loc**, **Pick Feature**:
  - First Part: Lower\_Arm
  - Second Part: ground
  - Location: HP4
  - Direction Vector (+z axis): Z-direction of marker .Lower\_Arm.bushing\_ref.

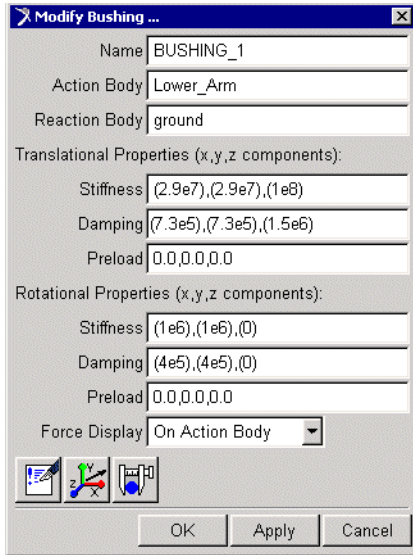
**Tip:** To easily find the marker .Lower\_Arm.bushing\_ref, display the Database Navigator, locate the marker under **Lower\_Arm**, and then select **Highlight** from the bottom of the Database Navigator. Back in your model, right-click at the marker location, and then select .Lower\_Arm.bushing\_ref.Z.

- 3 Modify the bushing to reflect the following properties:

$K_{matrix}$	$C_{matrix}$	Preload <sub>matrix</sub>
$\begin{bmatrix} 2.9e7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.9e7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 7.3e5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7.3e5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5e6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4e5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4e5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
Translational K	Translational C	
Rotational K	Rotational C	

# Workshop 16—Suspension-Steering System II...


The modify dialog box should look as follows:



- 4 Create the forward bushing with the following properties:
  - First Part: Lower\_arm
  - Second Part: ground
  - Location: HP5
  - Direction Vector (+z axis): Z-direction of marker .Lower\_Arm.bushing\_ref
- 5 Modify the bushing to reflect the properties given in Step 3 on page 235.

## Run a simulation to view the effect of adding the bushing

To run a simulation:

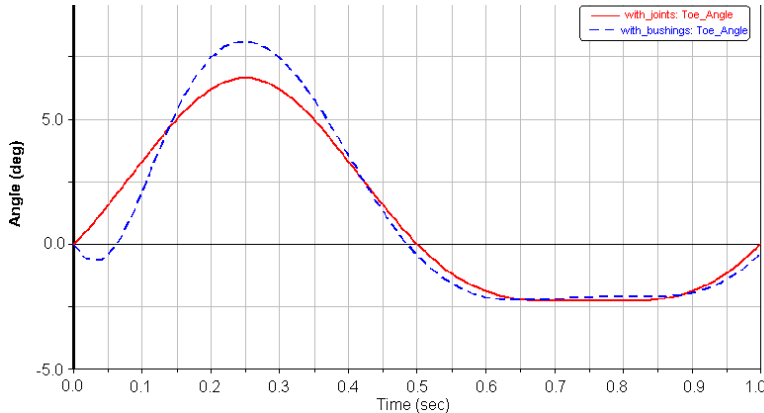
- 1 Run a static simulation () followed by a dynamic simulation for **1 second** with **50** output steps.
- 2 Save the simulation results as **with\_bushings**.

# Workshop 16—Suspension-Steering System II...

## Review the results

To review the results:

- 1 Launch ADAMS/PostProcessor.
- 2 Create a plot that contains the Toe\_Angle measure using the simulation results **with\_joint** and **with\_bushings** as a function of time.



- 3 Estimate the difference in the maximum toe angle between the two simulations and use it to answer Question 1 in [Module review](#), on page 238.

## Overlay animations

To overlay animations:

- 1 Set ADAMS/PostProcessor to **Animation** mode.
- 2 From the dashboard, select the **Overlay** tab.
- 3 Select both **with\_joint** and **with\_bushings** analyses.
- 4 In the **Offset** text box, enter **0.0, 40.0, 0.0**.
- 5 Select the **Animation** tab.
- 6 Set the **Speed Control** slider to approximately **50%**.
- 7 Select **Play**.

**Note:** To emphasize the difference, zoom in on the lower arm.

# Workshop 16—Suspension-Steering System II...

## Save your work

To save your work:

- 1 Save your 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 Replace the revolute joint between **Upper\_Arm** and ground with two bushings. Use the same bushing properties given for the bushing between **Lower\_Arm** and ground.
- 2 Run a static simulation followed by a dynamic simulation for **1 second** with **50** output steps.
- 3 Save the simulation results as **with\_all\_bushings**.
- 4 Compare these **toe\_angle** results with those from the previous two simulations.

## Module review

- 1 What was the approximate difference in the maximum toe angle that was a result of removing the revolute joint and replacing it with bushings?

---

---

- 2 Why did you perform a static simulation before the dynamic simulation after you added the bushings?

---

---

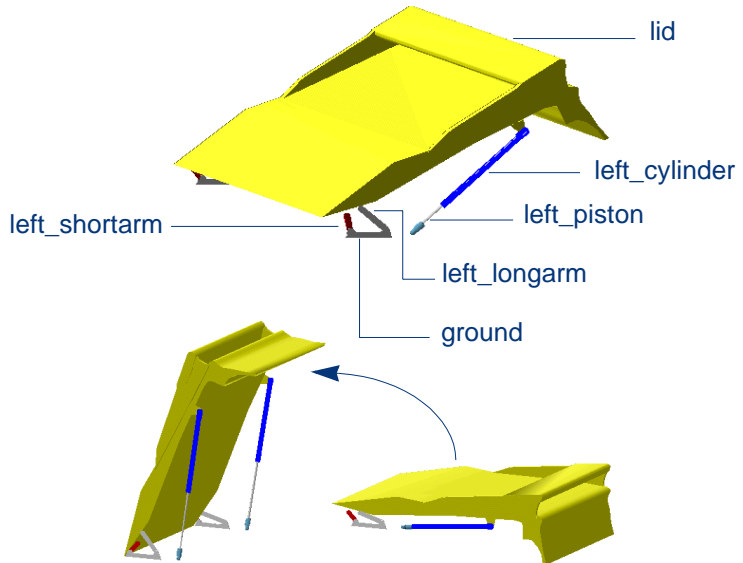
- 3 Why did you not have to perform a static simulation before the dynamic simulation when the **Lower\_Arm** was constrained with the revolute joint?

---

---

# 17 HATCHBACK I

Create the forces required to open the hatchback for the given Mazda MX-6 model.



## What's in this module:

- Impact Functions, 240
- Velocity Functions, 242
- Workshop 17—Hatchback I, 243
  - ◆ Module review, 250

# Impact Functions

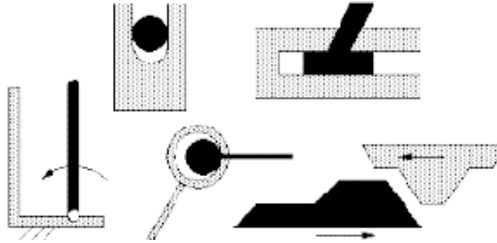
## Impact functions in MSC.ADAMS

- Are used with user-defined force elements to model contacts, impacts, collisions, and so on.
- Mimic nonlinear spring and damping forces that turn on and off depending on the distance between two objects.
- Just like a compression-only spring damper, MSC.ADAMS turns the force on when the distance between two objects,  $q$ , becomes less than the user-specified reference distance,  $q_0$ :

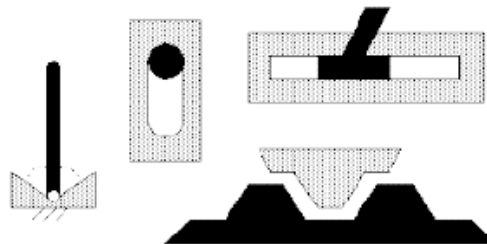
$$F_{\text{IMPACT}} = \text{Off, if } q > q_0$$

$$F_{\text{IMPACT}} = \text{On, if } q \leq q_0$$

## Applications of one-sided impact functions (IMPACT)



## Applications of two-sided impact functions (BISTOP)



# Impact Functions...

## Syntax for IMPACT function

IMPACT( $q, \dot{q}, q_o, k, e, c_{max}, d$ )

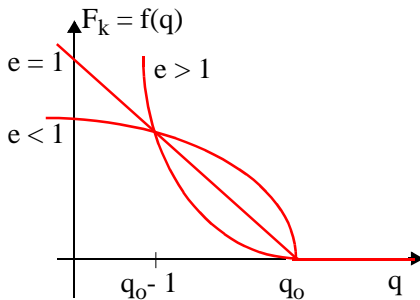
- $q$  - Actual distance between the two objects (defined with a displacement function)
- $\dot{q}$  - Time rate of change of the variable  $q$
- $q_o$  - Trigger distance used to determine when the contact force turns on and off; it should be specified as a real, constant value
- $k$  - Stiffness coefficient
- $e$  - Stiffness force exponent
- $c$  - Damping coefficient
- $d$  - Damping ramp-up distance

In MSC.ADAMS, the one-sided impact force is calculated as

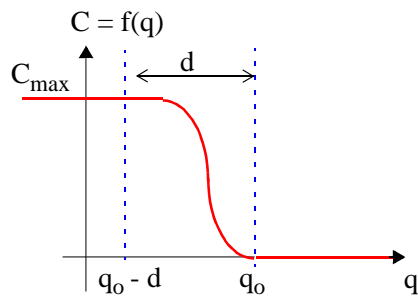
$$F = 0 \text{ if } q > q_o$$

$$F = k(q_o - q)^e - c_{max}\dot{q} * \text{STEP}(q, q_o - d, 1, q_o, 0) \text{ if } q \leq q_o$$

Compression-only spring force from one-sided IMPACT function



Compression-only damping force from one-sided IMPACT function



# Velocity Functions

---

## Definition of velocity and acceleration functions

- Returns scalar portions of velocity or acceleration vector components (translational or rotational).

## Syntax for velocity functions

- $VM(I,[J], [L])$
- $VR(I,[J], [L])$
- $VX, VY, VZ(I,[J],[R], [L])$

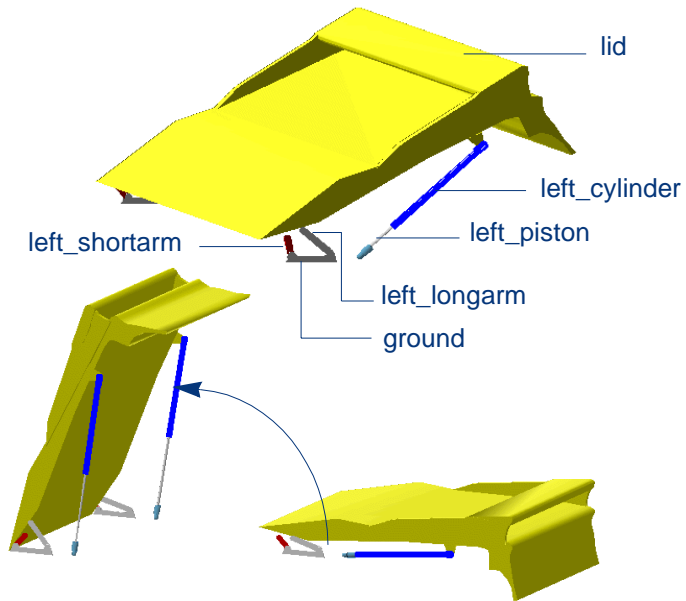
### **Notes:**

- Velocity function,  $VR$ , is used to define velocity along the line of sight, which is commonly used in spring dampers.
- If the markers are separating:  $VR > 0$ .
- If the markers are approaching:  $VR < 0$ .

# Workshop 17—Hatchback I

## Problem statement

Create the forces required to open the hatchback for the given Mazda MX-6 model.



# Workshop 17—Hatchback I...

## Model description

- When compressed, the force in each gas shock is 550 Newtons.
- The motion of the assembly is limited by stops in the gas shocks at full extension.
- Parts are constrained as shown next:

Location:	Parts:	Type:
POINT_1	left_shortarm and ground	Revolute
POINT_4	left_longarm and ground	Revolute
POINT_2	left_shortarm and lid	Spherical
POINT_6	left_cylinder and lid	Spherical
POINT_8	right_cylinder and lid	Spherical
POINT_3	lid and left_longarm	Hooke
POINT_5	ground and left_piston	Hooke
POINT_7	ground and right_piston	Hooke
POINT_56	left_piston and left_cylinder	Translational
POINT_78	right_piston and right_cylinder	Translational

## Start the workshop

To start the workshop:

- 1 Run ADAMS/View from the directory *exercise\_dir/mod\_17\_hatchback\_1*.
- 2 Import the model command file *hatchback\_start.cmd*.

## Deactivate movable parts not used for simulation

You must deactivate the parts on the right side of the model because they are not needed to constrain the model. The constraints from the left side of the model are sufficient to constrain this rigid-body model.

To deactivate parts:

- 1 Deactivate **right\_shortarm**.

**Tip:** Right-click the part and select **De(activate)**. Clear the selection of **Object Active**.

- 2 Deactivate **right\_longarm**.

## Create forces to represent gas shocks with limit stops

In this section, you will practice debugging your model. To ensure that your model is running correctly, run a simulation after you add each new force.

To create gas shocks:

- 1 Create a marker () named **lpiston\_ref**, at **POINT\_5**, located on **left\_piston**:

- **Add to Part**

- From the screen, select the **left piston** and then select **POINT\_5**.

- 2 Create a marker, named **lcyl\_ref** at **POINT\_6**, located on **left\_cylinder**.

You will use these markers to create the spring damper in the next step.

- 3 Create a spring damper between **left\_piston** (first body) and **left\_cylinder** (second body) using the markers **lpiston\_ref** (first location) and **lcyl\_ref** (second location), with the following parameters:

- **Stiffness:** 0.21578 (N/mm)

- **Damping:** 2.0 (N-sec/mm)


- 4 Modify the spring damper to add a preload of 550 N.
- 5 Create a marker, named **rpiston\_ref**, at **POINT\_7**, located on **right\_piston**.
- 6 Create a marker, named **rcyl\_ref** at **POINT\_8**, located on **right\_cylinder**.

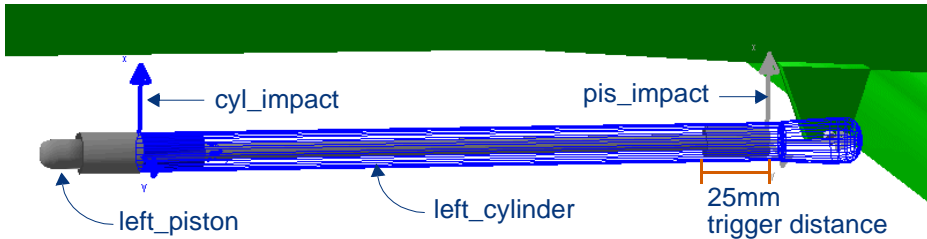
You will use these markers to create the spring damper in the next step.

## Workshop 17—Hatchback I...

- 7 Create a spring damper between **right\_piston** (first body) and **right\_cylinder** (second body) using the markers **rpiston\_ref** (first location) and **rcyl\_ref** (second location):
  - Stiffness: 0.21578 (N/mm)
  - Damping: 2.0 (N-sec/mm)
- 8 Modify the spring damper to add a preload of 550 N.

To create limit stops:

- 1 Create an SFORCE () on the left piston/cylinder, described by an impact function to stop the hatchback motion:
  - Use the **Two Bodies** method.
  - Use the existing markers, **pis\_impact** (located on **left\_piston**) and **cyl\_impact** (located on **left\_cylinder**), as shown next.



**Tip:** Make sure that you select the parts and markers in the same order. If you select the piston to be the action body, and the cylinder to be the reaction body, you should use the same order when selecting the action point (**pis\_impact**) and the reaction point (**cyl\_impact**).

## Workshop 17—Hatchback I...

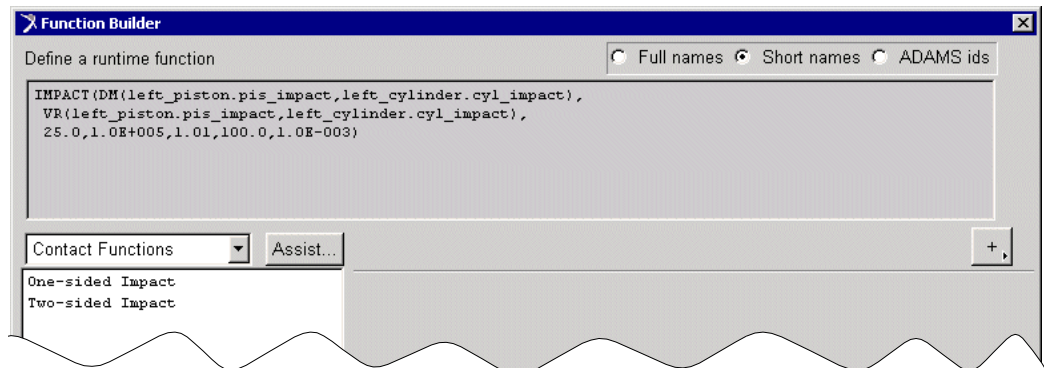
2 Modify the **SFORCE** and use the Function Builder to create a one-sided impact function. The impact functions are located in the Contact category in the Function Builder.

- The displacement parameter is equal to the magnitude of the distance between the markers, **pis\_impact** and **cyl\_impact** (use the DM function).
- The velocity parameter is equal to the velocity along the line of sight between the markers, **pis\_impact** and **cyl\_impact** (use the VR function).

**Note:** Do not enter units into the Function Builder.

- Stiffness Coefficient: 1e5 (N/mm)
  - Stiffness Force Exponent: 1.01
  - Damping Coefficient: 100 (N-sec/mm)
  - Trigger for Displacement Variable: 25 mm
  - Damping Ramp-up Distance: 1e-3 mm
- 3 While still in the Function Builder, verify the function to make sure that the syntax is correct.

The function should look as shown next:



## Workshop 17—Hatchback I...

---

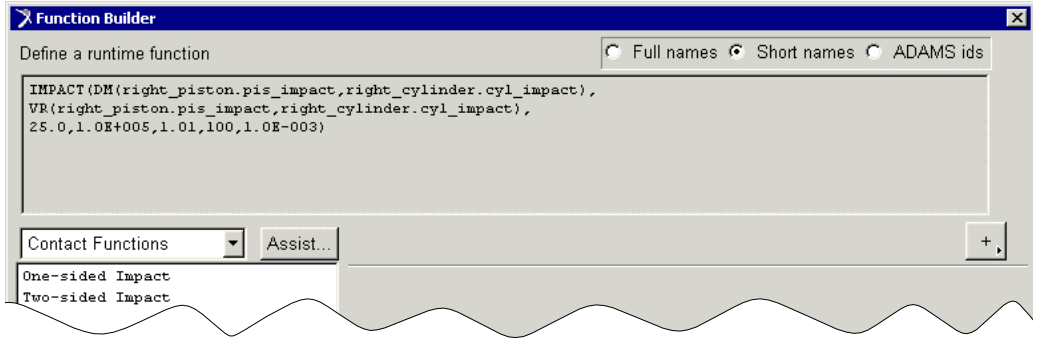
- 4 Create another **SFORCE**, on the right piston/cylinder, described by an impact function, to stop the hatchback motion:
  - Use the **Two Bodies** method.
  - Use the existing markers, **pis\_impact** (located on **right\_piston**) and **cyl\_impact** (located on **right\_cylinder**).

*Tip:* Make sure that you select the parts and markers in the same order. If you select the piston to be the action part, and the cylinder to be the reaction part, you should use the same order in the force definition.
- 5 Modify the **SFORCE** and use the Function Builder to create a one-sided impact function. The impact functions are located in the Contact category in the Function Builder.
  - The displacement parameter is equal to the magnitude of the distance between the markers, **pis\_impact** and **cyl\_impact** (use the DM function).
  - The velocity parameter is equal to the velocity along the line of sight between the markers, **pis\_impact** and **cyl\_impact** (use the VR function).
  - **Stiffness Coefficient:** 1e5 (N/mm)
  - **Stiffness Force Exponent:** 1.01
  - **Damping Coefficient:** 100 (N-sec/mm)
  - **Trigger for Displacement Variable:** 25 mm
  - **Damping Ramp-up Distance:** 1e-3 mm

## Workshop 17—Hatchback I...

- 6 While still in the Function Builder, verify the function to make sure that the syntax is correct.

The function should look as shown next:



**Note:** This syntax reflects the piston as the action part. If you chose the cylinder as the action part, the syntax should be opposite.

### Selecting parameters for a real-life model

For information on how to select parameters for a real-life model, see the following Knowledge Base articles:

- Modeling Impact: <http://support.adams.com/kb/faq.asp?ID=8230>
- Suggestions for debugging your IMPACT function: <http://support.adams.com/kb/faq.asp?ID=7301>
- Example of using Hertzian Contact Theory to estimate contact stiffness: <http://support.adams.com/kb/faq.asp?ID=8470>

# Workshop 17—Hatchback I...

---

## Simulate the model

Simulate the model to make sure that the hatchback opens and stops at a reasonable angle.

## Save your work

To save your work:

- 1 Save your 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.

## Module review

- 1 Are there any limitations to the trigger distance used in an IMPACT function? In other words, can you choose *any* value?

---

---

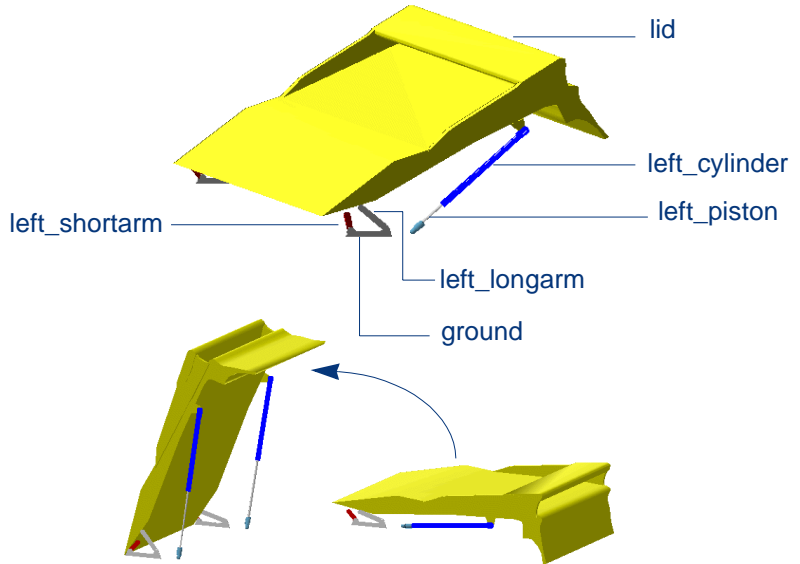
- 2 If you wanted to stop the hatchback from opening at a 45-degree angle, what steps would you take?

---

---

# 18 HATCHBACK II

Find the approximate maximum force at the winglet required to close the lid in three seconds, for the given Mazda MX-6 hatchback model.



## What's in this module:

- STEP Function, 252
- Scripted Simulations, 253
- ADAMS/Solver Commands, 254
- Workshop 18—Hatchback II, 255
  - ◆ Module review, 262

# STEP Function

## Definition of a STEP function

- In MSC.ADAMS, the STEP function approximates an ideal mathematical step function (but without the discontinuities).
- Avoid discontinuous functions because they lead to solution convergence difficulties.
- The STEP function steps quantities, such as motions or forces, up and down, or on and off.

**Note:** A STEP function is used when a value needs to be changed from one constant to another.

## Syntax for STEP function

STEP (q, q<sub>1</sub>, f<sub>1</sub>, q<sub>2</sub>, f<sub>2</sub>)

where:

q - Independent variable

q<sub>1</sub> - Initial value for q

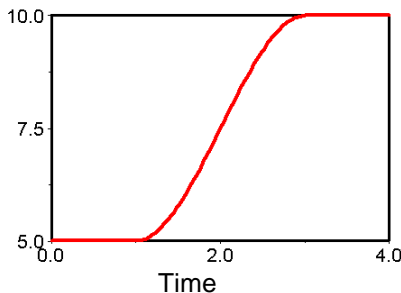
f<sub>1</sub> - Initial value for f

q<sub>2</sub> - Final value for q

f<sub>2</sub> - Final value for f

**Note:** q<sub>1</sub> < q<sub>2</sub>

## Example



STEP (time, 1, 5, 3, 10)

# Scripted Simulations

---

In ADAMS/View there are two ways to run a simulation

- Scripted
- Interactive

## Simulation scripts

- Let you program the simulation before submitting the simulation.
- Let you quickly repeat a simulation with the same set of parameters.
- Let you perform more sophisticated simulations.
- Are required for design studies, design of experiments, and optimization simulations.
- Simulation scripts are children of a model, and are, therefore, saved in a command file.

## Types of scripted simulations in ADAMS/View

- Simple run
- ADAMS/View commands
- ADAMS/Solver commands

## Scripted simulations based on ADAMS/Solver commands

- ADAMS/Solver commands let you perform sophisticated simulations, such as:
  - ◆ Changing model parameters during a simulation.
  - ◆ Using different output step sizes over different simulation intervals (versus specifying only one duration and output step size).
  - ◆ Using different solution parameters (such as convergence tolerance) over different intervals.
- Example of a simulation script that changes model topology while you work on your model:

```
simulate/dynamic, end=3.0, steps=30  
deactivate/joint, id=3  
simulate/dynamic, duration=2.0, steps=200
```

Before:



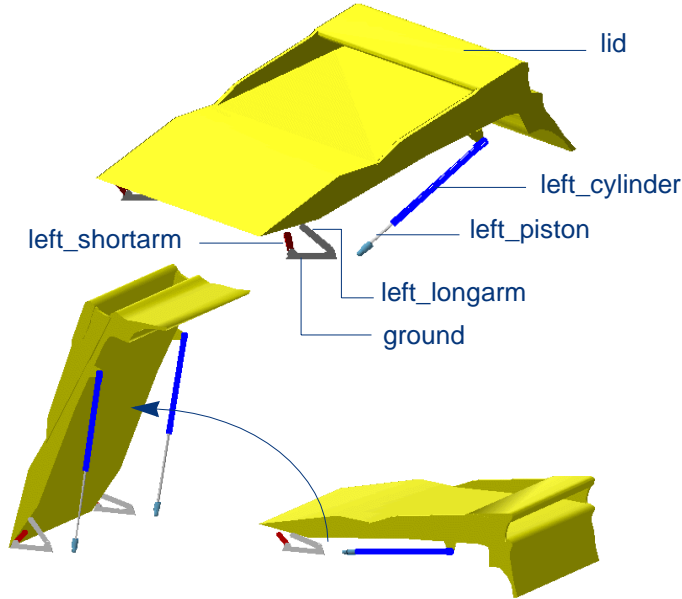
After:



# Workshop 18—Hatchback II

## Problem statement

Find the approximate maximum force at the winglet required to close the lid in three seconds, for the given Mazda MX-6 hatchback model.



## Model description

In this workshop, you will use the model you built in [Hatchback I](#), on page 239.

## Start the workshop

To start the workshop:


- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_18\_hatchback\_2*.
- 2 From the directory *exercise\_dir/mod\_17\_hatchback\_1*, import the model that you created in the previous module.

If you need a copy of the model, import the command file *hatchback\_1\_completed.cmd* from the directory *exercise\_dir/mod\_17\_hatchback\_1/completed*.

# Workshop 18—Hatchback II...

## Determine steady-state rotation of left\_shortarm

To measure the rotation of the lid:

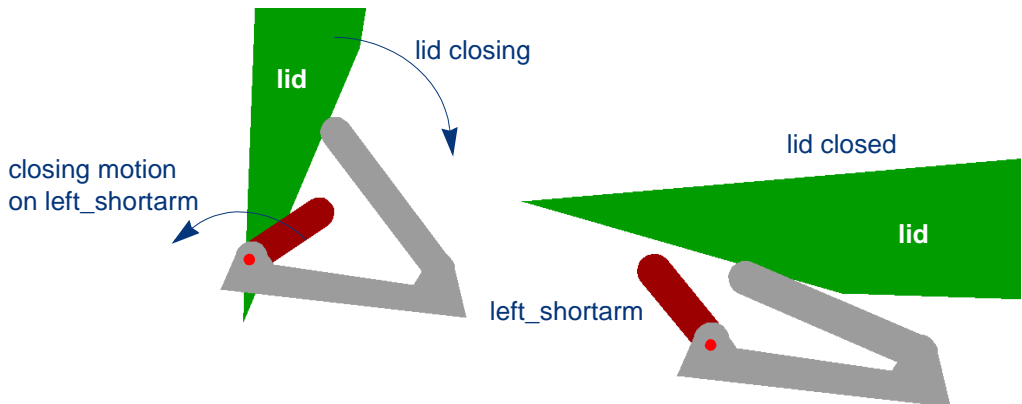
- 1 Right-click `l_shortarm_rev` joint and create a measure, named `shortarm_rotation`, of the rotational displacement:
  - Characteristic: `Ax/Ay/Az Projected Rotation`
  - Component: `Z`
  - From/At: `.ground.MAR_7`
- 2 Run a 5-second, 50-step simulation.
- 3 In ADAMS/PostProcessor, plot the `shortarm_rotation` versus `time`.
- 4 From the `shortarm_rotation` plot, use the **Plot Tracking** tool  to determine the steady-state angle of the `left_shortarm`.

The steady-state angle is 96.0693 d.

## Close the lid

Currently the lid opens because of the preload in the springs and stops opening because of the impact forces.

To close the lid, you will rotate the `left_shortarm` part back to its original position, as shown next. To rotate the `left_shortarm`, apply a joint motion to the `left_shortarm_rev` joint, as explained next.

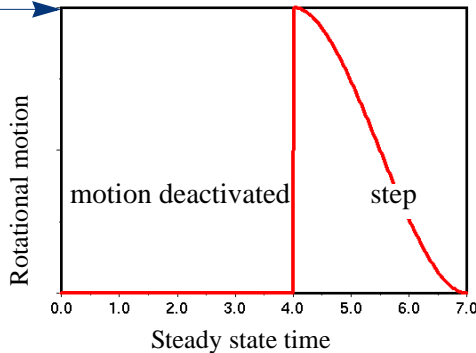


# Workshop 18—Hatchback II...

To create a motion to close the lid:

- 1 Create a joint motion on the joint `I_shortarm_rev`, named `closing_motion`.
- 2 Use a STEP function to modify the motion to drive the lid back to its closed position:
  - Start the STEP function at the steady-state rotation (determined in Step 4 on page 256) of the `left_shortarm` at 4 seconds.
  - End the STEP function at  $0^{\circ}$  rotation of the `left_shortarm` at 7 seconds.

Steady state  
opening angle



- The function should look as shown next:

```
STEP(time, 4.0, 96.0693d, 7.0, 0.0d)
```

## Perform a scripted simulation

In this section, you'll create a simulation script containing ADAMS/Solver commands that deactivate the motion and run a simulation, then activate the motion and run a second simulation.

To create the script:

- 1 From the **Simulate** menu, point to **Simulation Script**, and then select **New**.
- 2 Name the script, `script_1`.
- 3 Set **Script Type** to **ADAMS/Solver Commands**.

## Workshop 18—Hatchback II...

---

- 4 Enter the following ADAMS/Solver commands:  
DEACTIVATE/MOTION, id=1  
SIMULATE/DYNAMIC, END=4, STEPS=40  
ACTIVATE/MOTION, id=1  
SIMULATE/KINEMATIC, END=7, STEPS=30
- 5 Select OK.

To perform a scripted simulation:

- 1 From the **Simulate** menu, select **Scripted Controls**.
- 2 Enter the name of the script that you created, **script\_1**.
- 3 Select the **Play** tool.

### Measure the torque

You now create a measure of the torque required to close the lid. You then deactivate this measure because it is dependent on the motion that is deactivated when the scripted simulation starts.

To create the measure:

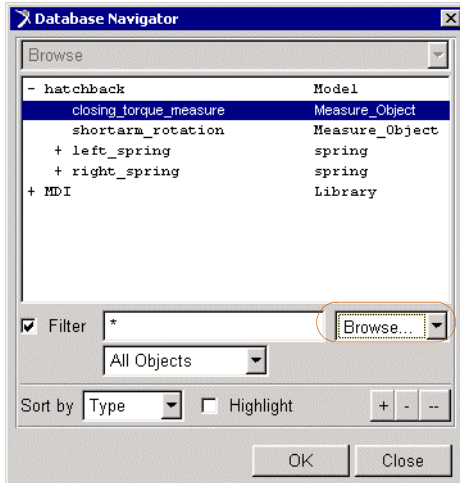
- Create a motion measure, named **closing\_torque\_measure** (right-click **closing\_motion**, and then select **Measure**):
  - ◆ **Characteristic:** Torque
  - ◆ **Component:** Z

To deactivate the measure:

- 1 By default, when you deactivate something using the **Edit** menu, ADAMS/View deactivates anything that is currently in your **Select** list. Therefore, first clear your **Select** list by selecting the **Select** tool, so you do not accidentally deactivate something else in your model.
- 2 From the **Edit** menu, select **Deactivate**.

## Workshop 18—Hatchback II...

- 3 Use the **Browse** option to filter only on measures.



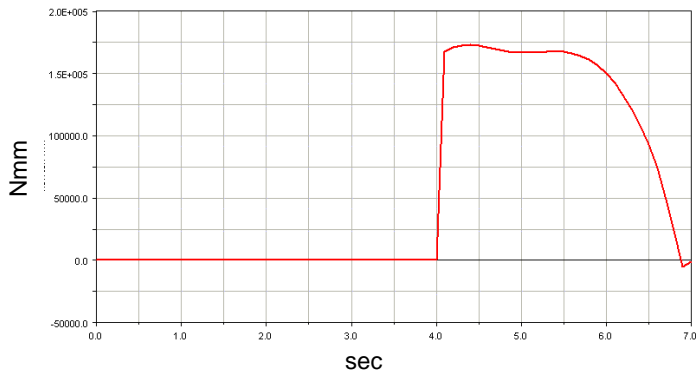
- 4 Select `closing_torque_measure`.
- 5 Select OK.


## Inspect the measure

To inspect the torque measure:

- 1 In ADAMS/PostProcessor, plot torque in the motion versus time (the `closing_torque_measure`).

**Figure 3. Plot of Approximate Torque Required to Close the Lid**

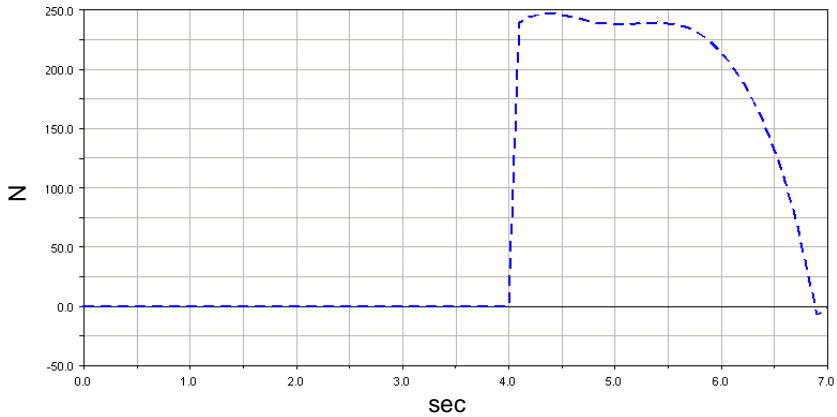


- 2 Edit the torque curve to find the approximate force required to lower the lid in three seconds. To find this force, use the **Scale a Curve** tool to divide the motion torque by a moment arm of 700 mm:
  - To display the **Scale a Curve** tool, from the **View** menu (inside ADAMS/PostProcessor), point to **Toolbars**, and then select **Curve Edit Toolbars**.  
A new toolbar appears.
  - Select the **Scale a Curve** tool .
  - Set **Scale** to **1/700**.
  - Select the curve.

## Workshop 18—Hatchback II...

- 3 To display only the force curve, delete the motion torque curve.

**Figure 4. Plot of Approximate Force Required to Close the Lid**



- 4 Note the approximate maximum force required to close the lid. Use the value to answer Question 1 in [Module review](#), on page 262.

### Save your work

To save your work:

- 1 Save your model.
- 2 Exit ADAMS/View.

# Workshop 18—Hatchback II...

---

## Module review

1 What is the approximate maximum force required to close the lid?

---

---

2 Is it possible to modify a force from one constant value to another *instantaneously*, such as shutting off a motor's torque?

---

---

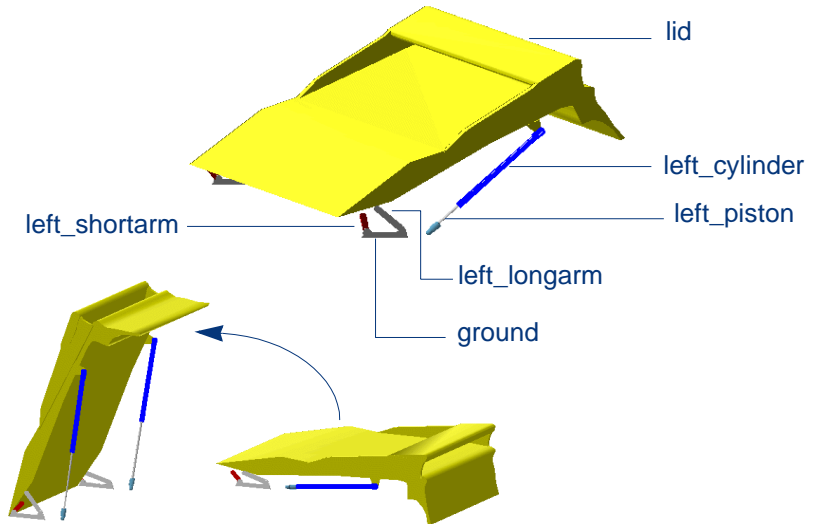
3 Is it possible to use different output step sizes over different intervals by submitting an interactive simulation?

---

---

# 19 HATCHBACK III

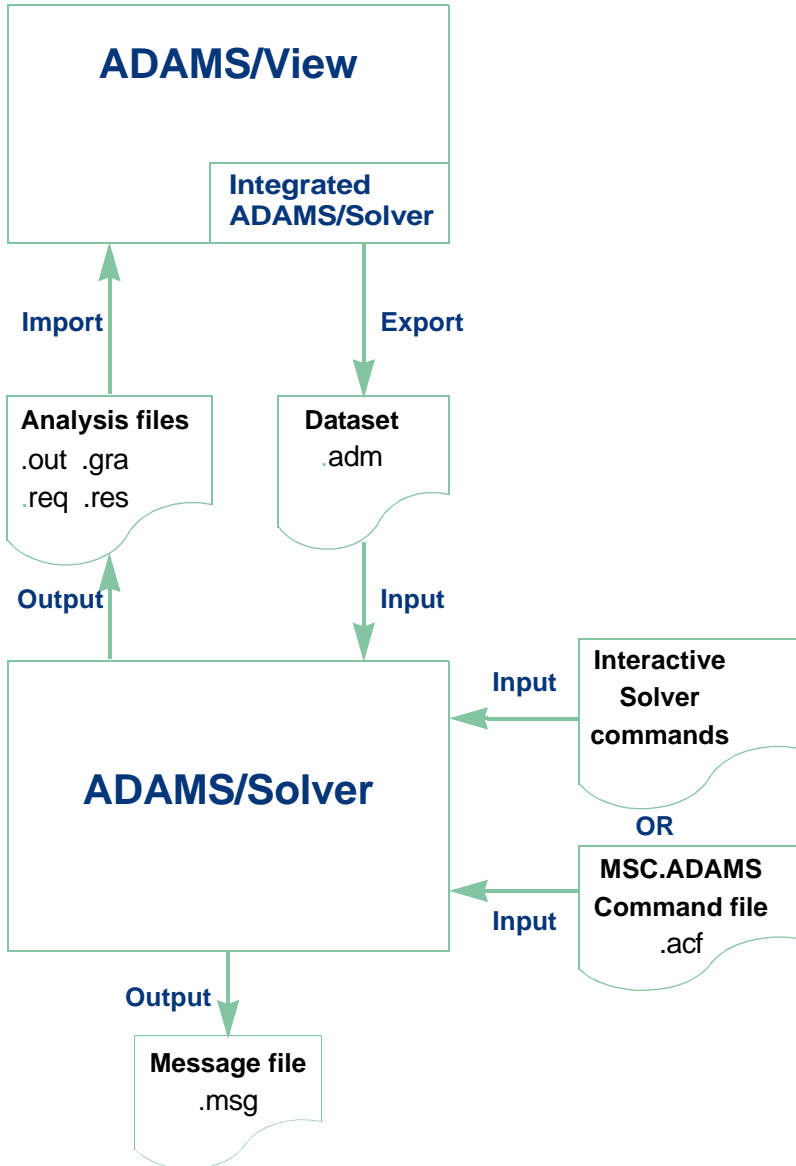
Use ADAMS/Solver to simulate the given Mazda MX-6 hatchback model.



## What's in this module:

- ADAMS/Solver Overview, 264
- Files in ADAMS/Solver, 265
- Example of an ADAMS/Solver Dataset (.adm) File, 266
- Stand-Alone ADAMS/Solver, 267
- Example: 2D Pendulum, 269
- Formulation of the Equations of Motion, 270
- Phases of Solution, 271
- Debug/Eprint (dynamics), 276
- Workshop 19—Hatchback III, 278
  - ◆ Module review, 285

# ADAMS/Solver Overview



## ADAMS/Solver dataset files (.adm)

- Statements define an element of a model such as a part, constraint, force, and so on.
- Functions are numeric expressions that define the magnitude of an element such as a force or motion.



For more information, see the ADAMS/Solver online help.

## ADAMS/Solver command files (.acf)

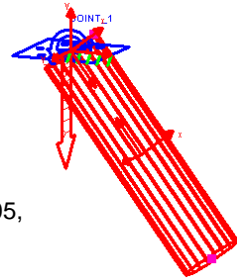
Commands define an action that needs to be taken during a simulation.

**See also:** [ADAMS/Solver Commands](#), on page 254

# Example of an ADAMS/Solver Dataset (.adm) File

Pendulum

```
!----- SYSTEM UNITS -----
UNITS/FORCE = NEWTON, MASS = KILOGRAM, ,LENGTH =
MILLIMETER, TIME = SECOND
!-----STATEMENTS FROM ORIGINAL DATASET -----
!
MATERIAL/1, NAME = steel, YOUNGS_MODULUS = 2.07E+005,
, POISSONS_RATIO = 0.29
, DENSITY = 7.801E-006
!
PART/1, GROUND
!
MARKER/1, PART = 1
!
MARKER/5, PART = 1, QP = 175, -225, 0
!
PART/2, MASS = 70.94, CM = 3, IP = 2.01E+006, 1.80E+005
, 2.01E+006, MATERIAL = steel
!
MARKER/2, PART = 2, REULER = 37.87498365D, 90D, 0D
!
MARKER/3, PART = 2, QP = 175, -225, 0, REULER = 37.87498365D, 0D, 0D
!
MARKER/4, PART = 2
!
GRAPHICS/1, CYLINDER, CM = 2, LENGTH = 570.08, RADIUS = 71.26
!
JOINT/1, REVOLUTE, I = 4, J = 1
!
REQUEST/1, DISPLACEMENT, I = 3, J = 5, RM = 5
ACCGRAV/JGRAV = -9806.65
OUTPUT/REQSAVE, GRSAVE
RESULTS/
!
MOTION/1, ROTATIONAL, JOINT = 1, FUNCTION = 30.0d * time
```



# Stand-Alone ADAMS/Solver

---

## Simulations in stand-alone ADAMS/Solver

- Interactive:
  - ◆ Not scripted: enter commands one by one.
  - ◆ Scripted: use an ADAMS/Solver command file (.acf).
- Batch - Run multiple jobs in the background using an ADAMS/Solver command file (.acf).

**Note:** ADAMS/Solver command files must start with the name of the model to be analyzed and must end with a **STOP** command.

You can run simulations externally in ADAMS/Solver from within ADAMS/View

# Solver Compatibility

---

With MSC.ADAMS 2003, the new ADAMS/Solver (C++) version has added significant functionality. With these additions, ADAMS/Solver (C++) now supports some entities that are not supported for ADAMS/Solver (FORTRAN). For this reason, a solver-compatibility check has been added. When using ADAMS/View, this check is called as each object is created.

The check is also called for:

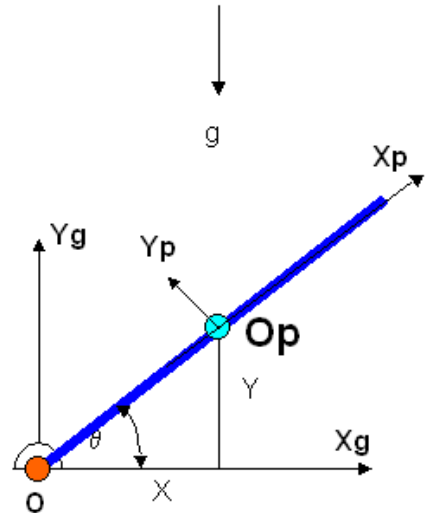
- Each object as it is created when a .cmd file is imported
- The entire model when an .adm file is imported
- The entire model before simulation

# Example: 2D Pendulum

## MSC.ADAMS Implementation: Euler-Lagrange Equations

### Description

- A link of mass  $M$ , moments of inertia  $I$ , and length  $2L$  is attached to ground using a revolute joint at the global origin  $O$ . The joint is oriented in such a way that motion is only allowed in the  $X$ - $Y$  plane of the global coordinate system.
- The coordinates of the center of mass of the link, with respect to the global origin, are represented by the states  $(x,y)$ .
- A coordinate system  $(O_p-X_p-Y_p)$  is attached at the center of mass of the link, such that  $X_p$  is along the length of the link. The angle between  $X_p$  and  $X_g$  is denoted by  $\theta$ .



Force balance equations	$p - \frac{\partial L}{\partial q} + \phi_q^T \lambda - H^T F = 0 \Rightarrow \begin{bmatrix} M\dot{U}_x + \lambda_1 = 0 \\ M\dot{U}_y + \lambda_2 - Mg = 0 \\ \dot{P}_\theta + \lambda_1 L \sin(\theta) - \lambda_2 L \cos(\theta) \end{bmatrix}$
Momenta equations (only in $\theta$ )	$p - \frac{\partial L}{\partial \dot{q}} = 0 \Rightarrow p_\theta - I\dot{\theta} = 0$ $U_x - \dot{x} = 0$
Kinematic differential equations	$u - \dot{q} = 0 \Rightarrow U_y - \dot{y} = 0$ $U_\theta - \dot{\theta} = 0$
Constraint equations	$x - L \cos(\theta) = 0$ $y - L \sin(\theta) = 0$

# Formulation of the Equations of Motion

Nonlinear system - Nine differential and algebraic equations (DAE's)

	Equations of motion	Unknowns
Force balance	$M\dot{U}_x + \lambda_1 = 0$	$U_x$
	$M\dot{U}_y + \lambda_2 - Mg = 0$	$U_y$
	$\dot{p}_\theta + \lambda_1 L \sin(\theta) - \lambda_2 L \cos(\theta) = 0$	$p_\theta$
Momenta	$p_\theta - I\dot{U}_\theta = 0$	$U_\theta$
Kinematics	$U_x - \dot{x} = 0$	$x$
	$U_y - \dot{y} = 0$	$y$
	$U_\theta - \dot{\theta} = 0$	$\theta$
	$x - L \cos(\theta) = 0$	$\lambda_1$
	$y - L \sin(\theta) = 0$	$\lambda_2$

$$G(Y, \dot{Y}, t) = 0 \quad Y = \{U_x, U_y, U_\theta, p_\theta, x, y, \theta, \lambda_1, \lambda_2\}$$

# Phases of Solution

---

## Task

Solve the differential and algebraic equation:  $G(y, \dot{y}, t) = 0, y(0) = y_0$

## Two major components: Predictor and Corrector

### Phase 1:

Predict an initial solution

### Phase 2:

Correct the prediction

### Phase 3:

Evaluate quality of solution (accept solution)

### Phase 4:

Prepare for next step

# Phases of Solution...

---

## Task

Solve the differential and algebraic equation:  $G(y, \dot{y}, t) = 0, y(0) = y_0$

### Phase 1:

#### **Predict an initial solution**

Predict an initial value using an explicit method:

- The predictor is simply looking at past values to guess the solution at the next time. The governing equations  $G$  are not satisfied.
- This is simply a good starting point for the next phase.

### Phase 2:

Correct the prediction

### Phase 3:

Evaluate quality of solution (accept solution)

### Phase 4:

Prepare for next step

# Phases of Solution...

## Task

Solve the differential and algebraic equation:  $G(y, \dot{y}, t) = 0, y(0) = y_0$

### Phase 1:

Predict an initial solution

### Phase 2:

#### Correct the prediction

Evaluate G. If G is near zero, corrector is finished. Go to phase 3.

Use the **Newton-Raphson method** to correct the prediction.

Solve for  $\Delta y$ . Update y.

Repeat iteration until  $\|\Delta y\| \leq \text{corrector error tolerance}$

Example:

$$f(q) = q^2 + \sin q - 1.841471 = 0$$

$$f' = 2q + \cos q$$

As a first guess, set  $q = 2$

STEP	q	f'(q)	f(q)
1	2	3.5838	3.0678
2	$2 - 3.0678/3.5838 = 1.1439$	2.7109	0.3775
3	$1.1439 - 0.3775/2.7109 = 1.004$	2.5451	0.0107

The exact answer is  $q = 1.0$

### Phase 3:

Evaluate quality of solution (accept solution)

### Phase 4:

Prepare for next step

# Phases of Solution...

## Task

Solve the differential and algebraic equation:  $G(y, \dot{y}, t) = 0, y(0) = y_0$

### Phase 1:

Predict an initial solution

### Phase 2:

Correct the prediction

### Phase 3:

**Evaluate quality of solution (accept solution)**

Estimate local truncation error

if estimated  $\leq (\epsilon_L)$

Yes  $\rightarrow$  Accept solution. Go to phase 4

No  $\rightarrow$  Reject solution and repeat phase 1 and 2 with new step size

**Global Error ( $\epsilon_G$ )**

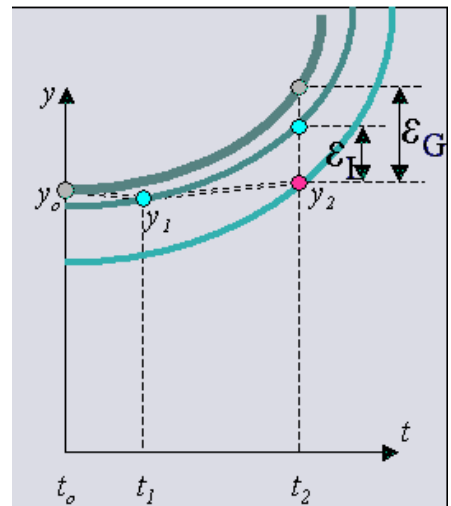
The difference between the current solution and the true solution

**Local Truncation Error ( $\epsilon_L$ )**

The error introduced in a single step

### Phase 4:

Prepare for next step



# Phases of Solution...

---

## Task

Solve the differential and algebraic equation:  $G(y, \dot{y}, t) = 0, y(0) = y_0$

### Phase 1:

Predict an initial solution

### Phase 2:

Correct the prediction

### Phase 3:

Evaluate quality of solution (accept solution)

### Phase 4:

#### **Prepare for next step**

Update higher order derivatives used in prediction for the next step

Determine step size and order for next step

Go back to phase 1, and start new step

# Debug/Eprint (dynamics)

---

Each GSTIFF integrator step consists of two phases:

Phase 1: a forward step in time (the predictor for dynamics)

- 1 The step number - A running count of the number of steps taken and can be used as a measure of how hard ADAMS/Solver is working.
- 2 The order of the predictor for dynamics - Corresponds to the order of the polynomial ADAMS/Solver uses to predict the solution at the end of the integration step.
- 3 The value of time at the beginning of the step.
- 4 The size of the step.

Phase 2: the solution of the equations of motion (the corrector for dynamics).

- 5 The cumulative number of iterations - A running count of the iterations needed to solve the equations of motion and can be used as a measure of how many computations ADAMS/Solver is performing.
- 6 The iteration number - One at the beginning of each step and increments by one until ADAMS/Solver converges to a solution or exceeds the maximum allowable number of iterations.
- 7 Absolute value of largest equation residual error - This number is an indicator of how far ADAMS/Solver is from a solution. This number should decrease after every iteration in healthy simulations.
- 8 Dataset element associated with #7 - The equation that has the largest equation residual error for the above dataset element.
- 9 Equation associated with #8.
- 10 Absolute value of largest change in a variable - The final iteration should not need to change variables very much. This number is an indicator of how far ADAMS/Solver needs to change variables to approach a solution. Ideally, this number should decrease after every iteration.
- 11 Dataset element associated with #10.
- 12 Variable with the largest change for #11.
- 13 Jacobian updates - If ADAMS/Solver has updated the Jacobian, YES appears under the Jacobian new? header.

# Debug/Eprint (dynamics)...

3. Time at beginning of step

Corrector information

1. Running count of successful steps

2. Order of predicting polynomial

Integration step No.	30, Order = 2	4. From 2.100000000E-01, Step = 1.00000E-02		5. Cumulative iterations of the corrector: 58			Jacobian
Iteration count	Residual (or equation error)	Change in the Variable	Maximum Element/ID	Equation	Maximum Element/ID	Variable new?	
6.	7.	8.	9.	10.	11.	12.	13.
1	-4.1E-08 PART/2	Psi Trq		-3.8E-10 PART/2	Y Vel		Yes
2	0.0E+00 PART/3	X Force		0.0E+00 PART/3	X Vel		

Time = 0.220. Step rejected due to impending contact. Limiting order, step-size.

Integration step No.	30, Order = 2	4. From 2.100000000E-01, Step = 1.44255E-03		5. Cumulative iterations of the corrector: 60			Jacobian
Iteration count	Residual (or equation error)	Change in the Variable	Maximum Element/ID	Equation	Maximum Element/ID	Variable new?	
1	2.1E+07 CONTACT/1	Fy		-2.2E+03 CONTACT/1	Fy		Yes
2	-1.4E+03 CONTACT/1	Fy		-1.6E+00 PART/3	X Vel		
3	-1.4E+03 CONTACT/1	Fy		-2.3E-04 PART/2	X Vel		
4	-1.4E+03 CONTACT/1	Fy		2.1E-04 PART/3	X Vel		
5	-1.4E+03 CONTACT/1	Fy		3.0E+00 PART/3	X Vel		Yes

All Newton-Raphson updates divided by: 65.715

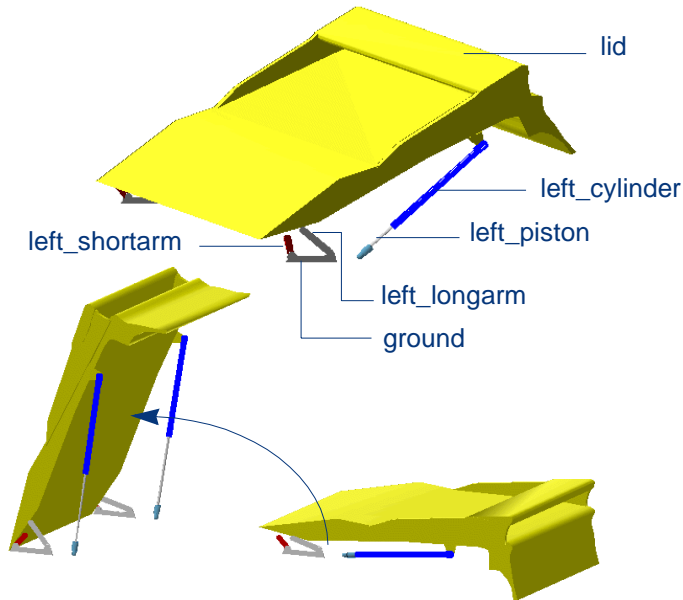
6 2.1E+07 CONTACT/1 Fy -4.7E+05 CONTACT/1 Fy

Time = 0.21144, Step = 1.44255E-03, Iterations = 6. Corrector did not converge.

# Workshop 19—Hatchback III

## Problem statement

Use ADAMS/Solver to simulate the given Mazda MX-6 hatchback model.



## Model description

In this workshop, you use the model you saved in [Workshop 18—Hatchback II](#), on page 255.

## Start the workshop

To start the workshop:

- 1 Start ADAMS/View from the directory *exercise\_dir/mod\_19\_hatchback\_3*.
- 2 From the directory *exercise\_dir/mod\_18\_hatchback\_2*, import the model that you created in the previous module.

If you need a fresh copy of the model, import the command file *hatchback\_2\_completed.cmd* from the directory *exercise\_dir/mod\_18\_hatchback\_2/completed*.

# Workshop 19—Hatchback III...

## Export a dataset (.adm) file

To export a dataset file:

- 1 From the File menu, select **Export**.
- 2 Enter the following, and then select **OK**:
  - File Type: **ADAMS/Solver Data Set**
  - File Name: **hatchback.adm**

ADAMS/View exports this file to your current working directory, `exercise_dir/mod_19_hatchback_3`.

## Create a command file (.acf)

To create a command file:

- 1 Open a text editor (UNIX: `vi` or `Jot`; Windows: Notepad or Wordpad), and create an ADAMS/Solver command file (.acf) that contains the following commands:

```
hatchback.adm (the .adm extension is optional)
hatchback_test1
OUTPUT/NOSEPARATOR
DEACTIVATE/MOTION, id=1
SIMULATE/DYNAMIC, END=4, STEPS=40
ACTIVATE/MOTION, id=1
SIMULATE/KINEMATIC, END=7, STEPS=30
STOP
```

Do not enter in acf.

- 2 Save the file as **hatchback.acf** in your current working directory, `exercise_dir/mod_19_hatchback_3`.

# Workshop 19—Hatchback III...

## Perform a simulation in stand-alone ADAMS/Solver

To perform a simulation in stand-alone ADAMS/Solver, you use the MSC.ADAMS Program Menu, a menu- and text-based interface that allows you to enter information on the command line.

To prepare to run ADAMS/Solver:

- 1 Depending on the platform you're on, do the following:
  - Windows: From the **Start** menu, select **Run** and open a command window by typing in `cmd`. Change directories to your working directory, `exercise_dir/mod_19_hatchback_3`.

**Table 1. Common Windows Commands**

change directory	cd <directory_name>
change disks	<drive_letter>: (for example, C:)
list	dir

- UNIX: Open a UNIX shell and change directories to your working directory, `exercise_dir/mod_19_hatchback_3`.

**Table 2. Common UNIX Commands**

change directory	cd <directory_name>
list	ls
list path	pwd

- 2 Leave the window open because you will be running ADAMS/Solver from this window.

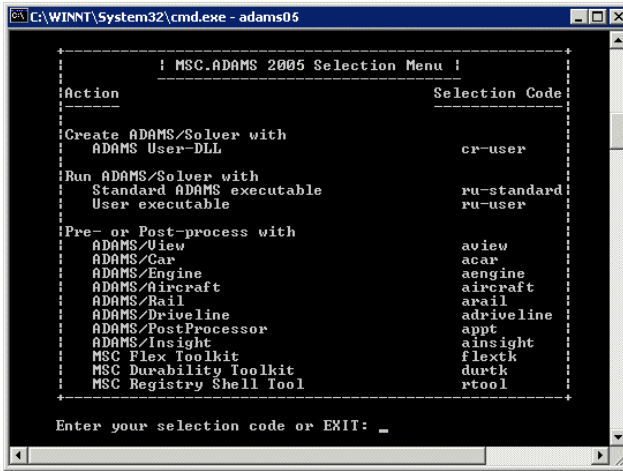
# Workshop 19—Hatchback III...

To perform the simulation:

1 Perform a simulation in stand-alone ADAMS/Solver using the command file you just created, `hatchback.acf`. Type the following commands in the window you just prepared:

■ Windows:

- ◆ `adamsxx` (where `xx` is the version number; for example, `adams05`) (displays the MSC.ADAMS Program Menu)



```

C:\WINNT\System32\cmd.exe - adams05
-----
! MSC.ADAMS 2005 Selection Menu !
-----
:Action                                     Selection Code
-----
:Create ADAMS/Solver with
  ADAMS User-DLL                             cr-user
:Run ADAMS/Solver with
  Standard ADAMS executable                 ru-standard
  User executable                          ru-user
:Pre- or Post-process with
  ADAMS/View                                aview
  ADAMS/Car                                  acar
  ADAMS/Engine                               aengine
  ADAMS/Aircraft                             aircraft
  ADAMS/Rail                                  arail
  ADAMS/DriveLine                            adriveline
  ADAMS/PostProcessor                         apt
  ADAMS/Insight                              ainsight
  MSC Flex Toolkit                            flextk
  MSC Durability Toolkit                      durtk
  MSC Registry Shell Tool                    rtool
-----
Enter your selection code or EXIT: _

```

- ◆ `ru-s` (runs ADAMS/Solver with standard MSC.ADAMS executable)
- ◆ `hatchback.acf` (identifies the `.acf` ADAMS/Solver file and runs the simulation)

■ UNIX:

- ◆ `adamsxx -c` (where `xx` is the version number; for example, `adams12 -c`) (displays the MSC.ADAMS Program Menu)
- ◆ `ru-s` (runs ADAMS/Solver with standard MSC.ADAMS executable)
- ◆ `i` (sets interactive mode)
- ◆ `hatchback.acf` (identifies the `.acf` ADAMS/Solver file and runs the simulation)
- ◆ `exit` (exits ADAMS/Solver)

**Note:** As common practice, you should open the message file (`.msg`) and search for errors and warnings. Correct the model or the `.acf` file to eliminate the errors and warnings.

2 Leave the window open because you will be using it again soon.

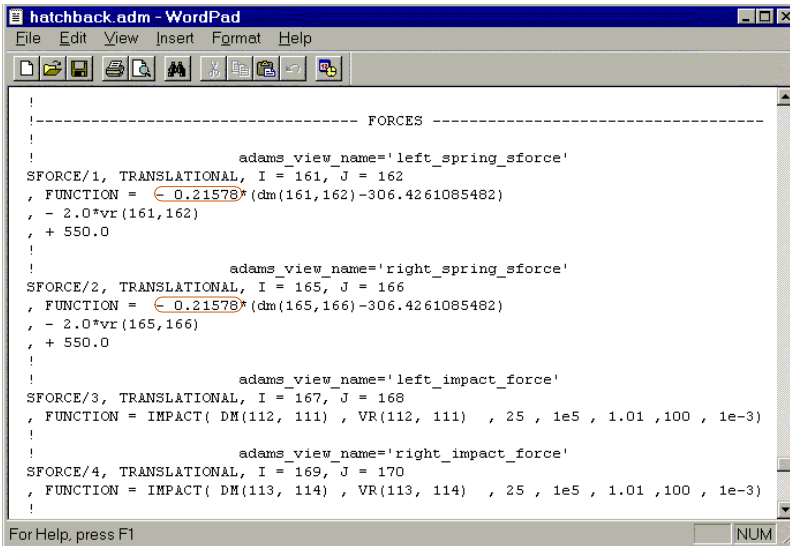
# Workshop 19—Hatchback III...

## Modify the dataset (.adm) file

Now change the spring stiffness in the .adm.

To change the spring stiffness:

- 1 In a text editor, open **hatchback.adm**.
- 2 Modify the value of spring stiffness (for both springs) to **-0.30 N/mm**.



```
!
!----- FORCES -----!
!
!           adams_view_name='left_spring_sforce'
SFORCE/1, TRANSLATIONAL, I = 161, J = 162
, FUNCTION = (0.21578)*(dm(161,162)-306.4261085482)
, - 2.0*vr(161,162)
, + 550.0
!
!           adams_view_name='right_spring_sforce'
SFORCE/2, TRANSLATIONAL, I = 165, J = 166
, FUNCTION = (0.21578)*(dm(165,166)-306.4261085482)
, - 2.0*vr(165,166)
, + 550.0
!
!           adams_view_name='left_impact_force'
SFORCE/3, TRANSLATIONAL, I = 167, J = 168
, FUNCTION = IMPACT( DM(112, 111) , VR(112, 111) , 25 , 1e5 , 1.01 ,100 , 1e-3)
!
!           adams_view_name='right_impact_force'
SFORCE/4, TRANSLATIONAL, I = 169, J = 170
, FUNCTION = IMPACT( DH(113, 114) , VR(113, 114) , 25 , 1e5 , 1.01 ,100 , 1e-3)
!
```

- 3 Save the file as **hatchback2.adm**.

# Workshop 19—Hatchback III...

## Modify the ADAMS/Solver command file (.acf)

Modify the .acf to run with hatchback2.adm.

To modify the .acf:

- 1 In a text editor, open **hatchback.acf**.
- 2 Modify the first and second lines of the .acf file so they are:  
**hatchback2**  
**hatchback\_test2**
- 3 Save the file as **hatchback2.acf**.

## Perform a simulation in stand-alone ADAMS/Solver:

- Using the new command file, perform a stand-alone ADAMS/Solver simulation.

## Compare the results of the two simulations in ADAMS/View:



Import both sets of results (**hatchback\_test1** and **hatchback\_test2**) into ADAMS/View, and then compare them.

To import and compare the results:

- 1 In ADAMS/View, from the **File** menu, select **Import**.
- 2 To import the results for **hatchback\_test1**, enter the following, and then select **OK**:
  - **File Type:** ADAMS/Solver Analysis (.req, .gra, .res)
  - **File to Read:** hatchback\_test1.res
  - **Model Name:** hatchback
- 3 To import the results for **hatchback\_test2**, repeat Step 2 above to import the results, **hatchback\_test2.res**.
- 4 Display the Database Navigator.
- 5 Change the **Filter** type from **Modeling** to **Analyses**.
- 6 Double-click the hatchback model to make sure that the two analyses, **hatchback\_test1** and **hatchback\_test2**, are children of the model.

# Workshop 19—Hatchback III...

To inspect both simulations:

- 1 Launch ADAMS/PostProcessor.
- 2 From the **Simulation** list, select the two analyses, `hatchback_test1` and `hatchback_test2`.
- 3 From the **Measure** list, select `closing_torque_measure`, and then select **Add Curves**.
- 4 Select the **Scale a Curve** tool  to individually scale both curves by the moment arm (1/700) (proceed just as you did in [Inspect the measure](#), on page 260).
- 5 Delete the old curves.
- 6 Use the **Subtract one curve from another** tool  to find the approximate difference between the maximum force values. Use this value to answer question Step 2 in [Module review](#), on page 285.
- 7 Save your 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.

- 8 Exit ADAMS/View.

## Optional tasks

Simulate the model in ADAMS/Solver:

- 1 Simulate interactively but without a script (no `.acf` file).
- 2 From a DOS prompt, enter the following commands, where `xx` is the current version of MSC.ADAMS:
  - `adamsxx`
  - `ru-standard`
  - `<CR>` (Because you do not have an ADAMS/Solver command file (`.acf`), press the enter key).

ADAMS/Solver starts.

# Workshop 19—Hatchback III...

---

- 3 In ADAMS/Solver, enter:
  - **hatchback** (the name of your MSC.ADAMS Dataset (.adm) file)
  - **hatchback\_test3** (the desired output file names .gra, .res, .out, and so on)ADAMS/Solver reads in the file and performs the assemble simulation.
- 4 At the MSC.ADAMS command prompt, enter commands one at a time in the same order in which they appear in the .acf file.
- 5 After entering all the commands, exit ADAMS/Solver and import your results into ADAMS/View so you can inspect them using animations and plotting.

## Module review

- 1 What is the difference between a statement and a command?

---

---

- 2 What is the maximum force difference between the two tests (**hatchback\_test1** and **hatchback\_test2**) you performed?

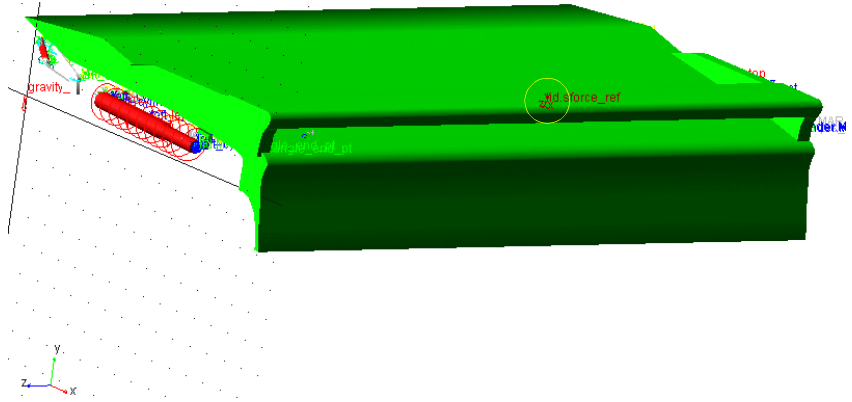
---



# 20 HATCHBACK IV

In *Workshop 19—Hatchback III*, on page 278, you determined the approximate force needed to close the lid. Now you are part of a group of design engineers for the Mazda MX-6 hatchback. Your goal is to make the model more realistic and ensure that it meets the following criteria:

- Lid opens completely in less than 4.0 seconds.
- Requires less than 210 N to close the lid.
- Takes no more than 3.0 sec to close the lid.



## What's in this module:

- Sensors, 288
- Design Variables, 289
- Workshop 20—Hatchback IV, 290
  - ◆ Module review, 297

## Sensors

- Monitor any quantity of interest in a model during a simulation, and take a specified action when the quantity reaches or exceeds a critical value.
- Take one of the following actions:
  - ◆ Completely stop the simulation.
  - ◆ If used with a script, sensors halt the current simulation and continue with the next command in the script.
  - ◆ Can be used to evaluate certain expressions when the required condition is met. You can access this value using the ADAMS/Solver function SENVAL. See the following Knowledge Base Articles:
    - ◆ Using SENVAL to count full rotations of a spinning part:  
<http://support.adams.com/kb/faq.asp?ID=10703>
    - ◆ Finding min/max of a state using a SENSOR:  
<http://support.adams.com/kb/faq.asp?ID=12377>
- A sensor basically represents an *If/Then* statement:  
*If* quantity = value (+/- tolerance)  
*Then* take a specified action

## Example of using sensors with scripts

- Monitor the reaction force in a constraint and deactivate the constraint when the force exceeds a specified value.
- Monitor the distance between two objects and reduce the solution step size just before contact, to avoid convergence problems.

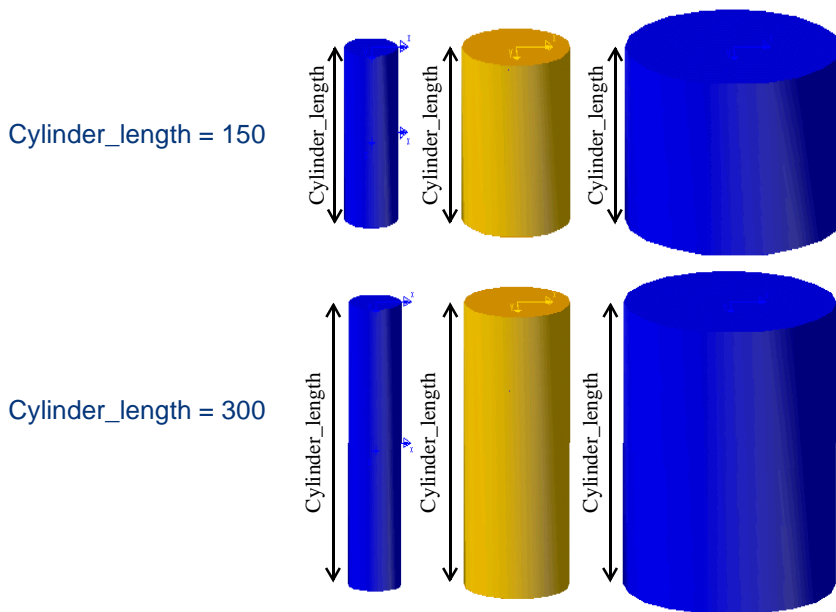
# Design Variables

## Design variables

- Define independent parameters that can be tied to objects.
- Organize the critical parameters of the design into a concise list of values that can be easily reviewed and modified.

## Example

You can create a design variable called `cylinder_length` to control the lengths of all three cylinders as shown next:



**Note:** You can also use parametric analyses to automatically run a series of simulations that vary your design variables, which you will do in [Workshop 22—Target Practice](#), on page 325.

# Workshop 20—Hatchback IV

## Problem statement

In [Workshop 19—Hatchback III](#), on page 278, you determined the approximate force needed to close the lid. Now you are part of a group of design engineers for the Mazda MX-6 hatchback. Your goal is to make the model more realistic and ensure that it meets the following criteria:

- Lid opens completely in less than 4 seconds.
- Requires less than 210 N to close the lid.
- Takes no more than 3.0 sec to close the lid.

## Start the workshop

In this workshop, you use the model you saved in [Workshop 19—Hatchback III](#), on page 278.

To start the workshop:

- 1 Start ADAMS/View from the directory `exercise_dir/mod_20_hatchback_4`.
- 2 From the directory `exercise_dir/mod_19_hatchback_3/completed`, import the command file named `hatchback_3_completed.cmd`.

## Make model more realistic


In [Workshop 19—Hatchback III](#), on page 278, you used a motion to close the lid. In this section, to make this model more realistic, you will deactivate that motion and replace it with a perpendicular force to close the lid.

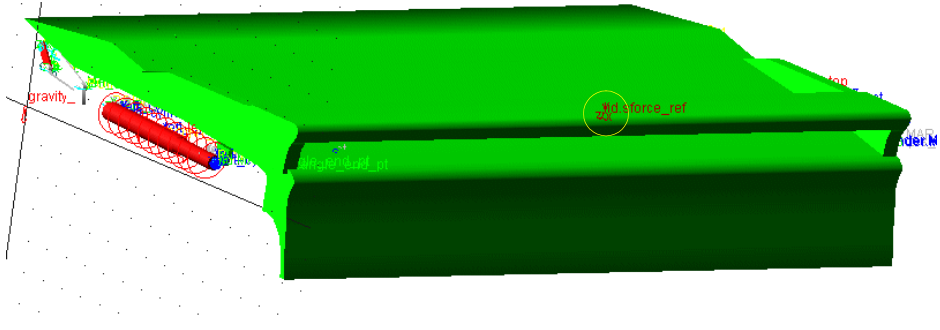
To deactivate the motion:

- Deactivate the motion `closing_motion` located on the revolute joint `I_shortarm_rev`.

# Workshop 20—Hatchback IV...

To create an SFORCE:

- 1 Create an SFORCE (  ) located at the marker, `sforce_ref`, at the tip of the lid:



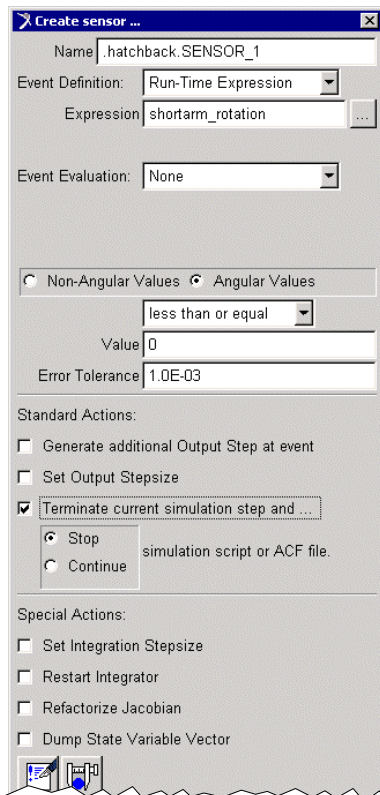
- Run-time Direction: Body Moving
  - Action body: lid
  - Location: `sforce_ref`
  - Direction: y direction of `sforce_ref`
- 2 Rename the SFORCE to `closing_force`.
  - 3 Modify the SFORCE function to be equal to the following STEP function:  
`STEP(time, 4, 0, 4.2, -247)`.  
**Note:** 247 N is the approximate force needed to close the lid, as you calculated in [Workshop 19—Hatchback III](#), on page 278.
  - 4 Run a 7-second, 100-step simulation to verify that the force closes the lid. Because you're not activating or deactivating any motions during the simulation, you can run a simple interactive simulation.  
**Note:** When using an SFORCE, there is no constraint stopping the rotation of the lid beyond the closed position.  
You will see that this approximate force is not large enough to close the lid.
  - 5 Modify the STEP function that defines the SFORCE to have a maximum value of 300 N:  
`STEP(time, 4, 0, 4.2, -300)`.

## Add a sensor

Add a sensor to stop the lid at the closed position.

To add a sensor:

- 1 Create a sensor based on the measure, `shortarm_rotation`:
  - From the **Simulate** menu, point to **Sensor**, and then select **New**.
  - Fill in the dialog box as follows:



- 2 Select **Apply**.

## Workshop 20—Hatchback IV...

- 3 Run another simulation to verify that the sensor works as expected.

Why does the sensor stop the simulation when the lid doesn't appear to be fully closed?

---

---

- 4 Back in the Create sensor dialog box, select **Generate Additional Output Steps at Event**.
- 5 Rerun the simulation.

### Modify spring preload and stiffness

Because the force required to close the lid is greater than the design criteria, you will modify the spring preload and stiffness for *each* spring.

To modify the preload:

- 1 For each spring, modify the preload to be equal to **400 N**.
- 2 Run a simulation.

The sensor was triggered at time  $3.5 \times 10^{-3}$ , meaning that instead of having the hatchback open, it dropped past the closing position because the springs were not strong enough to open the lid.

- 3 For each spring, modify the preload again to be **470 N**.

To modify the stiffness:

- 1 For each spring, modify the stiffness to be **.10 N/mm**.
- 2 Modify the STEP function for the **closing\_force** SFORCE to have a maximum value of **210 N**: **STEP(time, 4, 0, 4.2, -210)**.

This maximum value of 210N is the maximum force that we can use to close the lid, as defined in our design criteria from the problem statement.

- 3 Run a simulation.

While the springs now open the lid, the closing force is still too large.

# Workshop 20—Hatchback IV...

## Create design variables

Creating design variables in your model will help to speed up the design iteration process.

To create design variables:

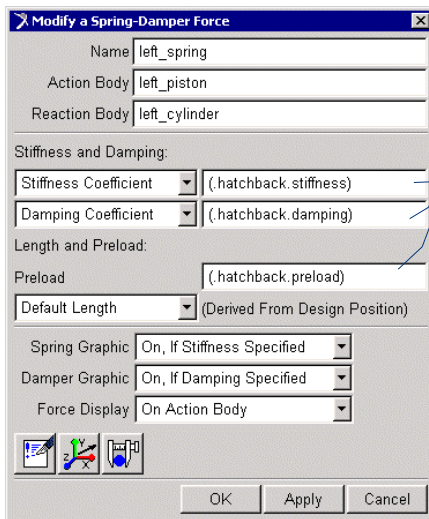
- 1 From the **Build** menu, point to **Design Variable**, and then select **New**.
- 2 Create three design variables as follows:

Name:	Units:	Standard Value:	Value Range By:	
			Min. Value	Max. Value
preload	force	460	300	600
stiffness	stiffness	.21578	.1	.5
damping	damping	2	.5	4

## Modify the springs to reference design variables

To modify design variables:

- 1 Modify the left spring as follows:



Right-click, point to **Parameterize**, select **Reference Design Variable**, and then double-click the appropriate design variable.

- 2 Repeat Step 1 for the right spring.

## Optimize design

Modify the SFORCE and the design variables to satisfy design criteria. To verify the parameters, simulate the model between the changes.

To optimize the design:

- 1 Modify the SFORCE to be equal to the design criteria (maximum force of 210 N).
- 2 Modify the standard value of the design variables until the lid opens and closes as required (**Build** → **Design Variables** → **Modify**).  
**Note:** There could be many parameter combinations that would meet the design criteria. Try a few different values to get a feel for the sensitivity of each parameter.
- 3 Save the model and exit ADAMS/View.

## Optional tasks

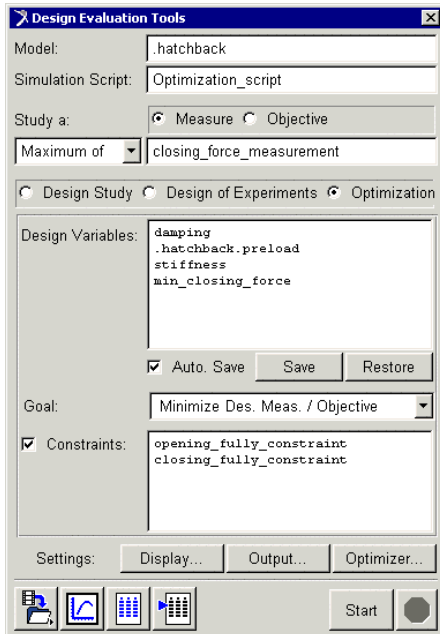
Run an optimization to find a set of values that meet the criteria of the problem statement.

To run an optimization:

- 1 Import the custom macro `/misc/optimization_optional_task.cmd`.

**Note:** This custom macro creates measurements, a design variable, a simulation script, and constraints needed to run this optimization. It also modifies your optimization settings and the values of your design variables to allow for the model to quickly optimize.

- 2 From the **Simulate** menu, select **Design Evaluation**.
- 3 Select **Optimization**.
- 4 Complete the dialog box as shown next:




- 5 Select **Optimizer**.
- 6 Set **Algorithm** to **OPTDES: SQP**.

# Workshop 20—Hatchback IV...

7 Select Start.

**Note:** Several sets of values will meet the design criteria. Further investigation is needed to determine the optimal design.

8 Select the tool **Create tabular report of results** .

ADAMS/View automatically updates the standard values for each design variable to the values found during the optimization process.

For this problem, the values that you receive from the optimization are very sensitive to the starting points of the design variables. If time permits, try to modify the starting values of the design variables and run the above optimization again. Did you find an optimal value? What went wrong?

**Hint:** Did the lid even open?

## Module review

1 Which parameters were most sensitive to meeting the design criteria?

---

---

2 How did the design variables help to speed the iteration process?

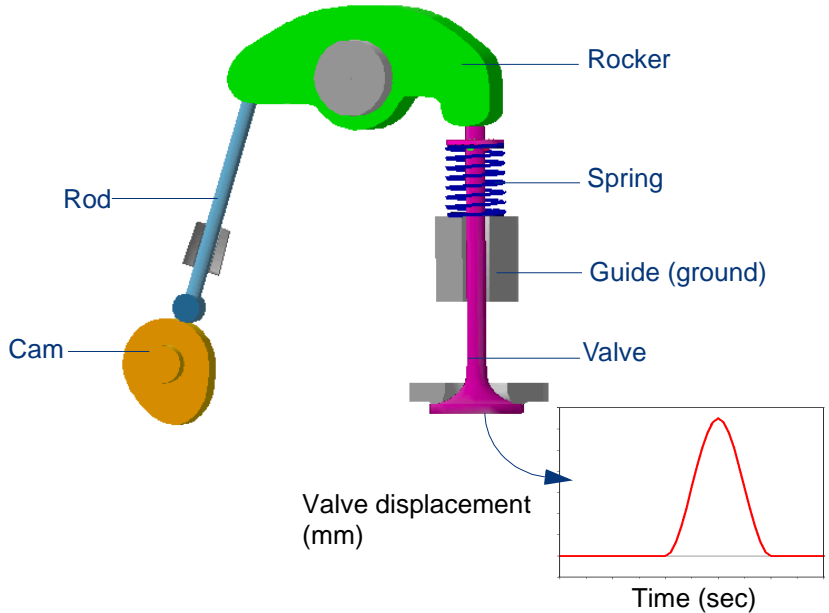
---

---



# 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, 300
- Curve Constraints, 301
- Automated Contact Forces, 302
- Flexible Parts—ADAMS/AutoFlex, 304
- Workshop 21—Cam-Rocker-Valve, 305
  - ◆ Module review, 317

## 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.

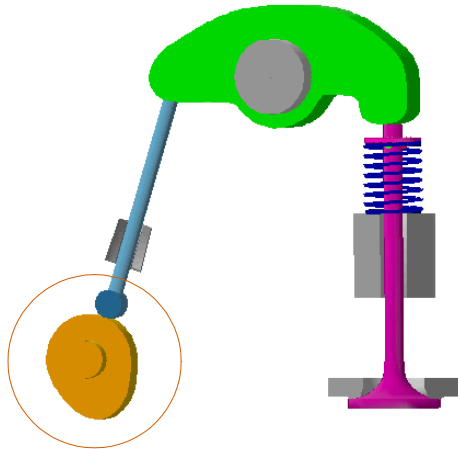
# Curve Constraints

## 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.

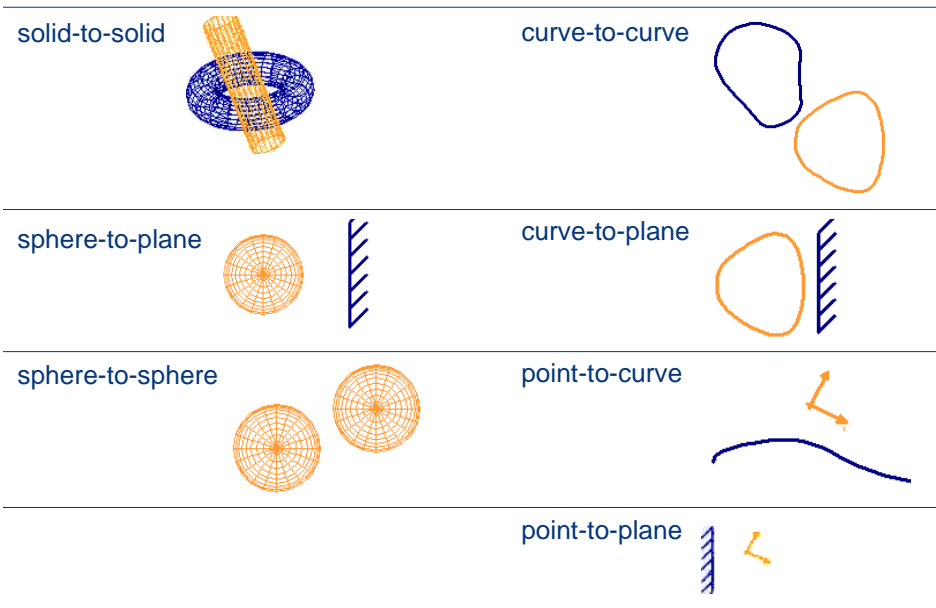
**See also:** [DOF removed by curve constraints](#), on page 376

# Automated Contact Forces

## 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



# 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.

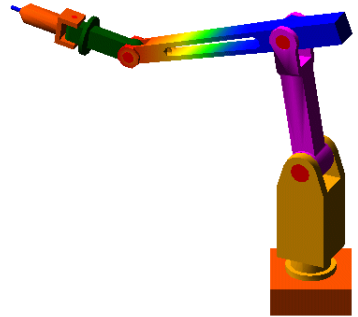
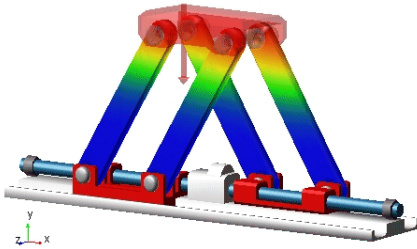
# Flexible Parts—ADAMS/AutoFlex

## 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

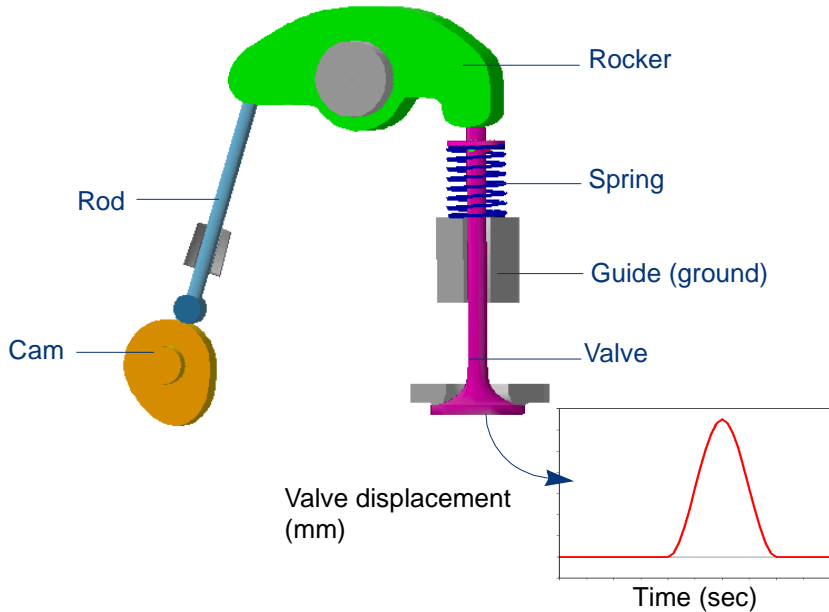


To run through a workshop, see the ADAMS/AutoFlex Training Guide. For more information, see the ADAMS/AutoFlex online help.

# 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.

# Workshop 21—Cam-Rocker-Valve...

## 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*.

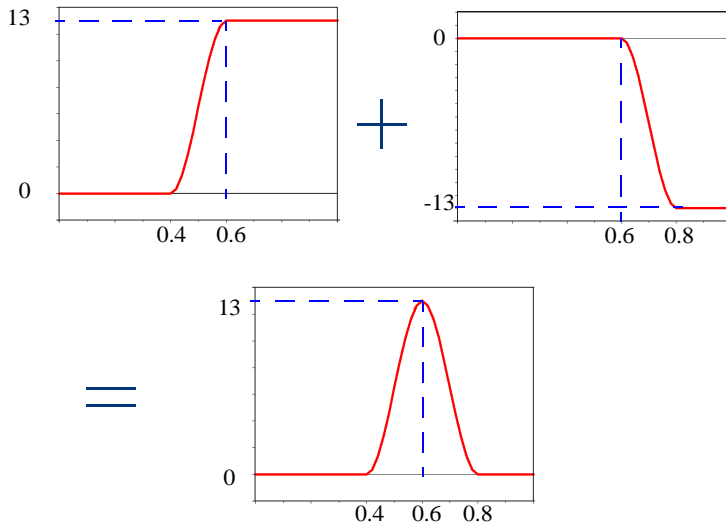
The file contains a model named *valve\_train*.

## 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:  $\text{STEP}(\text{time}, .4, 0, .6, 13) + \text{STEP}(\text{time}, .6, 0, .8, -13)$ .

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

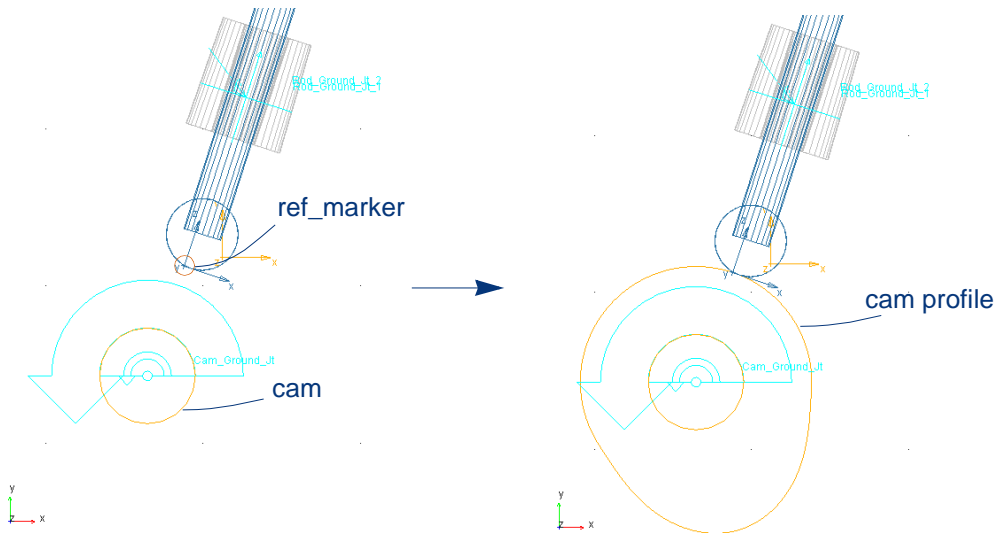
# Workshop 21—Cam-Rocker-Valve...

## 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.



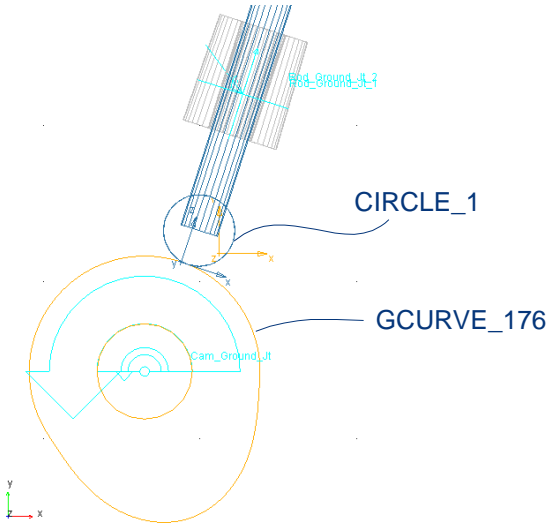
- 4 Run a simulation to verify that the **Rod** appears to move along the surface of the **Cam**.

# Workshop 21—Cam-Rocker-Valve...

## 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**.



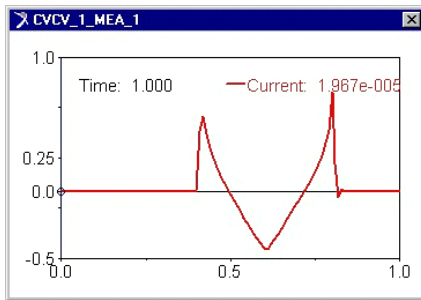
- 3 Run a simulation to verify that the new constraint works.

## 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.

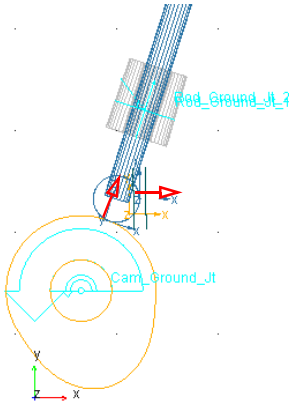
To replace the curve-on-curve constraint:

- 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 .

# Workshop 21—Cam-Rocker-Valve...

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 ( $\mu_s$ ): 0.08

## Workshop 21—Cam-Rocker-Valve...

---

- Dynamic Coefficient ( $\mu_d$ ): 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

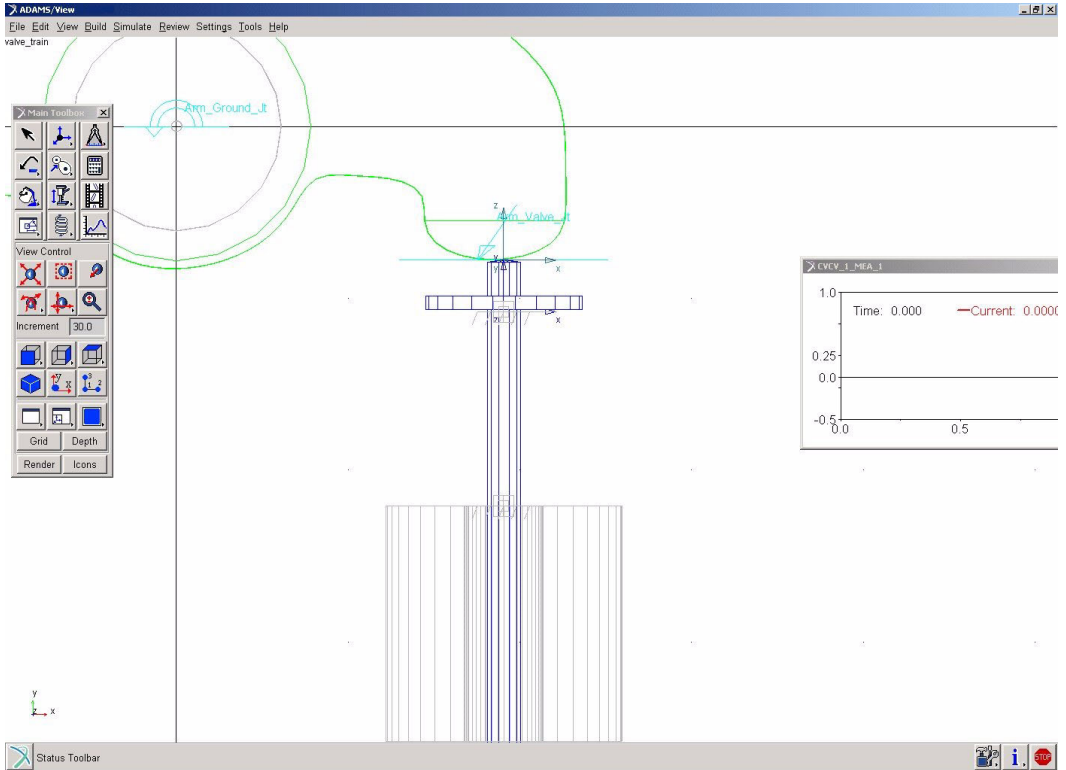
To prevent liftoff:

- 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**.

# Workshop 21—Cam-Rocker-Valve...

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.

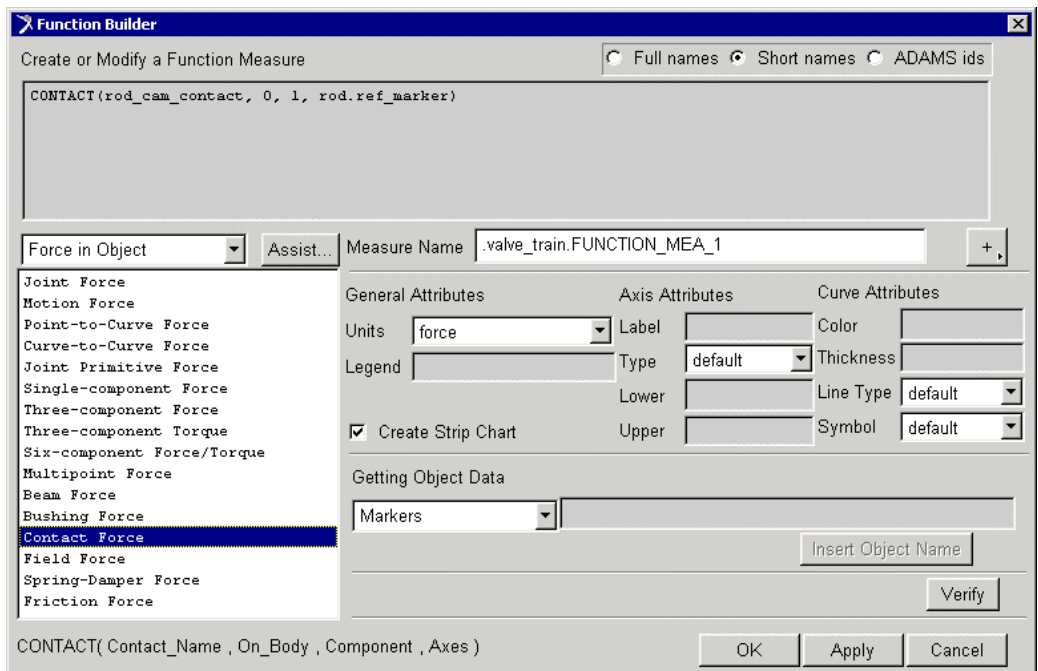
**Note:** You perform the static equilibrium to eliminate the transient effect that results from the time-dependent damping characteristic of the spring damper. In addition, positioning the model in static equilibrium establishes initial contact between the roller and the cam.

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

# Workshop 21—Cam-Rocker-Valve...

- 5 Modify the rotational motion on the cam to a speed of 3000 rpm. Enter the function as follows:  $-50 \cdot 360d \cdot \text{time}$ .
- 6 To view only one rotation of the cam, run a static equilibrium followed by a dynamic simulation for  $\text{end}=1/50$  seconds,  $\text{steps}=100$ . 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.

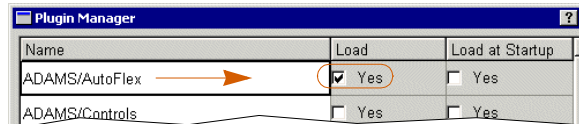
## Create and substitute the flexible part

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**.

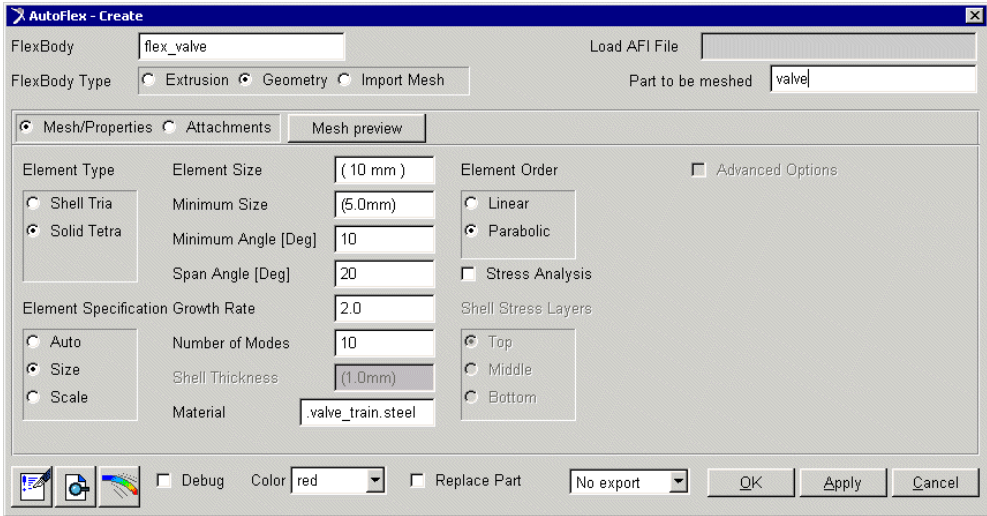


To create the mesh:

- 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

# Workshop 21—Cam-Rocker-Valve...

- 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.

# Workshop 21—Cam-Rocker-Valve...

## 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.

# Workshop 21—Cam-Rocker-Valve...

---

## 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?

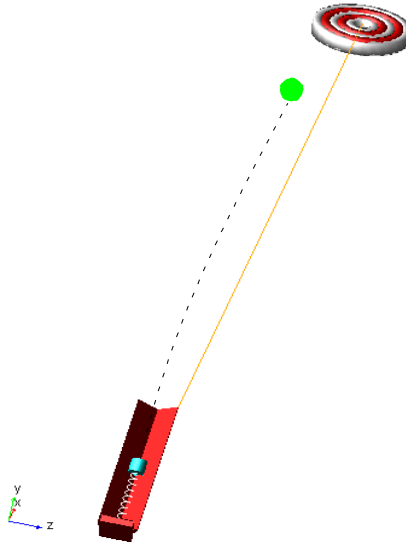
---

---



# 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, 320
- Design Studies, 322
- Workshop 22—Target Practice, 325
  - ◆ Module review, 336

# 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>I marker</i> and reaction force at floating <i>J marker</i> )
Vector components	Three translational
Orientations	Based on reference marker ( <i>R marker</i> )
Magnitudes ( $F_x, F_y, F_z$ )	User-defined

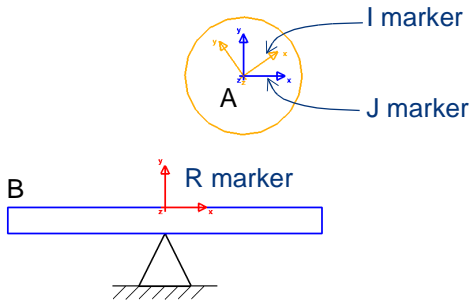
### Notes:

- 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

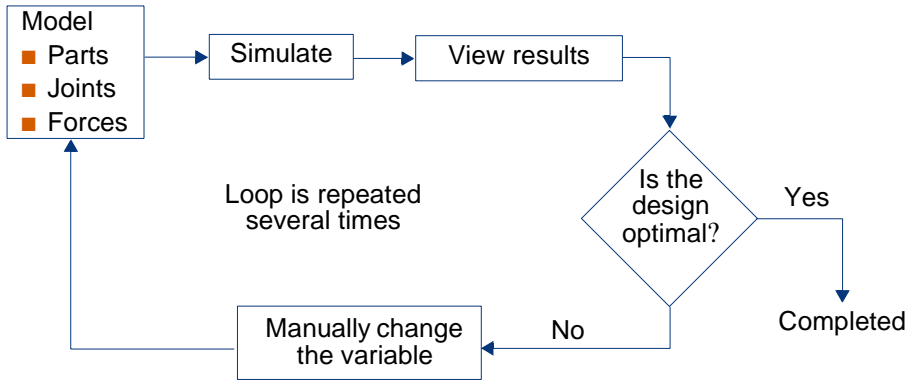
- Because the J marker belongs to part B, the force acts on part B when the bodies collide.
- Because the J marker moves with the I marker, part B knows where to apply the reaction force.

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

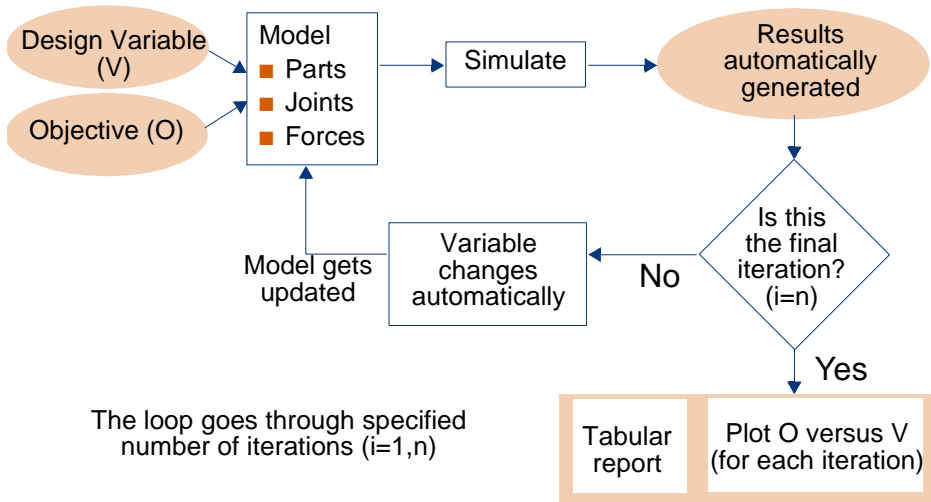
**See also:** Characteristics of a multi-component force, on page 375

# 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, $i$

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

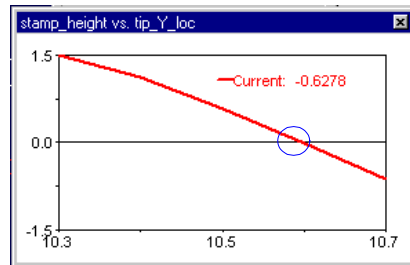
- Looking at Trial 4 ( $i = 4$ ):

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

$$S_4 = -6.0475$$

- $S_4$  is the approximate slope at Trial 4 ( $\text{tip\_y\_loc}=10.6$ ) in the plot.

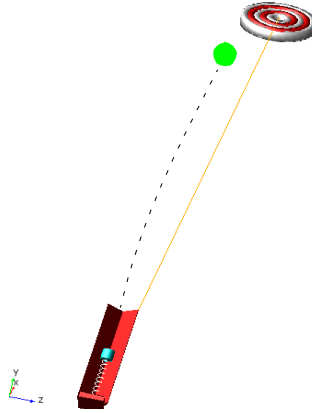
Design Study Summary			
Model Name : stamp			
Date Run : 15:48:55 23-Dec-98			
Objectives			
O1)	Minimum of stamp_height		
	Units	: inch	
	Maximum Value:	1.48945	(trial 1)
	Minimum Value:	-0.627838	(trial 5)
Design Variables			
V1)	tip_Y_loc		
	Units	: inch	
Trial	stamp_height	tip_Y_loc	Sensitivity
1	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`.

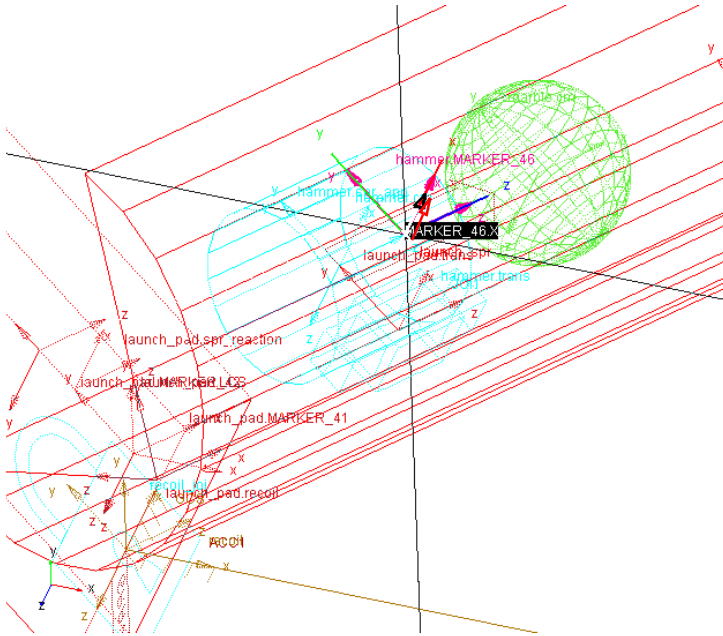
## Create a contact between the hammer and the marble

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.

# Workshop 22—Target Practice...

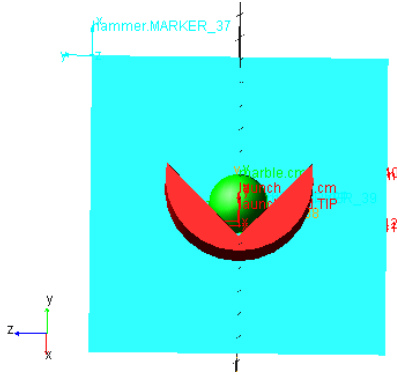
- 3 Orient the working grid along the XY-plane of the new marker, aligned with the face of the hammer (Settings → Working Grid)
  - **Set Location:** Pick (from your model, select the marker you just created, **MARKER\_46**)
  - **Set Orientation:** X-Y-Axes (first select the x- and then the y-axis of **MARKER\_46**, as prompted in the Status bar)




## Workshop 22—Target Practice...

- 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.



- 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.
- 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:

- From the **Settings** menu, point to **Solver**, and then select **Equilibrium**.
- Verify that **Error** is set to **1.0E-002**.
- Select **Close**.

# Workshop 22—Target Practice...


## 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

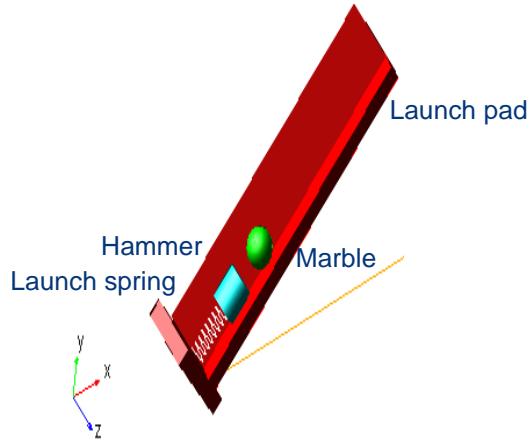
To create the contact:

- 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

# Workshop 22—Target Practice...

## 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:

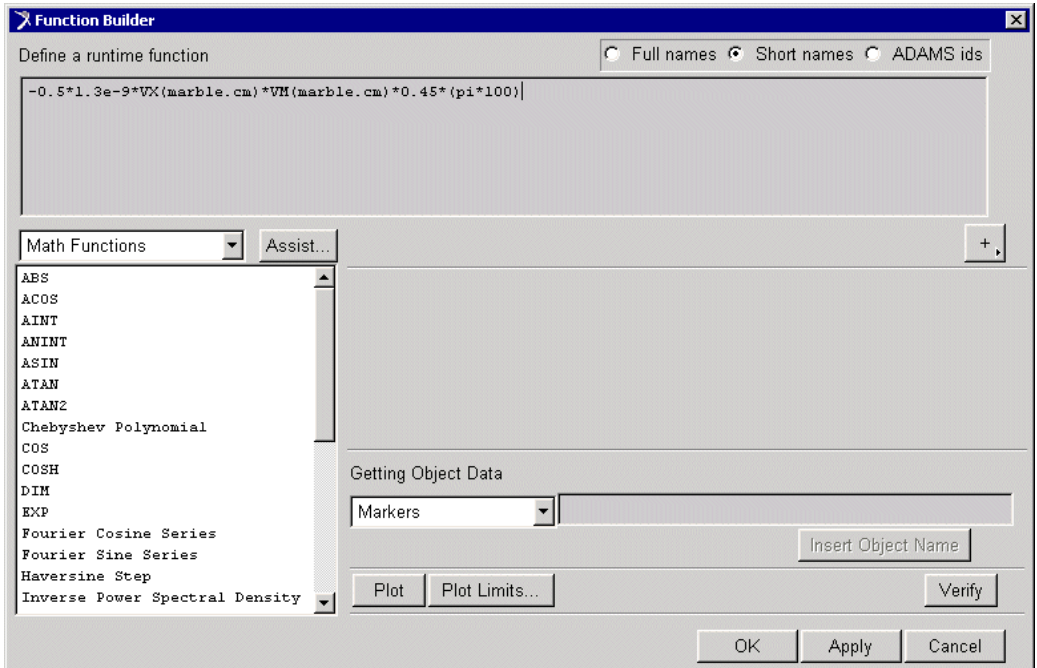
- ◆  $\rho = 1.3e-9 \text{ kg/mm}^3 = \text{density of air}$
- ◆  $V_x = \text{global x component of the marble's velocity}$
- ◆  $V_y = \text{global y component of the marble's velocity}$
- ◆  $V_m = \text{magnitude of the marble's velocity}$
- ◆  $C_d = 0.45 = \text{drag coefficient}$
- ◆  $A = \pi r^2 = \text{two-dimensional area of the marble face}$



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**.

## Workshop 22—Target Practice...

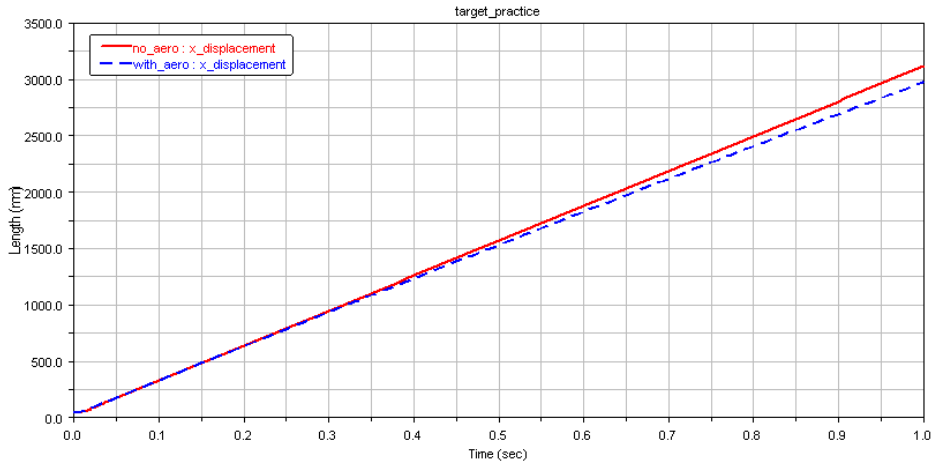
- 3 Enter the aerodynamic function as shown next:



- 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**.

# Workshop 22—Target Practice...

- 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:

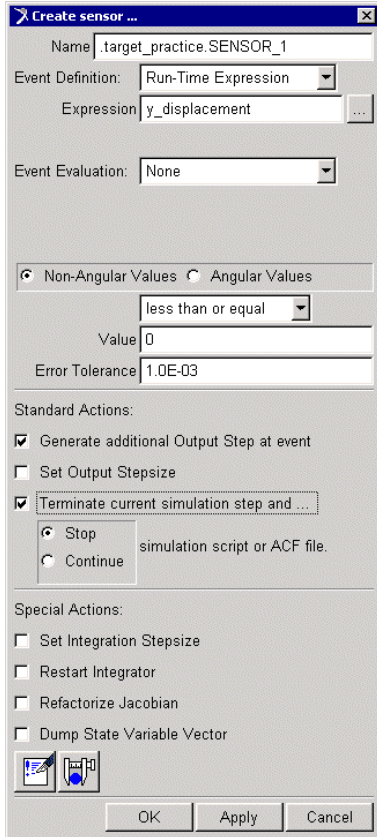
- 1 Create a measure, named **y\_displacement**, that tracks the global y displacement of the center of the marble.
  - Characteristic: CM position
  - Component: Y
  - From/At: ground

## Workshop 22—Target Practice...

- 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.

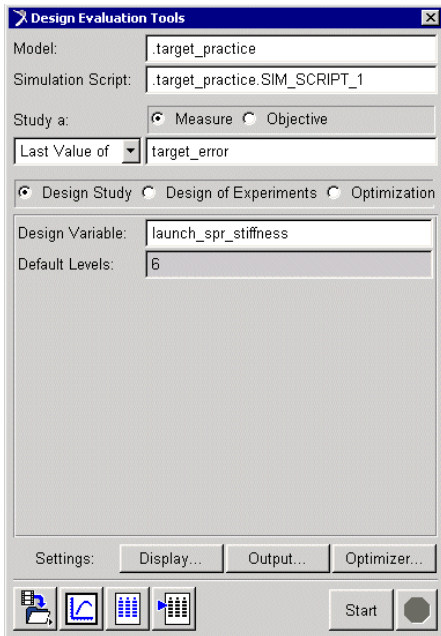


# Workshop 22—Target Practice...

## Set up a design study

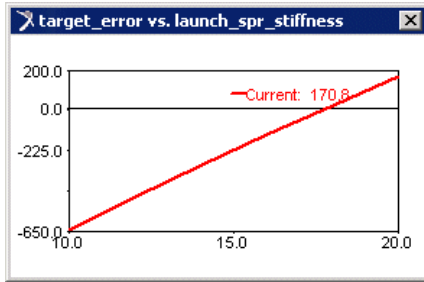
To set up a design study:

- 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:



# Workshop 22—Target Practice...

- 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?

---

---

# 23 RECOMMENDED PRACTICES

---

What are the recommended practices in debugging a model?

## What's in this module:

- General Approach to Modeling, 338
- Modeling Practices: Parts, 339
- Modeling Practices: Constraints, 340
- Modeling Practices: Compliant Connections, 341
- Modeling Practices: Run-time Functions, 342
- Debugging Tips, 344

# General Approach to Modeling

---

## Crawl-walk-run

- Try to understand the mechanism from a physical standpoint.
- Use building blocks of concepts that have worked in the past.
- Add enhancements to the model while testing periodically.
- Build kinematic models before building dynamic models.
- Use motions to check models before applying forces.
- Use motions which start with zero velocity.
- Verify enhancements to a complex model on a simpler model first.

# Modeling Practices: Parts

---

## Geometry associativity errors

Geometry may be added to the wrong part.

## Mass properties

- Using imported CAD-created geometry (IGES, STL, and so on) can yield inaccurate mass properties.
- Ensure inertia matrix is realistic.
- Use aggregate mass for a quick check of system mass and inertia.
- Use the Table Editor to do a quick check and potentially fix individual part masses and inertia.
- Small part mass and inertia lead to unrealistically high frequencies.

## Initial velocities

Check to see that part initial velocities are consistent (look in the .out file).

## Dummy parts

- Whenever possible, avoid using them.
- If absolutely needed, constrain all six DOF and assign a mass of 0.0 (not 1e-20).

## Design configuration

- Build a model close to assembled position.
- Build a model close to a stable equilibrium position, if possible.

# Modeling Practices: Constraints

---

## Fixed joints

- Not needed, since two or more parts can be combined or merged into a single part.
- An extra part with a fixed joint adds unnecessary equations to your system.
- When locking a part to ground, enormous torque may develop due to large moment arms.
- If absolutely needed, then add fixed joints at the center-of-mass (cm) location of lightest part.
- If locking a part to ground, consider assigning a very large mass/inertia to it so it can behave like ground.

**Note:** Whenever possible, avoid using fixed joints.

## Universal joints

When a universal joint is at  $90^\circ$ , you get a singular matrix.

## Motion

- Motion elements should only be functions of time.

**Note:** Avoid redundant constraints.

# Modeling Practices: Compliant Connections

---

## Spring dampers

- Ensure that the marker endpoints ( $DM(I,J)$ ) are never superimposed.
- Watch out for springs with very stiff spring constants.
- Watch out for springs with no damping.

## Bushings

Watch out for bushings with large rotations.

# Modeling Practices: Run-time Functions

## Function Builder

- Assists in building functions.
- Assists in function verification.
- Has function plot capability.

## Velocity

Make sure velocities are correct in force expressions. For example, in the damping function:  $-c*VX(i, j, j, \_)$ , the fourth term is missing.

## Splines

- Approximate forces with smooth, continuous splines.
- Extend the range of spline data beyond the range of need.
- Cubic splines (CUBSPL) work better on motions than Akima.
- Akima splines (AKISPL) work better on forces than Cubic.
- The Akima interpolation method is faster and can be defined as a surface, but its derivatives are generally discontinuous.

## IMPACTs/BISTOPs

- Do not use 1.0 for exponent on IMPACT or BISTOP functions.
- Models with IMPACTs/BISTOPs should have slight penetration in design position when doing statics.

## Measures

- Set up measures of your run-time functions.
- Set up measures of components of your run-time functions.
- Ensure that your function will not try to divide by zero.

# Modeling Practices: Run-time Functions

---

## Contacts

- Do not use 1.0 for exponent on IMPACT or BISTOP functions.
- Models with contacts should have slight penetration in design position when doing statics.

## Tires

- Models with tires should have slight penetration in model position when doing statics.
- If only rear tires penetrate, the static position could be a “handstand.”

## Units

- Use consistent units throughout the model (time, mass, stiffness, damping, and so on).
- Choose units (mass, force, time, and so on) that do not require using very large or very small numbers.
- Be wary when your model contains numbers like  $1e+23$  or  $1e-20$ .
- Use appropriate units—when modeling large models such as an aircraft landing on a runway, length units of millimeters may not be appropriate. Conversely, when modeling small models such as a power window switch (made up of small moving parts), using length units of meters may not be appropriate.
- Use reasonable time units—high frequencies may be better modeled with time units of milliseconds rather than seconds.

## Gravity

- Check magnitude and direction.
- Check for multiple gravity elements.

# Debugging Tips

---

## Model verify

- Lists number of moving parts, number of each type of constraint.
- Lists Gruebler's count and actual DOF count.
- Lists redundant constraints.
- Reports misaligned forces/force elements, joints, and so on.
- Helps identify and eliminate causes for input warning (don't ignore).

## Model topology

- Text or graphical model topology.
- Table Editor provides spreadsheet-like overview of model content.

## Icon feedback

Broken icon in design configuration probably means incorrectly defined joint or force.

## Table Editor

Convenient way to inspect and modify models (particularly large ones).

## Interactive simulation

By default, is turned on.

# Debugging Tips...

---

## Model display update

As ADAMS/Solver performs the simulation, you have the option to get immediate graphical feedback of the simulation at every:

- Output step
- Integration step
- Iteration

## Icons visible during simulation

This may help you monitor behavior of model components.

## Subroutines

- Check for their existence.
- While debugging a model, eliminate user subroutines so that they are not the source of the error.

## Gravity

Turning gravity off can accentuate modeling errors and make debugging easier.

## Statics

- When applicable, perform an initial static first.
- If static solution fails:
  - ◆ Turn on **Model display update = at every iteration** to provide additional insight.
  - ◆ Identify and eliminate the undesired static configuration—there could be more than one static configuration and ADAMS/Solver could be finding the undesired one.
- Check to see if there are any floating parts.
- Check the signs of applied forces.
- Experiment with *Alimit/Tlimit/Maxit/Stability*.
- Check if impacts are initially in contact; if not, they should be.
- Running an initial dynamic simulation can help you determine why the model is not finding static equilibrium.

# Debugging Tips...

---

## Dynamics

- If integrator fails to **start-up**:
  - ◆ Check sign and magnitude of forces.
  - ◆ Look at accelerations to understand what is happening.
  - ◆ Perform initial static analysis first.
  - ◆ Try a quasi-static simulation.
  - ◆ Try changing integrator parameter - HINIT.
  - ◆ Try a different integrator.
- If integrator fails in the **middle of a simulation**:
  - ◆ Look at animation and plots until failure, to understand simulation.
  - ◆ Decrease integrator parameter - HMAX.
  - ◆ Do not let the integrator step over important events.
  - ◆ Short duration events, such as an impulse can be captured by setting the maximum time step, HMAX, to a value less than the impulse width.
  - ◆ Use HMAX so ADAMS/Solver acts as a fixed-step integrator
  - ◆ Decrease error.
  - ◆ Try a different integrator.
- If integrator takes **very small steps**:
  - ◆ Look for sudden changes in force and motion input.
  - ◆ Rescale model to get more uniform numbers.

## Velocities at time=0

Check initial velocities using the .out file.

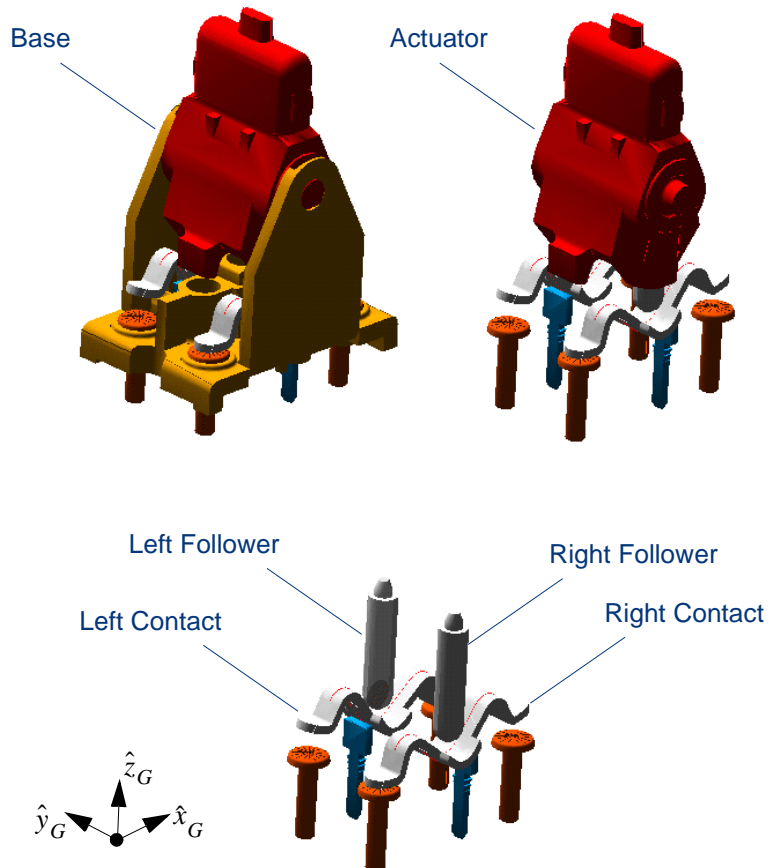


# 24 SWITCH MECHANISM WORKSHOP

## Problem statement

Determine the minimum force necessary to toggle the switch mechanism to the forward and rearward directions.

### Switch Mechanism



## Model description

The given switch model contains parts with shell geometry and mass properties:

<b>Part name:</b>	<b>Geometry (shells):</b>	<b>Mass (grams):</b>
base (ground)	base_main_geo	(ground)
	base_right_front_geo	
	base_right_mid_geo	
	base_right_rear_geo	
	base_left_front_geo	
	base_left_mid_geo	
	base_left_rear_geo	
actuator	actuator_geo	1.52
right_contact	right_contact_geo	0.51
left_contact	left_contact_geo	0.51
right_follower	right_follower_geo	0.1076
left_follower	left_follower_geo	0.1076

# Switch Mechanism Workshop...

The switch model contains construction points for adding the necessary modeling elements to address the problem statement. They are:

<b>Point:</b>	<b>Description:</b>
POINT_1	Actuator to base pivot location
POINT_2	right_follower to actuator spring lower location
POINT_3	left_follower to actuator spring lower location
POINT_4	right_follower to actuator spring upper location
POINT_5	left_follower to actuator spring upper location
POINT_6	Contains z-coordinate of base contact plane with left_contact and right_contact at four corners
POINT_7	left_contact to base idealized constraint location
POINT_8	right_contact to base idealized constraint location
POINT_9	Location of base rear contact surface with actuator
POINT_10	Location of base front contact surface with actuator
POINT_11	Location of actuator rear contact point with base
POINT_12	Location of actuator front contact point with base
POINT_13	Location of right_contact to base mid-contact point
POINT_14	Location of left_contact to base mid-contact point
POINT_15	Location of force application

The switch model is mounted such that the model's global negative x-axis defines forward and positive z-axis defines up (see [Problem statement](#), on page 349).

## Section I: Test the right half with constraints only

You can think of the switch mechanism as consisting of two halves. In this exercise, first constrain the right half of the mechanism and perform a kinematic simulation to visually verify correct motion. Then, add spring and contact forces to the right half to ensure that the mechanism actually toggles. Then, add more detail to the right half, introduce the left half, and then finally perform a system-level simulation.

This section emphasizes the crawl-walk-run method. In this section, you will crawl.

### To import the model:

- 1 Start ADAMS/View from the directory *exercise\_dir/switch\_workshop*.
- 2 From the directory *exercise\_dir/switch\_workshop*, import the model command file *switch\_start\_new.cmd*.

This file contains commands to build a model named *switch*.

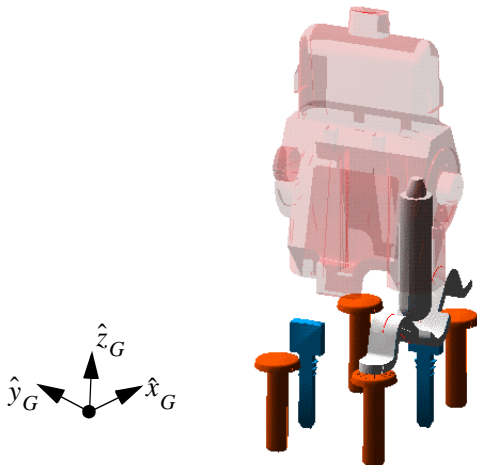
### To set up the modeling environment:

- 1 Turn the working grid off.
- 2 To modify the preset views in ADAMS/View so that they are relevant to the global coordinate system, from the **Tools** menu, select **Command Navigator**.  
The Command Navigator appears.
- 3 In the Command Navigator, from the **View** menu, select **Management**, and then select **Orient**.  
As mentioned previously, the switch model is mounted such that the model's global negative x-axis defines forward and positive z-axis defines up.
- 4 Preset the front view. All other views will be defined based on this front view. Ensure that the preset views (front, top, and so on) are correctly defined based on the global coordinate system.
- 5 To set the transparency of the actuator part to 80%, right-click the actuator, and then select **Appearance**.
- 6 Turn off the visibility of the base part geometry, *base\_main\_geo*.
- 7 Deactivate the *left\_contact* and *left\_follower* parts.
- 8 Turn off the visibility of the *left\_contact* and *left\_follower* parts.

# Switch Mechanism Workshop...

- 9 Set gravity in the global negative-z direction.

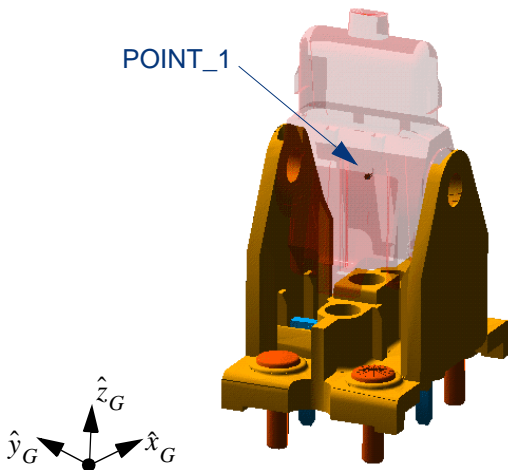
The model view should look as follows:



To add constraints:

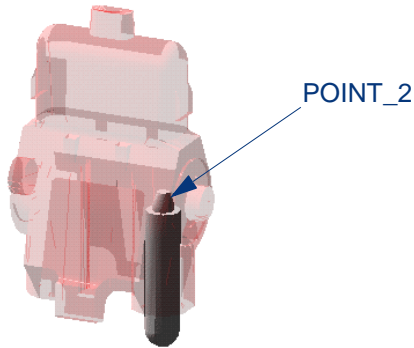
- 1 Constrain the **actuator** to the **base** at **POINT\_1** such that the only relative allowable degree of freedom is rotation about  $\hat{y}_G$ .

Establish a reference marker with global orientation on the base (ground) part that makes picking of global direction vectors easy. Setting the color and size of the marker helps in referencing it later.

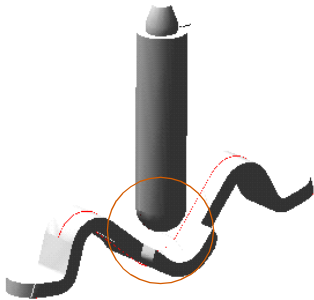


## Switch Mechanism Workshop...

- 2 Constrain the **right\_follower** to the **actuator** at **POINT\_2** such that the only relative allowable degree of freedom is translation along  $\hat{z}_G$ .



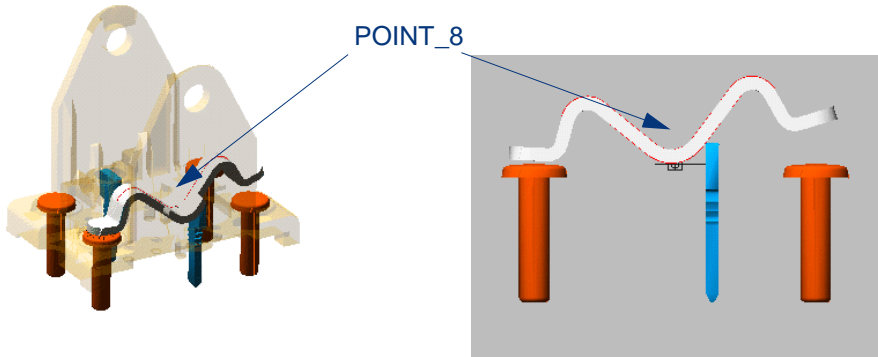
- 3 Constrain the tip of the **right\_follower** to the upper curve on the **right\_contact** part.



When creating the curve-to-curve constraint, select the red circle, **right\_follower.right\_follower\_circle\_geo**, at the tip of the **right\_follower** part, parallel to the global-xz plane, and then select **right\_contact\_upper\_bspline**.

# Switch Mechanism Workshop...

- 4 Constrain the **right\_contact** part to the **base** at **POINT\_8** such that the only relative allowable degree of freedom is translation along  $\hat{y}_G$ .



This might not seem intuitive, but it ensures that there are no redundant constraints in the model. It is a good modeling practice to remove all redundant constraints in your system prior to performing a simulation.

- 5 Add displacement **joint motion** to the actuator-to-base revolute joint such that the actuator oscillates sinusoidally with an amplitude of 15.1 degrees and one cycle per second.

## Test the model:

- 1 Verify the model.

Your system should have 0 degrees of freedom and no redundant constraints at this configuration. If it does not, inspect the model to determine the discrepancy.

- 2 Simulate the model kinematically to visually verify correct motion, using an end time of 1 second with 100 output steps.
- 3 Save your work.

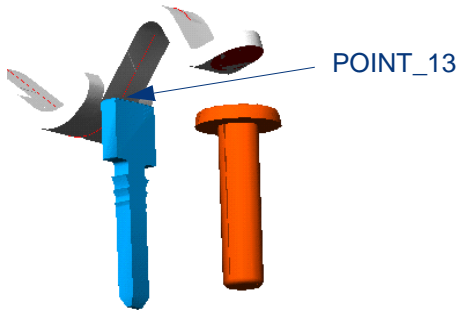
## Section II: Test the right half with front and rear contacts

Change the constraints on the `right_contact` part so that it can rotate and make contact with the right front and rear terminals on the base part (it will rock back and forth like a see-saw). Use the curve-to-curve constraint created earlier.

In this section you will start to walk.

To add detail to the connections between the `right_contact` and base parts:

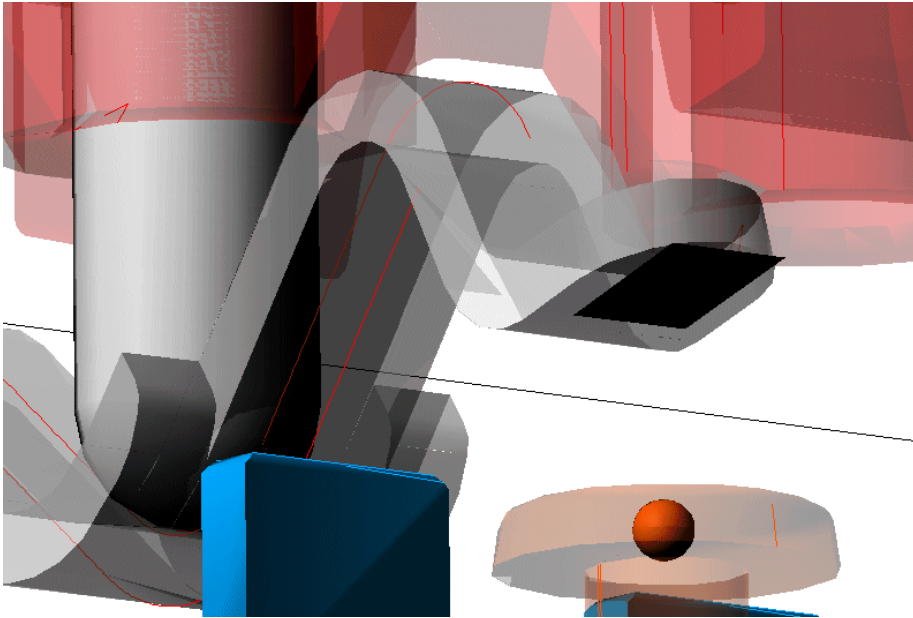
- 1 Remove the translational joint constraining the `right_contact` to the base at `POINT_8`.
- 2 Constrain the `right_contact` to the base at `POINT_13` such that the only allowable degree of freedom is rotation about  $\hat{y}_G$ .



## Switch Mechanism Workshop...

- 3 Create a sphere-to-plane contact force between the front end of the **right\_contact** part and the front right corner of the **base** part. Use **right\_contact.PLANE\_72** and **base.ELLIPSOID**.

Using the ellipsoid and plane geometries will improve run time when solving.



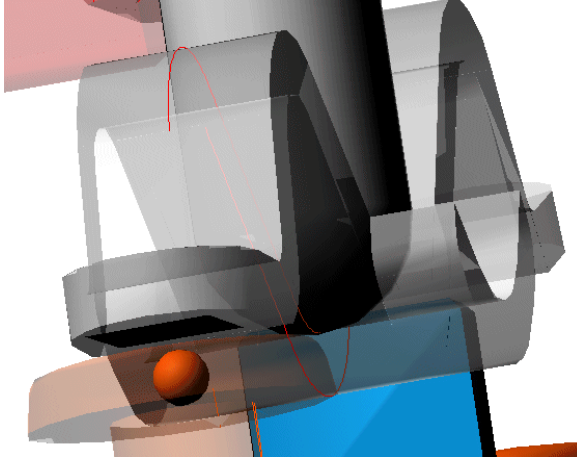
The contact parameter should be:

- Stiffness: 1e5 (milliNewton/mm)
- Force exponent: 2.2
- Damping: 1e2 (milliNewton-sec/mm)
- Penetration depth: 1e-3 mm
- Static friction: off
- Dynamic friction: off

# Switch Mechanism Workshop...

- 4 Create a sphere-to-plane contact force between the rear end of the **right\_contact** part and rear right corner of the **base** part. Use **right\_contact.PLANE\_128** and **base.ELLIPSOID\_73**.

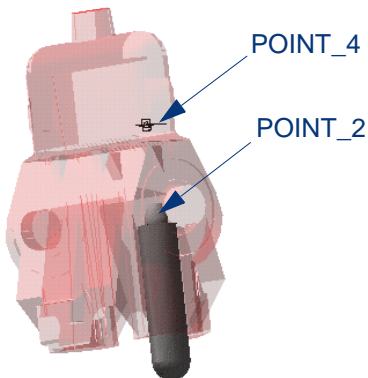
**Note:** Use the same contact parameters as in the sphere-to-plane force.



To add spring force to the right half:

- 1 Create a spring between the **right\_follower** at **POINT\_2** and **actuator** at **POINT\_4** using the following parameters:

- Stiffness: 600 (milliNewton/mm)
- Damping: 0.1 (milliNewton-sec/mm)
- Free length: 9 mm

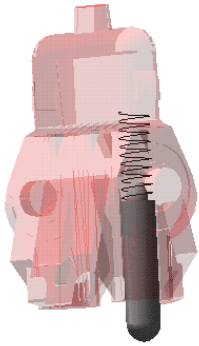


You need markers to create the spring. First create markers for each endpoint belonging to the appropriate parts.

# Switch Mechanism Workshop...

2 Override default spring geometry by using these custom parameters:

- Coil count: 10
- Diameter of spring: 2.5 mm
- Damper diameter at ij: 0, 0
- Tip length at ij: 0, 0
- Cup length at ij: 0, 0



With nothing selected, from the **Edit** menu, select **Modify**.

Filter on geometry, then double-click **SPRING\_1**, then select **spring\_graphic** (not **damper\_graphic**). To make it stand out, change the color to white.

To test the model:

- Verify the model.

The system should now have one degree of freedom and one redundant constraint.

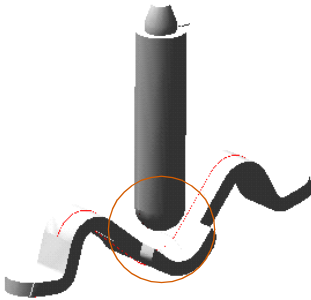
At this time, does the redundant constraint affect what you are doing?

# Switch Mechanism Workshop...


To replace the `right_follower` to `right_contact` curve-to-curve constraint with a force:

- 1 Remove the **curve-to-curve constraint** between the tip of the `right_follower` and the upper curve on the `right_contact` part.
- 2 Create a **curve-to-curve contact force** between the tip of the `right_follower` and the upper curve on the `right_contact` part. Use the same curves used in Step 3 on page 354 and the same contact parameters from Step 3 on page 357.

**Note:** After you fill in the I and J Curve text boxes, press Enter, in each text box, to activate the I and J Direction(s) text boxes.



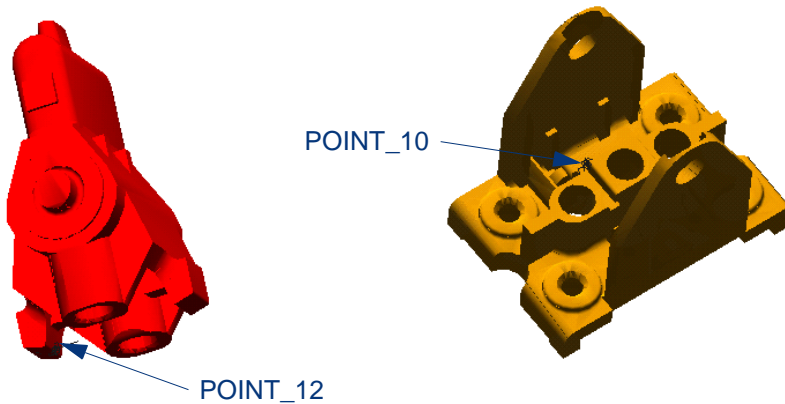
To test the model:

- 1 Verify the model.  
Your system should have two degrees of freedom and no redundant constraints.
- 2 Simulate the model:
  - Set **Update Graphics** to **Never**.
  - Turn on **Debug/EPRINT** so you can monitor the ADAMS/Solver performance.  
(Main Toolbox →  → from the bottom of the Main Toolbox, change NoDebug to EPrint using the pull-down menu)
  - Perform a **1-second, 200-step** dynamic simulation.
- 3 Animate the results to visually verify the correct motion.

# Switch Mechanism Workshop...

To create stops (forces) between the actuator and base:

- 1 Create a **sphere-to-plane** contact force between the front end of the **actuator** and the **base** part. As the actuator rotates, its sphere strikes a surface parallel to the global-*yz* plane on the base. Use the following parameters:
  - **Sphere**: sphere on the actuator part at POINT\_12 with a radius of 0.5 mm
  - **Plane**: parallel to global *yz*-plane at POINT\_10
  - **Contact parameters**: same as in Step 3 on page 357



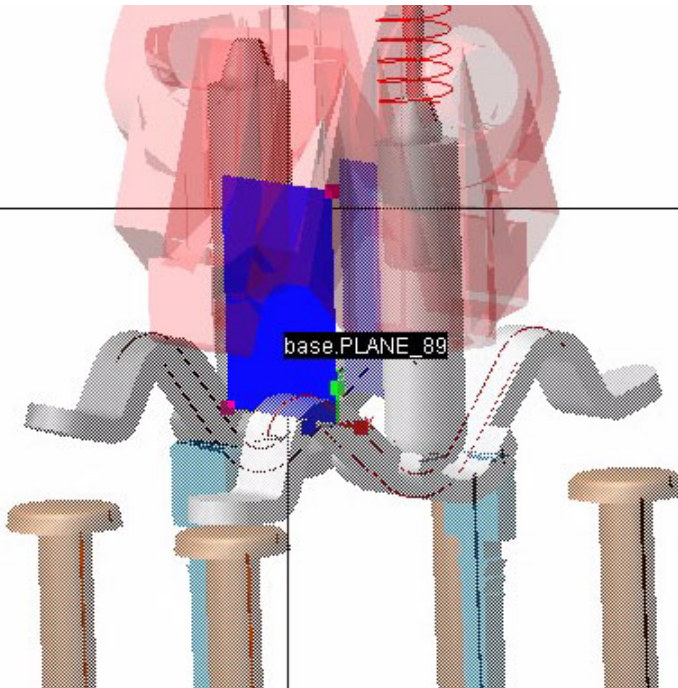
First create a sphere on the actuator part at POINT\_12, with a radius of 0.5mm. Use this sphere in the sphere-to-plane contact force.

# Switch Mechanism Workshop...

Then create a plane on the base part at POINT\_10.

**Note:** To create this plane, you will need to relocate and reorient the grid. Reset the location to be POINT\_10 with the orientation set to the global yz-plane. You may also need to decrease the size of the working grid (for example, size = 20 mm). Make sure that the points you snap to when creating the plane are on the working grid and not on the part geometry.

The plane should look as shown next.

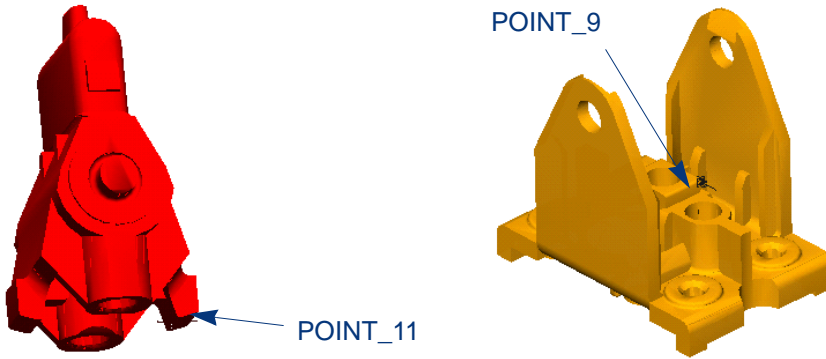


Again, use the same contact parameters as used in Step 3 on page 357.

- 2 After you've created the contact, make the plane and ellipsoid transparent.

## Switch Mechanism Workshop...

- 3 Create a sphere-to-plane contact force between the rear end of the actuator and the base part. Use the following parameters:
  - **Sphere:** sphere on the actuator part at POINT\_11 with a radius of 0.5 mm
  - **Plane:** parallel to global yz-plane at POINT\_9
  - **Contact parameters:** same as in Step 3 on page 357



First create a sphere on the **actuator** part at **POINT\_11**, with a radius of 0.5mm. Use this sphere in the sphere-to-plane contact force.

Then, create a **plane** on the base part at **POINT\_9**, parallel to the global yz-plane.

**Note:** To create this plane, you will need to relocate and reorient the grid. Reset the location to be **POINT\_9** with the orientation set to the global yz-plane. You may also need to decrease the size of the working grid (for example, size = 20 mm). Make sure that the points you snap to when creating the plane are on the working grid and not on the part geometry.

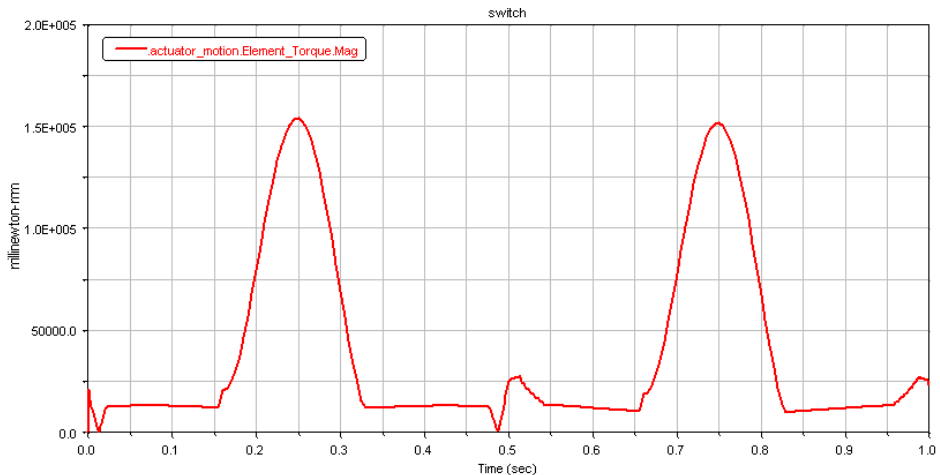
You will need to rotate the plane 180° such that the z-axis of the geometry anchor marker points toward the actuator.

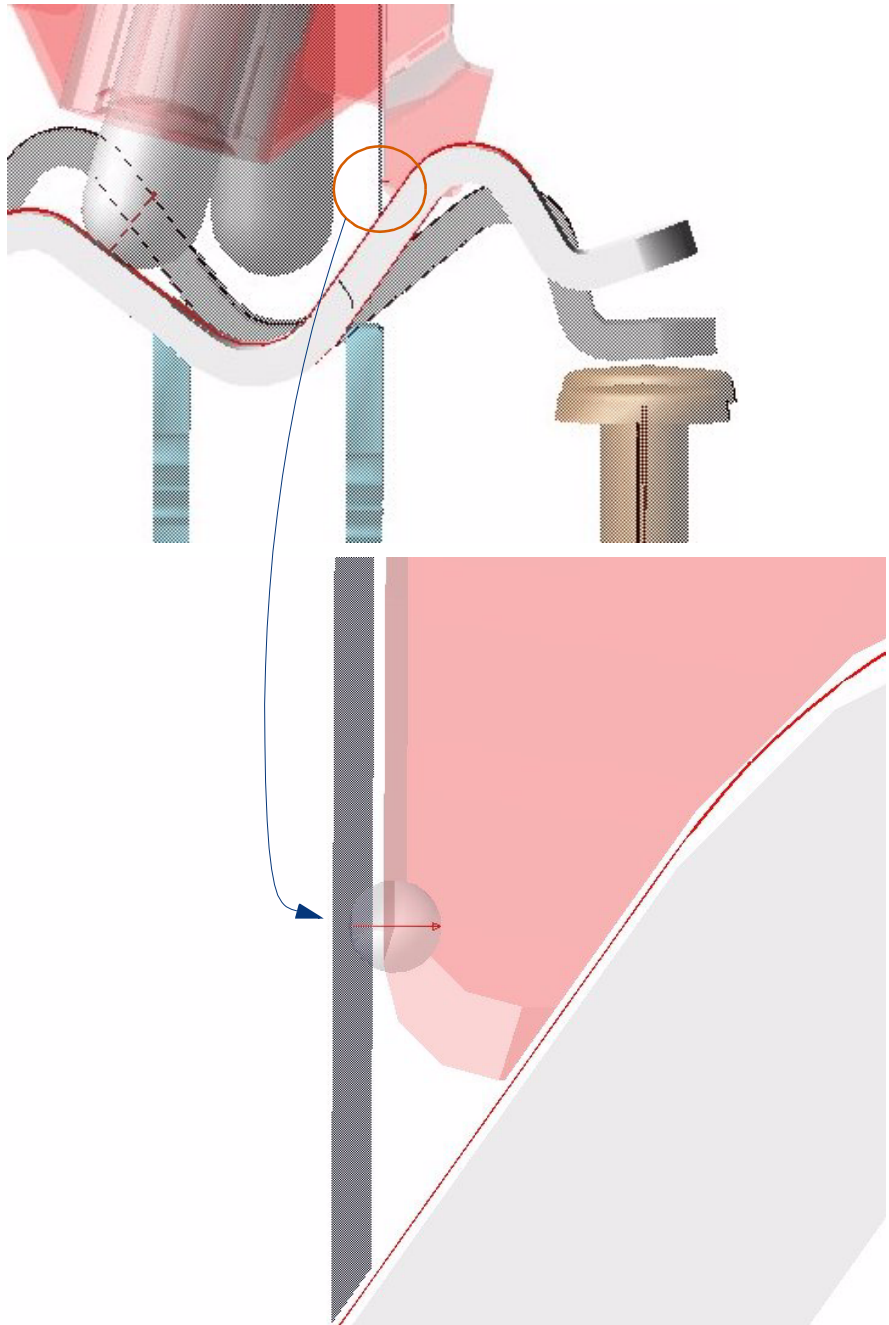
- 4 After you've created the contact, make the plane and ellipsoid transparent.

# Switch Mechanism Workshop...

To test the model using the actuator motion input:

- 1 Verify the model.  
Your system should have two degrees of freedom and no redundant constraints.
- 2 Simulate the model to visually verify correct motion.
  - ◆ Use the **GSTIFF, SI2 Formulation**.
  - Tip:** From the **Settings** menu, select **Solver**, then select **Dynamics**.
  - ◆ Perform a static simulation followed by **1-second, 200-step** dynamic simulation.
- 3 Plot the magnitude of the element force for the two contacts you just created. If these stop forces are not returning a nonzero value, inspect the model further.
- 4 In ADAMS/PostProcessor, plot the torque at the actuator's revolute joint (due to the motion input) versus time. On the same page, animate the model. Can you explain the shape of the curve? Is this intuitive?



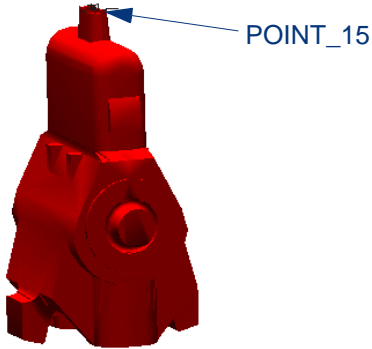


# Switch Mechanism Workshop...

To create force application:

- 1 Remove the **motion** applied to the revolute joint constraining the **actuator** to the **base**.
- 2 Apply a **force** to the actuator part at **POINT\_15** in the positive  $\hat{x}_G$  direction, moving with the body. Use the following function:

$$f(t) = -200 * \text{time}$$



To test the model:

- 1 Verify the model.  
Your system should have three degrees of freedom and no redundant constraints.
- 2 Create a function measure named **contact\_force**, based on the force magnitude of the right rear contact force between the **right\_contact** part and the **base** part.
- 3 Create a sensor that triggers when the force magnitude of the right rear contact force (measured in the above step) is greater than or equal to 1mN within a tolerance of 1e-3 mN.

When sensed, ADAMS/Solver should terminate the current simulation step and continue the simulation script.

Use the Function Builder to assist in referencing the expression you are monitoring. In the Function Builder, get object data for measures, select **Browse**, and then select **contact\_force**. Then insert the object name into the text box of the Function Builder.

# Switch Mechanism Workshop...

Remember, the force applied to the switch is a function of time. Before you run the simulation, you do not know how much force needs to be applied to toggle the switch; therefore, you do not know how long to simulate. For that reason, you create the sensor. You will purposely simulate for a larger amount of time than is needed, letting the sensor stop the simulation when the switch has been toggled.

- 4 Simulate the model to visually verify correct rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
INTEGRATOR/SI2,GSTIFF
EQUIL/STATIC,ALIMIT=0.1d,TLIMIT=1,MAXIT=50
SIMULATE/STATICS
SIMULATE/DYNAMIC, END=10.0, DTOUT=.01
DEACTIVATE/SENSOR, ID=<your right rear sensor id #>
SIMULATE/DYNAMIC, DURATION=0.5, DTOUT=.01
```

By using this simulation script, the model will simulate until the switch is toggled (assuming it toggles before 10 seconds), at which time the sensor is deactivated and the model simulates an additional 0.5 seconds to review follow-on transient behavior.

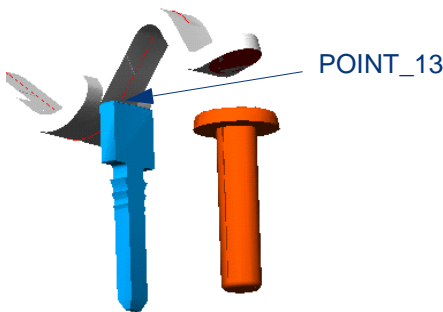
- 5 Save your work.

## Section III: Refine the right half of the mechanism

Replace the pivoting constraint at POINT\_13 (the lower\_contact to base revolute joint) with a more realistic connection that accounts for dynamic phenomena like sliding and liftoff.

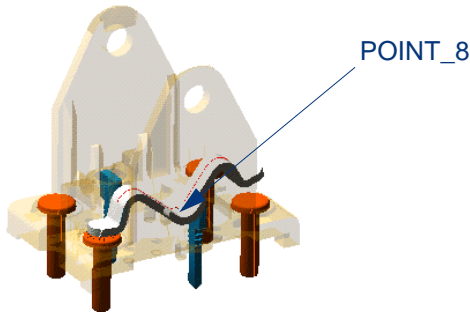
To refine right\_contact connections:

- 1 Remove the revolute joint constraining the right\_contact to the base at POINT\_13.



## Switch Mechanism Workshop...

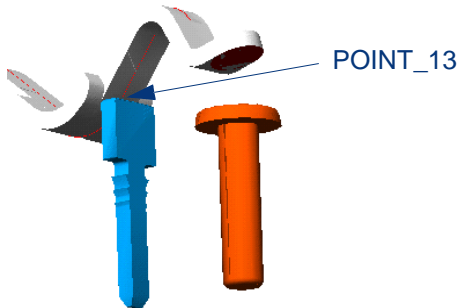
- 2 Constrain the **right\_contact** to the **base** at **POINT\_8** such that the only allowable degrees of freedom are translation along  $\hat{z}_G$  and rotation about  $\hat{y}_G$ .



This involves creating two joint primitives (inline and parallel).

You must ensure that the J marker of each primitive belongs to the base part, and not to the **right\_contact** part. This will absolutely affect the simulation. See the instructor if you do not fully understand this concept.

- 3 Create a **point-to-curve contact force** between the underside on the **right\_contact** part and the mid-contact point, **POINT\_13**, on the **base**. Use the same contact parameters as in Step 3 on page 357. However, set the stiffness to **1e8 N/mm**.



First create a marker on the base part at **POINT\_13**. Use this marker as the point marker in the point-to-curve contact force.

For the curve, use **right\_contact.right\_contact\_lower\_bspline**.

# Switch Mechanism Workshop...

---

To test the model:

- 1 Verify the model.

Your system should have four degrees of freedom and no redundant constraints.

- 2 Simulate the model to visually verify correct rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
INTEGRATOR/SI2,GSTIFF
EQUIL/STATIC,ALIMIT=1d,TLIMIT=1,MAXIT=50
SIMULATE/STATICS
SIMULATE/DYNAMIC, END=10.0, DTOUT=.01
DEACTIVATE/SENSOR, ID=<your right rear sensor id #>
SIMULATE/DYNAMIC, DURATION=0.5, DTOUT=.01
```

Note the force at which the switch toggles to the rearward direction when accounting for only the `right_follower` and `right_contact` parts and corresponding connections.

- 3 Save your work now.

In Sections IV and V you will incorporate the left half of the switch and add friction.

## Section IV: Add the left half

Since the right half of the switch mechanism is working properly at this time, apply the same steps to the left half of the mechanism. If desired, you can use a different crawl-walk-run method to connect parts in the left half.

Below is a copy of the key locations that will help you define the left half connections:

<b>Point:</b>	<b>Description:</b>
POINT_1	Actuator to base pivot location
POINT_2	right_follower to actuator spring lower location
POINT_3	left_follower to actuator spring lower location
POINT_4	right_follower to actuator spring upper location
POINT_5	left_follower to actuator spring upper location
POINT_6	Contains z-coordinate of base contact plane with left_contact and right_contact at four corners
POINT_7	left_contact to base idealized constraint location
POINT_8	right_contact to base idealized constraint location
POINT_9	Location of base rear contact surface with actuator
POINT_10	Location of base front contact surface with actuator
POINT_11	Location of actuator rear contact point with base
POINT_12	Location of actuator front contact point with base
POINT_13	Location of right_contact to base mid-contact point
POINT_14	Location of left_contact to base mid-contact point
POINT_15	Location of force application

To connect the left\_follower and left\_contact:

- Reintroduce the left\_follower and left\_contact parts by reactivating them and ultimately connect these parts to the switch mechanism the same way that you did for the right\_follower and right\_contact parts.

# Switch Mechanism Workshop...

To test the model:

- 1 Verify the model.

Your system should now have seven degrees of freedom and no redundant constraints.

- 2 In the **Message Window**, select **Settings**.

- 3 Change **Display messages only at or above severity level** from **Warning** to **Error**.

This prevents the Message Window from opening each time you verify or simulate due to the difference in orientation of the markers and curves used to define the pt\_cv contact.

- 4 Simulate the model to determine rearward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
INTEGRATOR/SI2,GSTIFF
EQUIL/STATIC,ALIMIT=1d,TLIMIT=1,MAXIT=50
SIMULATE/STATICS
SIMULATE/DYNAMIC, END=10.0, DTOUT=.01
DEACTIVATE/SENSOR, ID=<your right rear sensor id #>
SIMULATE/DYNAMIC, DURATION=0.5, DTOUT=.01
```

- 5 Note the force at which the switch toggles to the rearward direction now, when accounting for both halves of the mechanism?

- 6 Simulate the model to determine rearward toggle motion, and then switch to forward toggle motion using a simulation script based on the following ADAMS/Solver commands:

```
INTEGRATOR/SI2,GSTIFF
EQUIL/STATIC,ALIMIT=1d,TLIMIT=1,MAXIT=50
SIMULATE/STATICS
SIMULATE/DYNAMIC, END=10.0, DTOUT=.01
DEACTIVATE/SENSOR, ID=<your right rear sensor id #>
SIMULATE/DYNAMIC, DURATION=0.1, DTOUT=.01
SFORCE/<original input sforce id #> , FUNCTION=100*time
SIMULATE/DYNAMIC, DURATION=10.0, DTOUT=.01
```

Notice how you are reversing the input force applied to the actuator part through an ADAMS/Solver command in the simulation script, as opposed to in the model's design configuration. You can modify a force on-the-fly.

- 7 Note the force at which the switch toggles to the forward direction when accounting for both halves of the mechanism?
- 8 Save your work.

## Section V: Refine the switch

You will now refine your model to account for friction.

To add friction to the curve-to-curve contact forces:

- 1 Modify the curve-to-curve contact force between the tip of the **right\_follower** and the upper curve on the **right\_contact** part such that static and dynamic friction is accounted for. Use the following default parameters for contact friction:
  - **Static Coefficient:** 0.3
  - **Dynamic Friction Coefficient:** .1
  - **Stiction Transition Vel.:** 1 mm/sec
  - **Friction Transition Vel.:** 10 mm/sec
- 2 Modify the curve-to-curve contact force between the tip of the **left\_follower** and the upper curve on the **left\_contact** part such that static and dynamic friction is accounted for. Use the same contact array you used in the previous step.

To test the model:

- 1 Verify the model.

Your system should still have seven degrees of freedom and no redundant constraints.
- 2 Simulate the model to visually verify correct rearward toggle and forward motion using a simulation script based on the following ADAMS/Solver commands:

```
INTEGRATOR/SI2,GSTIFF
EQUIL/STATIC,ALIMIT=1d,TLIMIT=1,MAXIT=50
SIMULATE/STATICS
SIMULATE/DYNAMIC, END=15.0, DTOUT=.01
DEACTIVATE/SENSOR, ID=<your right rear sensor id #>
SIMULATE/DYNAMIC, DURATION=1.0, DTOUT=.01
SFORCE/<original input sforce id #>, FUNCTION=200*(time-10)
SIMULATE/DYNAMIC, DURATION=12.0, DTOUT=.01
```

Note the force at which the switch toggles to the forward and rearward directions when accounting for friction in the contact between the follower parts and the contact parts. Were the effects of friction negligible in this mechanism?

- 3 Save your work.

This appendix contains tables that describe the various elements in ADAMS/View.

## What's in this appendix:

- Constraints Tables (Incomplete), 374
- Forces Tables (Incomplete), 375
- Constraint Tables (Completed), 376
- Forces Tables (Completed), 378

# Constraints Tables (Incomplete)

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
<b>Fixed</b>							
<b>Revolute</b>							
<b>Translational</b>							
<b>Cylindrical</b>							
<b>Universal/ hooke/ Constant velocity</b>							
<b>Spherical</b>							
<b>Planar</b>							

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
<b>Point-to-curve</b>							
<b>Curve-to-curve</b>							

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
<b>Orientation</b>							
<b>Inline</b>							
<b>Parallel axis</b>							
<b>Inplane</b>							
<b>Perpendicular</b>							

# Forces Tables (Incomplete)

	Translational Spring-Damper:	Torsional Spring-Damper:	Bushing:	Beam:	Field:
<b>Number of Bodies Affected</b>					
<b>Points of Application</b>					
<b>Number of Components</b>					
<b>Direction/Orientation</b>					
<b>Magnitude</b>					

	Single-Component Forces			Multi-Component Forces		
	Between 2 Bodies Translational:	Between 2 Bodies Rotational:	1 Body - Space Fixed:	1 Body - Moving:	Vector Force/ Torque:	General Force:
<b>Number of Bodies Affected</b>						
<b>Points of Application</b>						
<b>Number of Components</b>						
<b>Direction/Orientation</b>						
<b>Magnitude</b>						

# Constraint Tables (Completed)

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Fixed	✓	✓	✓	✓	✓	✓	6
Revolute	✓	✓	✓	✓	✓		5
Translational	✓	✓		✓	✓	✓	5
Cylindrical	✓	✓		✓	✓		4
Universal/ hooke/ Constant velocity	✓	✓	✓			✓	4
Spherical	✓	✓	✓				3
Planar			✓	✓	✓		3

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Point-to-curve	✓	✓					2
Curve-to-curve	✓	✓					2

## Constraint Tables (Completed)...

	Translation along X:	Translation along Y:	Translation along Z:	Rotation about X:	Rotation about Y:	Rotation about Z:	Total:
Orientation				✓	✓	✓	3
Inline	✓	✓					2
Parallel axis				✓	✓		2
Inplane			✓				1
Perpendicular						✓	1

# Forces Tables (Completed)

	Spring-Damper Translational:	Spring-Damper Torsional:	Bushing:	Beam:	Field:
<b># Bodies Affected</b>	2	2	2	2	2
<b>Points of Application</b>	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)	2 (I & J markers)
<b>Number of Components</b>	1	1	6	6	6
<b>Direction/ Orientation</b>	Line of sight between the (I & J markers)	Z-axis of J marker	J marker	J marker	J marker
<b>Magnitude</b>	Defined by parameters, such as stiffness, damping, cross-sectional area.				

	Single-Component Forces				Multi-Component Forces	
	Between 2 Bodies Translational:	Between 2 Bodies Rotational:	1 Body - Space Fixed:	1 Body - Moving:	Vector Force/ Torque:	General Force:
<b>Number of Bodies Affected</b>	2	2	1	1	2	2
<b>Points of Application</b>	2 (I & J markers)	2 (I & J markers)	1 (I marker)	1 (I marker)	2 (I & J markers)*	2 (I & J markers) *
<b>Number of Components</b>	1	1	1	1	3	6
<b>Direction/ Orientation</b>	Line of sight between I and J markers	A-axis of J-marker	Z-axis of J-marker	Z-axis of J marker	R marker	R marker
<b>Magnitude</b>	Defined by whole functions of which the user must take ownership.					

\* The J markers created for a vector force/torque and a general force are floating markers.

### What's in this appendix:

- Answer Key for Workshop 1, 380
- Answer Key for Workshop 2, 380
- Answer Key for Workshop 3, 380
- Answer Key for Workshop 4, 381
- Answer Key for Workshop 5, 381
- Answer Key for Workshop 6, 381
- Answer Key for Workshop 7, 382
- Answer Key for Workshop 8, 382
- Answer Key for Workshop 9, 383
- Answer Key for Workshop 10, 383
- Answer Key for Workshop 11, 383
- Answer Key for Workshop 12, 383
- Answer Key for Workshop 13, 383
- Answer Key for Workshop 14, 384
- Answer Key for Workshop 15, 384
- Answer Key for Workshop 16, 384
- Answer Key for Workshop 17, 384
- Answer Key for Workshop 18, 385
- Answer Key for Workshop 19, 385
- Answer Key for Workshop 20, 385
- Answer Key for Workshop 21, 385
- Answer Key for Workshop 22, 385

# Answer Key...

---

## Answer Key for Workshop 1

Question 1, page 25: 269 mm

Question 2, page 25: 269 mm. This is the same as the previous results.

Question 3, page 25: 267.87 mm

Question 4, page 25: Six: 5 make up the stamper mechanism, while 1 makes up the part parcels.

Question 5, page 25: Eight: 7 are on the stamper mechanism, while 1 keeps the parcels moving translationally.

Question 6, page 25: Nothing: the conveyor is simply a graphic attached to ground. It adds nothing to the model other than for animation purposes.

## Answer Key for Workshop 2

Question 1, page 43: There are nine constraints (two revolute, one translational, three inplane, one orientation, one motion, one curve\_curve). Motions are considered constraints; these will be covered in detail later in the course.

Question 2, page 43: Yes

Question 3, page 43: No, geometry is a direct child of a part. Part geometry is a “grandchild” of a model.

Question 4, page 43: Status bar

Question 5, page 43: Our technical support staff prefers to receive .cmd files because they are smaller in size, and platform independent. Using .bin files is sometimes unavoidable, however.

## Answer Key for Workshop 3

Question 1, page 60: 1.635 pound mass based on geometry and density

Question 2, page 60: 100 lbf/foot\*second

Question 3, page 60: Approximately 8.6 lbf

# Answer Key...

---

## Answer Key for Workshop 4

Question 1, page 77: 4903 mm

Question 2, page 77: 9807 mm/sec

Question 3, page 77: 9807 mm/sec<sup>2</sup>

Question 4, page 77: Coordinate system *markers*

Question 5, page 77: The ground part is automatically created - it must exist in every model. It serves as a reference frame for the model.

Question 6, page 77: No, because MSC.ADAMS cannot calculate a volume for two-dimensional objects. You can, however, *assign* mass properties to a part that is made up of two-dimensional geometry by changing Defined by to User Input.

## Answer Key for Workshop 5

Question 1, page 90: ~1.06 sec (can vary slightly depending on the sampling rate chosen).

Question 2, page 90: ~3180 mm (can vary slightly depending on the sampling rate chosen).

Question 3, page 90: The system constraint takes precedence.

Question 4, page 90: You would have to constrain the stone to ground with a revolute (pin) joint.

## Answer Key for Workshop 6

Question 1, page 114:  $F_x = -29.9\text{N}$ ,  $F_y = 17.24\text{ N}$

Question 2, page 114: Approximately 0.61 Hz

Question 3, page 114: ADAMS/View will use the one specified by the connecting joint. This is because the initial conditions in the constraint always override the initial conditions of a part if these two differ.


Question 4, page 114: The marker names would be .human\_hip.femur.MAR\_1 and .human\_hip.hip\_bone.MAR\_2. Which one is I and which one is J depends on the order in which the parts were selected when creating the constraint.

Question 5, page 114: No, a constraint constrains two different bodies to one another.

# Answer Key...

## Answer Key for Workshop 7

Question 1, page 135: Between  $16.5^\circ$  and  $17^\circ$  (Exactly  $16.7^\circ$ ).

Question 2, page 135: You can tell because there is an icon () for the joints to which you can add friction in an automated way.



Question 3, page 135: I and J markers are automatically created when you add a joint, motion, or force to a system. MSC.ADAMS uses the I and J markers' relative displacement, velocity, and so on to define equations that describe part movement.

Question 4, page 135: Once the joint crosses the stiction threshold velocity, it exits the stiction phase and the maximum stiction displacement is ignored until the joint reenters the stiction phase (comes to rest). One of these two parameters is reached first, the other parameter is ignored until the joint enters the stiction phase again.

## Answer Key for Workshop 8

Question 1, page 153: Construction geometry is two-dimensional, and solid geometry is three-dimensional.

Question 2, page 153:

- Position: Move-Translate .
- Location Event: Right-click away from the model when prompted for a position.
- Working grid: Settings → Working Grid → Set Location.
- Precision Move: Edit → Move (or )

# Answer Key...

---

## Answer Key for Workshop 9

**Question 1**, page 164: I and J markers. The I marker belongs to the first body you selected when the creating the joint. The J marker belongs to the second body you selected.

**Question 2**, page 164: The orientation of the I and J markers. For example, if you added translational motion to a translational joint, the z-axis of the I and J markers would describe the axis of translation. The z-axis direction is positive.

**Question 3**, page 164: Yes. Even though they do not restrict movement, they still prescribe movement, therefore, removing degrees of freedom.

**Question 4**, page 164: Yes. You must measure the torque generated by the motion **not** the revolute joint.

## Answer Key for Workshop 10

**Question 1**, page 172: The order in which you chose the bodies (parts) should be the same as the order in which you chose the corresponding locations and orientations.

## Answer Key for Workshop 11

**Question 1**, page 182: A joint motion uses a joint to determine its direction and location. A point motion does not require a joint; it needs two bodies.

## Answer Key for Workshop 12

**Question 1**, page 194: No. The point-to-point measure is just a quicker and easier way to create a function measure of the displacement of one marker with respect to another.

**Question 2**, page 194: A CAD file represents geometry in a model. Therefore, it is a child of a part.

## Answer Key for Workshop 13

**Question 1**, page 206: You need to provide the two (or three) joints, and either the scalar coefficients, displacements, or user-defined constraint equations.

**Question 2**, page 206: Last\_run

# Answer Key...

---

## Answer Key for Workshop 14

**Question 1**, page 219: No, in the design configuration they do not have to be aligned. If they are not aligned, however, ADAMS/View warns you during a model verify or during the assemble simulation. Also, during the assemble simulation, MSC.ADAMS realigns the markers for you.

## Answer Key for Workshop 15

**Question 1**, page 230:

- First independent variable
- Second independent variable
- Spline name
- Derivative order

**Question 2**, page 230: ADAMS/Solver -> Function Expressions (see the online help for reference)

## Answer Key for Workshop 16

**Question 1**, page 238: ~1.46 degrees (.0255 radians)

**Question 2**, page 238: To remove the initial transient effects in the dynamic system because of mismatches in the preloading of the bushings.

**Question 3**, page 238: Because the model was kinematic, in this case (DOF=0), there is no initial transient response because you have specified the motion of the system for all points in time.

## Answer Key for Workshop 17

**Question 1**, page 250: Yes, it must be greater than zero (not negative and not zero).

**Question 2**, page 250: Reposition the I and J markers of each single-component force such that the trigger distance is reached sooner and the full impact is felt earlier. Measuring the rotation of the I marker of joint I\_shortarm\_rev will determine the angle.

# Answer Key...

---

## Answer Key for Workshop 18

Question 1, page 262: Approximately 247 N.

Question 2, page 262: Yes: you could use a simulation script with ADAMS/Solver commands to simulate for a while, then DEACTIVATE or ACTIVATE the force, and simulate again.

Question 3, page 262: Yes. For example, you could simulate the model with output step sizes of 0.01 seconds. When that simulation is complete, don't reset the model. Start another simulation with a step size of 0.001. The results of that simulation will be seamless, but you will notice a change when the step size changes. The animation changes speeds. A common reason for doing this is if you want the simulation to use smaller step sizes or be more accurate before a contact.

## Answer Key for Workshop 19

Question 1, page 285: A statement describes an element in a model, such as a part or force. A command tells ADAMS/Solver what to do with the model, such as simulate it or deactivate it.

Question 2, page 285: 7.57 N.

## Answer Key for Workshop 20

Question 1, page 297: Preload must be above 460 N to open the lid. Higher damping values increase the amount of time needed to close the lid. Higher stiffness values increase the resistance in closing the lid.

Question 2, page 297: It allowed for easy manipulation of the spring parameters. Changing the design variables changed the parameters for both springs at the same time.

## Answer Key for Workshop 21

Question 1, page 317: A curve-on-curve constraint removes two translational DOF.

Question 2, page 317: A curve-to-curve contact force removes no DOF.

## Answer Key for Workshop 22

Question 1, page 336: The reference marker (R marker)

Question 2, page 336: Between 17 and 18 N/mm





## A - B

### Acceleration

- calculating 73
- finding value of 75
- functions 242

### Accessing online help 13

### ADAMS/AutoFlex

- about 304
- loading 314
- working with 314

### ADAMS/PostProcessor

- leaving 53
- modes of 46
- modifying animation graphics in 59
- playing animations in 59
- starting 53
- using 53
- viewing animations in 58

### ADAMS/PostProcessor overview workshop

- module review 60
- problem statement 50

### ADAMS/PostProcessor tool, using 53

### ADAMS/Solver

- command files 265
- dataset files 265
- history of 10
- overview of 264
- simulations in stand-alone, about 267
- simulations in stand-alone, performing 280

### ADAMS/View

- starting on UNIX 21
- starting on Windows 20

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

ADAMS/View interface overview workshop

- module review 43
- optional tasks 42
- problem statement 34

**ADAMS/View** tool, using to start on UNIX 21

Add-on constraints

- about 196
- types of 196

AKISPL function

- creating 228
- example of 224
- syntax for 224

**Align & Rotate** tool, using to rotate objects 127

Angle measures

- about 100
- creating 107

Animation

- adding to plot 57
- compared to simulation 31
- how to perform 41
- modifying graphics in ADAMS/PostProcessor 59
- playing in ADAMS/PostProcessor 59
- viewing in ADAMS/PostProcessor 58

**Animation** tool, using 41

Arc geometry, about 143

Assembling models

- about 198
- how to 204

Assembly simulation, about 208

Binary files, saving as 89

Block

- creating 85
- geometry, about 144

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Body 313, example of 118

**Box** tool, using 85

**Bushing** tool, using 235

Bushings

characteristics of 232

creating 235

defined 232

recommended practices 341

## C - D

CAD-based geometry

importing, about 186

importing, how to 191

Cam profile, creating 307

Cam-rocker-valve workshop

module review 317

optional tasks 316

problem statement 305

**Chamfer** tool, using 151

**Change Direction** tool, using 310

Command files

about 32, 265

creating 279

modifying 283

**Compute Linear Modes** tool, using 217

Constraining a model, about 128

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Constraints

- add-on (complex) 196
- and DOF 96
- couplers 197
- curve-on-curve, creating 308
- curve-on-curve, replacing 309
- defined 94
- determining number and type 37
- diagrammed 95
- DOF they remove 376
- equations in MSC.ADAMS 95
- example 94
- types of 301

## Contact forces

- about 302
- creating solid-to-solid 329
- creating sphere-to-plane 329

## Contact pairs, types of 302

## Contact tool, using 309

## Coordinate systems

- defined 62
- types of 62

## Coupler

- creating 202
- defined 197
- example of 197
- modifying 203
- requirements for modeling 197

## Coupler tool, using 202

## Course, what you will achieve in 12

## Crawl-walk-run approach, about 175

## Create a New Page tool, using 53

## Curve-Curve Constraint tool, using 308

## Curve-on-curve constraint

- creating 308
- replacing 309

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Cylinder

- creating 149
- geometry, about 144

## Cylinder tool, using 149

## Cylindrical joints, DOF they remove 376

## Database files, about 32

## Dataset files

- about 265
- example of 266
- exporting 279
- modifying 282

## Debug, about 276

## Dependencies, about 66

## Design studies

- defined 323
- performing 24
- types of 322

## Design variables

- about 289
- creating 294
- example 289
- modifying 294

## Displacement

- calculating 72
- finding horizontal 88
- finding value of 74

## Displacement functions

- about 185
- example 185

## DOF

- constraints and DOF 96
- determining number of 96

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Dynamic simulation

about 210

performing 229

Dynamics, debugging tips 347

## E - F

Eprint, about 276

Equations of motion, formulation in ADAMS 269

Euler angles, defined 118

Extrusion geometry, about 144

Falling stone workshop

module review 77

optional tasks 76

problem statement 70

File formats, about 32

Fillet tool, using 150

Fixed joint tool, using 161

Fixed joints

DOF they remove 376

recommended practices 340

Flexible bodies

creating 314

defined 67

Forces

characteristics of 212

compare linear and nonlinear 229

contact forces 302

definition 212

friction 120, 121

multi-component 320

single component, about 222

spring dampers 213

vector 320

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Frequency

- determining 111
- finding natural 217

## Friction

- adding to joint 130
- effect of deformation on 122
- forces, about 120
- input forces to 122
- phases of 120

## Function Builder

- about 342
- using 189

## Functions

- acceleration 242
- AKISPL 224
- defined 158
- displacement 185
- IMPACT 240, 241
- spline 223
- STEP 252
- velocity 242

## G - H

GCON, using 111

General constrains, creating 112

General Constraints tool, using 112

## Geometry

- arc 143
- block 144
- cylinder 144
- extrusion 144
- importing CAD-based, about 186
- importing CAD-based, how to 191
- marker 142
- merging 99
- point 142
- polyline 143

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

- spline 143
- torus 144
- types of 141

Global components, determining 110

Graphical topology, checking 172

Gravity, recommended practices 343

Ground parts, about 67

Gruebler's count 96

Hatchback I workshop

- module review 250
- problem statement 243

Hatchback II workshop

- module review 262
- problem statement 255

Hatchback III workshop

- module review 285
- optional tasks 284
- problem statement 278

Hatchback IV workshop

- module review 297
- optional tasks 296
- problem statement 290

Help, online 13

Hierarchy of simulation 209

Hollow tool, using 152

Hooke joint tool, using 180

Hooke joint, creating 180

A - B
C - D
E - F
G - H
I - J
K - L
M - N
O - P
Q - R
S - T
U - V
W - Z

## I - J

### Icons

- animating with icons off 41
- animating with icons on 41

### IMPACT function

- about 240
- applications of one sided 240
- applications of two sided 240
- syntax 241

### Importing

- CAD-based geometry, about 186
- CAD-based geometry, how to 191
- files 35
- test data 223

### Inclination angle

- finding 133
- modifying 126

### Inclined plane workshop

- module review 135
- optional tasks 134
- problem statement 124

### Inertia properties, about 68

### Information tool stack, using 37

### Initial conditions

- about 80
- joint initial conditions 97
- locations and orientations 80, 208
- setting for joints 108
- setting for velocity 86
- velocity 81

### Inline joint primitives, about 166

### Inplane joint primitives, about 166

### Joint friction, about 120

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Joint motion

- about 156
- creating 162
- marker usage in 157

## Joint primitives

- inline 166
- inplane 166
- orientation 166
- parallel axis 166
- perpendicular 166
- perpendicular, usage of I and J markers 167
- types of 166

## Joint toolstack, using 105

## Joints

- friction 120
- initial conditions of 97

## K - L

## Kinematic simulation

- about 210
- performing 355

## Knowledge base, about 14

## Lift mechanism I workshop

- module review 153
- optional tasks 153
- problem statement 146

## Lift mechanism II workshop

- module review 164
- optional tasks 163
- problem statement 159

## Lift mechanism III workshop

- module review 172
- optional tasks 172
- problem statement 169

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Linear simulation

about 211

example 211

Link tool, using 102

Links, creating 102

Local coordinate system, about 62

Location Event, using 102

## M - N

Main Toolbox tool, using 52

Marker tool, using 107

Markers

defined 64

geometry, about 142

reference, creating 107

Mass properties

about 68

recommended practices 339

Mass, setting 71

Measures

angle 100

angle, creating 107

creating 72, 86

creating function 189

creating point-to-point 188

defined 69

function 184

in LCS 123

object, creating 106

point-to-point 184

recommended practices 342

representation 123

Merging geometry, about 99

Mesh, creating 314

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Model topology

- checking by constraints 162
- checking by parts 152

## Model topology by constraints tool, using 162

## Model topology by constraints tool, using to determine constraints 37

## Model topology by parts tool, using 152

## Models

- animating 41
- assembling 198
- clarifying topology 29
- constraining 128
- hierarchy of 28
- simulating 40
- verifying 39

## Modifying

- animation graphics 59
- coupler 203
- parts 126
- plot graphics 56
- plot legend 57
- spring stiffness 54

## Motion

- applying 181
- joint motion 156, 157
- point motion 174
- recommended practices 340
- types 156

## Move tool stack, using 132

## Moving objects, about 145

## MSC.ADAMS

- about 10
- history of 10
- list of products 11
- verification problems for 11

## MSC.ADAMS Full Simulation Package, about 9

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Multi-component forces  
    characteristics of 378  
    creating 331  
    types of 320

Naming convention, about 29

Nonlinear spring workshop  
    module review 230  
    optional tasks 230  
    problem statement 225

**O - P**

Object measures  
    creating 72  
    defined 69

Objects  
    creating a group of 131  
    rotating about an axis 119

One DOF pendulum workshop  
    module review 114  
    optional tasks 113  
    problem statement 101

Online help, accessing 13

Optimization study, performing 24

Optimizing a design, about 295

Orientation joint primitives, about 166

Orientation, verifying 171

Overlaying plots, how to 55

**Page layout** tool stack, using 57

Parallel axis joint primitives, about 166

Part coordinate system, about 63

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Parts

- building 71
- defined 65
- initial locations and orientations 80
- initial velocity 81
- renaming 37, 71
- rotating 127

## Perpendicular joint primitives

- about 166
- usage of I and J markers 167

Planar joints, DOF they remove 376

Plane tool, using 328

Plane, creating 328

Play tool, using to play simulation 40

Plot graphics, modifying 56

Plot legend, modifying 57

Plot statistics, obtaining 56

Plot Tracking tool, using 88

## Plots

- creating 53
- overlying 55

Plotting option, about 48

Point geometry, about 142

## Point motion

- about 174
- applying 181

## Point trace

- creating 88
- defined 82
- using 307

Point-to-Point tool, using 149

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Polyline geometry, about 143

**Precision Move** tool, using 132

Projectile motion workshop

    module review 90

    optional tasks 89

    problem statement 83

## Q - R

Range, finding 87

Reference markers, creating 107

Renaming objects, about 29

**Reset** tool, using to reset model 40

**Revolute joint** tool, using 105

Revolute joints

    creating 105

    deactivating 234

    DOF they remove 376

Rigid bodies, defined 67

**Rigid Body** tool stack, using 107

Rotating

    group of objects 132

    objects, about 119

    parts, how to 127

**Rotational Joint Motion** tool, using 162

## S - T

**Save simulation** tool, using 40

Saving

    command files 42

    model information 42

    simulation results 40

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Scripted simulations

- based on ADAMS/Solver commands 254
- creating script for 257
- in ADAMS/View 253
- performing 258

## Select tool, using 86

## Sensitivity, at iteration 324

## Sensors

- about 288
- adding to model 292
- example of using with scripts 288

## SFORCE, see Single-component forces 222

## Simulation

- about scripts 253
- compared to animation 31
- comparing results 283
- dynamic 210
- hierarchy 209
- how to perform 40
- kinematic 210
- linear 211
- saving results of 40
- static 210
- submitting 31
- types of 210, 253

## Simulation tool, using to simulate model 40

## Single-component forces

- about 222
- characteristics of 378
- creating 291

## Solution, phases of 271

## Space 313, example of 118

## Sphere tool, using 103

## Sphere, creating 103

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## Spherical joints

- creating 180
- DOF they remove 376

## Splines

- about 223
- from point trace 300
- geometry, about 143
- recommended practices 342

## Spring dampers

- characteristics of 213
- creating 216
- defined 213
- find force in 216
- recommended practices 341
- replacing predefined 226

## Spring stiffness

- coefficient, finding 54
- modifying 54

## Spring, changing linear to nonlinear 227

## Spring-damper workshop

- optional tasks 218
- problem statement 215

## Stamping mechanism workshop

- module review 25
- problem statement 19

## Stand-alone ADAMS/Solver

- about simulations in 267
- performing simulations in 280

## Starting ADAMS/View

- on Windows 20

## Static Equilibrium tool, using 216

## Static equilibrium, running simulation 216

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

- Static simulation
  - about 210
  - performing 216
- Statics, debugging tips 346
- Step Backward** tool, using 87
- Step Forward** tool, using 87
- STEP function
  - defined 252
  - example of 252
  - syntax for 252
- Stop** tool, using 87
- Suspension system I workshop
  - module review 182
  - optional tasks 182
  - problem statement 176
- Suspension system II workshop
  - module review 194
  - optional tasks 193
  - problem statement 187
- Suspension-steering system II workshop
  - module review 238
  - optional tasks 238
  - problem statement 233
- Suspension-steering system workshop
  - module review 206
  - problem statement 199
- Switch mechanism workshop, problem statement 349
- Target practice workshop
  - module review 336
  - optional tasks 336
  - problem statement 325
- Technical support, about 14
- Test data 223

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

Torus geometry, about 144

Training guide, organization of 12

Translating models, how to 73

**Translational Joint Motion** tool, using 306

**Translational Joint** tool, using 128

Translational joints

- creating 128
- DOF they remove 376

**Translational spring damper** tool, using 216

Transparent, making a part 172

**U - V**

Units, recommended practices 343

Universal joints, recommended practices 340

Vector force 320, 321

Velocity

- calculating 72
- finding value of 75
- functions 242

Verification problems, running through 11

**Verify** tool, using to verify model 39

Verifying models

- about 344
- how to 39

Viewing models, how to 36

Virtual prototyping process, diagrammed 18

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z

## W - Z

### Working grid

- adjusting [125](#)
- resetting to default position [328](#)
- set up to run through part center [149](#)
- setting display of [85](#)

### Zooming, how to [73](#)

A - B

C - D

E - F

G - H

I - J

K - L

M - N

O - P

Q - R

S - T

U - V

W - Z