UNIVERSAL MECHANISM 7.0

UM Input Program

User’s manual

2014

General methods of development of UM models are considered. Features in description of images, bodies, joints, force elements and so on are discussed

Contents

3. UM Input Program

3.1. What's new in 7.0?

Here new UM elements and features, which are related to UM Input program, are listed.

1. New function *bodyinertia* is added, which returns inertia parameters of bodies. The function can be included in expression of any type, Sect. 3.5.2.4.4.

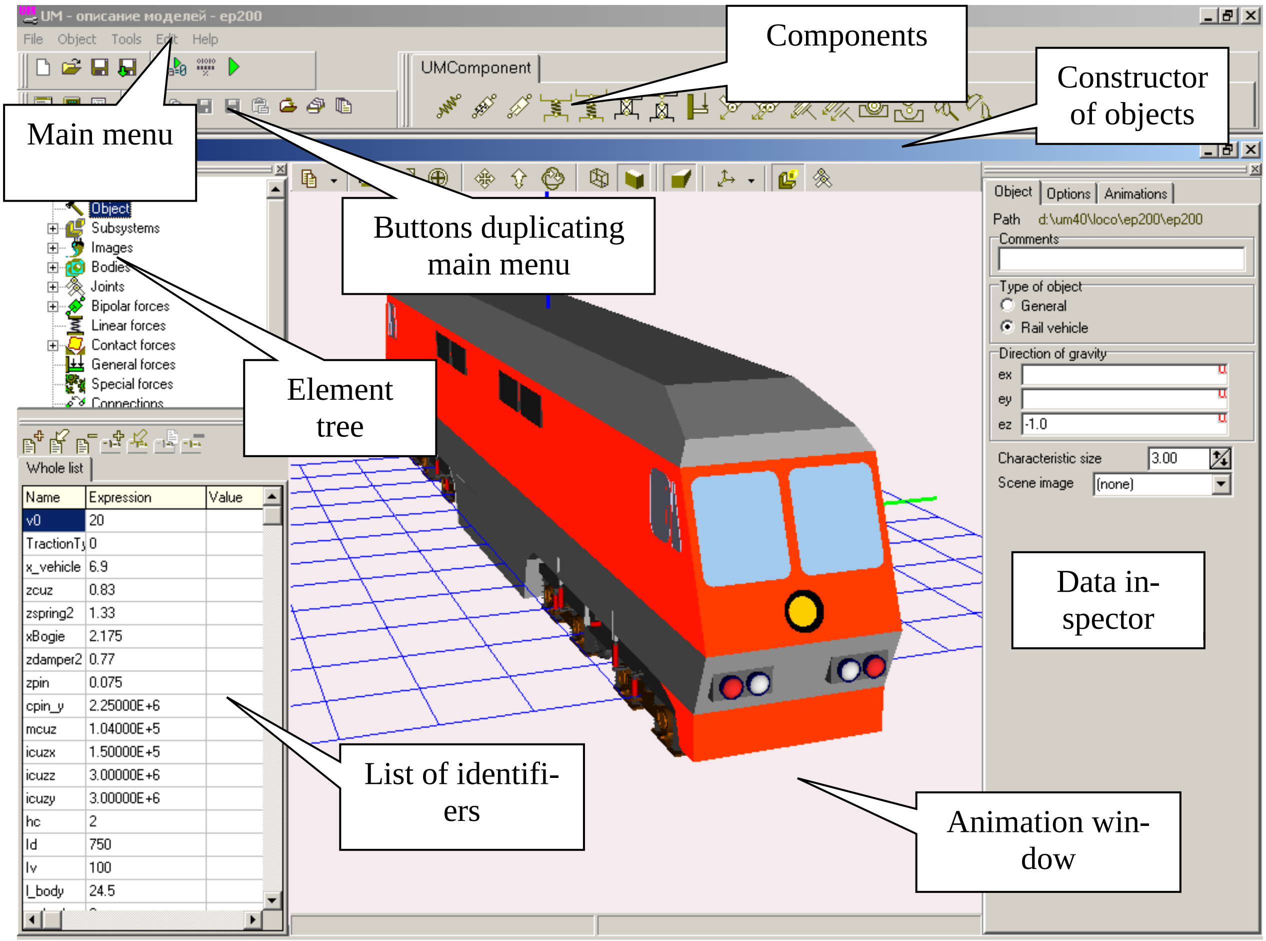
3.2. Data input program: general information

Program for development of models **UM Input** is intended for creation and correction of multibody systems as well as for optional automatic generation and compilation of equations of motion.

Basic elements of the program **(UM Input)** are the following (Figure 2.1):

- the main menu;
- the tool panel with buttons, which duplicate some most often used command of the main menu;
- the tabs with standard and user defined components;
- the object constructor is the main tool for the model development.

The data input program is a multitasking tool, which allows opening several constructors with description of different objects. An object, whose constructor is placed over all others, is the *active* one.



Name	Expression	Value
v0	20	
TractionTy	0	
x_vehicle	6.9	
zcuz	0.83	
zspring2	1.33	
xBogie	2.175	
zdamper2	0.77	
zpin	0.075	
cpin_y	2.25000E+6	
mcuz	1.04000E+5	
icuzx	1.50000E+5	
icuzz	3.00000E+6	
icuzy	3.00000E+6	
hc	2	
Id	750	
Iv	100	
l_body	24.5	

Path d:\um40\loco\ep200\ep200

Comments

Type of object
○ General
● Rail vehicle

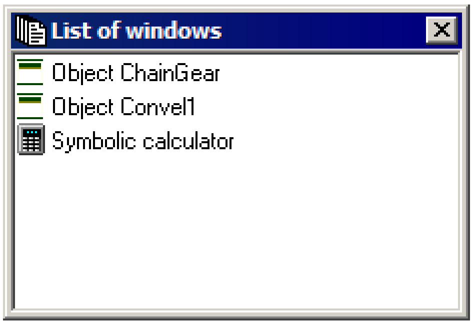
Direction of gravity
ex
ey
ez -1.0

Characteristic size 3.00

Scene image (none)

Figure 3.1. General view of UM Input program

To select an active object from the list of open models, the **List of windows** is used, Figure 3.2. The window is available by **the Tools | List of windows** menu command or by the *Alt+0* key.



List of windows

Object ChainGear
Object Convel1
Symbolic calculator

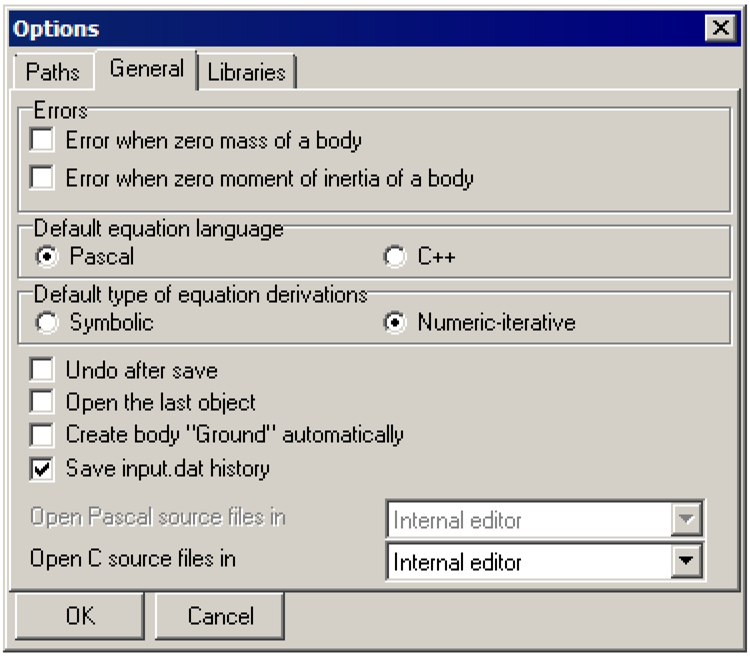
Figure 3.2. List of windows as well as open objects

3.3. Options of Input program

To set or modify options of the Input program

- run **UM Input**;
- call the option window with the **Tools | Options** main menu command;
- use the **OK** button of the window to store changes in the computer registry.

3.3.1. General options of the Input program



Options

Paths | General | Libraries

Errors
□ Error when zero mass of a body
□ Error when zero moment of inertia of a body

Default equation language
● Pascal ○ C++

Default type of equation derivations
○ Symbolic ● Numeric-iterative

□ Undo after save
□ Open the last object
□ Create body "Ground" automatically
☑ Save input.dat history

Open Pascal source files in: Internal editor
Open C source files in: Internal editor

OK Cancel

Figure 3.3. General options

Use the **General** tab (Figure 3.3) to specify the following parameters.

- **Error when zero mass of a body**

If the key is checked, zero mass is considered as an input error, else as a warning, Sect. 3.6.9.2.

- **Error when zero moment of inertia of a body**

If the key is checked, zero inertia moment is considered as an input error, else as a warning;

- The **default equation language**

The option is taken into account if symbolic generation of equations is assigned to the object, Sect. 3.9.2. Here the language for automatically generated equations of motion and programming in the UM environment is selected. The choice depends on the presented compiler (Sect. 3.3.2);

- **Default type of equation derivations:**
    - *Symbolic* – the equations are generated either in Pascal or C files and should be compiled as a DLL with an external compiler; programming in the Control file is available;
    - *Numeric-iterative* – the equations of motion are generated during the simulation in the numeric form; no external compiler is required; programming in the Control file is not available;
- **Undo after save –** if not checked, the undo of old operations are not available after saving the model;
- **Open the last object** – if checked, the latest active object will be opened automatically when the Input program starts;
- **Create body 'Ground' automatically**

If checked, the fictitious 'ground' body is created automatically; this body is rigidly connected to the SC0; the ground body is used for visual coupling of model bodies to the SC0 by joints or force elements with the help of connection points preliminary assigned to the ground.

- **Save input.dat history** – creates a copies of the model description file input.dat in the object directory after each saving the modified model;
- **Open Pascal source files in** – allows selecting an editor for programming on Pascal language.
- **Open C source files in** – allows selecting an editor for programming on C language.

Remark. Handling zero inertia parameters as an error is recommended for beginners to avoid the degeneration of the mass matrix at the simulation of objects. Besides, the option is useful if the models simulation is done with the parallel solver on multi-core processors, which requires all mass and moment of inertia to be nonzero.

3.3.2. Setup of symbolic generation of equations of motion

UM **optionally** generates equations of motion of objects with the help of a built-in specialized computer algebra system. To simulate the object dynamics, the equations should be compiled with the help of an external compiler (Delphi 4.0-7.0, Visual C++ 5.0-6.0, Borland C++ Builder 4.0-7.0, BDS 2005-2006), which is not delivered with UM. First of all, make sure that a proper compiler is installed on your computer or on a server. Then use the **General** tab to set the default external compiler. In fact, *UM can use numeric-iterative method of the generation of equations of motion without an external compiler*, Sect. 3.9.

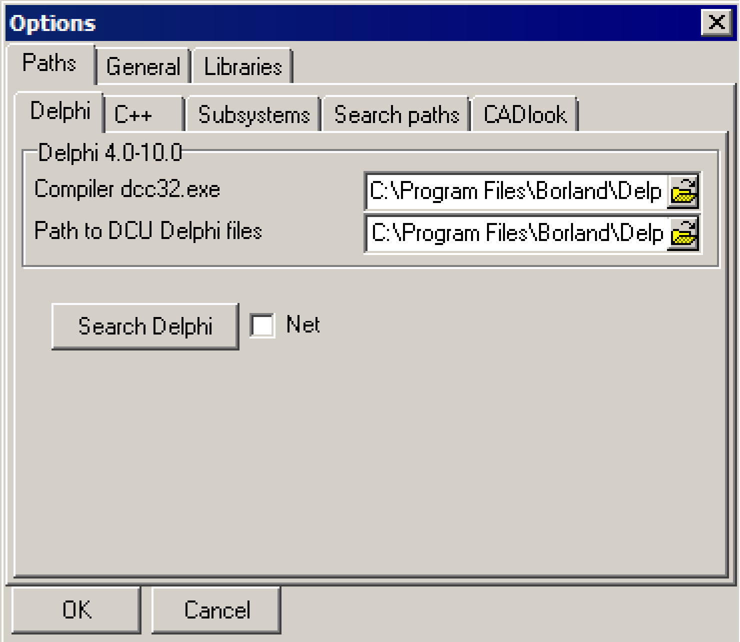
3.3.2.1. Delphi

Use the **Paths | Delphi** tab (Figure 3.4) to specify paths to a Delphi compiler and the Delphi VCL files. If the current computer has Delphi 4.0-7.0 (or higher) installed, it is enough to click the **Search Delphi** button to set the paths. If Delphi can be found in the local net try to find it automatically using the same button and the option **Net** turned on. In this case the paths will be found if the registry of the corresponding net computer is available for reading. If reading the registry is not allowed, the paths should be set manually. To do this, use the 📂 buttons in the right hand side of two boxes and find

- Delphi compiler dcc32.exe (usually [path to Delphi]\bin),
- Directory containing Delphi VCL *.dcu files (usually [path to Delphi]\lib).

Example:

- D:\Delphi7\bin\dcc32.exe,
- D:\Delphi7\lib.



Compiler dcc32.exe: C:\Program Files\Borland\Delp

Path to DCU Delphi files: C:\Program Files\Borland\Delp

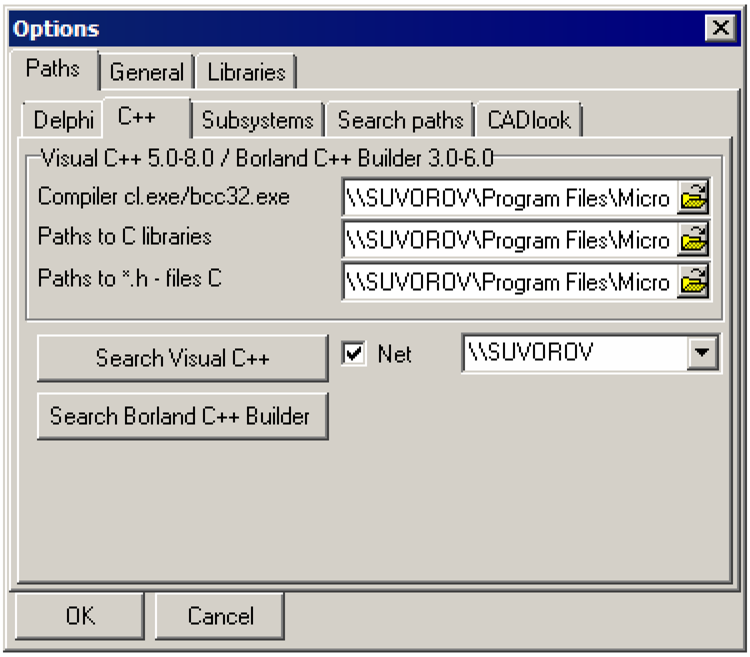
□ Net

Figure 3.4. Paths to Delphi

3.3.2.2. C++

Use the **Paths | C++** tab (Figure 3.5) to specify paths to a C++ compiler. If the current computer has Visual C++ 5.0-8.0 or Borland C++ Builder 3.0-6.0, installed, it is enough to click the corresponding button to set the paths. If C++ can be found in the local net, try to find it automatically using the same button and the option Net turned on. In this case the paths will be found if the registry of the corresponding net computer is available for reading. If reading the registry is not allowed, the paths should be set manually. To do this, use the 📂 buttons in the right hand side of two boxes and find

- **C** compiler cl.exe/bcc32.exe;
- Paths to **C** libraries (lib-files);
- Paths to the standard *.h files.



Visual C++ 5.0-8.0 / Borland C++ Builder 3.0-6.0

Compiler cl.exe/bcc32.exe \\SUVOROV\Program Files\Micro

Paths to C libraries \\SUVOROV\Program Files\Micro

Paths to *.h - files C \\SUVOROV\Program Files\Micro

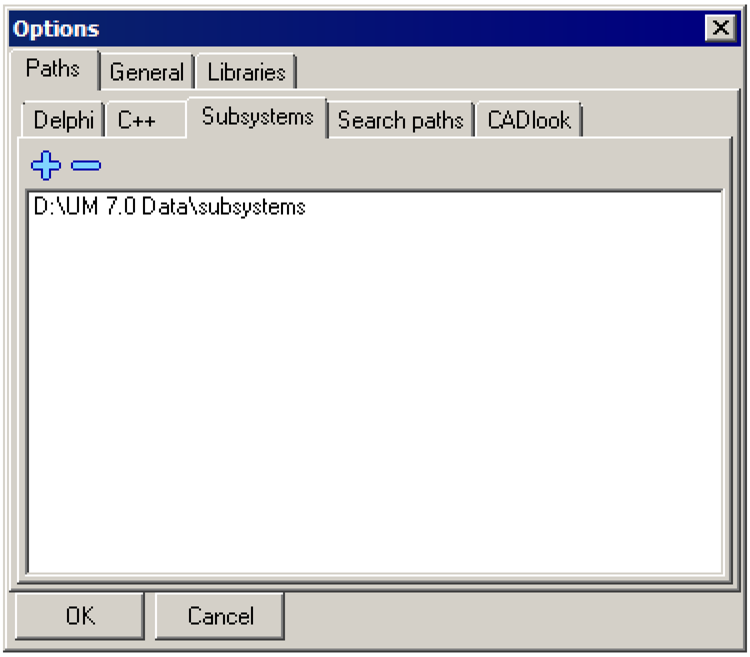
☑ Net \\SUVOROV

Figure 3.5. Paths to C++

Example for Visual C++ installed on computer *Suvorov*:

- \\SUVOROV\Program Files\Microsoft Visual Studio\VC98\bin\cl.exe
- \\SUVOROV\Program Files\Microsoft Visual Studio\VC98\lib
- \\SUVOROV\Program Files\Microsoft Visual Studio\VC98\Include

3.3.3. Paths to external subsystems



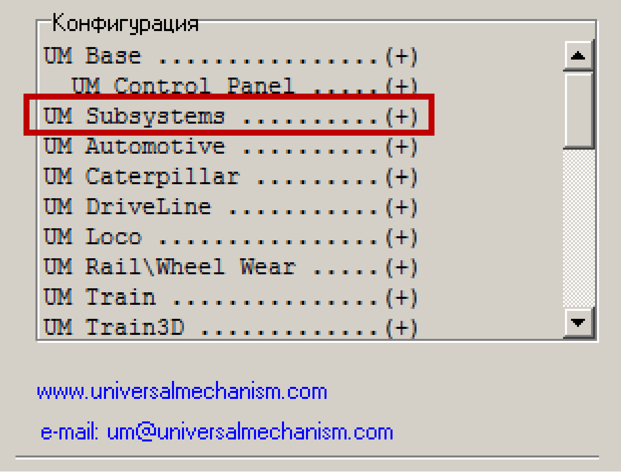
D:\UM 7.0 Data\subsystems

Figure 3.6. Paths to external subsystem

Use the **Paths | Subsystems** tab (Figure 3.6) to add/delete paths to directories with *external subsystems*. UM uses these paths to search external subsystems added to the current UM model.

Remark 1. The option is important if the *Subsystems* module is included in UM configuration, figure 3.7.

Remark 2. The path must leads to a directory, which contain UM models, and not directly to an external subsystem.



```
UM Base ................(+)
  UM Control Panel .....(+)
UM Subsystems ..........(+)
UM Automotive ..........(+)
UM Caterpillar .........(+)
UM DriveLine ...........(+)
UM Loco ................(+)
UM Rail\Wheel Wear .....(+)
UM Train ...............(+)
UM Train3D .............(+)
```

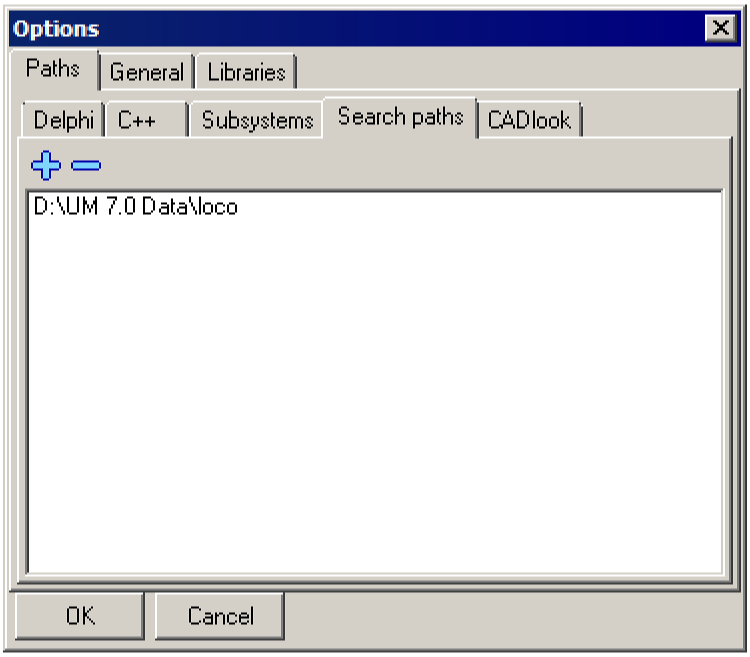
www.universalmechanism.com

e-mail: um@universalmechanism.com

Figure 3.7. UM configuration in the **About** window

3.3.4. Paths to user's files

Use the *Paths* | *Search paths* (Figure 3.8) tab to add/delete paths to directories with files, which are used as parts of user's code (programming in UM environment). The paths are used by the external compiler if equations are generated in symbolic form, Sect. 3.9.

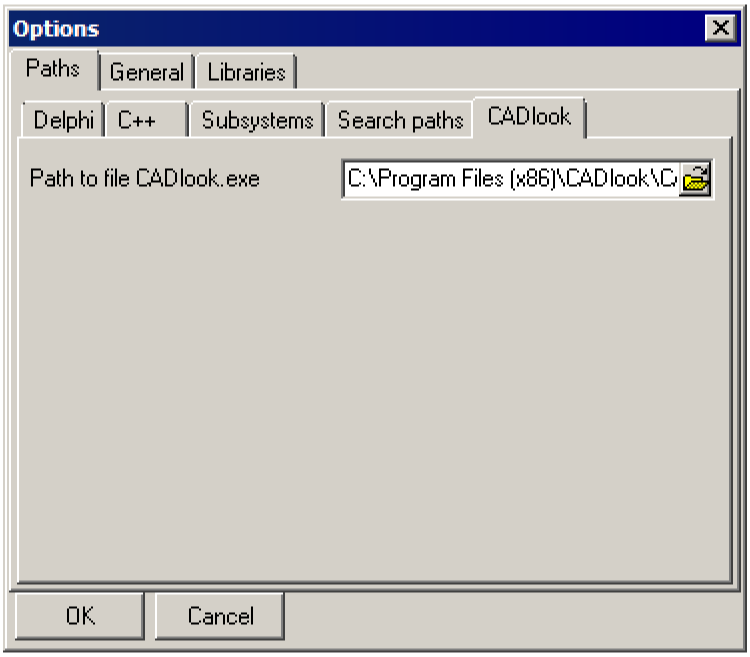


D:\UM 7.0 Data\loco

Figure 3.8. Paths to user's units and files

3.3.5. Path to CADlook program

Use the *Paths* | *CADlook* (Figure 3.9) tab to add/delete path to commercial viewer of CAD files CADLook for conversion of data from STEP (both AP203 and AP214), IGES, X_T (Parasolid), SAT formats, see Chapter 9 of the user's manual.

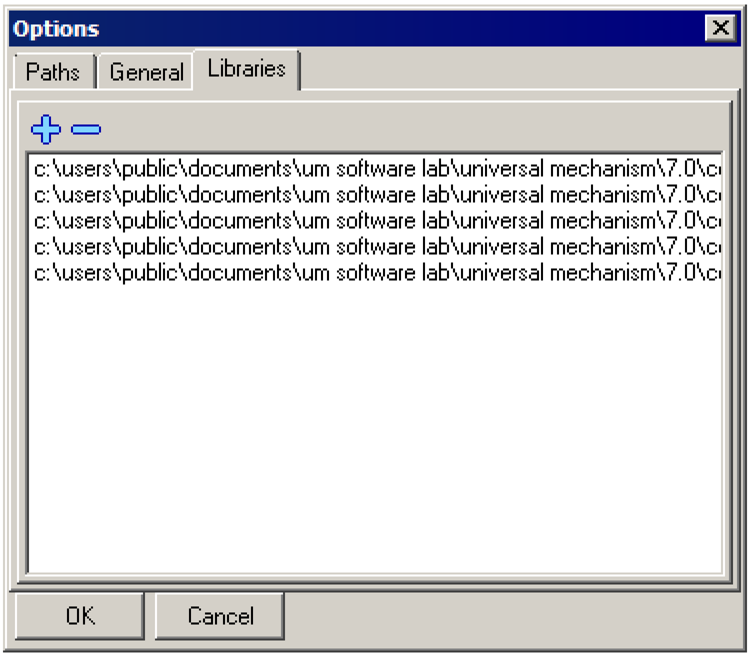


Path to file CADlook.exe C:\Program Files (x86)\CADlook\C

Figure 3.9. Paths to CADlook.exe

3.3.6. Component libraries

Use the **Libraries** tab to add or remove component library files, Sect. 3.7. To add a new component library, click the + button and select the necessary file using the standard dialog window **Open**. To remove a selected library, use the − button.



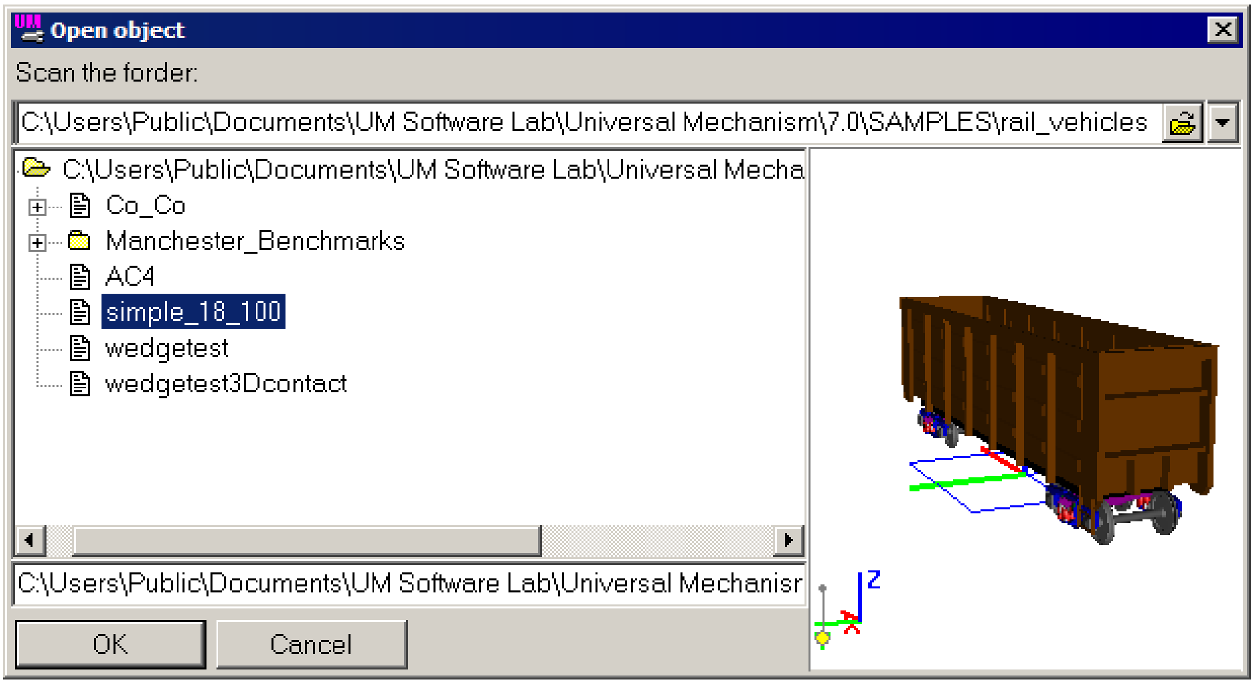
```
c:\users\public\documents\um software lab\universal mechanism\7.0\c
c:\users\public\documents\um software lab\universal mechanism\7.0\c
c:\users\public\documents\um software lab\universal mechanism\7.0\c
c:\users\public\documents\um software lab\universal mechanism\7.0\c
c:\users\public\documents\um software lab\universal mechanism\7.0\c
```

Figure 3.10. Component libraries

3.4. Main menu commands and tool panel

3.4.1. File

- **New object** (*Ctrl+N*) – creates a new model with the default name **UmObj[Index]** and opens constructor.
- **Open object** (*Ctrl+O*) – calls a special dialog box for choice of an existing object (Figure 3.11). The dialog box contains a tree of objects found in the *root directory*. To set the default path to the root directory use the button in the top edit box to select a directory and the *Accept as default* button right after that.



C:\Users\Public\Documents\UM Software Lab\Universal Mechanism\7.0\SAMPLES\rail_vehicles

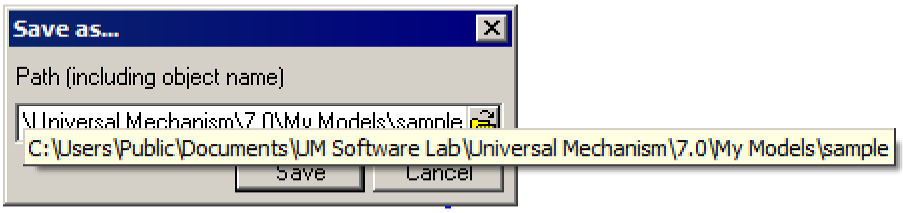
C:\Users\Public\Documents\UM Software Lab\Universal Mecha
- Co_Co
- Manchester_Benchmarks
- AC4
- simple_18_100
- wedgetest
- wedgetest3Dcontact

C:\Users\Public\Documents\UM Software Lab\Universal Mechanisr

Figure 3.11. Open object dialog box

Use the **F5** button or the pop-up menu to refresh the tree of objects in the *Open object* dialog box. Use the upper edit box for changing the current directory.

- **Open *.umd** – open file of object imported from CAD and save in the intermediate format umd, Sect. 3.10.2.
- **Import MSC.ADAMS** – reading ADAMS files *.cmd, *.adm and conversion the model in UM format, Sect. 3.11.
- **Reopen** – allows the user to open recently used objects.
- **Save** (*Ctrl+S*) – saves the active object in the object directory. The command is executed if the active object has been modified and the object directory has already been created. If the directory does not exist, the **Save as…** command is executed.
- **Save as…** – **saves** the active object in a directory pointed out by the user. Use the Save as... dialog box to select or enter a path to the object **including its name**, Figure 3.12. If necessary, new directories are created. The object takes the last directory in the path as its own name.



Path (including object name)

\Universal Mechanism\7.0\My Models\sample

C:\Users\Public\Documents\UM Software Lab\Universal Mechanism\7.0\My Models\sample

Figure 3.12. Window contains path for save the model

Remark. Since UM 6.0 the object name (i.e. the last directory in the path) can include any characters.

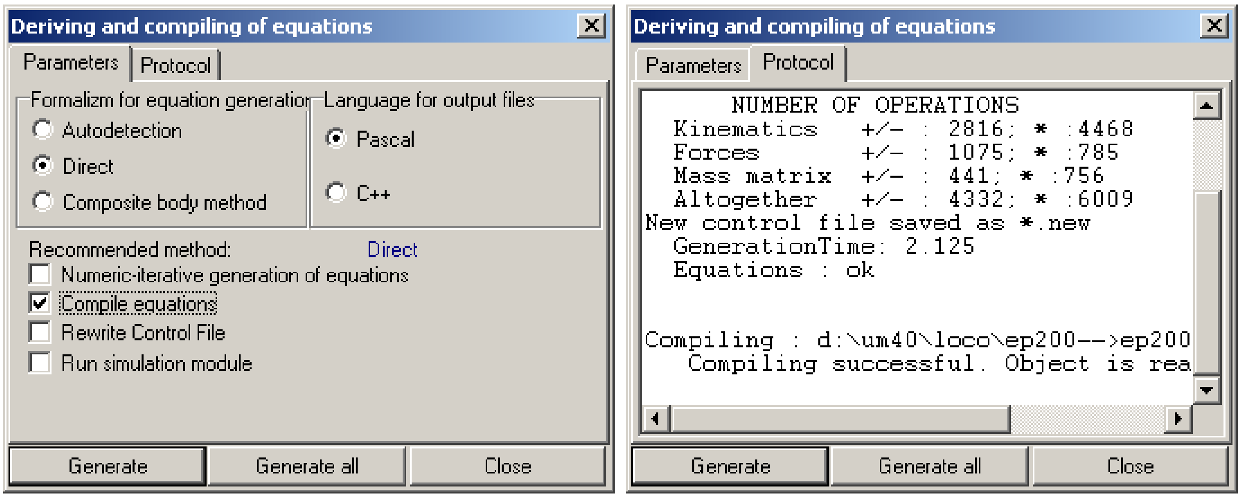
- **Save as component** – saving the model in a file with any name and extension to include the model in the list of components, Sect. 3.7.1, or to merge the model with another one by the **Edit | Read from file** menu command, Sect. 3.6.2.7.
- **Exit** (*Alt+X*) – close the program.

3.4.2. Edit

- **Copy to clipboard** – copy parameters of selected elements to clipboard.
- **To clipboard as component** – write parameters of the selected element and the assigned graphic object to clipboard. If the element has no graphical object the option is disabled.
- **Copy into file…** – write parameters of the selected element into file.
- **Save as component…** – save parameters of the selected element and the assigned graphic object in a file.
- **Insert** – insert element/component from clipboard.
- **Read from file…** – add all elements from file to the object.
- **UnDo, ReDo** – commands of UNDO system.

3.4.3. Object

- **Verify** data (*F7*) – verifies correctness of the object description.
- **Generate equations…** (*F8*) – saves modified active objects, deletes old UMTask.dll file of equations, and verifies the object description. If no errors are found, the window for generating and compiling equations starts. The tool is used if generation of equations in symbolic form is assigned to the active object, Sect. 3.5.2.1/



Formalizm for equation generation
○ Autodetection
● Direct
○ Composite body method

Language for output files
● Pascal
○ C++

Recommended method: Direct
□ Numeric-iterative generation of equations
☑ Compile equations
□ Rewrite Control File
□ Run simulation module

```
       NUMBER OF OPERATIONS
  Kinematics    +/- : 2816; * :4468
  Forces        +/- : 1075; * :785
  Mass matrix   +/- : 441; * :756
  Altogether    +/- : 4332; * :6009
New control file saved as *.new
  GenerationTime: 2.125
  Equations : ok


Compiling : d:\um40\loco\ep200-->ep200
   Compiling successful. Object is rea
```

Figure 3.13. Window for generation of equations in symbolic form

- **Compile equations** (*Ctrl+F9*) – compiling equations if they are generated in symbolic form.
- **Simulation…** (*F9*) – verifies whether the model is ready for simulation and runs the **UM Simulation** program.

3.4.4. Add

The tree of menu command allows adding a new element with simultaneous assignment of its type, Figure 3.14. Analogous commands are available in the tree of elements, Sect. 0.

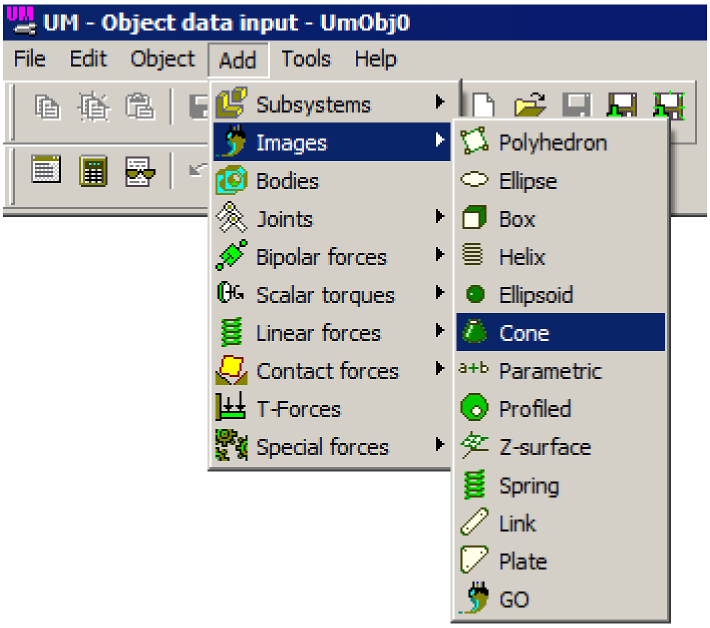


Figure 3.14. Example: adding a graphic element “Cone”

3.4.5. Tools

- **Editor…** – runs the built-in text editor.

- **Calculator of expressions…** – runs the calculator of chains of symbolic expressions (Sect. 3.5.2.4.2).
- **Inspector** (*F12*) – brings to front the *Data Inspector* for the active object if it is located on a separate window (Sect. 3.5.2).
- **List of elements** (*F11*) – brings to front the *Tree of elements* for the active object if it is located on a separate window (Sect. 3.5.1.1).
- **Identifiers** (*Alt+I*) brings to front the *List of Identifiers* of the active object if it is located on a separate window.
- **List of windows** (*Alt+0*) – calls the window containing the list of open windows.
- **Control File…** (*Alt+C*) – opens the Control file for the active object in the text editor.
- **File of elements…** – creates a file n[NameOfObject].txt in the object directory and opens it in the text editor. The file contains lists of all the object elements (bodies, joints, identifiers, force elements etc.) and their names.
- **Import form CAD…** – conversion CAD assemblies to UM format or reading files in *ucf,*, *3ds* and *stl* formats. Sect. 3.10.
- **Transformation of coordinates…** (*Alt+T*) – opens the forms for transformation of coordinates of points into different system of coordinates (Sect. 3.6.10.5).
- **Wizard of components…** – tool for edition of component libraries.
- **List of components…** – open window with the list of loaded components.
- **Button GO** – a tool for creation of a small bmp file based on an animation window; can be applied by adding a component to the list, Sect. 3.7.
- **Options** – opens a dialog box with UM options: paths to external compiler, standard and user's libraries etc. (Sect. 3.3).

Special tools (available if the corresponding UM module is included in the current configuration).

- **Import wheel profile from CAD** – tool for conversion in UM format of a railway wheel profile created in a CAD program; requires UM Loco; see file 08_UM_LOCO.pdf, Sect. Import wheel profile from CAD for more details.
- **Train wizard** – automatic generator of a train model with simplified one-dimensional vehicles; UM Train module is required; see file 15_UM_Train.pdf, Sect. *Development of train model* for more details.
- **Wizard of flexible subsystems** – a tool for preparing data of flexible subsystems; UM FEM module is required; see file 11_UM_FEM.pdf, Sect. *Wizard of flexible subsystems*.

3.4.6. Help

- **About UM context Help…** the command opens the file with implementation of the context UM help, file {UM Data}\MANUAL\UM_Context_Help.pdf.
- **Getting started** – list of help files with examples of starting UM modules: UM Base, UM Loco, UM Experiments, UM Automotive.
- **User's manual** – list of command to call of the manual pdf files.
- **About…** – short information about UM version and the list of developers.

3.4.7. Tool panel

Buttons located on the tool panel have the following functions:

creates a new object;

opens an existing object;

saves the active object;

saves the active object with a new name;

opens the text editor;

opens symbolic calculator;

verifies correctness of the active object description;

generates and compile equations for the active object;

compiles equations for the active object;

runs simulation of the active object.

import data from CAD;

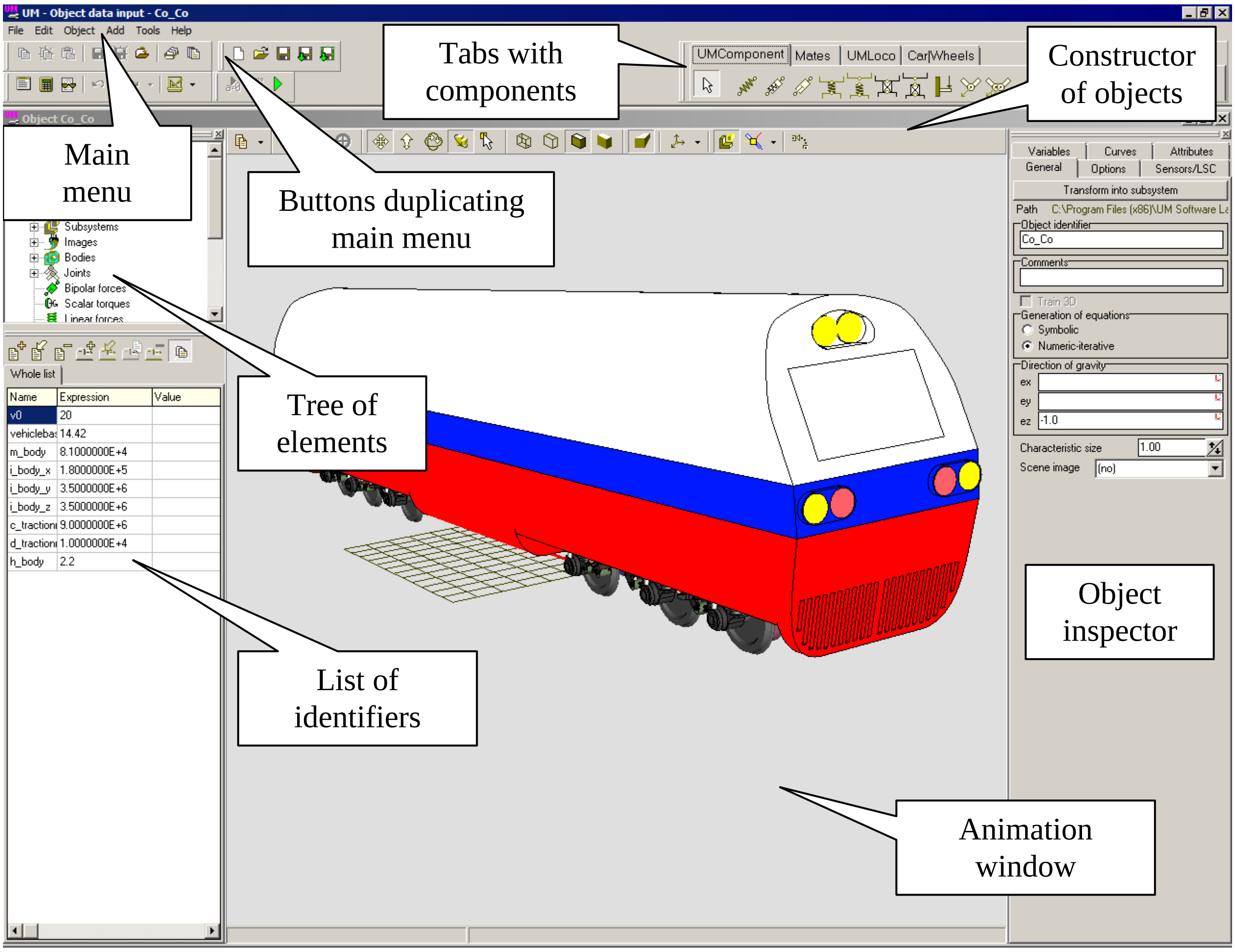
wizard of components, Sect. 3.7;

list of components, Sect. 3.7.2;

buttons and menu of UNDO and REDO operations.

3.5. Object constructor

3.5.1. Basic elements of constructor



Name	Expression	Value
v0	20	
vehiclebas	14.42	
m_body	8.1000000E+4	
i_body_x	1.8000000E+5	
i_body_y	3.5000000E+6	
i_body_z	3.5000000E+6	
c_traction	9.0000000E+6	
d_traction	1.0000000E+4	
h_body	2.2	

Path C:\Program Files (x86)\UM Software La

Object identifier: Co_Co

Comments:

□ Train 3D

Generation of equations
○ Symbolic
● Numeric-iterative

Direction of gravity
ex:
ey:
ez: -1.0

Characteristic size: 1.00

Scene image: (no)

Figure 3.15. Object constructor

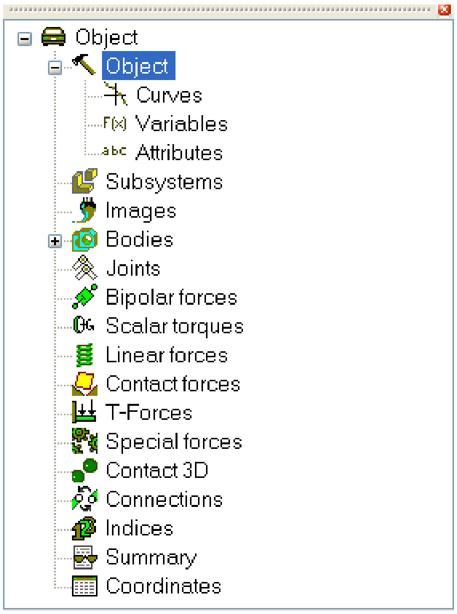
The object constructor allows describing an object or a multibody system as a set of standard elements: bodies, joints, force elements. Basic parts of the constructor are the following.

- **Object inspector** is used for input and modification of object elements as well as some other information about the object.
- **Animation window** displays the object or its part according to the active elements presented in the inspector. It can be also used for visual construction of models.
- **Tree of elements** presents the lists of all object's elements and organizes access to parameters of elements
- **List of identifiers** is used for modification of identifiers of the model. The list is the base of the full parameterization of UM models (Sect.3.5.2.4.2).
- **Tabs with components** can be also considered as a useful tool of the constructor. This tool allows adding to the model some simple standard elements.

The *drag-and-dock* technology is used for the element tree, list of identifiers and the inspector. They can be removed from the constructor window and placed on a separate window with the help of the mouse. If these tools are located in separate windows, the hot keys *F11*, *F12*, *Alt+I* are used to bring them in front, Sect. 3.4.5.

3.5.1.1. Tree of elements

An object is a multibody system, which consists of separate typical elements. Access to the elements is realized by means of the list of elements (Figure 3.15), visually (3.5.1.2) or with the help of hot keys (3.5.3).



- Object
  - Object
    - Curves
    - Variables
    - Attributes
  - Subsystems
  - Images
  - Bodies
  - Joints
  - Bipolar forces
  - Scalar torques
  - Linear forces
  - Contact forces
  - T-Forces
  - Special forces
  - Contact 3D
  - Connections
  - Indices
  - Summary
  - Coordinates

Figure 3.16. List of object elements

Click on a list item (figure 3.16) calls the corresponding information in the object inspector (3.5.2). The tree contains the following items:

- **Object** – general object options, gravity, background color, lists of variables, curves, sensors etc. (3.5.2.1).
- **Subsystems** – list of subsystems. For UM version with subsystem technique only, Sect. 3.6.3.
- **Images** – list of images, which are used for visualization of the scene, bodies and force elements (0).
- **Bodies** – list of **bodies** and their parameters (mass, moments on inertia, coordinates of centers of mass etc., Sect.3.6.9).
- **Joints** – **input** of joints (rotational, translational etc.) as well as coordinates of bodies (Sect.3.6.11).
- **Bipolar forces** – list of bipolar forces, i.e. forces acting along the axis of element, which connects two points of bodies (Sect.3.6.12.1). The force element is used for modeling dampers, leads etc.
- **Scalar torques** – a list of torques, acting by rotation of body relative to some axis, Sect. 3.6.12.3.

- **Linear forces** – list of generalized linear force elements described by 6×6 stiffness and damping matrices (Sect.3.6.12.3). The element is used for modeling springs, linear bushings etc.
- **Contact forces** – list of force elements, which models contact interaction between bodies (Sect.3.6.12.6).
- **T - forces** – list of forces and torques which components can be functions of time or kinematic functions, Sect. 3.6.12.7.
- **Special forces** – models of special force interactions (gearing, cams, combined friction, bushings etc. Sect. 3.6.12.8).
- **3D Contact** – setting for 3D contact model. Reserved for future use. For more detailed description of using 3D Contact see Sect. 3.6.9.7.
- **Connections** – a tool for assignment of attachment points for external force elements. For UM version with subsystem technique only, Sect. 3.5.2.3.1, 3.6.3.3.1.
- **Indices** – internal UM indices of object elements and coordinates, Sect. 3.5.2.3.2.
- **Summary** – contains information about correctness of the object description as well as lists of errors and warnings, Sect. 3.5.2.3.3.
- **Coordinates** – a list of current values of model coordinates; the tool is used for change of the coordinates, Sect. 3.5.2.3.4.

3.5.1.2. Animation window

3.5.1.2.1. Visualization of object elements

The whole object or active elements are visualized in the animation window depending on the window mode (Sect.3.5.1.2.2). The following types of visual elements are used for visualization of different elements:

- **Graphic objects (GO)** created by the user – for bodies, bipolar and generalized linear force elements, special force elements spring, rod constraint images (Sect. 3.6.8);

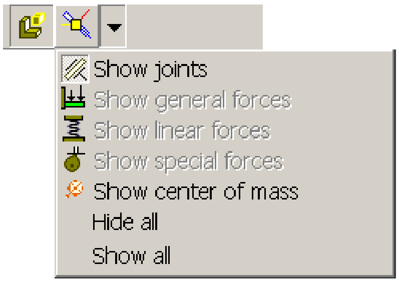


Figure 3.17. Visualization commands

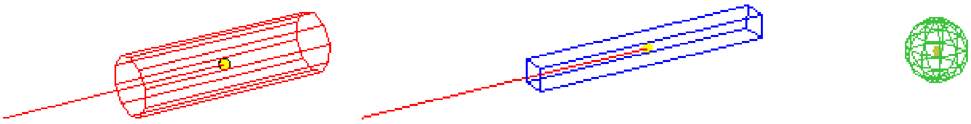


Figure 3.18. Icons for joints:

- **Icons** – for joints, T-force, linear and special force elements, bushings, external elements etc (Figure 3.17);
- **Points**.

Every type of listed visual elements has active regions, which are used for visual selection of the corresponding object elements by the mouse.

- **GO** – the active region is the whole image;
- **Icon** – the active region is a small neighborhood of the left bottom part pointed out by the arrow, e.g. for the joint icon:

- **Point** – active region is a small neighborhood of the point.

3.5.1.2.2. Modes of animation window

Animation window has two modes of visualization of an object. The button or the **Mode | Object/Element** command of a pop-up menu are used to switch them.

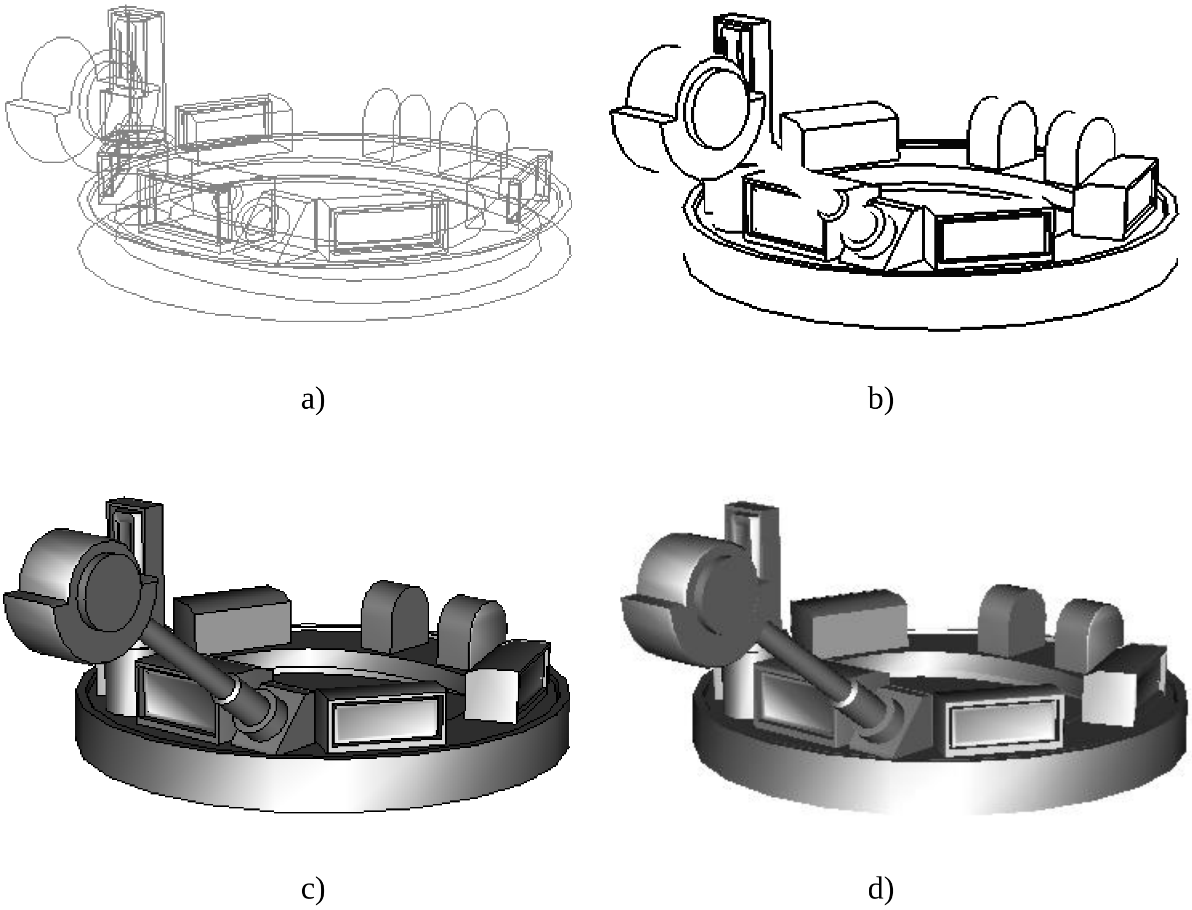
- **Whole object mode**

The whole object is visualized. In this mode, a mouse click on the active region of an image makes the corresponding element active (body, joint and force element, Sect. 3.5.1.2.1.

- **Single element mode**

A separate element is visible in this mode: GO, body, joint or force element (together with connected bodies).

Graphic modes



a) b)

c) d)

Figure 3.19. Different modes of visualization

- wired graphics, Figure 3.19a
- wired, invisible edges are hidden, Figure 3.19b
- surfaces with edges, Figure 3.19c
- surfaces without edges, Figure 3.19d

Perspective type is switched by the **Perspective** command of the popup menu or by the button on the tool panel.

Parameters of the perspective are changed in the window available by the **Window parameters** command of the popup menu.

3.5.1.2.3. Basic system of coordinates, pop-up menu

The basic system of coordinates (SC0) is optionally presented in the animation window. Use the **Coordinate system** command of the pop-up menu to visualize of hide the axes. Coordinates of all elements attached the base must be given in this SC. A color principle is used to identify the SC0 axes (RGB):

- axis X is red;
- axis Y is green;
- axis Z is blue.

A coordinate grid coincides with one of the coordinate planes. Use the **Window parameters** command of the pop-up menu to change the grid size and step.

Use the right mouse button to call a pop-up menu (Figure 3.20).

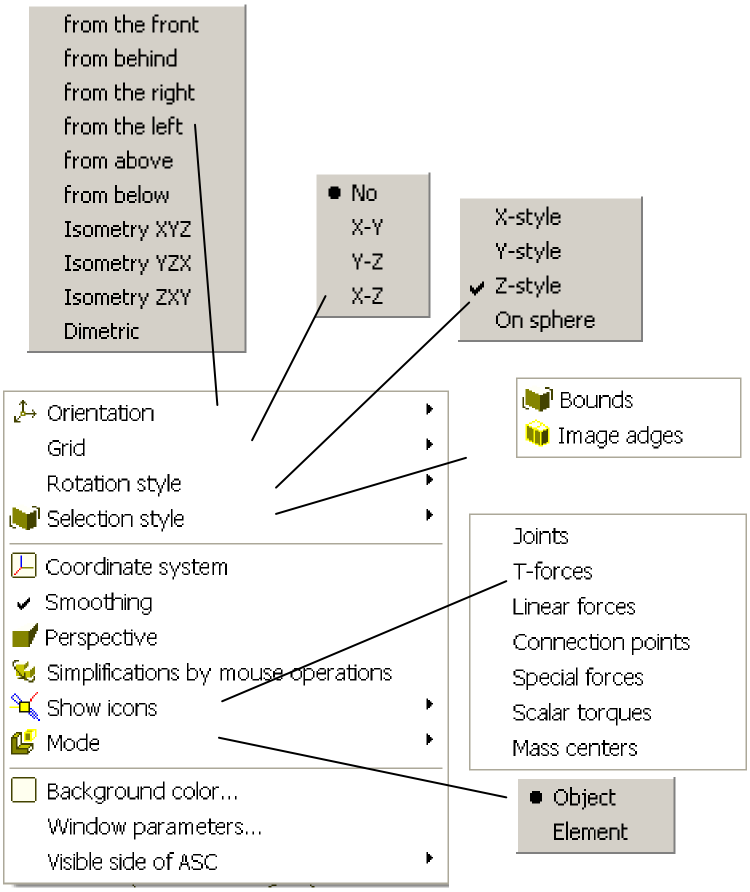


Figure 3.20. Popup menu of animation window

Menu commands:

- **Orientation** – choice of one of the standard object orientations.
- **Grid** – choice of one of the standard grid locations.
- Click an image of the SC0 axis to set the grid perpendicular to the corresponding axis
- **Rotational style** – choice the style of rotation for objects in the animation window: **Z-style** is used by default (from UM 3.0), **On sphere** – the style usually used in CAD systems.
- **Selection style** – the style of graphical visualization of an active element of the object (image contours or box rounding the element image).

- **Coordinate system** – turn on/off visualization of SC0.
- **Smoothing** – turn on/off of smoothing mode.
- **Perspective** – turn on/off of orthogonal projection.
- **Contour graphic mode** – is used to obtain contrast black-and-white image which is suitable for printing.
- **Show icons** – is used for visualization of icons for joints, force element of general type, generalized linear force element etc. (for the whole object mode in the animation window only, Sect.3.5.1.2.2).
- **Mode** – switching the animation window modes (whole object / active element, Sect.3.5.1.2.2).
- **Window parameters** – call of a window with perspective and grid parameters.
- **Background color** – setting the background color of the animation window.
- **Visible side of ASC** – mode of drawing the ASC surfaced imported from CAD. One side drawing is faster.

3.5.1.2.4. Tool bar

To clipboard
To file

– Copy the window to clipboard or to a file (bmp).

– Zoom in the selected area of animation window.

– Show all (F9).

– Zoom in/out of a selected point on the object: click the button and immediately click the left/right mouse button on an image point to zoom in/out the object and to shift the point in the window center. Use also *Alt+Shift + mouse click* on an object point.

– Shift mode (*Ctrl + left mouse button*).

– Zoom mode (*Shift + left mouse button*).

– Rotation mode (left mouse button).

- Mode of simplified drawing by mouse operation in animation window (rotation, shift). A parameter of the simplification is set on the **Object | Options | Animation** tab of inspector, the parameter **Bound of visibility by simplified drawing**, Sect. 3.5.2.1.2.

- switch on/off visual operations with mouse in animation window; the mode must be off is the visual adapter is too slow for such mouse operations.

– graphic mode in animation window, Sect. 3.5.1.2.2.

– Turn on/off perspective.

– Choice of one of standard views.

– Switch full object / single element mode (Sect.3.5.1.2.2).

– Show element icons in the full object mode.

- The buttons call the procedure for computations of constraint equations it the model has closed kinematic loops and cut joints, See Charter 2, Sect. *System graph. Closed kinematical loops; Theoretical foundations for solving constraint equations*.

See also Sect.3.5.5.3 for a list of hot keys.

3.5.1.2.5. Additional possibilities for object translation/rotation by mouse

Modes for shift, rotation of objects, zoom and so on can be activated by the mouse using buttons on the tool panel of animation window, Sect. 3.5.1.2.4. Here we consider some additional possibilities, in particular using keyboard keys.

Rotations:

- if the mouse cursor is , press the left mouse button, keep it and move the cursor on the windows;
- if the mouse cursor points a definite element of the object like a body, press the mouse wheel and keeping it shift the mouse cursor.

Shifts: press the Ctrl and keep it. If the mouse cursor is , press the left mouse button, keep it and move the cursor on the windows; if the mouse cursor points a definite element of the object like a body, press the mouse wheel and keeping it shift the mouse cursor.

Zoom:

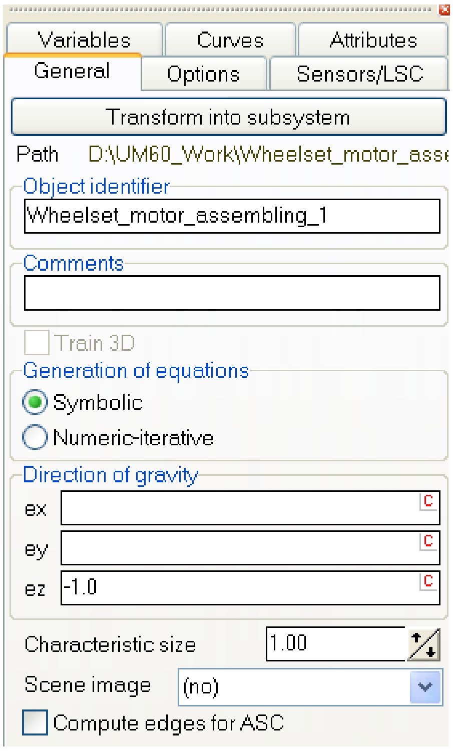
- rotate the mouse wheel forward/backward;
- press the *Shift* key and keep it; If the mouse cursor is , press the left mouse button, keep it and move the cursor on the windows up and down;
- press the *Shift+Alt* key and keep it. Move a bit the cursor until it looks like ⊕; point the cursor on the point for zoom and click the left (zoom in) or right (zoon out) mouse button.

3.5.2. Data inspector and some features of object element description

3.5.2.1. Object parameters and options

Use the Object tab of the inspector to set some parameters and options for the current model as well as to define lists of variables, sensors, curves etc.

3.5.2.1.1. “General” tab



Path D:\UM60_Work\Wheelset_motor_asse

Object identifier

Wheelset_motor_assembling_1

Comments

□ Train 3D

Generation of equations

● Symbolic

○ Numeric-iterative

Direction of gravity

ex

ey

ez -1.0

Characteristic size 1.00

Scene image (no)

□ Compute edges for ASC

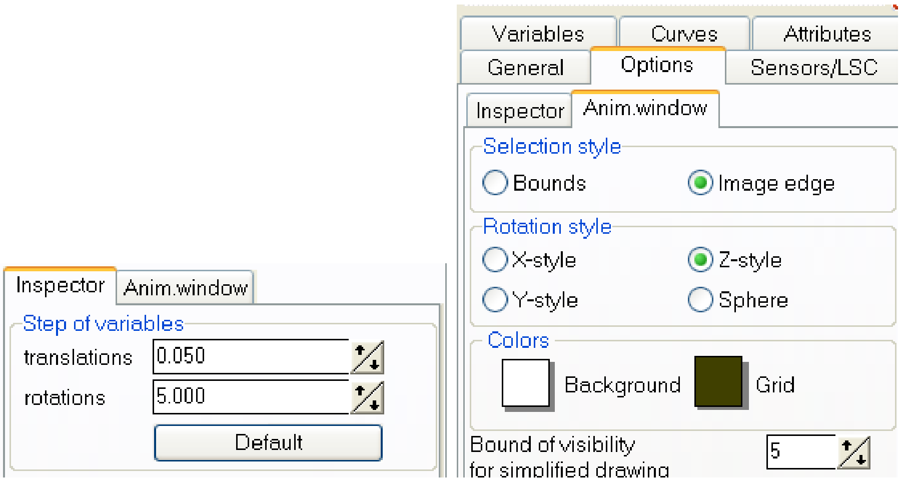
Figure 3.21. Object general parameters and options

The following parameters are set in the Object | General tab of the inspector, Figure 3.21.

- The **Transform into subsystem** button transforms all elements of the object into an included subsystem of the object. This tool is used by development of an object as a tree of subsystems as well as for creating components in particular suspensions in the module UM Tracked Vehicle. After running the operation, the active object includes one subsystem, which is equivalent to the object before making the operation.
- The **Path** label contains the path to the current model.
- **Object identifier** is used only if equations are generated in symbolic form. The identifier is included in the names of files with equations as well as in names of some standard structures within these files. In earlier version of um before UM6.0, the name of the object was used as the identifier, which imposed restrictions of the object name. Now the restrictions on the object name are removed.
- **Generation of equation** group allows the user to set either symbolic or numeric-iterative methods for generation of equations of motion, Sect. 3.9.

- **Direction of gravity** is set by a vector, which specifies the direction of gravity force relative to SC0. Vector components can be set as constant expressions or identifiers. To turn off the gravity, set zero value for the vector components. If the length of the vector is not equal to unity, the gravity acceleration decreases or increases proportionally. The direction of gravity for all subsystems of the objects is imposed by the main object. This means that the directions specified in the subsystems (both included and external) are ignored.
- **Characteristic size** allows decreasing/increasing the default size of images in the animation window. In particular, this parameter is used for obtaining proper vector sizes for small or large objects.
- **Scene image** drop-down menu is the assignment of a graphic object, which corresponds to fixed elements of the object as well as to environment. Press the **Delete** key to cancel the assignment of the scene image.
- **Compute edges for ASC** is visible if the model includes graphic elements of the ASC type imported from CAD programs without conversion of edges. The key can be used for automatic computation of edges and improving the images, see Sect. 0.

3.5.2.1.2. “Option” tab



Step of variables
translations 0.050
rotations 5.000

Selection style
○ Bounds ● Image edge

Rotation style
○ X-style ● Z-style
○ Y-style ○ Sphere

Colors
□ Background ■ Grid

Bound of visibility for simplified drawing 5

Figure 3.22. Tabs of the inspector and animation window options

The **Object | Options** tab contains values of stepwise changing angular and linear variables

10.00000000

when special buttons in edit boxes are used

Angular variables are measured in degrees within the Input program and in radians in the simulation program.

Use the **Object | Animation window** tab to set

- the selection style either by bounds or edges, Figure 3.23;
- mouse rotational style;
- the background and grid colors; click the color box by the mouse to choose the color;
- bound of visibility in percents by simplified drawing during rotations and shifts of object by the mouse.

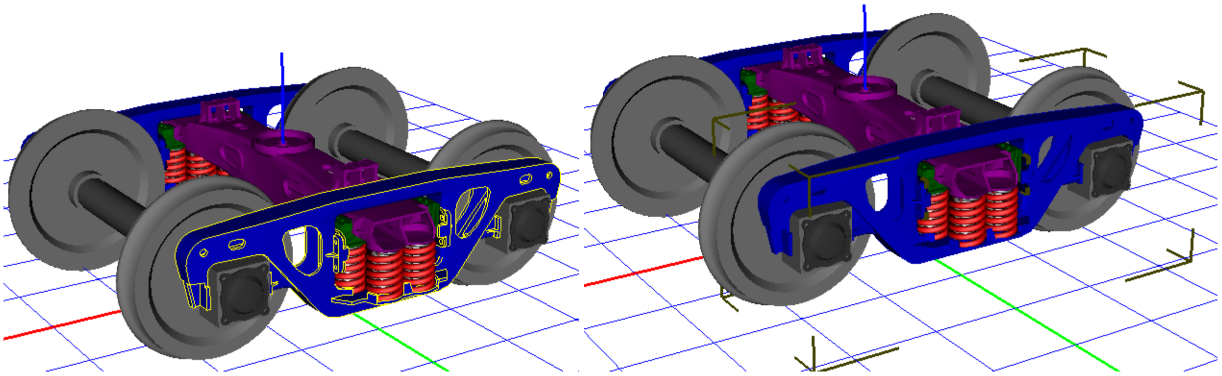
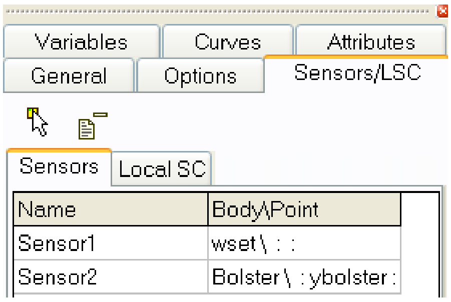


Figure 3.23. Selection by edges (left) and bounds (right)

3.5.2.1.3. “Sensors/LSC” tab



Name	Body\Point
Sensor1	wset\ : :
Sensor2	Bolster\ : ybolster :

Figure 3.24. List of sensors

A list of sensors is created on the **Object | Sensors/LSC | Sensors** tab, Figure 3.24. The **Object | Sensors/LSC | Local SC** tab is reserved for the future applications and is not used in UM7.0.

A sensor is a point with parameterized coordinates relative to some of the bodies. During the simulation, the user can get plots of kinematic characteristics of the body related to the sensor: coordinates, velocity, acceleration of the sensor point.

The advantage of using the sensors in comparison with usual kinematic variables created with the Wizard of variables consists in the possible parameterization of point coordinates, see Chapter 4, Sect. *Wizard of variables* | *Sensors*.

Steps by creation of a sensor:

- create necessary connection points for a body, Sect. 3.6.9.6;
- click on the button ↖ in inspector; connection points appear in animation window, Figure 3.25;

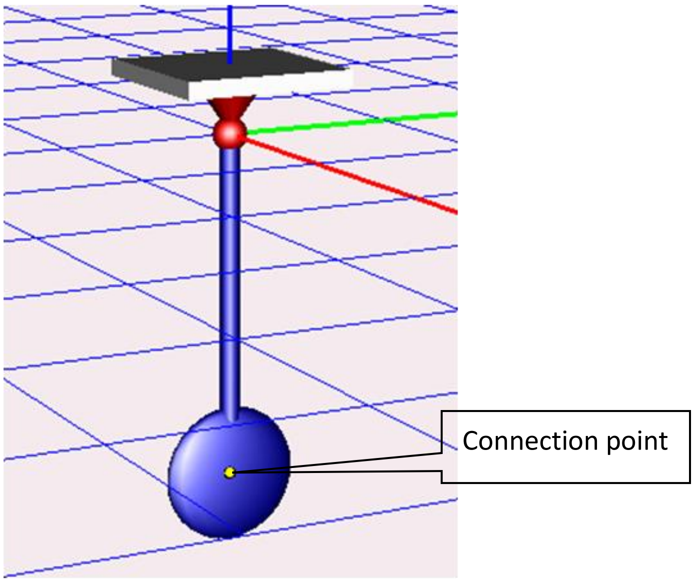


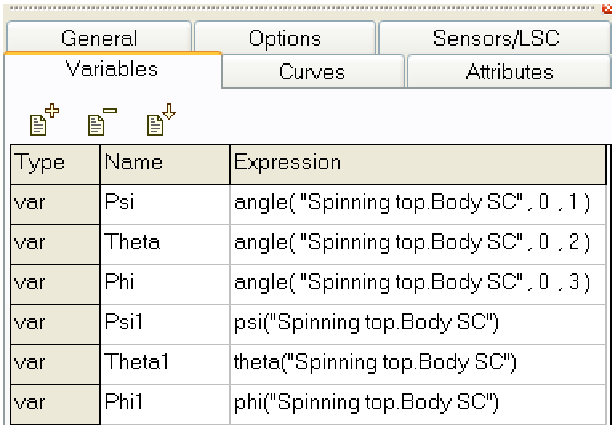
Figure 3.25. Connection point in animation window

- select a point by the mouse to add it to the list as a sensor;
- rename the sensor if necessary.

Remark. Creating the parameterized kinematic variables is possible with the elements of the *list of variables*, Sect. 3.5.2.1.4.

Model: {UM Data}\SAMPLES\LIBRARY\Pendulum.

3.5.2.1.4. “Variables” tab

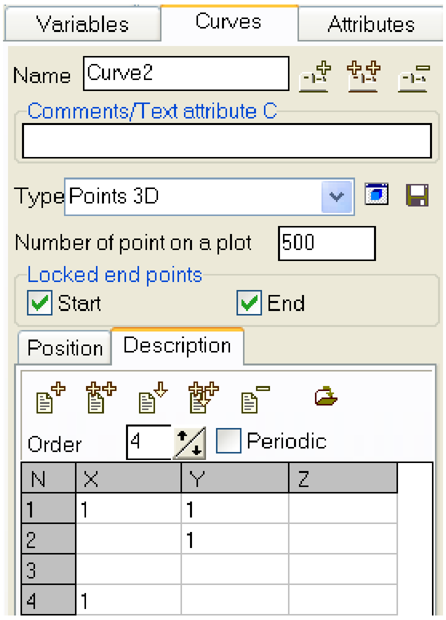


Type	Name	Expression
var	Psi	angle("Spinning top.Body SC" , 0 , 1)
var	Theta	angle("Spinning top.Body SC" , 0 , 2)
var	Phi	angle("Spinning top.Body SC" , 0 , 3)
var	Psi1	psi("Spinning top.Body SC")
var	Theta1	theta("Spinning top.Body SC")
var	Phi1	phi("Spinning top.Body SC")

Figure 3.26. Example of list of variables

The **Object | Variables** tab, Figure 3.26, is used for development of a list of variable functions, which extend considerably description of non-standard force interactions, surfaces and kinematic variables, Sect. 3.5.2.4.8.

3.5.2.1.5. “Curves” tab



Name Curve2

Type Points 3D

Number of point on a plot 500

☑ Start ☑ End

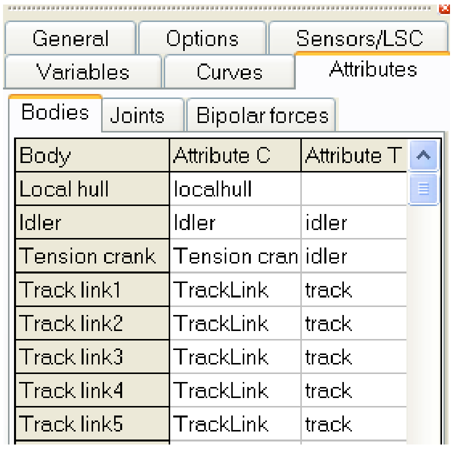
Order 4 □ Periodic

N	X	Y	Z
1	1	1	
2		1	
3			
4	1		

Figure 3.27. Example of a curve

With the **Object | Curves** tab, Figure 3.27, the user develops a set of 3D curves, which are used in the Point-Curve contact force element, Sect. 3.6.12.6.3. Detailed information about creating a curve can be found in Sect. 3.6.7.

3.5.2.1.6. “Attributes” tab



Body	Attribute C	Attribute T
Local hull	localhull	
Idler	Idler	idler
Tension crank	Tension cran	idler
Track link1	TrackLink	track
Track link2	TrackLink	track
Track link3	TrackLink	track
Track link4	TrackLink	track
Track link5	TrackLink	track

Figure 3.28. Example of use of attributes in the model of a tracked vehicle

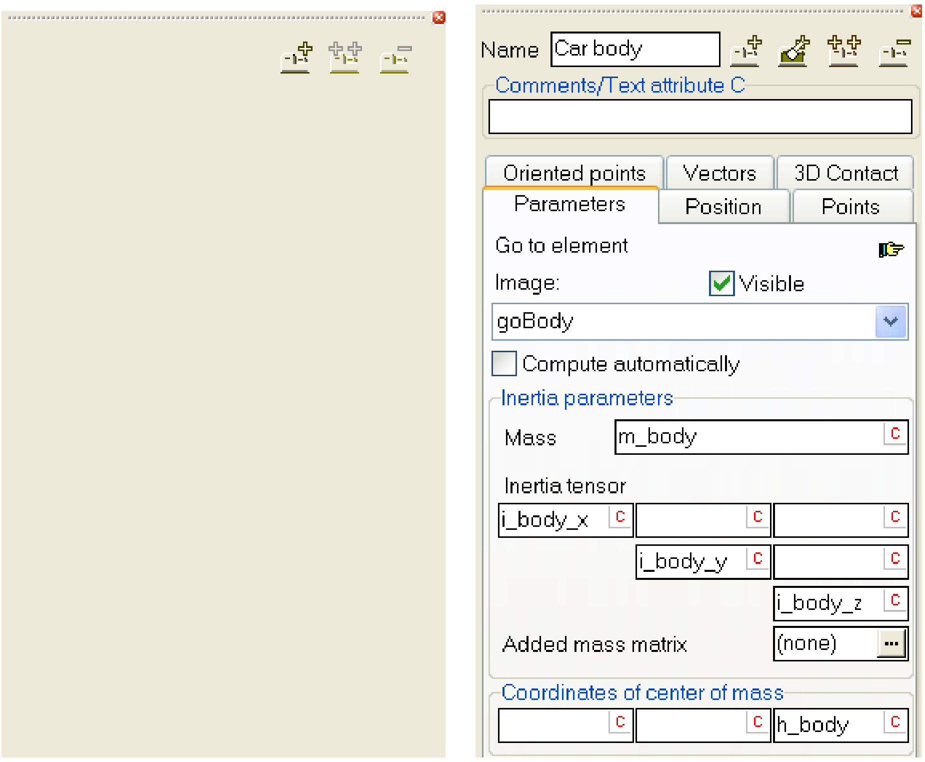
The **Object | Attributes** tab, Figure 3.28, is used for internal identification of some elements of models of tracked vehicle, Chapter 18.

3.5.2.2. Lists of elements of a definite type

Each object (multibody system) consists of sets of elements, most of which are grouped as lists. Every list contains elements of a single type, e.g., lists of bodies, joints, bipolar force elements and so on. Each element of a list has a *name*, which is an arbitrary set of symbols. The name of element is the base of its identification, and the name must be *unique within the corresponding list*, that is, it is not allowed setting one name for two elements of the same type (e.g., for two bodies). Elements of different lists as well as elements, which belong to different subsystems, may have the same names. So, the body and its image (or a joint) can have equal names.

The tree of elements of the model is used for access to parameters of an element of a list, Sect. 3.5.1.1.

Standard interfaces are used to manage the lists within the data inspector.



Name: Car body

Comments/Text attribute C

Oriented points | Vectors | 3D Contact | Parameters | Position | Points

Go to element

Image: ☑ Visible

goBody

□ Compute automatically

Inertia parameters

Mass: m_body

Inertia tensor: i_body_x, i_body_y, i_body_z

Added mass matrix: (none)

Coordinates of center of mass: h_body

a b

Figure 3.29. Lists

Figure 3.29a shows an empty list in the inspector, Figure 3.29b shows an element of the list of bodies.

Edit box for the name and three buttons are located in the top of the tab:

adds a new element to the list;

creates an exact copy of the current element and adds it to the list;

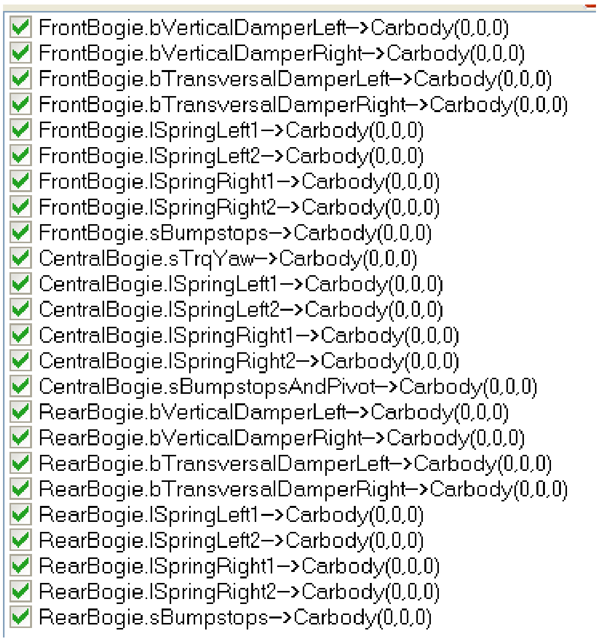
deletes the current element.

See also Sect. 3.5.5.4.

Remark. Press the *Enter* key after modification of the name else the changes can be lost.

3.5.2.3. Auxiliary tabs in inspector

3.5.2.3.1. External connections

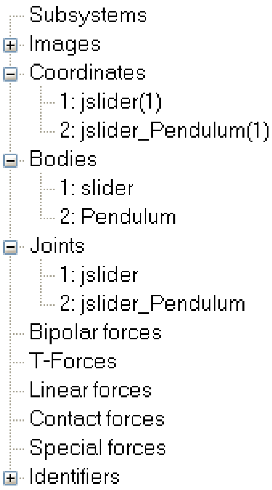


☑ FrontBogie.bVerticalDamperLeft–>Carbody(0,0,0)
☑ FrontBogie.bVerticalDamperRight–>Carbody(0,0,0)
☑ FrontBogie.bTransversalDamperLeft–>Carbody(0,0,0)
☑ FrontBogie.bTransversalDamperRight–>Carbody(0,0,0)
☑ FrontBogie.lSpringLeft1–>Carbody(0,0,0)
☑ FrontBogie.lSpringLeft2–>Carbody(0,0,0)
☑ FrontBogie.lSpringRight1–>Carbody(0,0,0)
☑ FrontBogie.lSpringRight2–>Carbody(0,0,0)
☑ FrontBogie.sBumpstops–>Carbody(0,0,0)
☑ CentralBogie.sTrqYaw–>Carbody(0,0,0)
☑ CentralBogie.lSpringLeft1–>Carbody(0,0,0)
☑ CentralBogie.lSpringLeft2–>Carbody(0,0,0)
☑ CentralBogie.lSpringRight1–>Carbody(0,0,0)
☑ CentralBogie.lSpringRight2–>Carbody(0,0,0)
☑ CentralBogie.sBumpstopsAndPivot–>Carbody(0,0,0)
☑ RearBogie.bVerticalDamperLeft–>Carbody(0,0,0)
☑ RearBogie.bVerticalDamperRight–>Carbody(0,0,0)
☑ RearBogie.bTransversalDamperLeft–>Carbody(0,0,0)
☑ RearBogie.bTransversalDamperRight–>Carbody(0,0,0)
☑ RearBogie.lSpringLeft1–>Carbody(0,0,0)
☑ RearBogie.lSpringLeft2–>Carbody(0,0,0)
☑ RearBogie.lSpringRight1–>Carbody(0,0,0)
☑ RearBogie.lSpringRight2–>Carbody(0,0,0)
☑ RearBogie.sBumpstops–>Carbody(0,0,0)

Figure 3.30. Example of the list of external connections

The tab contains a list of external elements of the object, which are included in subsystems, Figure 3.30. The list is used for assignment of the second bodies as well as connection points to external force elements, Sect. 3.6.3.3.

3.5.2.3.2. Indices



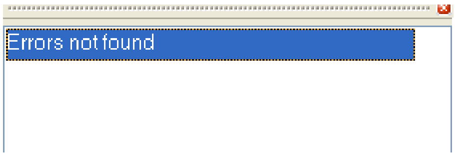
```
Subsystems
⊞ Images
⊟ Coordinates
    1: jslider(1)
    2: jslider_Pendulum(1)
⊟ Bodies
    1: slider
    2: Pendulum
⊟ Joints
    1: jslider
    2: jslider_Pendulum
Bipolar forces
T-Forces
Linear forces
Contact forces
Special forces
⊞ Identifiers
```

Figure 3.31. Example of the list of element indices

The tab contains a list of object elements including all elements of subsystems. Indices of elements are used by programming in UM environment (obsolete), see Chapter 5.

3.5.2.3.3. Summary

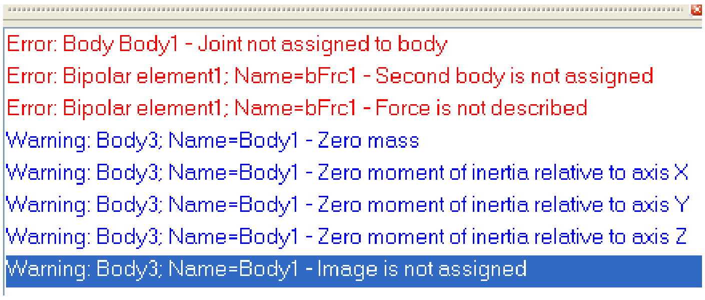
The summary tab contains information about correctness of description of the object.



Errors not found

Figure 3.32. Description of the object does not contain errors

The model is ready for simulation if no errors found, Figure 3.32. Warnings do not affect the status of ready object but can lead to simulation errors.

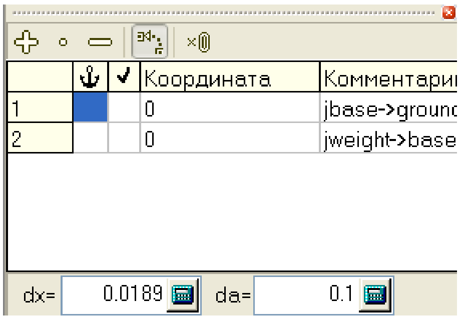


Error: Body Body1 - Joint not assigned to body
Error: Bipolar element1; Name=bFrc1 - Second body is not assigned
Error: Bipolar element1; Name=bFrc1 - Force is not described
Warning: Body3; Name=Body1 - Zero mass
Warning: Body3; Name=Body1 - Zero moment of inertia relative to axis X
Warning: Body3; Name=Body1 - Zero moment of inertia relative to axis Y
Warning: Body3; Name=Body1 - Zero moment of inertia relative to axis Z
Warning: Body3; Name=Body1 - Image is not assigned

Figure 3.33. Model contains errors and warnings

When errors found, Figure 3.33, the simulation is not allowed for the object. Click on the error of warning line opens the corresponding element in the inspector.

3.5.2.3.4. Coordinates



	⚓	✓	Координата	Комментари
1			0	jbase->grounc
2			0	jweight->base

dx= 0.0189 da= 0.1

Figure 3.34. Example of the list of coordinates

The **Coordinates** tab is used for changing values for joint coordinates and body positions, Figure 3.34. The list contains all coordinates of the model including coordinates in subsystems.

See Chapter 4, Sect. *Choice and automatic calculation of the initial conditions* for more details.

If the coordinate tab is active, the user can change positions of bodies by the mouse in the animation window, Figure 3.35:

- move the mouse cursor to the desired body until it changes to ✥;

- press the left mouse button and drag the body.

Remark. Shift of bodies by the mouse is usually applied in the case of simple models with small number of bodies. The operation is not recommended for models of rail vehicles, cars, and tracked vehicles.

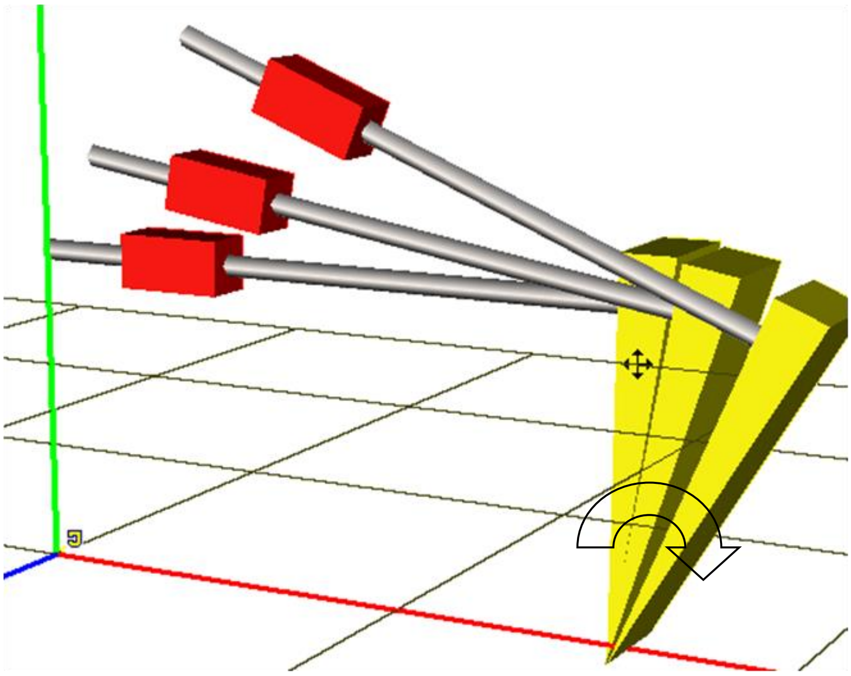


Figure 3.35. Example of shift of bodies by mouse

3.5.2.4. Data types

Information for each element of an object (element parameters) is entered in boxes in the object inspector. UM uses several standard data types. The user must know features of each data type to work with UM correctly.

A very important feature of object description using UM is the data parameterization. This means that many element parameters could be set not only by its numeric values but by expressions including numbers, identifiers, operations and functions. Consider the basic types of data presented in UM.

3.5.2.4.1. Numeric constants

UM uses standard syntax for numbers.

Examples: 1.23, 0.256e-3

3.5.2.4.2. Identifiers

Identifier is a set of symbols, which includes Latin letters, digits and character “_”.

The first symbol in the identifier cannot be a digit or character “_”.

Identifiers with the character “_” as the first symbol are reserved for internal presentation of identifiers in equations of motion generated by the program.

Reserved words of Pascal and C languages cannot be used as identifiers.

The program verifies syntax of entered expressions. If a new identifier is found, it is added to the *list of identifiers* of the object (Sect. 3.5.3).

Example of correct identifiers:
mass_1 length_of_rod cdiss cstiff

Examples of wrong identifiers:
2mass – the first symbol is the digit 2;
length – the first symbol is the character “”;
mass% - prohibited character “%”;
do, as, while – reserved words of the Pascal language.

There exist two types of identifiers:

- Identifier – number;
- Identifier – expression.

Values of identifiers of the first type can be changed both in the Input and in the Simulation programs. Identifiers of the second type are presented by arbitrary expressions, which include

- numbers;

- identifiers of the first and the second types;
- standard functions (Sect. 3.5.2.4.3).

Chains of calculations may be programmed with the help of identifiers of the second type.

An example of a chain including identifiers of the both types is shown below.

Name	Expression	Value	Comments
Mass	1.12		Mass of rod
length	0.55		Length of rod
ix	mass*length^2/12	0.0282333333	Moment of inertia of the rod relative to X axis
iy	ix	0.0282333333	Moment of inertia of the rod relative to Y axis

Remark. An expression can only include identifiers located above the current identifier.

The same principle is used in the built-in calculator (the menu **Tools | Symbolic calculator** command).

3.5.2.4.3. Standard functions and constants

The following **standard functions** can be used for description of data of several types (explicit function, identifiers-expressions):

- *sin, cos* – trigonometric functions, arguments are set in radians;
- *arcsin, arccos, arctan* – inverse trigonometric functions (rad);
- *arctan2(x, y)* computes an angle $\alpha,\ \tan\alpha = x / y$ in radians in the interval from -π to π; quadrant for the angle is defined by signs of arguments *x,y* as if $x = \sin\alpha,\ y = \cos\alpha$;
- *exp* – natural exponent;
- *ln* – natural logarithm;
- *abs* – absolute value;
- *sign* – $$sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$$
- ^ – power function, the expression *a^b* corresponds to a^b, the exponent must be an integer if the base is negative;
- *sqr* – square;
- *sqrt* – root square;
- *Heavi* – *Heavi(x)* = $\begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$
- $$\text{if(c , v1 , v2 , v3)} = \begin{cases} v1, c < 0 \\ v2, c = 0 \\ v3, c > 0 \end{cases}$$

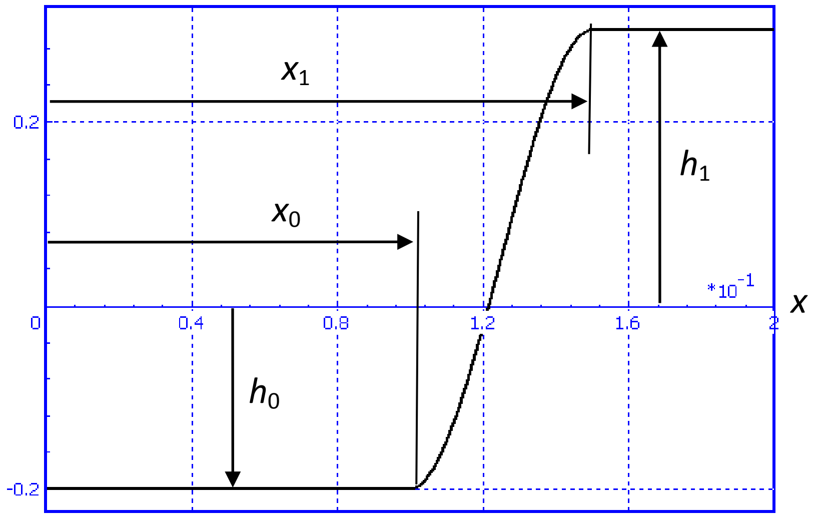


Figure 3.36. Step function

- step(x , x0 , h0 , x1 , h1)= $\begin{cases} h_0, \; x < x_0 \\ h_0 + (h_1 - h_0)d^2(3-2d), \; d = \dfrac{x - x_0}{x_1 - x_0} \\ h_1, \; x > x_1 \end{cases}$

As a rule, the Step function is used for a smooth but fast transition of expression from one value to another. Example of the function step(t , 0.1 , -0.2 , 0.15 , 0.3) is shown in Figure 3.36.

- The ‘bodyinertia’ function is considered as a standard function as well, and can be included in expression of any type, see Sect. 3.5.2.4.4.

Standard constants

pi : number $\pi = 3.1415926536...$

e : number e=2.7182818285

rtod: factor converting radians to degrees, e.g. arctan(1)*rtod=45;

dtor: factor converting degrees to radians, e.g. 90*dtor=pi/2;

itom = 0.0254: factor converting Inch TO Meter;

mtoi = 1/itom: factor converting Meter TO Inch;

pton = 453.6/1000*9.81: factor converting Pound-force TO N;

ntop = 1/pton: factor converting inch to meter N TO Pound-force.

3.5.2.4.4. Function bodyinertia – inertia parameters

‘Bodyinertia’ is a standard function, which allows the user to get inertia parameters of bodies and include them in expression of any types: identifiers (Sect. 3.5.3), force element parameters, variables (Sect. 3.5.2.4.8). Function syntax:

bodyinertia(_inertia_name , _body_name)

Here

_inertia_name is a standard designation of inertia parameter from the following list (please do not mix up with identifiers from the identifier list!):

m – mass

ix – moment of inertia relative to the X axis;

iy – moment of inertia relative to the Y axis;

iz – moment of inertia relative to the Z axis;

ixy, ixz, iyz – centrifugal moments of inertia;

cx, cy, cz – coordinates of center of mass relative to the body-fixed SC.

The function returns numerical values, which exactly correspond to the inertia parameters specified by the user for the corresponding body, Sect. 3.6.9.2.

_body_name – name of body.

The function is recommended to be used when inertia parameters are computed by the body image.

Function writing can be effectively done with the help of the expression editor, Sect. 0.

Example:

sqr(2*pi*f)*bodyinertia(m , "bogie1.body1")

The mass of body ‘bogie1.body1’ is included in this expression.

bodyinertia(ix , "body")

This function returns the moment of inertia of the body ‘bogie1.body1’.

Example of a model:

{Path to UM data}\Samples\Library\BodyInertia_test

In this model, the function ‘bodyinertia’ is used in the identifier list, in expression for bipolar force, and in the list of variables.

3.5.2.4.5. Constant symbolic expressions

A constant symbolic expression is an expression, which contains

- identifiers;
- numbers;
- additions, subtractions, divisions and multiplication;
- standard functions (Sect. 3.5.2.4.3, 3.5.2.4.4).

It is not allowed the use of identifier *t* (time).

Example of correct constant symbolic expressions:
sqrt(2)*b1+sqrt(a1+a2)/2

The constant symbolic expressions can be used for the most of the element parameters (inertia and geometric parameters, coordinates of attachment points of force elements, sizes of graphic elements, coefficients of stiffness and damping and so on). The corresponding edit boxes in the data inspector have the standard interface

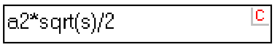

a2*sqrt(s)/2

Figure 3.37. Exit box for constant expressions

The letter ‘c’ in the right top part of the box points out that the parameter can be a *constant* expression. Double click on the box or click on the ‘c’ letter calls a tool for visual construction of the expressions, Sect. 0.

3.5.2.4.6. Expression – explicit function

The expression of this type includes

- numbers;
- identifiers;
- standard functions;
- standard variables (*t*, *x*, *v*, *p1*, *p2*, *p* – depending on type of function).

Double click the edit box to call a tool for writing the expressions, Sect. 0. The corresponding window contains the lists of identifiers and allowed variables as well as buttons with standard functions.

Consider types of explicit functions

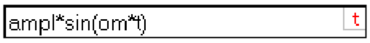
3.5.2.4.6.1. Function of time - t

The standard variable is t – time. The function is used for description of joints

- joint of generalized type, elementary transformation of types *tt*, *rt* (Sect. 3.6.11.7.4);
- rotational and translational joints in the cases when the joint coordinate is an explicit function of time (Sect. 3.6.11.4);

- components of a T-force, Sect. 3.6.12.7.

The corresponding edit boxes in the data inspector have the following standard interface:



ampl*sin(om*t) t

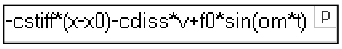
The letter ‘t’ in the right top part of the box points out that the expression is a time function.

3.5.2.4.6.2. Scalar force and torques of the Expression type: x, v, t

See Sect. 3.6.12.2.8. *Standard variables are t* (time) as well as two additional variables *x, v*. The function is used for description of mathematical models of forces in the following cases:

- *Bipolar force element*; *x* is the length of the element; v is the time derivative of the length, Sect. 3.6.12.3;
- *Scalar torque*; *x* is the rotation angle; v is the time derivative of the angle, Sect. 3.6.12.4;
- *Joint forces* in the case of *joint of general type* (elementary transformations *rv*, *tv*, type of force is expression, Sect. 3.6.11.7.3, as well as for translational and rotational joints, Sect. 3.6.11.4; *x* is the value of coordinate, *v* is its time derivative;
- Description of an axle force in the case of a special force of the *Combined friction type*;
- Components of *bushing* force elements of the generalized type, Sect. 3.6.12.8.6.2.

The corresponding edit boxes in the data inspector have the following standard interface (Figure 3.38).



-cstiff*(x-x0)-cdiss*v+f0*sin(om*t) p

Figure 3.38. Exit box for Pascal/C expressions

The letter ‘p’ – Pascal – in the right top part of the box points out that the data is a function of *x*, *v*, *t*.

3.5.2.4.6.3. Curves as expressions

The standard identifier *p* is used for parameterization of curves in the following cases:

- 2D profile or axis curves as expressions in the case of a profiled graphic element, Sect. 3.6.8.2.8;
- 3D curves as expressions, Sect 3.6.7.2.

3.5.2.4.6.4. Functions of description of parametrical graphic elements

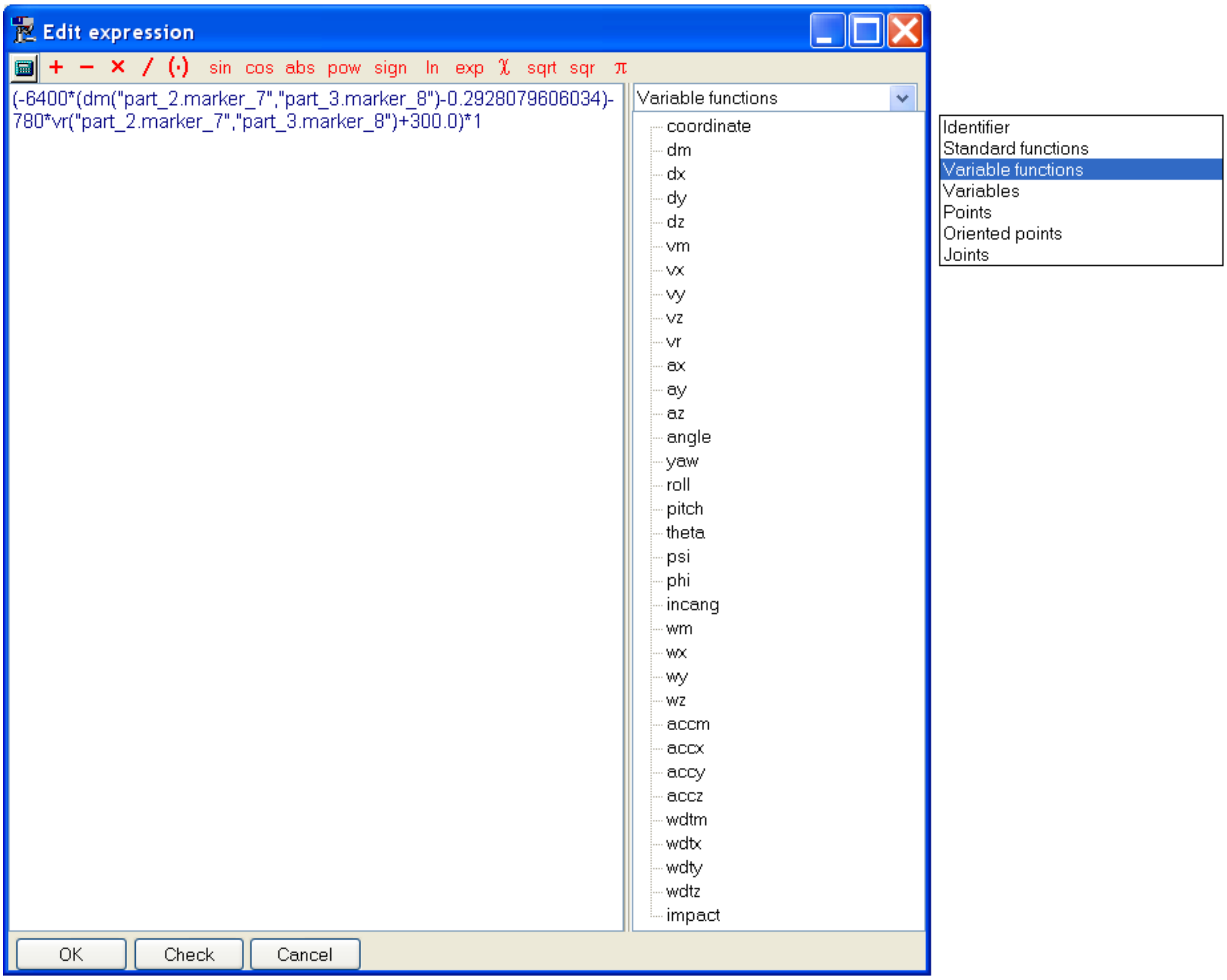
Standard variables are *p1*, *p2* parameterize a surface or a curve, Sect. 3.6.8.2.7.

3.5.2.4.6.5. Z –surfaces

Surfaces $z = f(x, y)$ in 3D space are used in description of *Point -- Z –surface, Circle -- Z – surface and Sphere - Z – surface* contact force elements, Sect. 3.6.12.6.8.

Standard variables are *p1*, *p2* parameterize a Z-surface. The corresponding edit boxes in the data inspector have the standard interface shown in Figure 3.38.

3.5.2.4.7. Kinematic functions



(-6400*(dm("part_2.marker_7","part_3.marker_8")-0.2928079606034)-780*vr("part_2.marker_7","part_3.marker_8")+300.0)*1

Figure 3.39. Expression editor and list of kinematic functions

The user can use kinematic functions depending on relative positions and velocities of bodies by description of some force elements.

The following force elements allow using the following kinematic functions:

- scalar force of the **Expression** type, Sect. 3.6.12.2.8 in description of force elements
    - bipolar force, Sect. 3.6.12.1,
    - joint forces, Sect. 3.6.11.7.3,
    - scalar torque, Sect. 3.6.12.3,
    - axial force in models of combined friction, Sect. 3.6.12.8.7.5;
    - components of force and torque in case of bushing of generalized type, Sect. 3.6.12.8.6.2;

- components of T-force in the case of the **Expression** type, Sect. 3.6.12.7.

Access to the kinematic functions is realized by the expression edit window, Figure 3.39, Sect. 0. The function template is added to the expression after double click on the function name in the list. The user should set real parameters of function in the template. Example of template:

vx(_to_point , [_from_point] , [_SC_component] , [_SC_deriv]).

The following designations are used as template arguments:

_to_point is the first connection point;

_from_point is the second connection point;

_SC_component is the local system of coordinates (SC), in which vector components are defined;

_SC_deriv is the local system of coordinates (SC), in which vector time derivative is evaluated;

_to_SC is the first local SC;

_from_SC is the first local SC.

All systems of coordinates are specified by the oriented connection points, Sect. 3.6.9.6.2.

Optional arguments are written in square brackets. To omit the optional argument, a white space can be set. Example (the second argument is omitted):

dy(“Bogie.Frame.Point 2”, , “Body.Local SC”)

3.5.2.4.7.1. Function coordinate

The function returns a joint coordinate or first/second time derivative of the coordinate.

Template of the function:

coordinate(_joint , _index , _type)

Arguments:

_joint – name of joint;

_index –index of the coordinate starting with 1;

_type – type of output (0 - coordinate, 1 – first time derivative of the coordinate, 2 – second time derivative of the coordinate).

Examples

coordinate(“jBody” , 4 , 0)

The function returns a value of the fourth coordinate in joint jBody.

coordinate(“Bogie1.jFrame” , 2 , 1)

The function returns the first derivative in time of the second coordinate in joint Bogie1.jFrame.

3.5.2.4.7.2. Functions dm, dx, dy, dz

These functions determine the length (dm) or projections (dx, dy, dz) of a vector $\overrightarrow{\mathrm{BA}}$, which connects two fixed points A, B of a pair of bodies, Figure 3.40. The projections are evaluated relative to the SC C.

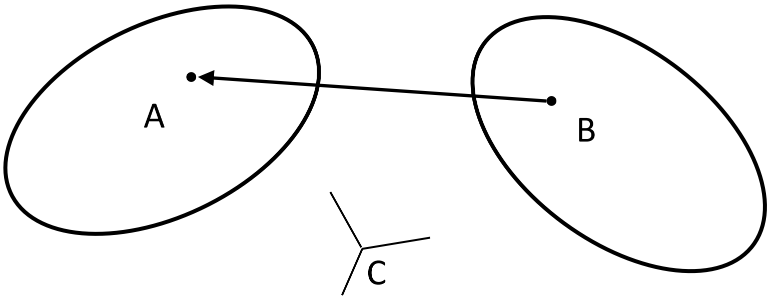


Figure 3.40. Vector connecting fixed points of two bodies

Mathematical model

The vector $\overrightarrow{\mathrm{BA}}$ is computed according to the formula

$$\overrightarrow{\mathrm{BA}} = \mathbf{r}_A - \mathbf{r}_B ,$$

where $\mathbf{r}_A, \mathbf{r}_B$ are radius vectors to points A, B with respect to SC0.

Function templates:

dm(_to_point , [_from_point])

dx(_to_point , [_from_point] , [_SC_component])

dy(_to_point , [_from_point] , [_SC_component])

dz(_to_point , [_from_point] , [_SC_component])

Arguments:

_to_point is name of connection point A;

_from_point is name of connection point B;

_SC_component is name of oriented connection point C.

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, it is located in the origin of SC0.

If SC C is omitted, the projection is computed on axis of SC0.

Examples.

dx("Car body.Center of mass")

The function returns the coordinate x of connection point “Center of mass” of body “Car body” in SC0.

dm("Car body.Center of mass", “Bogie1.Frame.Point 2”)

The function returns the distance between two points.

dy(“Bogie1.Frame.Point 2”, "Car body.Center of mass", “Car body.Local SC”)

The function returns the y projection of vector connecting points “Bogie1.Frame.Point 2”, "Car body.Center of mass" on the SC specified by the oriented connection point “Car body.Local SC”.

3.5.2.4.7.3. Functions vm, vx, vy, vz, vr

These functions determine the magnitude (vm), projections (vx, vy, vz) or the radial component (vr) of a difference of velocities of two points A, B $\mathbf{v}_{\mathrm{BA}}$ of a pair of bodies, Figure 3.40. The projections are evaluated relative to the SC C. Velocities are computed relative to SC D (not shown in the figure).

Mathematical model

Velocity $\mathbf{v}_{BA}$ is evaluated according to the formula

$$\mathbf{v}_{\mathrm{BA}} = \mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}.$$

where $\mathbf{v}_A, \mathbf{v}_B$ are the velocities of points A, B relative to SCD. The result depends on the body, which the SCD belongs to, and does not depend on the position and orientation of SCD relative to this body.

The bipolar or radial velocity is equal to the separation velocity between points A, B. It is computed as

$$v_{\mathrm{BA},r} = (\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}}) \cdot \mathrm{e}_{\mathrm{BA}},$$

where e_{BA} is the unit vector from point B to point A. The result does not depend on SCD, and velocities are computed relative to SC0.

Function templates

vm(_to_point , [_from_point], [_SC_deriv])

vx(_to_point , [_from_point] , [_SC_component] , [_SC_deriv])

vy(_to_point , [_from_point] , [_SC_component] , [_SC_deriv])

vz(_to_point , [_from_point] , [_SC_component] , [_SC_deriv])

vr(_to_point , [_from_point])

Arguments:

_to_point: name of connection point A;

_from_point: name of connection point B;

_SC_component: name of oriented point C.

_SC_deriv: name of oriented connection point D, i.e. SC relative to which the time derivative are computed.

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, it is located in the origin of SC0.

If SC C is omitted, the projection is computed on axis of SC0.

If SC D is omitted, velocities are calculated relative to SC0.

Examples.

vy(“Bogie1.Frame.Point 2” , “Car body.Local SC” , “Car body.Local SC”, “Car body.Local SC”)

The function returns the y component of the velocity of point “Bogie1.Frame.Point 2” relative to SC “Car body.Local SC”.

vm("Car body.Center of mass")

Magnitude of velocity of point "Car body.Center of mass" relative to SC0.

3.5.2.4.7.4. Functions ax, ay, az

These functions return angles of rotation of SCA about one of the axis of SCB, Figure 3.41. The angle of rotation relative to a definite axis is calculated correctly, if rotations about two other axes are small and do not exceed 10 degrees.

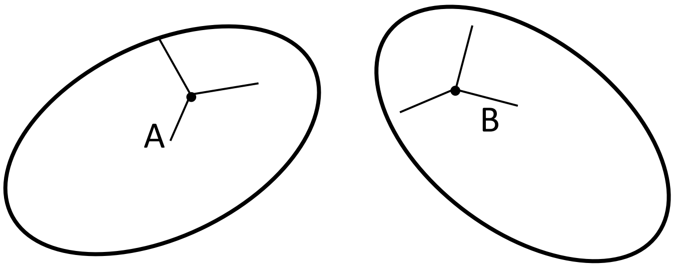


Figure 3.41. On definition of systems of coordinates

Mathematical model

Angles of rotation are calculated as

$$\text{ax} : \alpha = \arctan2(-\mathbf{k}_A \cdot \mathbf{j}_B,\ \mathbf{k}_A \cdot \mathbf{k}_B),$$

$$\text{ay} : \beta = \arctan2(-\mathbf{k}_A \cdot \mathbf{i}_B,\ \mathbf{k}_A \cdot \mathbf{k}_B),$$

$$\text{az} : \gamma = \arctan2(\mathbf{i}_A \cdot \mathbf{j}_B,\ \mathbf{i}_A \cdot \mathbf{i}_B),$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the unit vectors along the axes of SCA and SCB.

Function templates

ax(_to_SC , [_from_SC])

ay(_to_SC , [_from_SC])

az(_to_SC , [_from_SC])

Arguments

_to_SC is name of oriented point A;

_from_SC is name of oriented point B.

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, rotation of SCA relative to SC0 is considered.

Example

az("Body.Local SC")

The function returns the angle of rotation of body Car body relative to the axis Z of SC0.

3.5.2.4.7.5. Function angle

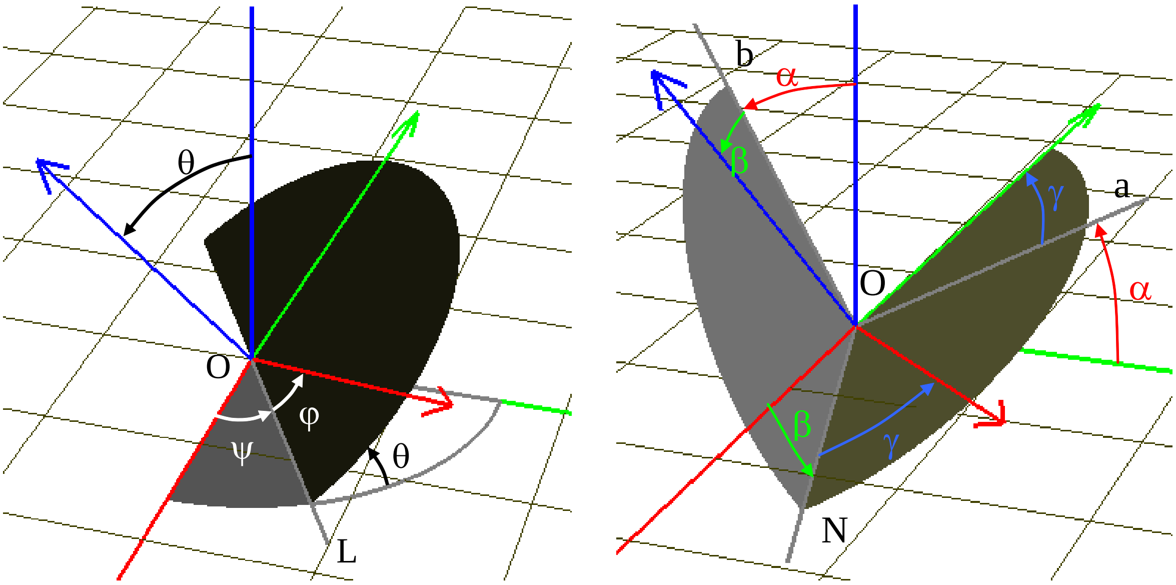


Figure 3.42. Euler angles (left) Cardan angles (right)

The function returns one of three orientation angles of SCA relative to SCB, Figure 3.41. Orientation angles as a sequence of three rotations are introduced in Chapter 2, Sect. *Joint with six degrees of freedom*. Examples of orientation angles: Euler angles (ψ, θ, φ), sequence of rotations 3,1,3; Cardan angles (α,β,γ), sequence of rotations 1,2,3, Figure 3.42.

Table 1

Types of orientation angles

_angle_type	Sequence	Comments
0	(3,1,3)	Euler angles: precession, nutation, intrinsic rotation
1	(1,2,3)	Cardan angles
2	(1,3,2)	
3	(3,2,1)	Yaw, pitch, roll
4	(3,1,2)	Wheelset angles
5	(2,1,3)	
6	(2,3,1)	
7	(1,2,1)	
8	(1,3,1)	
9	(2,1,2)	

10	(2,3,2)	
11	(3,2,3)	

Function template

angle(_to_SC, _angle_type, _index, [_from_SC])

Arguments

_to_SC: name of oriented point A;
_angle_type: type of orientation angles, Table 1;
_index: index of angle 1, 2 or 3;
_from_SC: name of oriented point B;

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, orientation of SCA relative to SC0 is considered.

Example

angle("Car body.Local SC", 3, 1)
angle("Car body.Local SC", 3, 2)
angle("Car body.Local SC", 3, 3)
The function returns yaw, pitch or roll angles of Car body relative to SC0.

Models:

{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Yaw_Pitch_Roll;
{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Euler_angles.

3.5.2.4.7.6. Functions yaw, pitch, roll

The functions return yaw, pitch or roll angle. Sequence of rotations: (3,2,1), Sect. 3.5.2.4.7.5.

Function templates

yaw(_to_SC, [_from_SC])
pitch(_to_SC, [_from_SC])
roll(_to_SC, [_from_SC])

Arguments

_to_SC is name of oriented point A;
_from_SC is name of oriented point B;

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, orientation of SCA relative to SC0 is considered.

Correspondence to the *angle* function:

yaw(_to_SC, [_from_SC]) = angle(_to_SC, 3, 1, [_from_SC]);

pitch(_to_SC, [_from_SC]) = angle(_to_SC, 3, 2, [_from_SC]);
roll(_to_SC, [_from_SC]) = angle(_to_SC, 3, 3, [_from_SC]);

Model:
{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Yaw_Pitch_Roll.

3.5.2.4.7.7. Functions psi, theta, phi (Euler angles)

The functions return one of the Euler angles. Sequence of rotations: (3,1,3), Sect. 3.5.2.4.7.5.

Function templates
psi(_to_SC, [_from_SC])
theta(_to_SC, [_from_SC])
phi(_to_SC, [_from_SC])

Arguments
_to_SC is name of oriented point A;
_from_SC is name of oriented point B;

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.
If point B is omitted, orientation of SCA relative to SC0 is considered.

Correspondence to the *angle* function:
psi(_to_SC, [_from_SC]) = angle(_to_SC, 0, 1, [_from_SC]);
theta(_to_SC, [_from_SC]) = angle(_to_SC, 0, 2, [_from_SC]);
phi(_to_SC, [_from_SC]) = angle(_to_SC, 0, 3, [_from_SC]);

Model:
{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Euler angles.

3.5.2.4.7.8. Functions wm, wx, wy, wz

The functions return the magnitude (wm) and projections (wx, wy, wz) of a difference of angular velocities of SCA and SCB ω_{BA} of a pair of bodies (angular velocity of body A relative to body B), Figure 3.40. The projections are evaluated relative to the SC C.

Mathematical model

Velocity ω_{BA} is evaluated according to the formula

$$\omega_{BA} = \omega_A - \omega_B$$

where ω_A, ω_B are the angular velocities of SCA and SCB.

Function templates

wm(_to_SC , [_from_SC])

wx(_to_SC , [_from_SC] , [_SC_component])

wy(_to_SC , [_from_SC] , [_SC_component])

wz(_to_SC , [_from_SC] , [_SC_component])

Arguments:

_to_SC: name of oriented point A;

_from_SC: name of oriented point B.

_SC_component: name of oriented point C.

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, rotation of SCA relative to SC0 is considered.

If SC C is omitted, the projection is computed on axis of SC0.

Example

wx("Car body.Local SC")

The function returns angular velocity of the Car body relative to the axis X of SC0.

wz("Bogie1.Frame.Local CK", "Car body.Local SC", "Car body.Local SC")

The function returns angular velocity of "Bogie1.Frame" relative to body “Car body”, projection on the axis Z of the local SC "Car body.Local SC".

3.5.2.4.7.9. Function incang

The function calculates the angle between two vectors $\overline{BA}$ and $\overline{BC}$. The angle value lies in the interval from 0 to π. If one of the vectors is zero, the function returns zero value.

Mathematical model

$$\alpha = \arccos\left(\frac{\overline{BA}\cdot\overline{BC}}{\left|\overline{BA}\right|\left|\overline{BC}\right|}\right).$$

Function template

incang(_to_point1 , _from_point , _to_point2)

Arguments

_to_point1 : name of connection point A;

_from_point2 : name of connection point B;

_to_point2 : name of connection point C;

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

3.5.2.4.7.10. Functions accm, accx, accy, accz

Functions return the magnitude (accm) or projections (accx, accy, accz) of vector a_{BA} - the acceleration of point A relative to SCB, Figure 3.41. Projections are calculated on axes of SCC (not shown in the figure).

Mathematical model

Vector $\mathbf{a}_{BA}$ is computed as

$$\mathbf{a}_{BA} = \mathbf{a}_A - \mathbf{a}_B - \boldsymbol{\varepsilon}_B \times (\mathbf{r}_B - \mathbf{r}_A) - \boldsymbol{\omega}_B \times \boldsymbol{\omega}_B \times (\mathbf{r}_B - \mathbf{r}_A) - 2\boldsymbol{\omega}_B \times \mathbf{v}_{BA}$$

where $\mathbf{a}_A, \mathbf{a}_B$ are the accelerations of points A and B, $\boldsymbol{\omega}_B, \boldsymbol{\varepsilon}_B$ are the angular velocity and acceleration of SCB, $\mathbf{v}_{BA}$ is the velocity of point A relative to SCB.

Function templates

accm(_to_point , [_from_point])
accx(_to_point , [_from_point] , [_SC_component])
accy(_to_point , [_from_point], [_SC_component])
accz(_to_point , [_from_point], [_SC_component])

Arguments:

_to_point: name of connection point A;
_from_point: name of connection point B.
_SC_component: name of connection point C.

Names of connection points include long names of bodies. As a rule, the name should be put in quotes.

If point B is omitted, acceleration of point A relative to SC0 is considered.

If SC C is omitted, the projection is computed on axis of SC0.

Remark. These kinematic functions cannot be included in description of force elements.

3.5.2.4.7.11. Functions wdtm, wdtx, wdty, wdtz

Functions return the magnitude (wdtm) or projections (wdtx, wdty, wdtz) of vector $\boldsymbol{\varepsilon}_{BA}$ - the angular acceleration of SCA relative to SCB, Figure 3.41. Projections are calculated on axes of SCC (not shown in the figure).

Mathematical model

Vector $\boldsymbol{\varepsilon}_{BA}$ is computed as

$$\boldsymbol{\varepsilon}_{BA} = \boldsymbol{\varepsilon}_A - \boldsymbol{\varepsilon}_B - \boldsymbol{\omega}_B \times \boldsymbol{\omega}_A$$

where $\boldsymbol{\varepsilon}_A, \boldsymbol{\varepsilon}_B$ are the angular accelerations of SCA and SCB, $\boldsymbol{\omega}_A, \boldsymbol{\omega}_B$ are the angular velocities of SCA and SCB.

Function templates

```
wdtm( _to_SC , [_from_SC] )
wdtx(_to_SC , [_from_SC], [_SC_component]  )
wdty(_to_SC , [_from_SC], [_SC_component]  )
wdtz(_to_SC , [_from_SC], [_SC_component]  )
```

Arguments:

_to_SC: name of oriented point A;

_from_SC: name of oriented point B.

_SC_component: name of oriented point C.

Names of connection points include long names of bodies. As a rule, the name should be written in quotes.

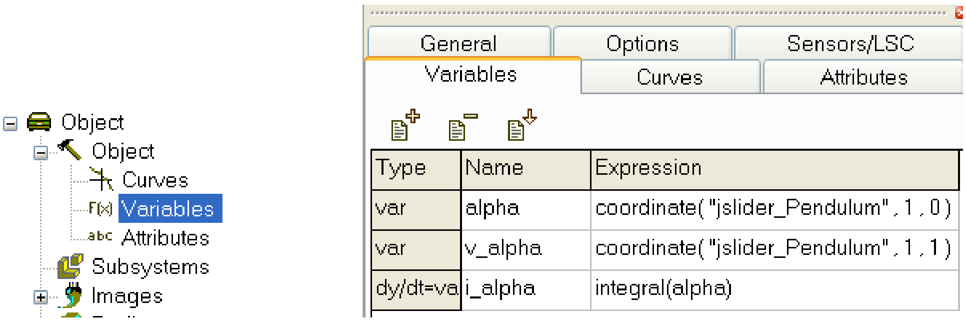
If point B is omitted, angular acceleration of SCA relative to SC0 is considered.

If SC C is omitted, the projection is computed on axis of SC0.

Remark. These kinematic functions cannot be included in description of force elements.

3.5.2.4.8. List of variables

List of variables is an efficient tool for development by the user of non-standard force interactions, surfaces, curves as well as kinematic variables evaluated during the object simulation. List of variables is used by conversion of MSC.ADAMS models in UM format, Sect. 3.11.



Type	Name	Expression
var	alpha	coordinate("jslider_Pendulum",1,0)
var	v_alpha	coordinate("jslider_Pendulum",1,1)
dy/dt=va	i_alpha	integral(alpha)

Figure 3.43. List of variables in inspector

The **Object | Variables** tab is used for development of a list of variables, Figure 3.43. The following buttons add and delete variables:

- add a new variable to the end of the list;

- delete selected variable;

- insert a new variable before the selected one.

A variable is an expression, which includes

- numbers,
- identifiers from the list of identifiers of the current model, Sect. 3.5.2.4.2, 3.5.3,
- standard identifier of time t, arguments of scalar force *x*, *v*, parameter of a curve *p*, parameters of a surface *p*1, *p*2;
- kinematic functions, Sect. 3.5.2.4.7;
- variables from the current list;
- standard functions and constants, Sect. 3.5.2.4.3
- function **Integral**, Sect. 3.5.2.4.9.

The program determines the type of variable. Allowed types:

- function of time, Sect. 3.5.2.4.6.1;
- function x, v, t for use in description of scalar forces, Sect. 3.5.2.4.6.2;
- kinematic functions, Sect. 3.5.2.4.7;
- function of argument p for description of a curve, Sect. 3.5.2.4.6.3;
- function of arguments p1, p2 for description of a surface, Sect. 3.5.2.4.6.4, 3.5.2.4.6.5.

The variables can be used by description of the following elements of the model:

- forces listed in Sect. 3.5.2.4.7;
- graphic objects, Sect. 3.6.8.2.7, 3.6.8.2.8;
- contact surfaces (Z-Surfaces, Sect. 3.6.12.6.8) and curves (Sect. 3.6.7, 3.6.12.6.3).

Remark. The user must avoid circling of variable description.

Examples of use of the variables:

- control force in the model of inverted pendulum, the model {UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_functions\Inverted_pendulum UM;
- plots of kinematic variables by simulation, the models;

{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Yaw Pitch Roll;
{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_Functions\Euler angles.

3.5.2.4.9. Integral function and adding differential equations

The **Integral** function allows the user to specify additional ordinary differential equations (ODE)

$$\frac{dy}{dt} = f(t, y, \ldots) .$$

Format of the corresponding variable in the list of variables is as follows, Sect. 3.5.2.4.8:

[Name of variable], Integral(f(t,[Name of variable],…)).

Arguments of the right hand side of the equations f are

- t,
- the variable itself,
- any variable in the list, which type is a kinematic variable or an integral.

Therefore, the user can specify both a single ODE, and a system of ODE.

Additional differential equations specified by the user are solved together with equations of motion of the model. To make this process compatible, the first order ODE is replaced by the equivalent second order ODE

$$\frac{d^2 z}{dt^2} = f(t, y, \ldots), y = \frac{dz}{dt} .$$

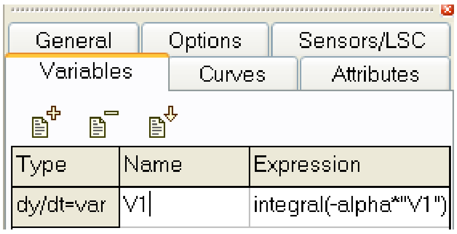
Initial values for additional variables are specified in the standard manner. It is important to know that the initial value y(0) correspond to the *velocity value* in the table of initial conditions.

Examples.

1. ODE

$$\frac{dy}{dt} = -\alpha y$$

is specified by the variable *V*1 in Figure 3.44. In this example the variable name is *V*1, and the parameter α is set by the identifier *alpha*.



Type	Name	Expression
dy/dt=var	V1	integral(-alpha*"V1")

Figure 3.44. Variable as a solution of differential equation

2. Example of a variable, which is an integral of an rotations angle in a joint, is shown in Figure 3.43, see the model {UM Data}\SAMPLES\LIBRARY\Variables and Kinematic functions\Inverted pendulum UM;

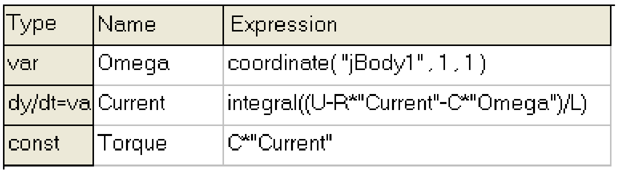
3. Simplified model of a direct current motor

Consider a simplified linear model of a direct current motor

$$L\frac{dI}{dt} = U - RI - C\omega,$$

$$M = CI$$

Here *L*, *R* are the inductance and the active coil resistance, *I* is the current in the armature coil, *U* is the voltage, *M* is the torque on the motor shaft, ω is the shaft angular velocity, *C* is motor constant. The implementation of the model as a list of variables is shown in Figure 3.45. It is supposed that the shaft angular velocity is equal the angular velocity in joint jBody1. The torque is computed as the variable of the same name and must be applied to the shaft as the joint torque.



Type	Name	Expression
var	Omega	coordinate("jBody1",1,1)
dy/dt=va	Current	integral((U-R*"Current"-C*"Omega")/L)
const	Torque	C*"Current"

Figure 3.45. Implementation of the motor model

Remark. Use of arguments of a scalar force x, v is not allowed in the differential equations. The corresponding kinematic variables must be applied instead, Sect. 3.5.2.4.8. For instance, if x and v are the joint *coordinate* and its time derivative, the coordinate function is recommended, Sect. 3.5.2.4.7.1.

3.5.2.4.10. External functions

Using *external function* is directly connected with programming in the UM environment based on a *Control file*. As a rule, these functions are used when the corresponding mathematical model is too complicated for description as an implicit function (Sect. 3.5.2.4.6).

Using external functions requires generation of equations of motion in symbolic form as well as external compiling the equations, Sect 3.9.2.

There exist three types of external functions, which differ in arguments.

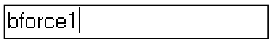
1. Time functions (*t*) are used for joint of generalized type, elementary transformation of types *tt,rt* (Sect. 3.6.11.7.4) as well as for rotational and translational joints in the cases when the joint coordinate is an explicit function of time (Sect. 3.6.11.4).

2. Functions of three arguments (*x, v, t*) are used for description of scalar force of **external** type; Sect. 3.6.12.2.8. The list of the corresponding force elements is

- *Bipolar force element*; *x* is the length of the element; *v* is the time derivative of the length, Sect. 3.6.12.3;
- *Scalar torque; x* is the rotation angle; *v* is the time derivative of the angle, Sect. 3.6.12.4;
- *Joint forces* in the case of *joint of general type* (elementary transformations *rv, tv*, type of force is *expression*, Sect. 3.6.11.7.3, as well as for translational and rotational joints, Sect. 3.6.11.4; *x* is the value of coordinate, *v* is its time derivative;
- Description of an axle force in the case of a special force of the *Combined friction* type;
- Components of *bushing* force elements of the generalized type, Sect. 3.6.12.8.6.2.

3. Function of two arguments (*p1, p2*) are used for description of Z-surfaces (surfaces given by the function *z = f(x, y))* in the cases of graphic element (type - Z-surfaces) (Sect. 3.6.8.2.9) and contact forces (Z-sphere).

To describe an external function, the user should enter its name (identifier) in the corresponding edit box of the inspector without arguments, for instance,


bforce1

Syntax rules for name of function as the same as for identifier (Sect. 3.5.2.4.2).

UM generates a *template* for each external function in the control file. This means, that special functions will be added to the control file, where the external function will be initialized by zero values. The user should rewrite the corresponding procedures.

Consider a template of a time function. Let the identifier *alpha* be used as the name of an external function. UM inserts the following procedure in the control file Cl[IdentifierOfObject]:

```
procedure alpha( _isubs : integer; _t : real; var _Value, _dValue, _ddValue : real_ );
var
  _ : _platfVarPtr;
begin
  _ := _PzAll[SubIndx[_isubs]];
  _Value := 0;
  _dValue := 0;
  _ddValue := 0;
end;
```

The input parameters are *_isubs* (the global index of subsystem), *_t* – the current time value. The output: value of function (identifier *_Value*) as well as its first and second derivatives *(_dValue, _ddValue).*

Wrong programming of derivatives leads to wrong simulation results.

Consider a template for a function of *(t, x, v)*. Let the identifier *bforce1* was used for external function corresponding to a scalar force. UM inserts the following function in the control file Cl[IdentifierOfObject]:

```
function bforce1( _isubs : integer; _t, _x, v : real ) : real_;
var
  _ : _vehicleVarPtr;
begin
  _ := _PzAll[SubIndx[_isubs]];
  Result := 0;
end;
```

The input parameters are *_isubs* (the global index of subsystem), *_t* is the current time value, *_x*, *_v* are the current *x* and *v* values. The user should change the function code to calculate the output value *Result*.

Remarks

1. External functions require generation of equations of motion in the symbolic form. In the case of numeric-iterative generation of equations, an error is detected.
2. Functions of the one and same type, which have coinciding identifiers, are identified. That is, only one template of function or procedure will be generated for them in the control file. Different identifiers must be used for external functions of different types.
3. Detailed information about the control file and programming in the UM environment can be found in Chapter 5 of the user's manual.
4. The Simulation program calls external functions automatically.
5. After adding or deleting external functions, the user should verify the correctness of the *old* control file.

3.5.2.4.11. Time function using text file

Here we consider how to set dependences on time of angular and translational coordinates with the help of text files. The file can contain both full-scale test and simulation results.

Coordinates as time functions are realized in the following joints

- generalized joint, elementary transformations *tt*, *rt* (3.6.11.7.4);
- translational and rotational joints when the coordinate is a time function (3.6.11.4).

Format of a text file

A text file with a time function should contain two columns separated by space symbols. The first column contains time in seconds starting with zero or a small value. The second column contains the corresponding values of the function in meters for a translational coordinate and in radians for an angular coordinate.

First symbol in comment lines should be %.

The file should be created beforehand and located in the directory of the model, which uses it.

If UM does not find the file, zero value is set for the corresponding function.

Creation of files with a time function as a simulation result

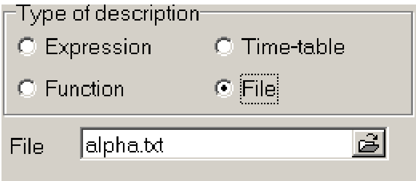
Each plot in graphic windows can be saved in a text file after simulation of a UM model (Chapter 4, Sect. *Graphical window | Copying graphs to clipboard, text file and file of calculated variables*). The file format matches the above requirements if

- % symbol is set as a prefix for comments (Chapter 4. Sect. *Options of simulation program | General*), otherwise comments should be deleted from file manually;
- one variable is saved;
- time is laid off as abscissa.

Fragment of an automatically generated compatible text file with a time function:

```
%
% 1 - time
% 2 - dyWheelset4 [Lateral position of Wheelset4]
%
2.00000002337219E-7 2.82372854E-15
1.03125004097819E-2 2.03257468E-6
2.09375005215406E-2 7.46718570E-6
3.21874991059303E-2 1.76279409E-5
4.21875007450581E-2 3.07774899E-5
```

Standard interface for setting the file



Type of description
○ Expression ○ Time-table
○ Function ● File

File: alpha.txt

Figure 3.46. Time function from file

To set the name of the file, use the 📂 button or write the name directly.

Note 1. UM uses a spline interpolation of discrete file data to get function value in intermediate time moments as well to compute the first and the second derivative, which are necessary for simulation.

Note 2. The user should take care of a sufficient smoothness of data in file.

Note 3. When the current simulation time exceeds the latest time point in the file, the function value is constant equal the latest one in the file.

3.5.2.4.12. Timetable as a method of description of time functions

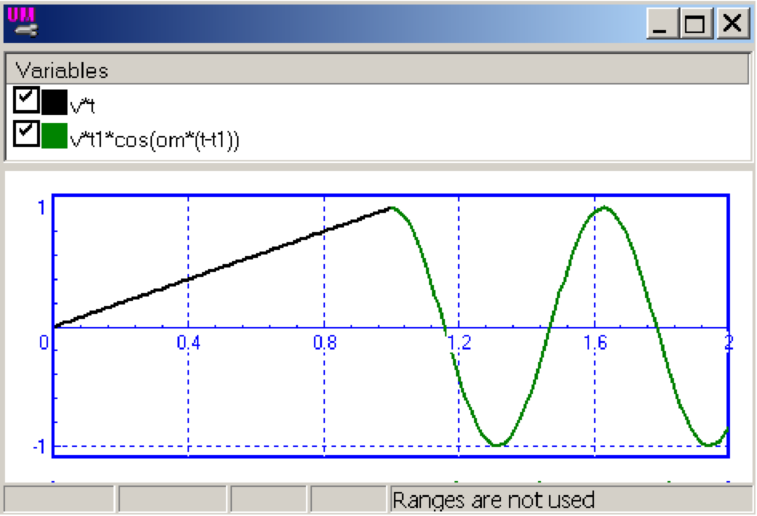


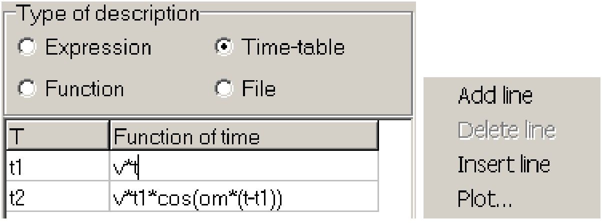
Figure 3.47. Example of a timetable function

Timetable is a generalization of time function description by an expression. This method is used if the function can be described by different symbolic expressions on several time intervals.

For instance, the function in Figure 3.47 satisfies the following relations:

$$f(t) = \begin{cases} vt, \; t \in [0, t_1] \\ vt_1 \cos(\omega(t - t_1)), \; t \in t[_1, t_2] \end{cases}$$

A standard interface is used for setting such dependencies:



Type of description
○ Expression ● Time-table
○ Function ○ File

T	Function of time
t1	v*t
t2	v*t1*cos(om*(t-t1))

Figure 3.48. Interface for timetable

Use a pop-up menu to add, delete insert a line into the timetable, Figure 3.48. The *Plot* item is used for plotting the functions.

The table can contain any number of lines.

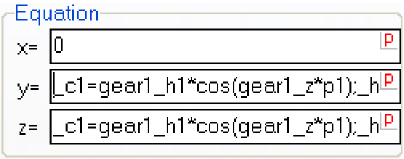
Time in the left column can be set by expressions (identifiers *t*1, *t*2 in our example).

Note 1. The user should take care of a continuity of the function.

Note 2. When the current simulation time exceeds the latest time point in the timetable, the function value is constant equal to the latest one in the table.

3.5.2.4.13. Expression editor

A special editor can be used for writing long expressions. To call the editor, double click on the edit box with the expression. Figure 3.49 shows edit boxes for description of a surface by parametric expressions (a) and the content of one of the boxes in the editor (b).

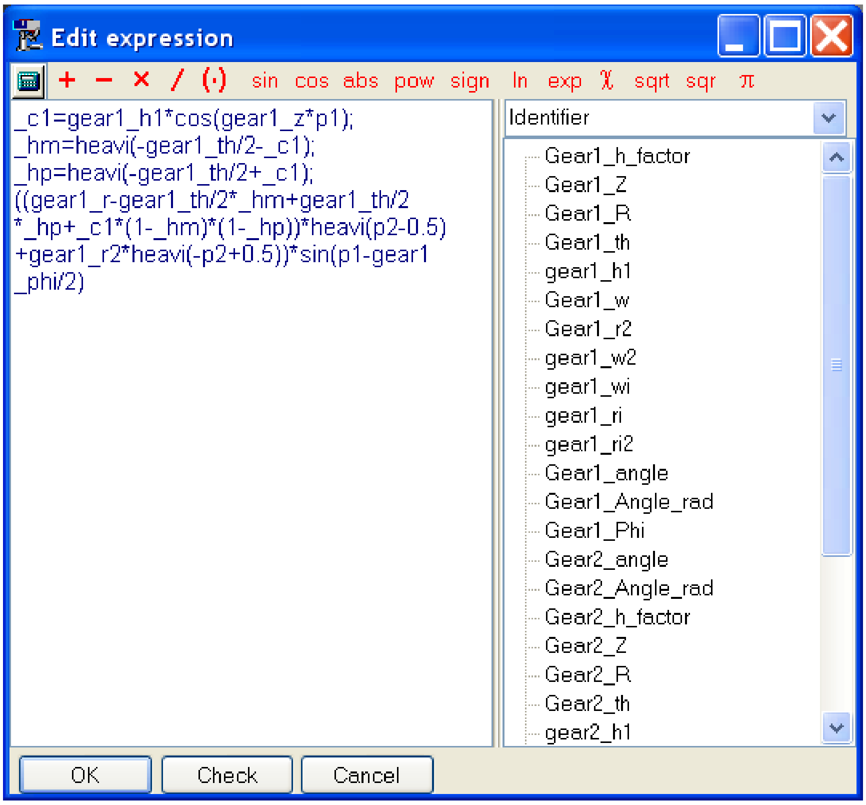


x= 0

y= _c1=gear1_h1*cos(gear1_z*p1);_h

z= _c1=gear1_h1*cos(gear1_z*p1);_h

a



```
_c1=gear1_h1*cos(gear1_z*p1);
_hm=heavi(-gear1_th/2-_c1);
_hp=heavi(-gear1_th/2+_c1);
((gear1_r-gear1_th/2*_hm+gear1_th/2
*_hp+_c1*(1-_hm)*(1-_hp))*heavi(p2-0.5)
+gear1_r2*heavi(-p2+0.5))*sin(p1-gear1
_phi/2)
```

b

Figure 3.49. Edit box (a) and expression editor (b)

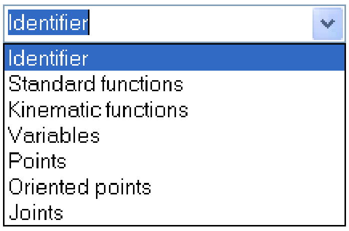


Figure 3.50. Selection of available elements of expressions

Depending on the expression type, the editor organizes access to data, variables and functions. In the most general case the following lists are available in the right part of the window:

Identifier

List of object identifiers, Sect. 3.5.3.

Standard functions

List of the standard functions, Sect. 3.5.2.4.3.

Kinematic functions

List of kinematic functions, Sect. 3.5.2.4.7.

Variables

List of variables created by the user, Sect. 3.5.2.4.8.

Points

List of connection points for all of the bodies, including simple points, vectors and oriented points, Sect. 3.6.9.6.

Oriented points

List of all oriented points assigned to bodies, Sect. 3.6.9.6.2.

Joints

List of joints, Sect. 3.6.11.

Double click of an element of a list adds the element or function template to the expression in the cursor position.

Remark 1. The *Points, Oriented points* and *Joints* lists contain all elements in the object including subsystems. Lists of *identifiers* and *variables* do not include elements from subsystems

Remark 2. Names of elements containing non-standard symbols like a space must be put in quotes.

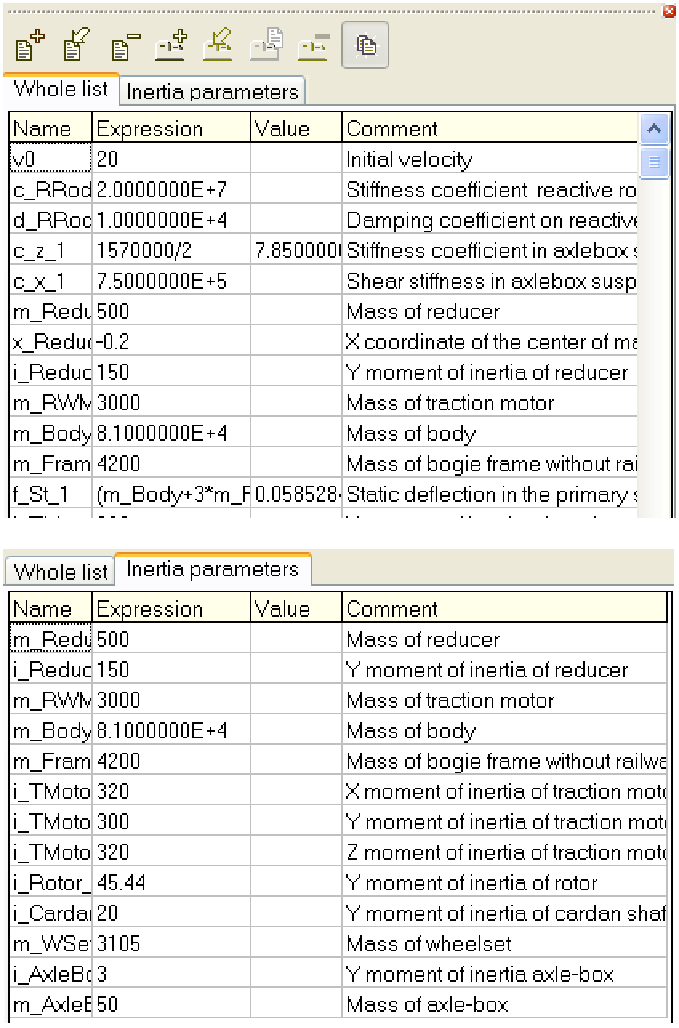
The **Check** button calls syntax analysis of the expression. The **Cancel** button closes the editor without saving modifications in the expression.

3.5.3. List of identifiers

A list of identifiers is the main tool of parameterization of UM models, Sect. 3.5.2.4.2.

3.5.3.1. Window with list of identifiers

Window with the list of identifiers is intended for creation and modification of identifiers, changing numeric values and expressions.



Name	Expression	Value	Comment
v0	20		Initial velocity
c_RRod	2.0000000E+7		Stiffness coefficient reactive ro
d_RRoc	1.0000000E+4		Damping coefficient on reactive
c_z_1	1570000/2	7.850000	Stiffness coefficient in axlebox s
c_x_1	7.5000000E+5		Shear stiffness in axlebox susp
m_Redu	500		Mass of reducer
x_Redu	-0.2		X coordinate of the center of ma
i_Reduc	150		Y moment of inertia of reducer
m_RWM	3000		Mass of traction motor
m_Body	8.1000000E+4		Mass of body
m_Fram	4200		Mass of bogie frame without rai
f_St_1	(m_Body+3*m_F	0.058528	Static deflection in the primary s

Name	Expression	Value	Comment
m_Redu	500		Mass of reducer
i_Reduc	150		Y moment of inertia of reducer
m_RWM	3000		Mass of traction motor
m_Body	8.1000000E+4		Mass of body
m_Fram	4200		Mass of bogie frame without railwa
i_TMoto	320		X moment of inertia of traction moto
i_TMoto	300		Y moment of inertia of traction moto
i_TMoto	320		Z moment of inertia of traction moto
i_Rotor_	45.44		Y moment of inertia of rotor
i_Carda	20		Y moment of inertia of cardan shaf
m_WSe	3105		Mass of wheelset
i_AxleBo	3		Y moment of inertia axle-box
m_AxleE	50		Mass of axle-box

Figure 3.51. Sheets of the list of identifiers

3.5.3.2. Modification of identifiers

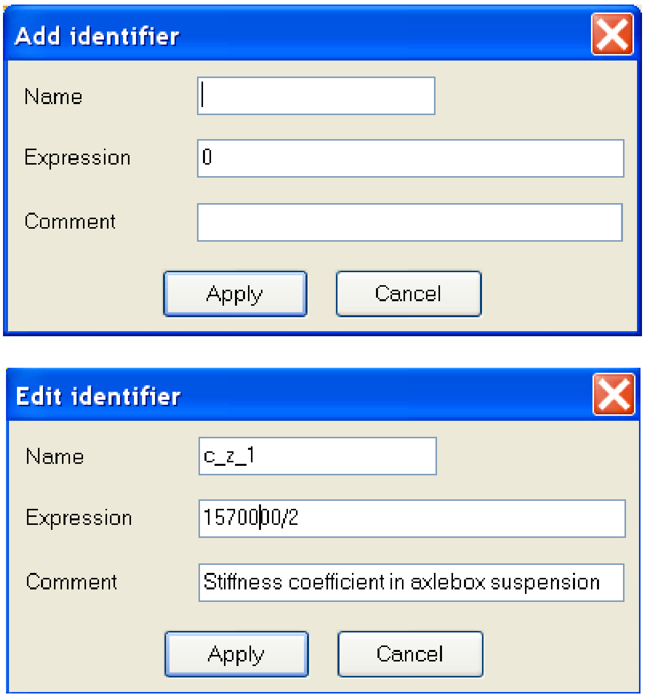
The tool panel contains a number of buttons with the following functions.

Adding a new identifier, Figure 3.52. Hot key: *Insert*.

Editing selected identifier, Figure 3.52. Hot key: *Enter*; mouse: double click.

Deleting selected identifier. After deleting the identifier in expressions is replaced by its current numeric value.

Renaming identifiers is possible in the editor window, Figure 3.52. The new name replaces automatically the old one throughout in the model.



Name

Expression 0

Comment

Name c_z_1

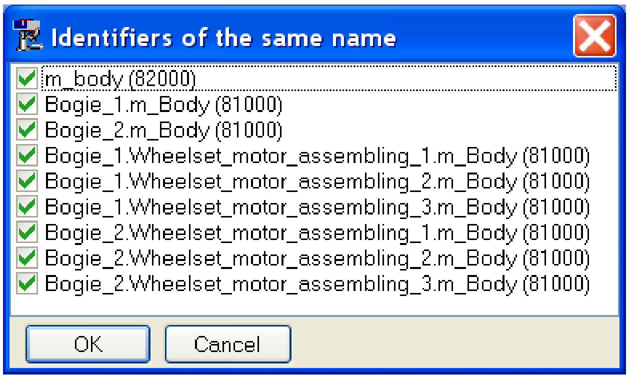
Expression 1570000/2

Comment Stiffness coefficient in axlebox suspension

Figure 3.52. Adding and modifying an identifier

Changing values of identifiers of the same name in subsystems

If the user changes the value of an identifier and subsystems of the model have identifiers of the same name, a special window with the list of these identifiers appears, Figure 3.53. The user may assign the new value to the selected identifiers from the list. In the case shown in Figure 3.53, the new value 82000 will be assigned to all identifiers.



☑ m_body (82000)
☑ Bogie_1.m_Body (81000)
☑ Bogie_2.m_Body (81000)
☑ Bogie_1.Wheelset_motor_assembling_1.m_Body (81000)
☑ Bogie_1.Wheelset_motor_assembling_2.m_Body (81000)
☑ Bogie_1.Wheelset_motor_assembling_3.m_Body (81000)
☑ Bogie_2.Wheelset_motor_assembling_1.m_Body (81000)
☑ Bogie_2.Wheelset_motor_assembling_2.m_Body (81000)
☑ Bogie_2.Wheelset_motor_assembling_3.m_Body (81000)

Figure 3.53. Example of a list of identifiers of the same name

See Sect. 3.5.3.5.

3.5.3.3. Identifier sheets

A sheet is a group of selected identifiers. The sheets are used for logical separation of the identifiers into groups, e.g. inertial, geometrical parameters, suspension parameters etc.

Control buttons for managing the sheets:

Adding a new sheet.

Renaming the current sheet.

Modification of the current identifier group. The tool allows adding and removing identifiers with the special dual window. In particular, double clicks are used for adding/removing identifiers.

Deleting the current sheet.

3.5.3.4. Object refresh after change of identifier value

The button switches the modes of immediate/postponed refresh of the object elements after change of numeric value of an identifier. The refresh is recommended to be postponed if it is made too slow in case of large models.

3.5.3.5. Pop-up menu of identifier list

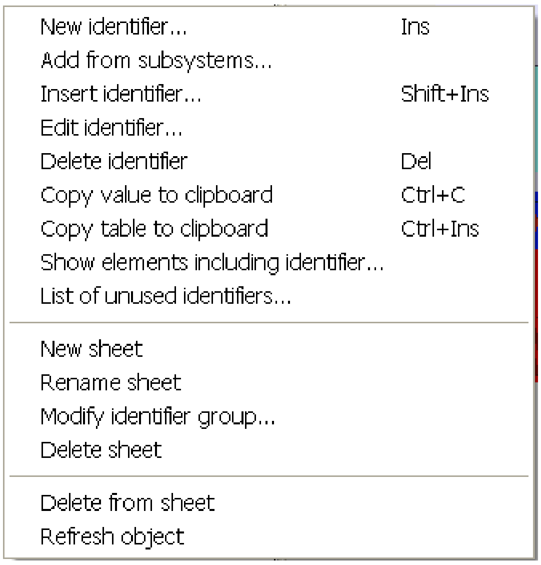
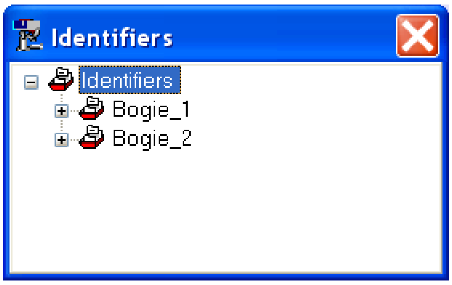

New identifier... Ins
Add from subsystems...
Insert identifier... Shift+Ins
Edit identifier...
Delete identifier Del
Copy value to clipboard Ctrl+C
Copy table to clipboard Ctrl+Ins
Show elements including identifier...
List of unused identifiers...
New sheet
Rename sheet
Modify identifier group...
Delete sheet
Delete from sheet
Refresh object

Figure 3.54. Commands of pop-up menu of identifier list

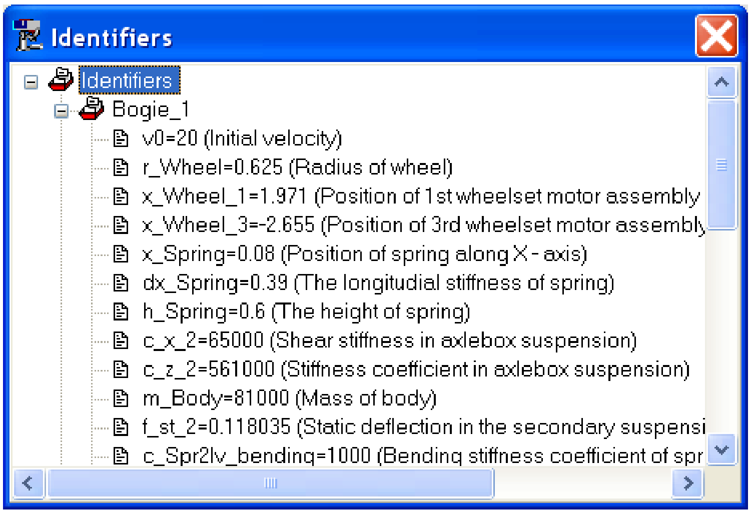
The pop-up menu appears after click of the right mouse button on the list of identifiers, Figure 3.54. The menu contains the following commands.

- **New identifier:** the command opens the window for editing identifiers, Figure 3.51, and adds the new identifier to the end of the list.
- **Add from subsystem:** if subsystems are presented in the current UM object, the command opens a window with the list of identifiers included in subsystems (Figure 3.55) and allows the user to add identifiers in the current list by clicking on the desired identifiers in the window by the left mouse button. The command is often used to add to the main object identifiers from subsystems, which values are frequently changed during the simulation. In such the way the process of access to identifiers is simplified.



Identifiers

- Identifiers
  - Bogie_1
  - Bogie_2



Identifiers

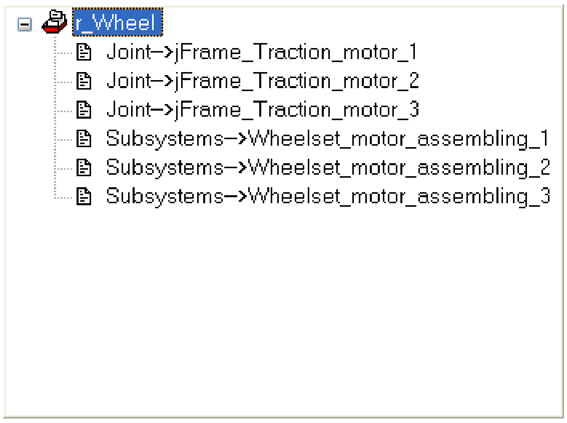
- Identifiers
  - Bogie_1
    - v0=20 (Initial velocity)
    - r_Wheel=0.625 (Radius of wheel)
    - x_Wheel_1=1.971 (Position of 1st wheelset motor assembly
    - x_Wheel_3=-2.655 (Position of 3rd wheelset motor assembly
    - x_Spring=0.08 (Position of spring along X - axis)
    - dx_Spring=0.39 (The longitudial stiffness of spring)
    - h_Spring=0.6 (The height of spring)
    - c_x_2=65000 (Shear stiffness in axlebox suspension)
    - c_z_2=561000 (Stiffness coefficient in axlebox suspension)
    - m_Body=81000 (Mass of body)
    - f_st_2=0.118035 (Static deflection in the secondary suspensi
    - c_Spr2lv_bending=1000 (Bending stiffness coefficient of spr

Figure 3.55. Example of identifier tree

- **Insert identifier:** the command inserts a new identifier immediately before the line corresponding to the position of the mouse cursor.
- **Edit identifier:** the command opens the window for editing the identifier in the position of the mouse cursor, Figure 3.51.
- **Delete identifier:** the command deletes an identifier in the position of the mouse cursor.
- **Copy value to clipboard:** the command copies to clipboard the value of the identifier in the position of the mouse cursor.
- **Copy table to clipboard:** the command copies to clipboard as text all identifiers from the current sheet. Example:

Name	Expression	Value	Comments
m_bus		1.5000000E+4	Mass
ixx_bus		1.0000000E+4	Moment of inertia
izz_bus		2.5000000E+5	

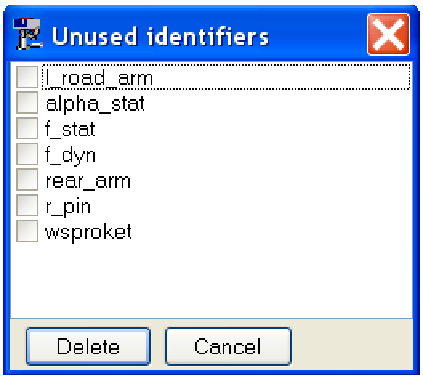
- **Show elements including identifier:** the command opens a window with the list of elements, which includes the identifier in the position of the mouse cursor, Figure 3.55. The user opens the element in inspector by clicking on the element name in the list.



r_Wheel
- Joint–>jFrame_Traction_motor_1
- Joint–>jFrame_Traction_motor_2
- Joint–>jFrame_Traction_motor_3
- Subsystems–>Wheelset_motor_assembling_1
- Subsystems–>Wheelset_motor_assembling_2
- Subsystems–>Wheelset_motor_assembling_3

Figure 3.56. Example: list of elements including identifier r_Wheel

- **List of unused identifiers:** the command opens a window with the list of identifiers, which are not used in description of object elements, Figure 3.57. In particular, this window is used for deleting some of unused identifiers: the checked identifiers are deleted by the **Delete** button in the bottom of the window.



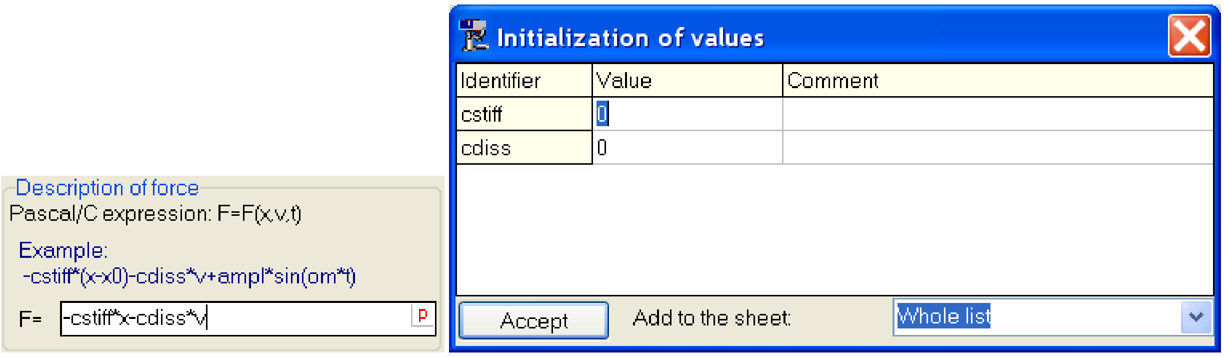
□ l_road_arm
□ alpha_stat
□ f_stat
□ f_dyn
□ rear_arm
□ r_pin
□ wsproket

Figure 3.57. Example of unused identifiers

- **New sheet, Rename sheet, Modify identifier group, Delete sheet:** the commands manage the identifier sheets, Sect. 3.5.3.3.
- **Delete from sheet:** the command removes an identifier located under the mouse cursor from the current sheet.
- **Refresh:** the command refresh the object, Sect. 3.5.3.4.

3.5.3.6. Adding new identifiers by element description

The user may add new identifiers directly during description of any model parameters and elements like inertia parameters, force elements and so on, Figure 3.58.



Description of force

Pascal/C expression: F=F(x,v,t)

Example:
-cstiff*(x-x0)-cdiss*v+ampl*sin(om*t)

F= -cstiff*x-cdiss*v

Initialization of values

Identifier	Value	Comment
cstiff	0	
cdiss	0	

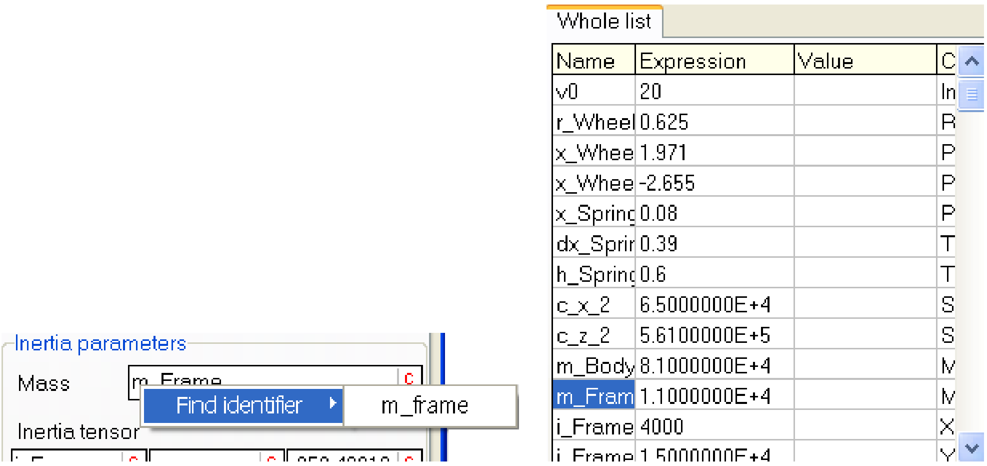
Figure 3.58. Example of adding new identifiers by description of expression

By analysis of expression the program detects new identifiers and suggests the user to assign numeric values and comments to the identifiers in a special window. If necessary, the new identifiers can be placed on one of the existing sheets or on a new sheet, Sect. 3.5.3.3. The list “Add to the sheet” is used for selection of existing sheet. Write the name of a new sheet in the same edit box to place the identifier in a new created sheet.

3.5.3.7. Jump to identifier from the element parameter

If an identifier is used by description of a parameter of the object, the user can find it in the list of identifiers. To find the identifier, click the right mouse button on the edit box and select the **Find identifier** commend of the pop-up menu, Figure 3.59. The identifier will be selected in the list.

The function is useful in the case of large lists of identifiers.



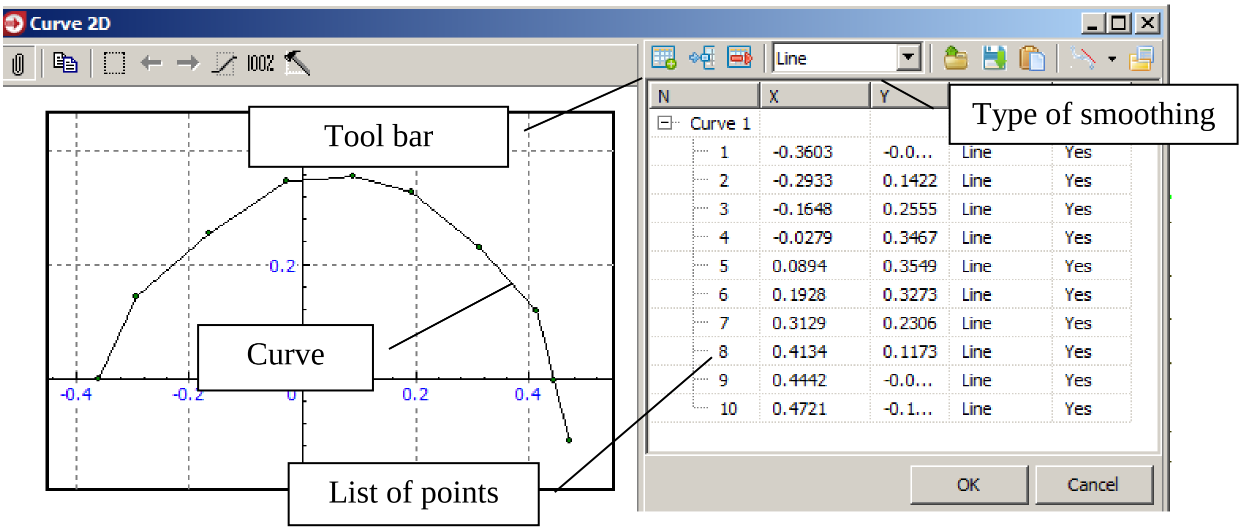
Inertia parameters

Mass m_Frame

Name	Expression	Value	C
v0	20		In
r_Wheel	0.625		R
x_Whee	1.971		P
x_Whee	-2.655		P
x_Spring	0.08		P
dx_Sprir	0.39		T
h_Spring	0.6		T
c_x_2	6.5000000E+4		S
c_z_2	5.6100000E+5		S
m_Body	8.1000000E+4		M
m_Fram	1.1000000E+4		M
i_Frame	4000		X
i_Frame	1.5000000E+4		Y

Figure 3.59. Example of jump to the identifier M_Frame

3.5.4. 2D curve editor



N	X	Y		
Curve 1				
1	-0.3603	-0.0...	Line	Yes
2	-0.2933	0.1422	Line	Yes
3	-0.1648	0.2555	Line	Yes
4	-0.0279	0.3467	Line	Yes
5	0.0894	0.3549	Line	Yes
6	0.1928	0.3273	Line	Yes
7	0.3129	0.2306	Line	Yes
8	0.4134	0.1173	Line	Yes
9	0.4442	-0.0...	Line	Yes
10	0.4721	-0.1...	Line	Yes

Figure 3.60. Curve editor and its elements

2D curve editor is a tool for input of data in a graphic form. The editor allows the user to create a plane curve, a function or a set of curves by a set of points. Here we consider some features of working with the editor. Additional information about usage of this tool for creation of 2D graphic images can be found in Sect.3.6.8.6.

3.5.4.1. Modes of curve editor

There exist two modes of the editor depending on the problem to be solved.

- Mode of creation of a set of curves
- Mode of creation of a function

The second mode imposes a number of restrictions and additional features:

- one curve only;
- ordering points according to abscissa value;
- plots the first and second derivative as well as a curvature in a separate window is available.

3.5.4.2. Tool bar

Consider functions of buttons on the tool bar. Note that sets of buttons differ for different modes of the editor.

← left shift depending on the mouse button;

→ right shift depending on the mouse button;

📎 - popup /fixed panel;

📋 - copy to clipboard;

⬚ - zoom in by frame;

⟋ - select curve sections by frame;

100% - show all;

- settings;

- add new point;

- insert new point;

- delete selected points;

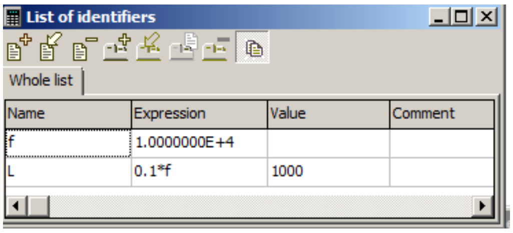
- read data from file;

- save data to file;

- paste;

- tangent and curvature (1st derivative, 2nd derivative, curvature, smoothed curvature);

- list of identifiers (Figure 3.61).



Name	Expression	Value	Comment
f	1.0000000E+4		
L	0.1*f	1000	

Figure 3.61. Window of model parameters

3.5.4.3. Adding, positioning, and deleting separate point on a curve

There exist two methods for **adding** a point to a curve.

1. Double click by the left mouse button in the position of the adding point.

With this method a point can be added both to begin and the end of a non-closed curve, as well as inside the closed or non-closed curve. When a point is added to the beginning or to the end of a curve, it is recommended to put it near the corresponding first or last point of the curve and then to drag it to the desirable position.

2. Button in the top of the list of points (Figure 3.60).

With this method you can add point to the end of the curve only.

Positioning the point means setting its desirable position. Two methods are realized for this purpose.

1. Positioning by the list of points.

Find the point in the list, e.g. by clicking on its image in the plot area, and set its new coordinates.

2. Positioning by dragging.

This is the most often used method for approximate positioning points. Move the mouse cursor near the point image. The cursor must change to ↖. Press the left mouse button and drag the point to the desirable position.

To **delete** a point, either select it in the list of point and click the button, or move the mouse cursor near the point image until it changes to , call the pop-up menu by clicking the right mouse button and select the *Delete* menu item.

3.5.4.4. Selecting, copying, deleting and moving fragments and curves

To select a fragment (a set or points) draw a rectangle region in the plot area by dragging the mouse cursor (Figure 3.62).

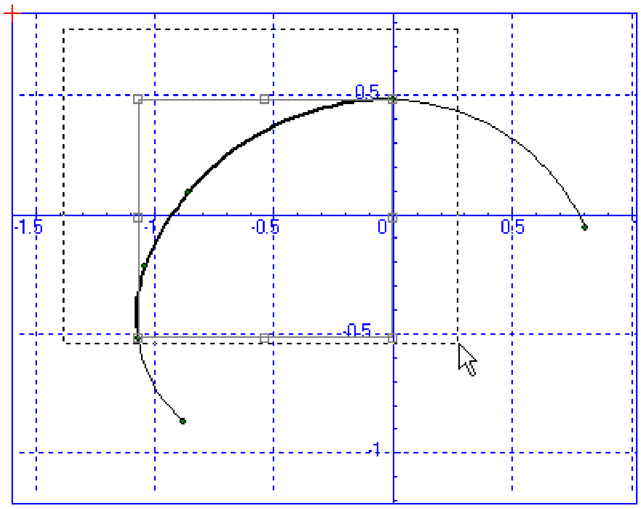
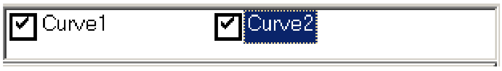


Figure 3.62. Fragment selection by mouse

There exist to methods for the **selection of a curve**.

1. Select the name of a curve in the list of curves (Figure 3.63).



☑ Curve1 ☑ Curve2

Figure 3.63. Selection of a curve in the list of curves

2. Move the mouse cursor near the curve until it changes to , call the pop-up menu by the right mouse button and select the *Select whole curve* menu item.

To **select all points** call the pop-up menu and select the *Select all* menu item.

To **remove a selection**, click by the left mouse button anywhere outside the selection rectangle.

To **move a fragment or a curve**, select it, move the mouse cursor until it changes to , press the left mouse button and drag the fragment. *Moving a fragment is forbidden in the mode of creation of a function.*

To **delete a fragment or a curve**, select it and press the *Delete* key.

To **copy a fragment or a curve** select it, press the *Ctrl+C* and *Ctrl+V* hot keys, and move the copied fragment into a desirable position.

3.5.4.5. Closing curve

Two methods of closing a curve:

1. move one of the curve end point by the mouse to a small neighborhood of the another end;
2. use the ☑ Closed key in the top of the list of points.

3.5.4.6. Smoothing

To smooth a curve or a fragment, select it and choose one of the smoothing type from the list (Figure 3.60)

3.5.4.7. Using the clipboard for creating curves and functions

For input from the clipboard, points should be written as a text in two columns. The first column contains abscissa values, the second one corresponds to the ordinate values

-68.9	11.7
-66.4	8.88
-63.9	6.98
-61.4	6.48
-58.9	5.99

…….

To get points from the clipboard

- Copy the new data to the clipboard from any text editor in a standard manner;
- Activate the curve editor by the mouse and paste data from the clipboard (*Ctrl+V* or *Shift+Insert*).

3.5.5. Hot keys

3.5.5.1. Constructor hot keys

- *Ctrl+Alt+X* – make the tree of elements active, Sect. 3.5.1.1;
- *F11* – bring to front the list of elements (if it is located as a separate window);
- *F12* – bring to front the data inspector (if it is located as a separate window).

3.5.5.2. Inspector hot keys

Open element data (a tab of the constructor):

- *Ctrl+Alt+O* – object;
- *Ctrl+Alt+S* – subsystems;
- *Ctrl+Alt+B* – bodies;
- *Ctrl+Alt+G* – graphic objects;
- *Ctrl+Alt+J* – joints;
- *Ctrl+Alt+F* – bipolar forces;
- *Ctrl+Alt+M* – scalar torques;
- *Ctrl+Alt+L* – linear forces;
- *Ctrl+Alt+C* – contact forces;
- *Ctrl+Alt+T* – T-forces;
- *Ctrl+Alt+A* – special forces;
- *Ctrl+Alt+Z* – external connections;
- *Ctrl+Alt+I* – indices;
- *Ctrl+Alt+P* – protocol;
- *Ctrl+Alt+R* – coordinates.

3.5.5.3. Animation window hot keys

In the mode of element selection by the mouse button (the button must be down, Sect. 3.5.1.2.4), mouse move over a body image allows the user to get coordinates of the corresponding points of the body relative to SC0. If the *Shift* key is pressed, the coordinates will be given in the body-fixed SC. If the *Ctrl* key is pressed, the coordinates of body-fixed SC origin in SC0 are shown.

3.5.5.4. Inspector tab with a list

Ctrl+Alt+GrayPlus – add element;
Ctrl+Alt+GrayMinus – delete the current element;
*Ctrl+Alt+Gray** – copy the current element;
Ctrl+Alt+N – edit name.

For elements connecting a pair of bodies (joints, force elements):
Ctrl+Alt+1 – choose the first body from the list of bodies;
Ctrl+Alt+2 – choose the second body from the list of bodies;

Ctrl+Alt+T – choose element type.

3.6. Data Input

3.6.1. Data Input Sequence

The following sequence of object data input is recommended.

1. External and/or included subsystems, if UM Subsystems module is available, Sect. 3.6.3.

It is recommended to use subsystems for development of complex models. Structuring a model as a tree of subsystems is efficient for simulation of rail vehicles, trains, and tracked vehicles.

2. Bodies, their graphical images and the corresponding joints.

The sequence of description of bodies is usually defined by the kinematical scheme of the object. At first, the bodies connected with the base body (Base0, SC0) as well as the corresponding joints are described, and then the bodies connected with already described bodies and so on. By such description the sequence of kinematic scheme of object all its bodies are drawn not only in the single element mode of animation window but also in the whole object mode of the animation window (Sect. 3.5.1.2.2). It is important to remember, that in the mode of whole object animation, the described body is drawn in the animation window if there exists a path from the current body to the base body through the described joints.

3. Force elements and their graphical images

After describing the object kinematical scheme, force element images are drawn in the animation window both in the current element and the full object animation mode, which allows the user to control geometrical parameters of force elements visually.

3.6.2. Methods for adding elements to object

3.6.2.1. Direct creation of single element

A direct creation of an element such as subsystem, graphic object, body, joint, bipolar force and so on with the help of the standard interface of element lists is considered in Sect. 3.5.2.2.

3.6.2.2. Adding elements by visual components

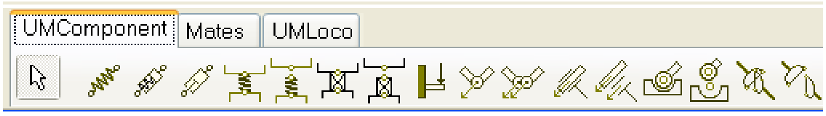


Figure 3.64. Tabs with visual components

Lists of components give an effective tool for visual adding some of elements, Figure 3.64. The method is described in Sect. 3.7.2.

3.6.2.3. Main menu commands for adding a typed element

A type element is an element with assigned type. The typed element can be added by the **Add** menu command, Figure 3.65.

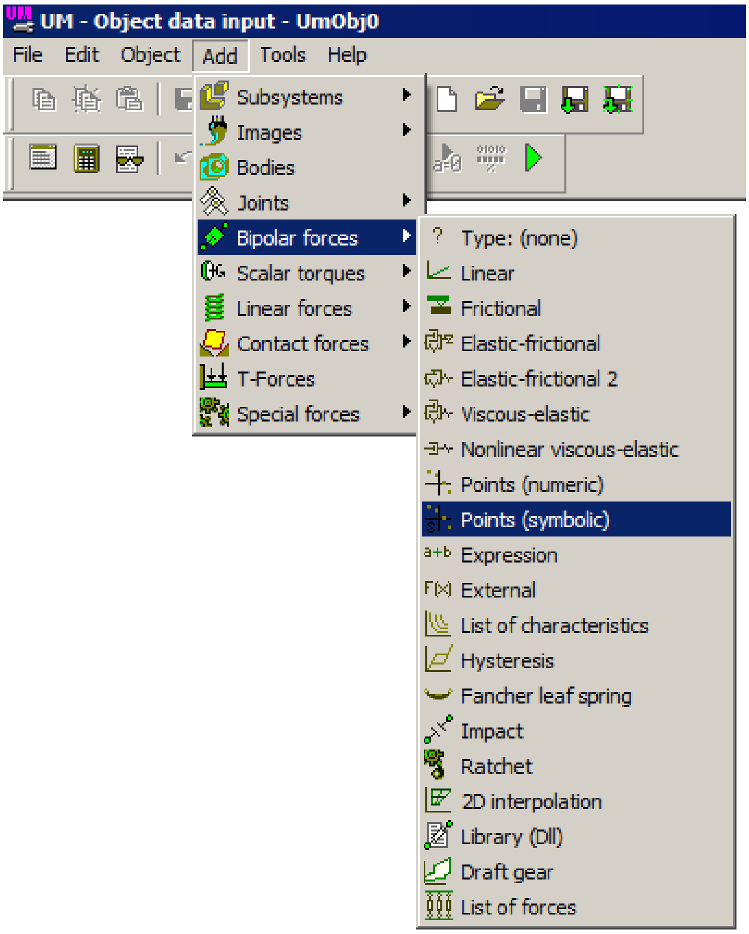


Figure 3.65. Adding a typed element by menu command

3.6.2.4. Adding typed elements by pop-up menu of element tree

In this case,

- select the root item of element list in the element tree (e.g. *Images*),
- call a pop-up menu by the right mouse button and select the desired command, Figure 3.66.

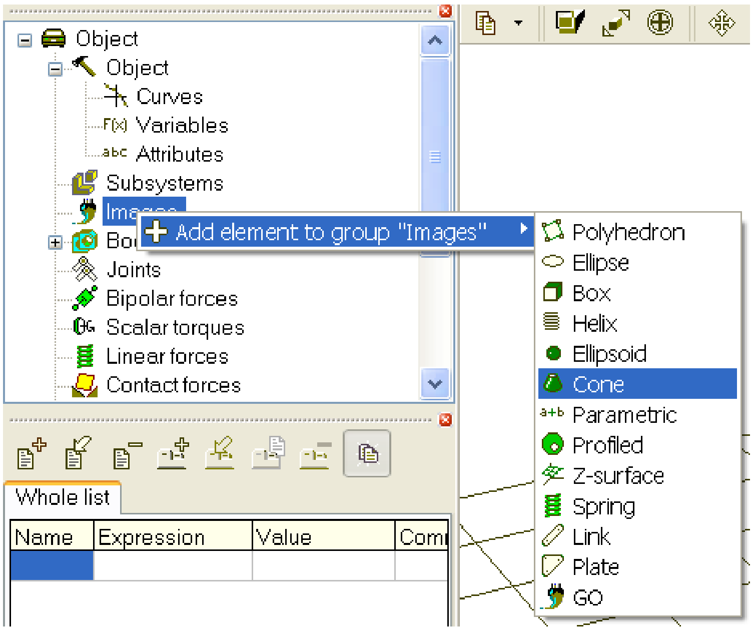


Figure 3.66. Adding typed element from element tree

The same result is obtained by the main menu command **Add**, Sect. 3.6.2.3.

3.6.2.5. Insert of element from other model by clipboard

The following steps are required:

- open the model, which element must be copied;
- select the element in the element tree;
- copy the element in the clipboard by

a) **Edit | Copy to clipboard** menu command or by the 📋 button on the tool panel if no image of the element must be copied;

b) **Edit | To clipboard as component** menu command or by the 📋 button on the tool panel to copy both the element and the image assigned to it;

- make active the model for adding the element and run the **Edit | Insect** menu command or click on the 📋 button on the tool panel;
- if necessary, correct numeric values of identifiers added together with the element.

3.6.2.6. Insert of element from other model by file

The following steps are required:

- open the model, which element must be copied;
- select the element in the element tree;
- copy the element in a file by

a) **Edit | Copy to file** menu command or by the 💾 button on the tool panel, if no image of the element must be copied;

b) **Edit | Save as component** menu command or by the 💾 button on the tool panel to copy both the element and the image assigned to it;

- make active the model for adding the element and run the **Edit | Read from file** menu command or click on the 📂 button on the tool panel;
- if necessary, correct numeric values of identifiers added together with the element.

3.6.2.7. Merging models

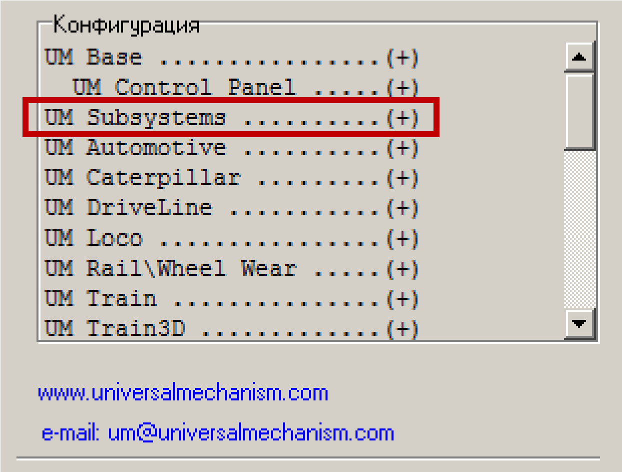
One model can be completely included into another one:

- open the model to be included and save it to a file with any name and extension by the **File | Save as component** menu command or by the 💾 button;
- open the model which will include the first one;
- read the saved file by the **Edit | Read from file** menu command or by the 📂 button.

3.6.3. Subsystems

Subsystems give a powerful tool for development of complex models such as rail vehicle, train, articulated truck and so on, see Chapter 2, Sect. *Subsystems*. In particular, flexible bodies are added to a model as subsystems.

Subsystems are available if the **UM Subsystems module** is included in the configuration. To verify whether the module is presented in the current UM version, use the menu command **Help | About…**, Figure 3.67.



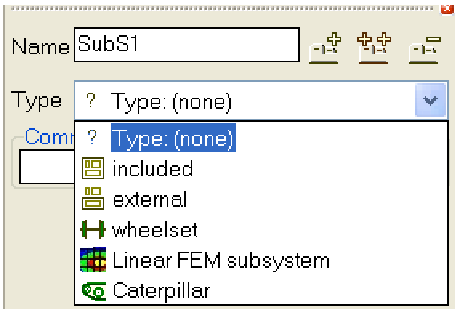
```
Конфигурация
UM Base ................(+)
  UM Control Panel .....(+)
UM Subsystems ..........(+)
UM Automotive ..........(+)
UM Caterpillar .........(+)
UM DriveLine ...........(+)
UM Loco ................(+)
UM Rail\Wheel Wear .....(+)
UM Train ...............(+)
UM Train3D .............(+)
```

www.universalmechanism.com

e-mail: um@universalmechanism.com

Figure 3.67. List of modules included in the current UM configuration

3.6.3.1. Adding a subsystem



Name SubS1

Type ? Type: (none)

Figure 3.68. Selection of subsystem type

Subsystem list management buttons are, Figure 3.68

– create a subsystem;

– copy an existing subsystem;

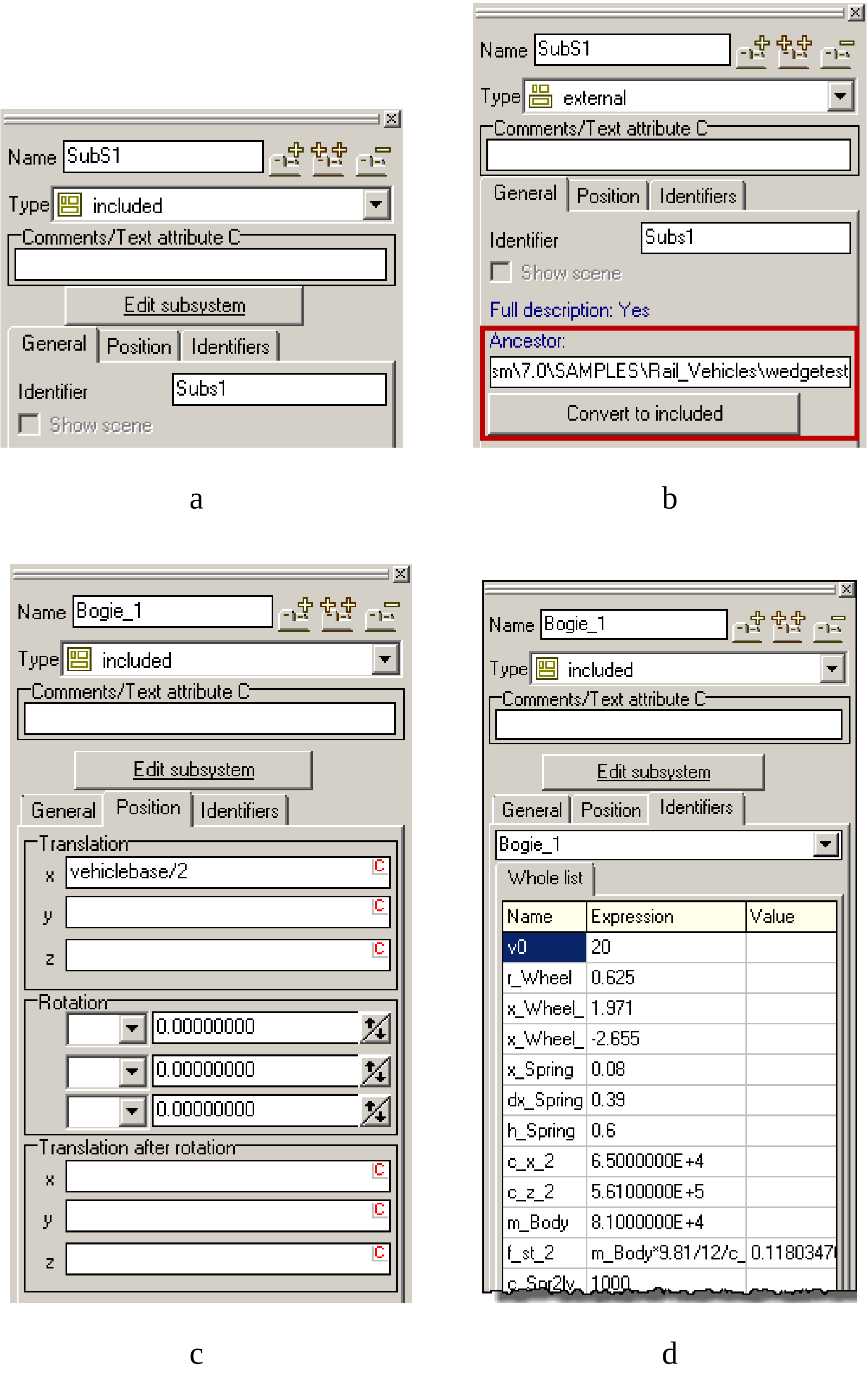
– delete a current subsystem.

There are three types of subsystems: **included, external** and **special**.

Special subsystems such as wheelset, FEM subsystem and Caterpillar are used in UM modules UM Loco, UM FEM and UM Tracked vehicle respectively. Working with special subsystems is described in details in the corresponding parts of the user's manual. Here we consider included and external subsystems only.

After choice of the subsystem type (included or external) the user must open an UM object which will be included as a subsystem in the current object.

Main parameters of external and included subsystems are almost the same, Figure 3.69. They are name, comments, position of the subsystem (Sect. 3.6.4), list of identifiers used in the subsystem (Sect. 3.5.3), and the identifier of the subsystem used for programming. Also an external subsystem has one more parameter Ancestor (Figure 3.69b), which is the path to the folder with ancestor of the subsystem. To convert an external subsystem to an included one, click the button Convert to include on the tab General.



a

Name SubS1

Type included

Comments/Text attribute C

Edit subsystem

General | Position | Identifiers

Identifier Subs1

□ Show scene

b

Name SubS1

Type external

Comments/Text attribute C

General | Position | Identifiers

Identifier Subs1

□ Show scene

Full description: Yes

Ancestor:

sm\7.0\SAMPLES\Rail_Vehicles\wedgetest

Convert to included

c

Name Bogie_1

Type included

Comments/Text attribute C

Edit subsystem

General | Position | Identifiers

Translation

x vehiclebase/2

y

z

Rotation

0.00000000

0.00000000

0.00000000

Translation after rotation

x

y

z

d

Name Bogie_1

Type included

Comments/Text attribute C

Edit subsystem

General | Position | Identifiers

Bogie_1

Whole list

Name	Expression	Value
v0	20	
r_Wheel	0.625	
x_Wheel_	1.971	
x_Wheel_	-2.655	
x_Spring	0.08	
dx_Spring	0.39	
h_Spring	0.6	
c_x_2	6.5000000E+4	
c_z_2	5.6100000E+5	
m_Body	8.1000000E+4	
f_st_2	m_Body*9.81/12/c_	0.1180347(
c_Spr2ly	1000	

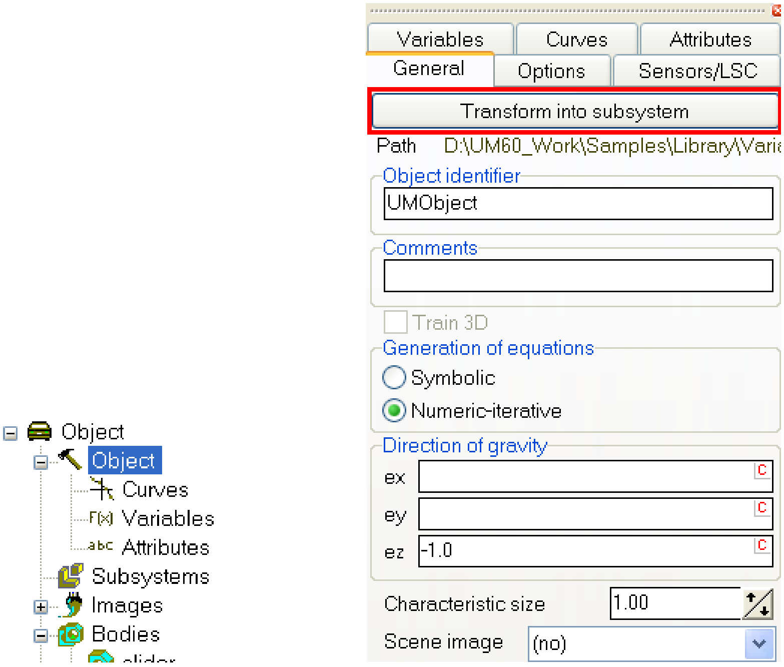
Figure 3.69. Tabs of the object inspector for subsystems: a) General tab for included subsystem; b) General tab for external subsystem; c) Position tab, d) Identifier tab.

3.6.3.2. Transformation of model into subsystem

The transformation of the model into subsystem is used

- by development of a model as a tree of subsystems;
- by development of a component, which must be added to the model as a subsystem (example: suspension subsystems of tracked vehicle, see 18_UM_Caterpillar.pdf).

To transform the model into a subsystem, open the **Object** tab of the inspector and click on the **Transform into subsystem** button, Figure 3.70. After the transformation, all elements will be removed from the model. Thereby, the model will include only one subsystem, which content is identical to the model before the transformations.



Path D:\UM60_Work\Samples\Library\Varia

Object identifier
UMObject

Comments

□ Train 3D

Generation of equations
○ Symbolic
● Numeric-iterative

Direction of gravity
ex
ey
ez -1.0

Characteristic size 1.00

Scene image (no)

Figure 3.70. Transformation button

3.6.3.3. Interconnection of subsystem. Use of external elements

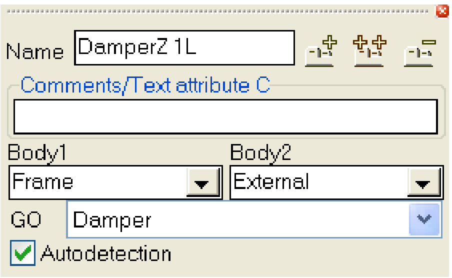
Joints and force elements can connect bodies of different subsystems by two different ways:

1. the element (joint, force element) is created in the head object; one or two bodies from the subsystems are assigned to it;
2. **external elements** described in the subsystem are used, Sect. 3.6.10.1.

3.6.3.3.1. External elements. Autodetection

External element is a joint or a force element described in a subsystem, which second body is **External**, Figure 3.71. This means that the second body belongs to either another subsystem or to the 'upper' object. Use of external elements is the base for rapid development of an object as a tree of subsystems.

The second body is assigned in an upper subsystem or in a head object with the help of connection points, Sect. 3.6.3.3.2.



Name DamperZ 1L

Comments/Text attribute C

Body1 Frame

Body2 External

GO Damper

☑ Autodetection

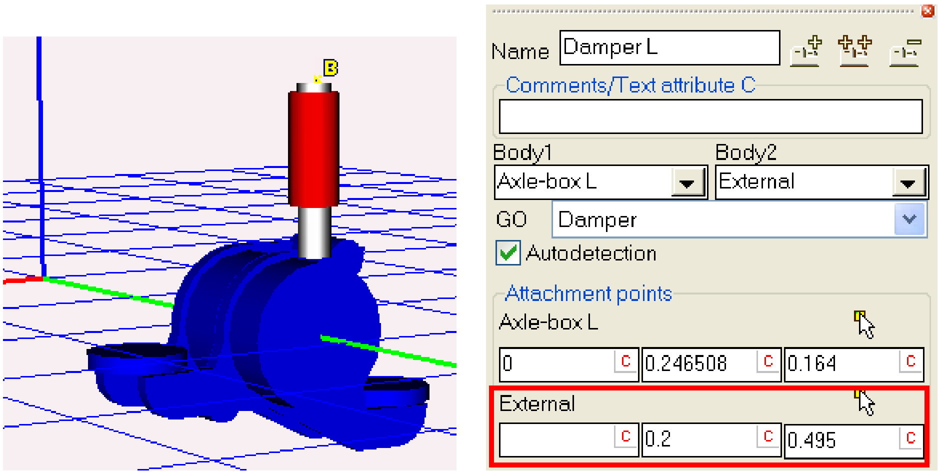
Figure 3.71. Example of an external element

Remark 1. External elements are ignored if the second body is not assigned.

Remark 2. Method of a fictitious body is recommended instead of use of external joints, Sect. 3.6.3.3.3.

Autodetection mode of elements allows a significant simplification and acceleration of working with external elements. If the autodetection mode is on (Figure 3.71), the coordinates of the attachment point for the *second* body are set in the SC of the *first* body **for zero values of all object coordinates**. Thereby, the coordinates of the second attachment point can be defined in the subsystem, in which the external element is described.

Consider a damper connecting an axle box and a frame, Figure 3.72. The damper is described in a subsystem, which does not include the frame, and the second body is the **External** one. The autodetection mode is on, and the coordinates of the second (upper) end of the element are specified in the SC of the axle box. As a result, the coordinates of this point in SC of the frame will be computed automatically.



Name Damper L

Comments/Text attribute C

Body1 Axle-box L

Body2 External

GO Damper

☑ Autodetection

Attachment points

Axle-box L

0 | 0.246508 | 0.164

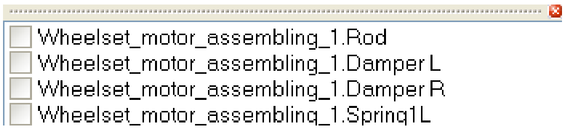
External

 | 0.2 | 0.495

Figure 3.72. Example of an external element with the autodetection mode

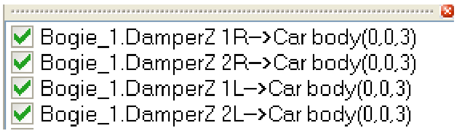
3.6.3.3.2. Assignment of second bodies to external elements

The **Connections** tab of the inspector is used for assignment of second bodies to external elements, Figure 3.72. The tab includes the list of external elements included in subsystem tree, Figure 3.73. If second bodies are already assigned to some external elements in subsystems, such the elements are excluded from the list.



□ Wheelset_motor_assembling_1.Rod
□ Wheelset_motor_assembling_1.Damper L
□ Wheelset_motor_assembling_1.Damper R
□ Wheelset_motor_assembling_1.Spring1L

Figure 3.73. Example of external elements. Second bodies are not assigned



☑ Bogie_1.DamperZ 1R->Car body(0,0,3)
☑ Bogie_1.DamperZ 2R->Car body(0,0,3)
☑ Bogie_1.DamperZ 1L->Car body(0,0,3)
☑ Bogie_1.DamperZ 2L->Car body(0,0,3)

Figure 3.74. Example of external elements. Second bodies are assigned

The following methods are used for assignment of the second bodies to external elements.

1. **External element without autodetection**

- A connection point must be preliminary defined for the body, which will be assigned as the second body to the external element, Sect. 3.6.9.6. Coordinates of this point correspond to the attachment point of the element.
- Select the external element in the list. If a graphic image is assigned to the element, it is selected in the animation window. For *visual assignment* of the connection, click by the mouse on the point image, Figure 3.75. For *selection of a point from the list*, double click on the element name or run the **Assign point** command of the pop-up menu, Figure 3.76. The list of connection points replaces the list of external elements, Figure 3.77. Assign the point by the mouse click.

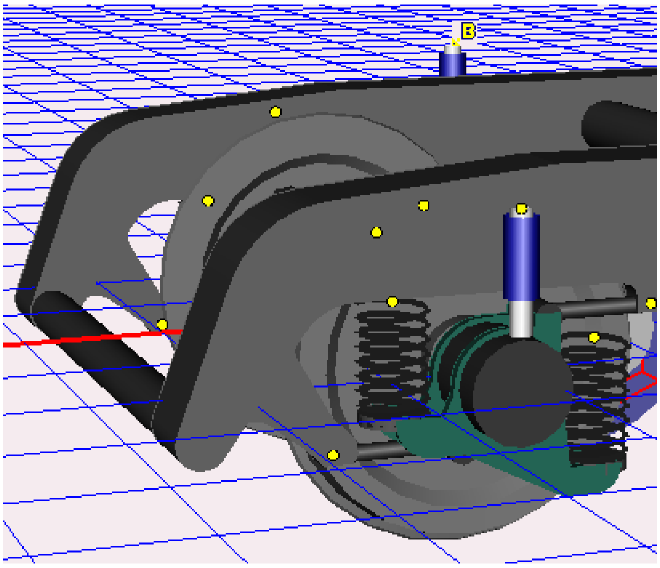
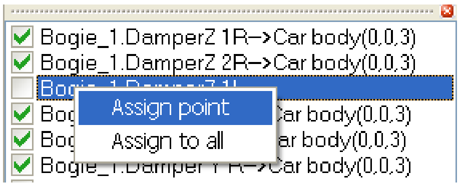
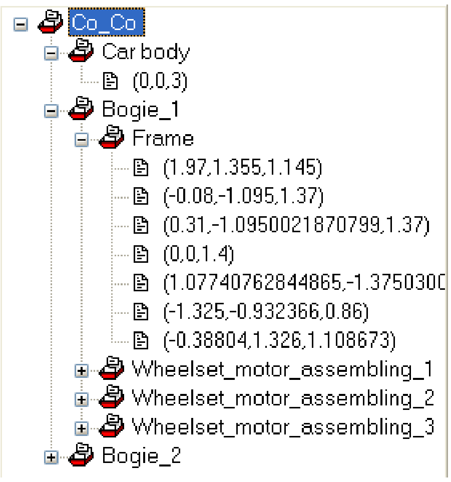


Figure 3.75. Yellow points mark connection points



☑ Bogie_1.DamperZ 1R–>Car body(0,0,3)
☑ Bogie_1.DamperZ 2R–>Car body(0,0,3)

Figure 3.76. Pop-up menu of the list of external elements



Co_Co
Car body
(0,0,3)
Bogie_1
Frame
(1.97,1.355,1.145)
(-0.08,-1.095,1.37)
(0.31,-1.0950021870799,1.37)
(0,0,1.4)
(1.07740762844865,-1.3750300
(-1.325,-0.932366,0.86)
(-0.38804,1.326,1.108673)
Wheelset_motor_assembling_1
Wheelset_motor_assembling_2
Wheelset_motor_assembling_3
Bogie_2

Figure 3.77. List of connection points

2. External element with autodetection

Advances in use of external elements with autodetection consist in fact that any connection point of the necessary body can be assigned to the element, because the coordinates of the attachment point are computed automatically according to data specified in the subsystem. As a result, only one connection point is necessary for assignment of any number of external elements to a body. The assignment process in this case is quite similar to that described above for an element without autodetection.

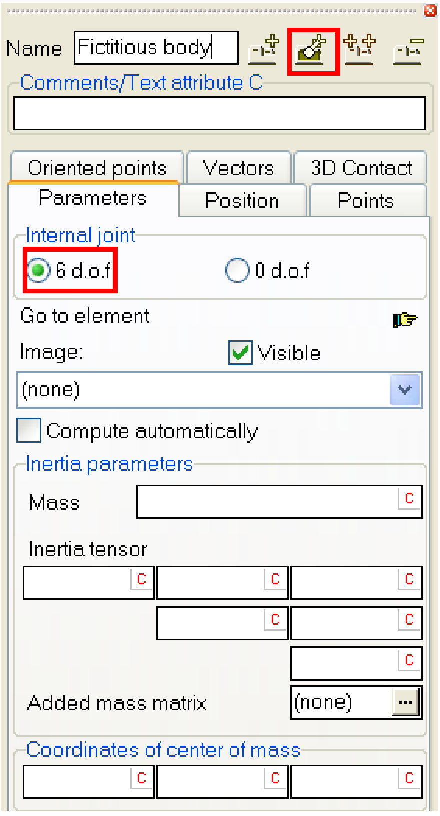
In a particular case when autodetection is set for all external elements and all the elements are connected with the same second body, the **Assign to all** menu command is highly efficient, Figure 3.76. In this case, all external connections are assigned by one operation!

Remark. An external element is ignored if the second body is not assigned.

3.6.3.3.3. Using fictitious body instead of external elements

A **fictitious body** is an alternative to external elements in the case of **included** subsystems. The following method is recommended.

1. An additional body with six degrees of freedom is added to the model by the button. The body model is this case contains an internal joint, Sect. 3.6.9.3, Figure 3.78. Inertia parameters of the body are zeroes.



Name: Fictitious body

Internal joint: ● 6 d.o.f ○ 0 d.o.f

Image: ☑ Visible

(none)

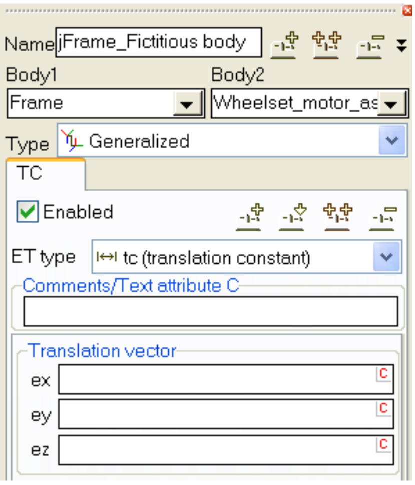
□ Compute automatically

Added mass matrix: (none)

Figure 3.78. Body with internal joint

2. All external elements are connected to this body.
3. In the upper subsystem or in the head object, the fictitious body is rigidly connected to the necessary body by a joint with zero degrees of freedom. It is recommended to use a generalized joint with one elementary transformation of the type *tc* (translation, constant), Figure 3.79, or a 6 d.o.f. joint with disabled coordinates. If necessary, additional rotation is introduced by elementary transformation of the type *rc* or *rt*. Translation and rotation allows a correct positioning of the fictitious body relative to the corresponding body.

After the fictitious body is connected to the external body, the internal joint with six degrees of freedom is automatically removed (ignored).



Name: jFrame_Fictitious body

Body1: Frame

Body2: Wheelset_motor_as

Type: Generalized

TC

☑ Enabled

ET type: tc (translation constant)

Figure 3.79. Joint rigidly connecting a frame with a fictitious body

Fictitious bodies are successfully used, e.g., in the module UM Tracked Vehicle.

3.6.4. Standard interface for setting local system of coordinates

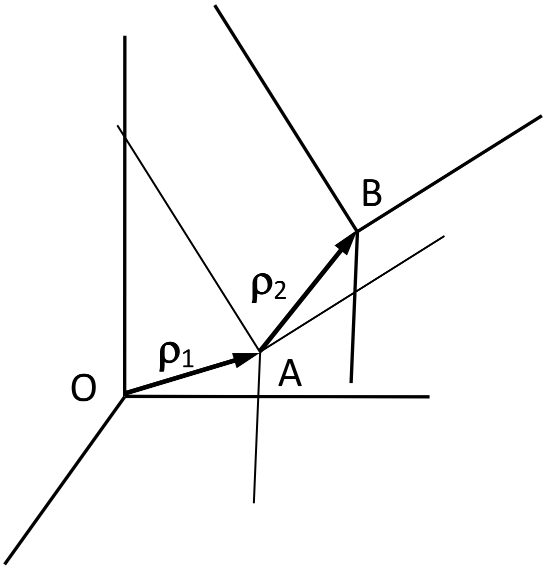
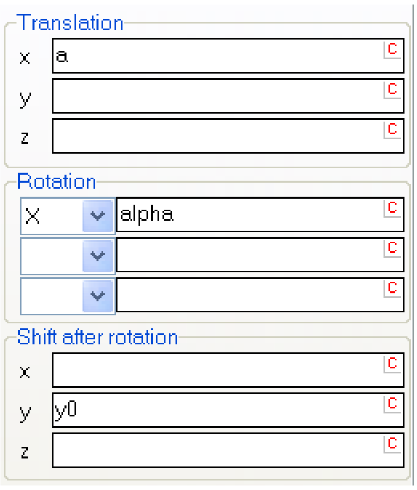


Figure 3.80. Setting position of local SC

The interface allows the user to define a fully parameterized position of a local system of coordinates (LSC) relative to SC of a body, a graphic object or a graphic element. Consider a SC with an origin in point O (SCO). It is necessary to define the position of SCB relative to SCO.

First, an auxiliary SC with the origin in point A (SCA) is introduced, which position relative to SCO is set by the shift vector ρ_1 and by a set of up to three sequential rotations. After that the position of point B relative to SCA is specified by the vector ρ_2. Thus, axis of SCA and SCB are parallel. In particular case $\rho_2 = 0$ and SCB coincides with SCA.

The auxiliary SCA is sometimes useful when shifts could be specified simpler relative to already rotated exes, i.e. relative to SCA.



Translation

x a

y

z

Rotation

X alpha

Shift after rotation

x

y y0

z

Figure 3.81. Window for specifying an LSC

The standard interface for setting LSC is shown in Figure 3.81. Data are entered in three groups.

- **Translation**. Projections of vector ρ_1 in SCO are entered.

- **Rotation**. Sequence of rotations specifies orientation of SCA and SCB relative to SCO. It is allowed up to three rotations. Angles of rotation are set here in degrees.

- **Shift after rotation**. Projections of vector ρ_2 in SCA are entered, i.e. shifts along axes of SCA.

Both projections of vectors and angles of rotation can be parameterized.

The interface is used for input of the following data types.

- Position of a subsystem, Sect. 3.6.3.1.
- Position of local SC of a curve, Sect. 3.6.7.
- Position of a graphic object, Sect. 3.6.8.1.
- Position of a graphic element relative to SC of graphic object, Sect. 3.6.8.4.
- Setting local SC for special force element of the “Bushing” type, Sect. 3.6.12.8.6.
- Setting local SC for scalar torques, Sect. 3.6.12.3.
- Setting local SC for 6 d.o.f. joint, Sect. 3.6.11.6.

3.6.5. Assigning Graphical Image to Object Element

Graphical images (graphical objects, GOs) can be assigned to the most of UM elements, for example, to the scene, bodies, some kinds of joints (e.g. a rod), some kinds of force elements (bipolar, linear) and so on. Clicking on a graphical image of element by mouse, the user can open the element description in the object inspector. For this purpose the *whole object animation mode* must be set in the animation window (Sect.3.5.1.2.2).

Remark. Visual operations with the mouse in animation window are available if the button ↖ on the tool panel is set to the 'down' state.

Graphical image *is assigned* to an object element from the list of the previously described GOs using a drop-down list in the inspector of the current element (body, for example).

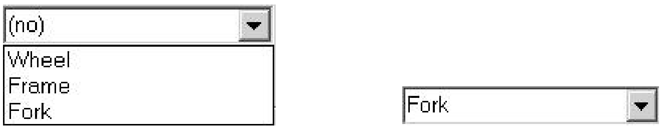


Figure 3.82. Assignment of a GO

The drop-down list contains *names* of GOs, which have been set by the user while describing graphical objects or by default. To make choosing a GO easier, it is recommended to give sensible names to GOs (Figure 3.82).

Activate the corresponding drop-down list and press the *Delete* key to *cancel* the assignment of a GO to the element.

There are some features by assigning GO to the object element.

GO can be assigned to each of the bodies; the same GO may correspond to several bodies (this is the main principle). The GO system of coordinate is automatically superposed with the SC of the corresponding body. All elements of mechanical system, which are fixed relative to the base SC0 (e.g. plane supporting a rolling ball, obstacles etc.), can be presented by a single GO, which is assigned to the scene image on the **Object | General** inspector tab, the parameter **Scene image**. The scene image can be assigned to the body **Ground** as well, Sect. 3.6.9.7.

3.6.6. Assignment of graphic images to rods, linear and bipolar force elements

A GO can also be assigned to a linear or bipolar force element (Sect. 3.6.12.3, 3.6.12.1) or to a weightless rod constraint (Sect. 3.6.11.9). These elements link two points of different bodies. UM automatically puts the assigned GO between the points. There is an obligatory condition while creating such a GO: a GO corresponding to a linear or bipolar force element or to a rod constraint must be located along the Z axis of the SC GO.

Consider two methods for development of GO.

1. Simplified

In this case the button must be in the ‘up’ state. The program puts the point (0, 0, **0**) of GO to the first attachment point to the first body, and the point (0,0,**1**) is put to the second attachment point. Usually, the total length of the GO must be 1m. The GO is stretched or compressed according to the real length of the element, Figure 3.83.

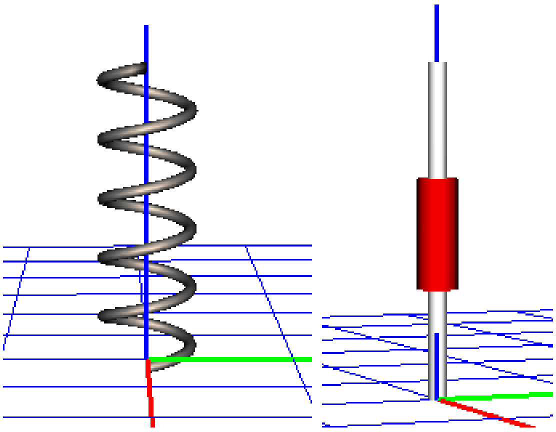


Figure 3.83. Examples of simplified images of a cylindrical spring and a damper. The length of images is 1m

2. **An advanced** method allows escaping stretching or compressions of some parts of GO, Figure 3.84.

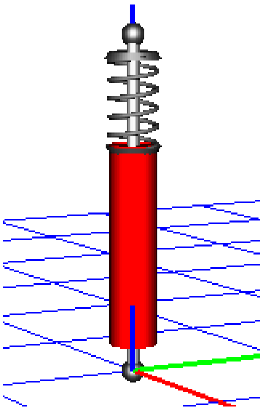
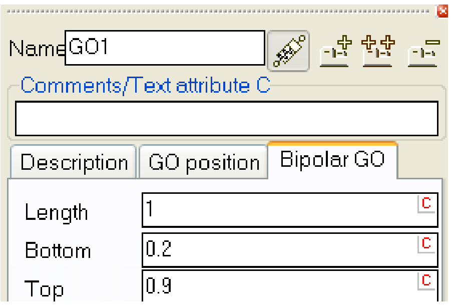


Figure 3.84. Example of an advanced image of a shock absorber

In this case, the button state must be ‘down’. Drawing of the image depends on the parameters in the **Bipolar GO** tab: Length, Bottom, Top, Figure 3.85.



Name GO1

Comments/Text attribute C

Length 1

Bottom 0.2

Top 0.9

Figure 3.85. Parameters of a bipolar GO

The program locates the image in such a way that the point (0,0, **Bottom**) of GO is set into the first attachment point, and the point (0,0,**Top**) of GO coincides with the second body point. Orientation and length of the image is computed respectively. Thus, in spite of the image is oriented along the Z axis in SC GO, an arbitrary orientation of the element in the model can be obtained.

Use the ☑ Not stretch key to prohibit the stretching some of the graphic elements included in the image. In this case the □ top key indicates the position of the element in the top or in the bottom part of the image.

3.6.7. Input of 3D curves

3D curves are used in the point-curve contact element, Sect. 3.6.12.6.3, Chapter 2, Sect. *Point-Curve contact*.

In general, the curve equation is the dependence of the radius-vector on the scalar parameter p in some SC as

$$\rho = \rho(p), \quad p \in [p_{\min}, p_{\max}],$$

The same curve in a scalar form is set by three relations

$$x = x(p),\ y = y(p),\ z = z(p),$$

$$\rho(p) = (x(p), y(p), z(p))^T$$

The following types of 3D curves are considered:

- open curve, which end points differ, $\rho(p_{\min}) \neq \rho(p_{\max})$;
- closed curve with equal end points, $\rho(p_{\min}) = \rho(p_{\max})$;
- periodic curve is a closed curve, which have smooth derivative at the end points (tangents coincide) $\rho'(p_{\min}) = \rho'(p_{\max})$; the stroke here corresponds to derivative with respect to the parameter *p*.

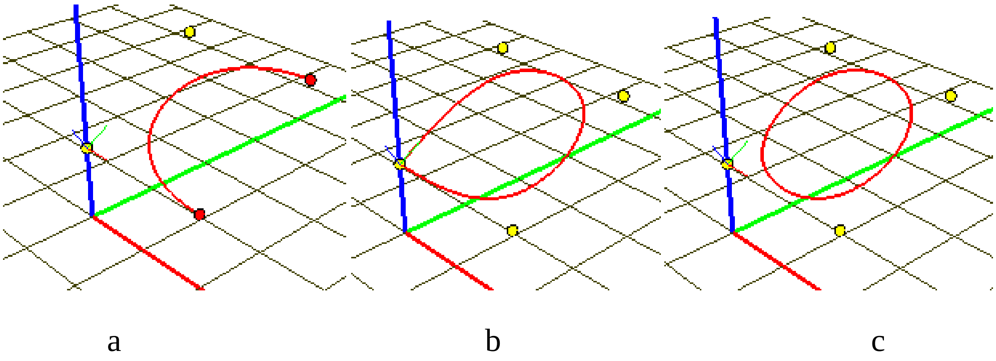


Figure 3.86. Curves of different types

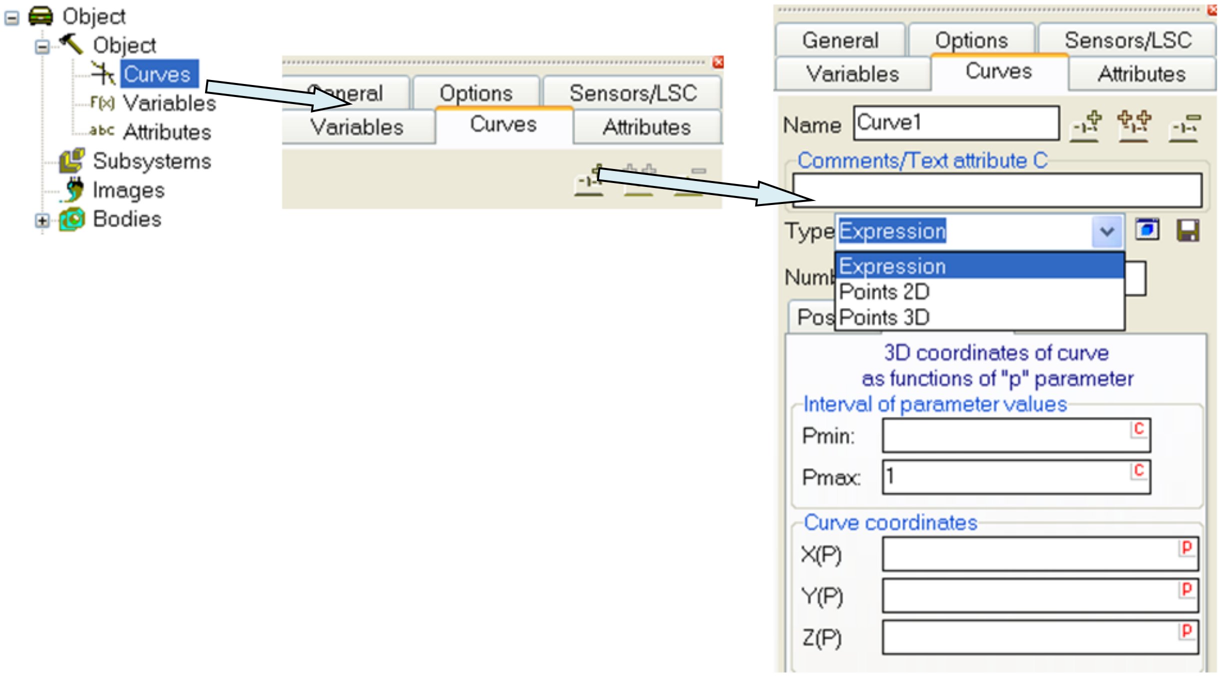
Figure 3.86 shows an open (a), closed (b) and periodic (c) curves.

If the curve is used in the contact element, each of the two end points of an open curve can be either *locking* or *unlocking*. The locking end point keeps the contact point on the curve, while the contact disappears by passing through the unlocking point.

Three methods are used for description of curves in UM:

- analytic expressions; elements of list of variables can be used in the expressions, Sect. 3.5.2.4.6.3;
- plane pointwise curve;
- 3D pointwise curve.

3.6.7.1. Adding new curve



Name Curve1

Type Expression

Pmax: 1

Figure 3.87. New curve

To create a new curve, open the curve editor in the inspector by the Object | Curves item of the element tree and add a curve by the button, Figure 3.87.

Type of curve

Select one of the three types of curve in the drop-down menu:

- Expression
- Points 2D
- Points 3D

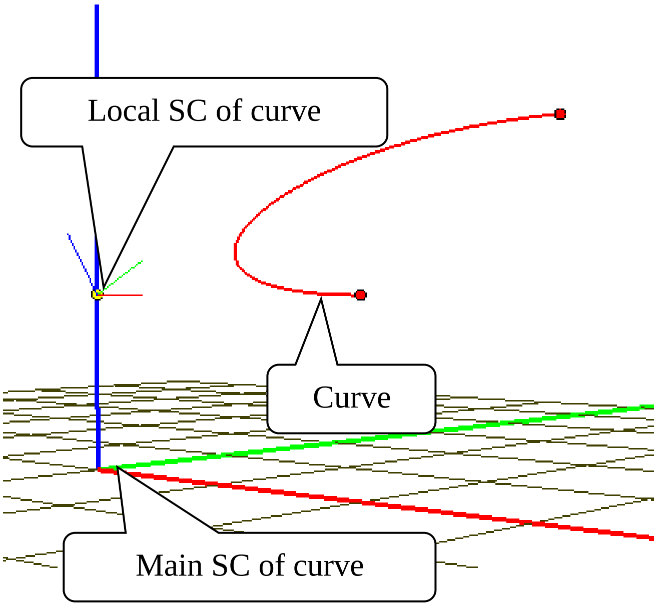


Figure 3.88. Systems of coordinates related to curve

Main and local SC of curve

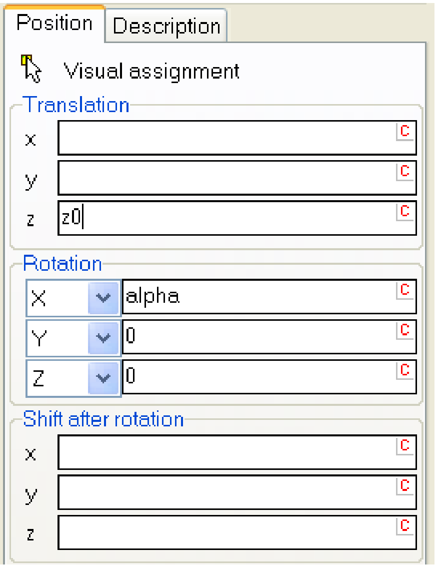
Two systems of coordinates are related to a curve, Figure 3.88:

- **main SC of curve** will coincide with SC of the body containing the curve;

- **local SC of curve** is the system of coordinates in which the curve is defined by an expressions or a set of points.

The local SC can coincide with the main one.

The ▣ button is used for visualization of the curve, Figure 3.87, right. The main SC is drawn by thick lines, the local one by thin lines, Figure 3.88.



Translation

x

y

z z0

Rotation

X alpha

Y 0

Z 0

Shift after rotation

x

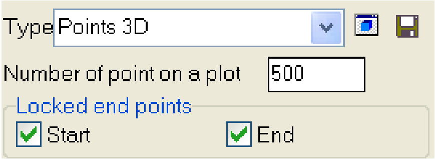
y

z

Figure 3.89. Position of local SC relative to main one

Position of the local SC relative to the main one

To set the parameterized position of the local SC relative to the main one, the standard interface is used, Figure 3.89, Sect. 3.6.4.



Type Points 3D

Number of point on a plot 500

Locked end points

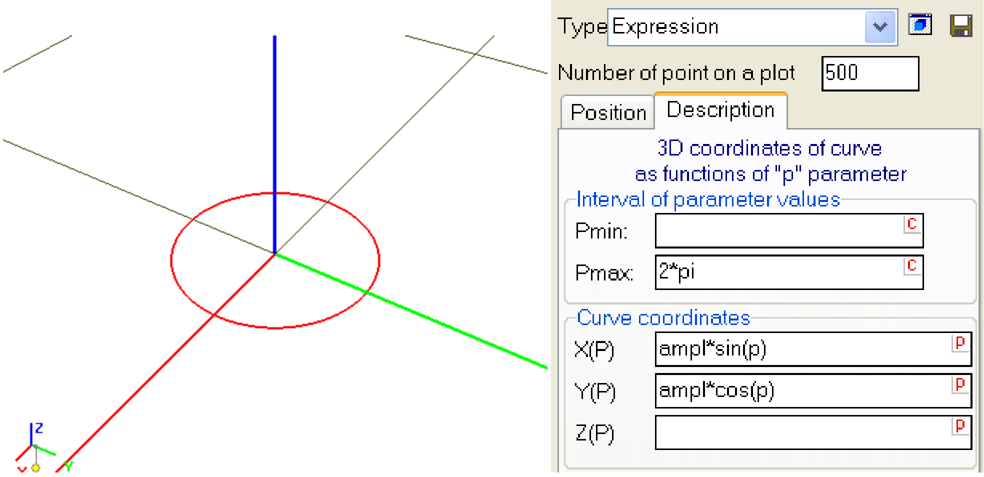
☑ Start ☑ End

Figure 3.90. Setting locking end points

Locking end points

The “Locking end points” group is used for locking or unlocking the curve end points, Figure 3.90. Checked points are locked. Locked points are drawn by thick points, Figure 3.88.

3.6.7.2. Setting curves by analytic expressions



Type Expression

Number of point on a plot 500

Position | Description

3D coordinates of curve
as functions of "p" parameter

Interval of parameter values

Pmin:

Pmax: 2*pi

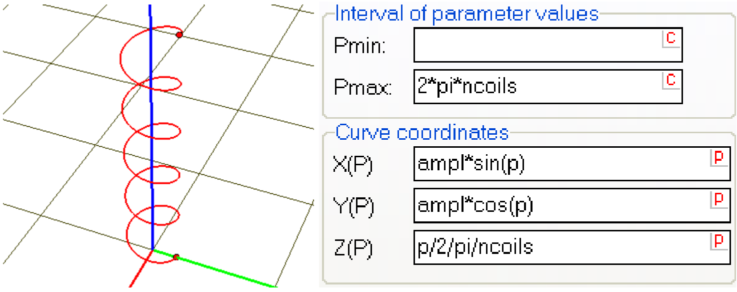
Curve coordinates

X(P) ampl*sin(p)

Y(P) ampl*cos(p)

Z(P)

Figure 3.91. Expressions for a circle



Interval of parameter values

Pmin:

Pmax: 2*pi*ncoils

Curve coordinates

X(P) ampl*sin(p)

Y(P) ampl*cos(p)

Z(P) p/2/pi/ncoils

Figure 3.92. Analytical description of a spiral with parameterized number of coils

A big variety of classical plane and spatial curves can be described by analytic dependence on p parameter, Figure 3.91, Figure 3.92.

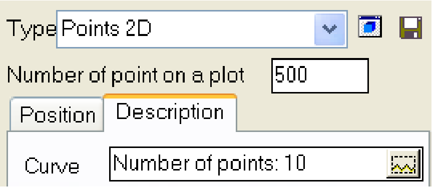
The curve is specified by

- interval of parameter *p* values,
- dependencies of the curve coordinates on *p* in the local SC of the curve; elements of the list of variables can be used in the expressions, Sect. 3.5.2.4.8.

Both identifiers and standard functions are allowed in the expressions, Sect. 3.5.2.4.3.

3.6.7.3. Setting curves by 2D points

This type of curve is used in the case when all points of the spatial curve lie in a plane coinciding with the XY plane of the local SC, Figure 3.93.

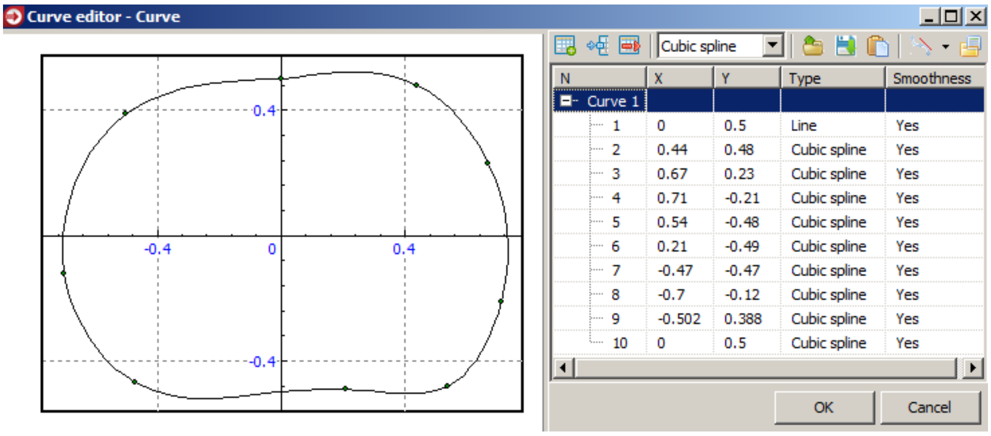


Type Points 2D

Number of point on a plot 500

Curve Number of points: 10

Figure 3.93. Curve as a set of points in a plane



N	X	Y	Type	Smoothness
Curve 1				
1	0	0.5	Line	Yes
2	0.44	0.48	Cubic spline	Yes
3	0.67	0.23	Cubic spline	Yes
4	0.71	-0.21	Cubic spline	Yes
5	0.54	-0.48	Cubic spline	Yes
6	0.21	-0.49	Cubic spline	Yes
7	-0.47	-0.47	Cubic spline	Yes
8	-0.7	-0.12	Cubic spline	Yes
9	-0.502	0.388	Cubic spline	Yes
10	0	0.5	Cubic spline	Yes

Figure 3.94. 2D curve editor

To define the plane curve, click on the 📈 button and open the curve editor, Figure 3.94. See Sect. 3.5.3 for more details.

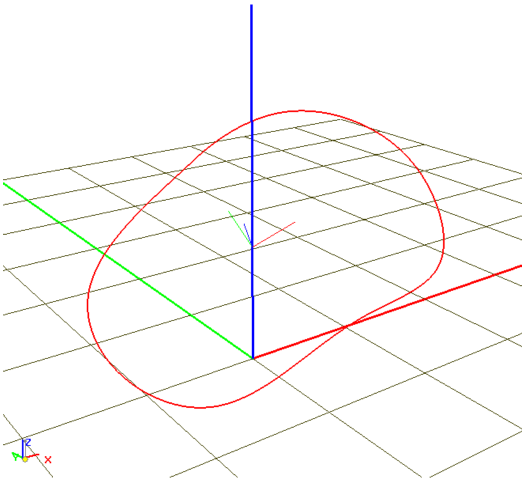
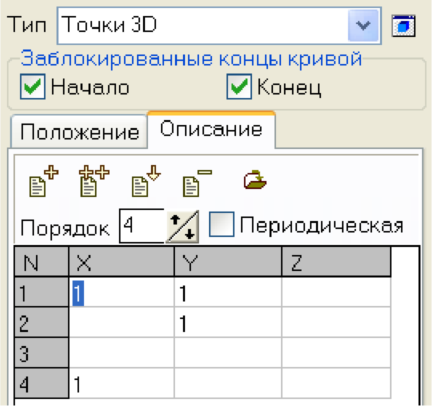


Figure 3.95. Spatial positioning of plane curve relative to main SC

Remark 1. Coordinates of points can be parameterized.

Remark 2. Shift and rotation of the local SC relative to the main SC allows the spatial positioning of the curve, Figure 3.95.

3.6.7.4. Setting curves by 3D points. B-spline



Тип Точки 3D

Заблокированные концы кривой

☑ Начало ☑ Конец

Положение | Описание

Порядок 4 □ Периодическая

N	X	Y	Z
1	1	1	
2		1	
3			
4	1		

Figure 3.96. Set of points for 3D curve

The B-splines of order m=2-10 are used for description of 3D curves by a set of points - vertices. B-splines are power functions of order (m-1). Thus, B-splines of second order are polylines passing through the given vertices. The B-spline of the forth order is the most frequently used one; in general this spline has smooth second derivatives.

Editing of list of vertices

The following buttons are used for development of the list of vertices (Figure 3.96):

- add a vertex with zero coordinated to the end of the list;

- copy the current vertex and add it to the end of the list;

- insert a vertex with zero coordinates before the current selected vertex;

- delete the current vertex;

- read a set of points from a text file; the file must contain Cartesian coordinates of vertices in three columns like

1.47333433255E-1	1.3808948848	0.1
1.5971679659E-1	1.4227401843	0.12
1.7768028748E-1	1.4645854838	0.12
2.5001219032E-1	1.5064307834	0.1

………..

Parameterization of vertices

Coordinates of vertices can be parameterized, i.e. identifiers and standard functions can be used.

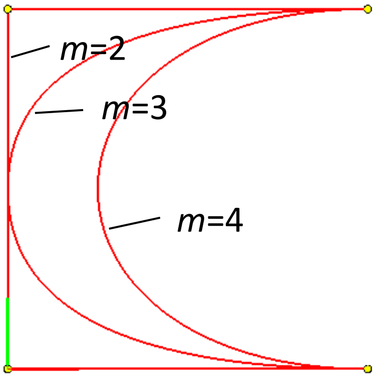
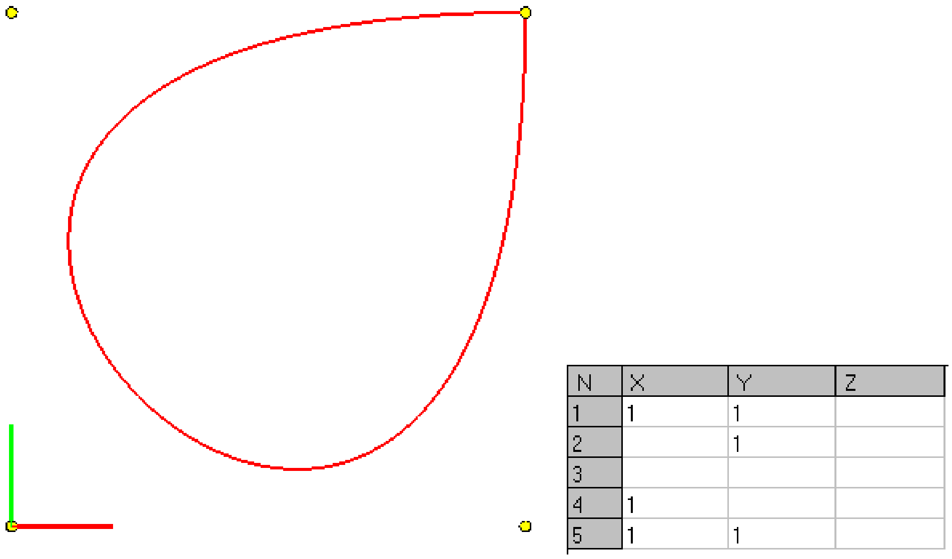


Figure 3.97. Open B-splines of different order for the same set of points

Order of B-spline

Order of a B-spline specifies the smoothness of the curve: the higher is the order the smoother is the curve, Figure 3.97. The minimal value m=2 (polyline), the maximal order is m=10 (UM restriction). Besides, the order cannot exceed the number of vertices in the list. As a rule, if m>2, the curve does not pass through the vertices exactly. There exists a simple trick how to make the curve passing exactly through a separate vertex.

The recommended value of the order is m=4.

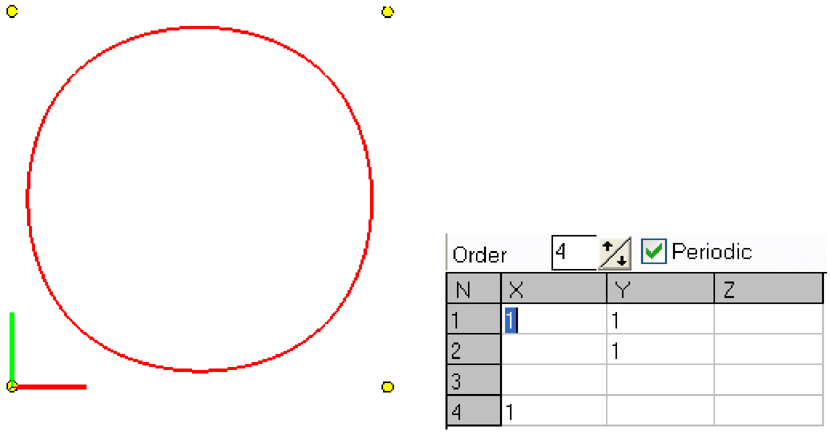


N	X	Y	Z
1	1	1	
2		1	
3			
4	1		
5	1	1	

Figure 3.98. Closed unperiodic B-spline, m=4

Closed unperiodic B-spline

To create a closed unperiodic B-spline, the first and the last vertices in the list must be equal, Figure 3.98. In the closure point the tangent is not smooth, which corresponds to the derivative discontinuity.



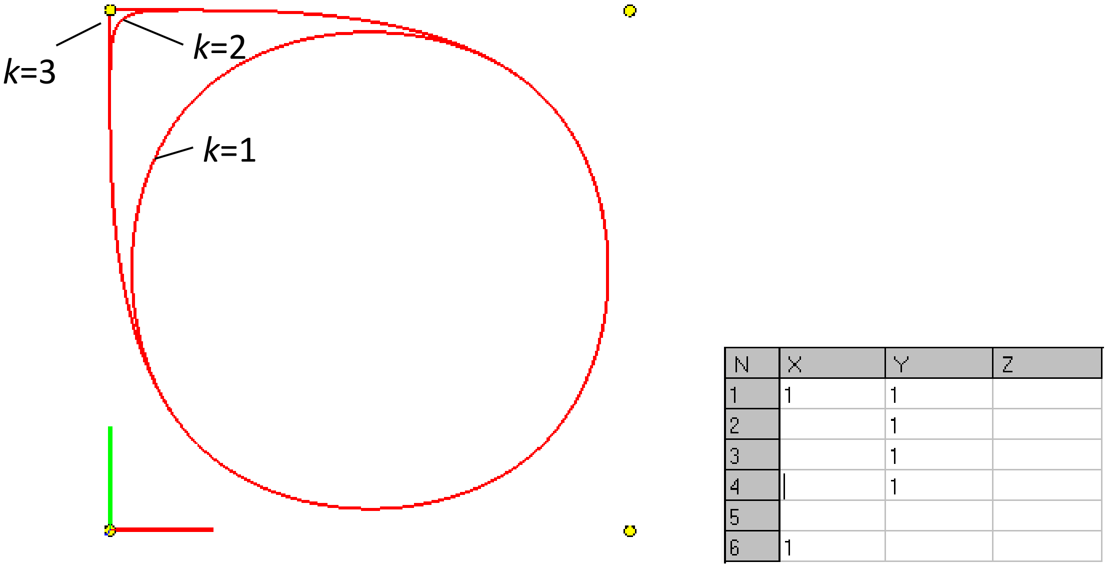
Order 4 ☑ Periodic

N	X	Y	Z
1	1	1	
2		1	
3			
4	1		

Figure 3.99. Periodic B-spline, m=4

Periodic B-spline

To specify a periodic B-spline, check the **Periodic** key. As opposed to the closed unperiodic spline, as a rule the fist and the last vertices are different, Figure 3.99.



N	X	Y	Z
1	1	1	
2		1	
3		1	
4		1	
5			
6	1		

Figure 3.100. Curves with different multiplicity k of the second vertex, m=4 (left). List of vertices in which the second vertex has multiplicity 3

Multiplicity of vertex

Multiple vertices, i.e. the successive vertices with equal coordinates, lead to decrees of the curve smoothness in this point. If the multiplicity of vertex is equal to *m*-1, the curve exactly pass through this point with a jump of the tangent.

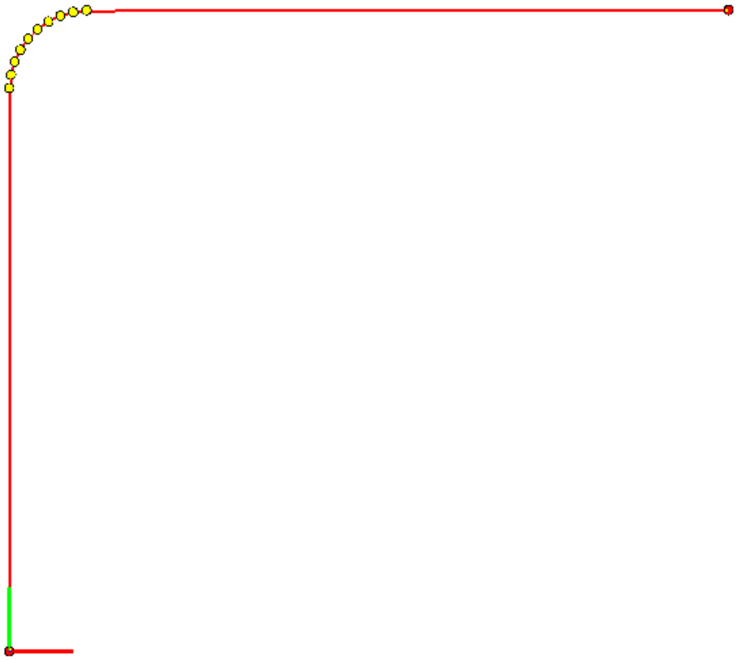


Figure 3.101. Approximation of curve consisting of three sections

Approximation of curve with B-spline

To apply the B-spline for a sufficiently good approximation of a curve, it is required a big enough number of vertices. For instance, approximation of the curve in Figure 3.101 having three sections requires a big number of vertices on the arc part.

3.6.8. Input of Graphical Objects

For description of graphical objects (GO), a built-in graphical editor is used. It is accessible with the inspector tab **Images**. Notice that it is recommended to describe the rest of object elements *after* the complete description of graphical images. In this case the visual verification of input data is possible, and it allows avoiding a lot of input errors.

Remark. Recently the most part of images of bodies are created by the user in one of the CAD programs, Sect. 3.10. Nevertheless, images of force elements like strings and dampers is recommended to develop by the standard UM tools. Besides, sometimes full parameterization of UM images is important by the development of models.

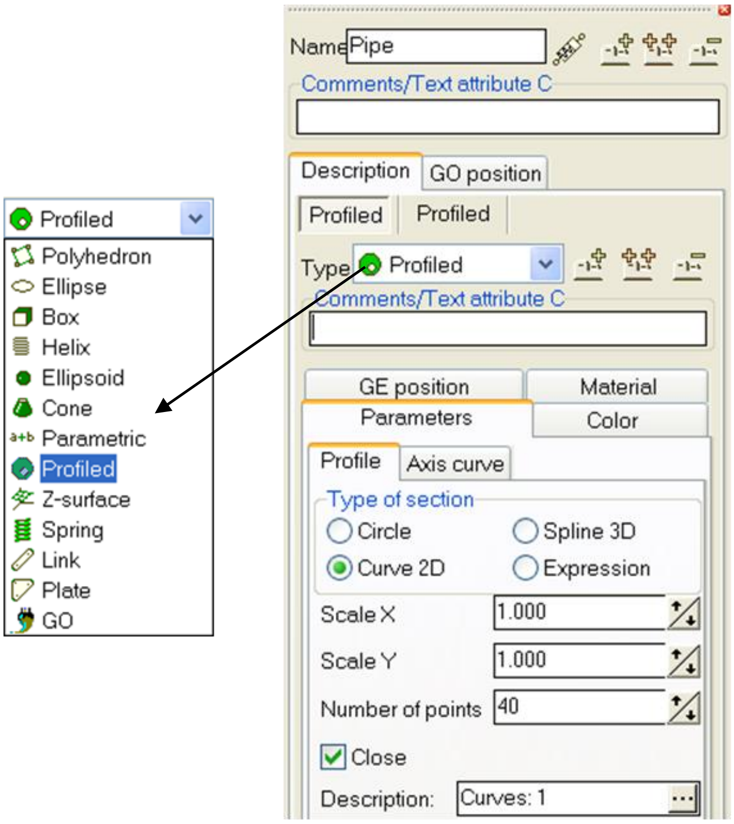
3.6.8.1. Lists of Graphical Objects and Graphical Elements

To visualize any UM element (body, joint, force element and others; bodies will only be mentioned below), it is necessary to assign a *graphical object* to it. GO is a set of *graphical elements* (GEs), which should be created using the built-in graphical editor. The same GO may belong to different (geometrically identical) bodies. For example, for description of GO of a bogie having four similar wheels (even if they have different masses), it is enough to create two GOs: a body and a wheel. The same wheel GO is assigned to each of the wheels.

It is desirable to create a full set of GOs before input of data for bodies, joints etc. In this case the visual verification of the object development process is possible.

After assigning a GO to an object element, UM superposes the SC of GO with the element SC in a certain rule, but it should be remembered, that the shape and sizes of GO *are not connected at all* with inertia parameters of body, stiffness of spring and other parameters of the object elements, which should be described separately. Exclusion presents the case when the option for automatic calculation of body inertia parameters is turned on.

To add or modify *graphical objects or elements (GOs or GEs)* use the **Images** tab of the object inspector, Figure 3.102.



Name: Pipe

Type: Profiled

Type of section: ○ Circle ● Curve 2D ○ Spline 3D ○ Expression

Scale X: 1.000

Scale Y: 1.000

Number of points: 40

☑ Close

Description: Curves: 1

Figure 3.102. Graphical object in the data inspector

Each GO is a set (list) of *graphical elements* (GE). Lower buttons are intended for:

- adding a new GE;
- copying (duplicating) the current GE;
- deleting the current GE.

To set (change) the type of the current GE a drop-down box is used, which contains names of the standard GEs:

- Polyhedron, Sect. 3.6.8.2.1;
- Ellipse, Sect. 3.6.8.2.2;
- Box, Sect. 3.6.8.2.3;
- Spiral, Sect. 3.6.8.2.4;
- Ellipsoid, Sect. 3.6.8.2.5;
- Cone, Sect. 3.6.8.2.6;
- Parametrical GE, Sect. 3.6.8.2.7;
- Profiled GE, Sect. 3.6.8.2.8;
- Z-surface, Sect. 3.6.8.2.9;
- Spring, Sect. 3.6.8.2.10;
- Link, Sect. 3.6.8.2.11;
- Plate, Sect. 3.6.8.2.12;
- Reference to GO, Sect. 3.6.8.2.13.

While describing almost all parameters of GEs, identifiers and symbolic expressions may be used.

Remark. Changing a type deletes previously entered data for the current GE.

3.6.8.2. Input of graphical elements (GE)

Each GE has four groups of parameters located in the different *tabs* (Figure 3.102):

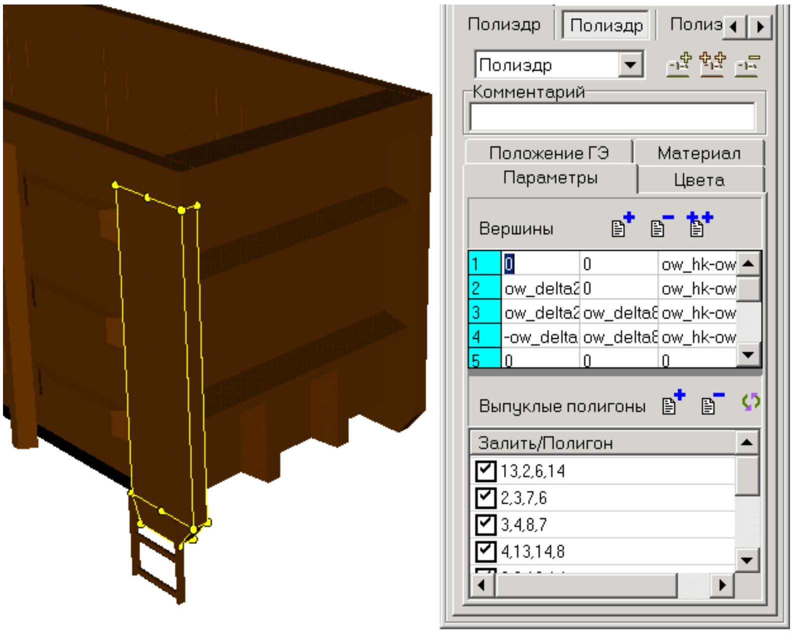
- **Parameters** – a tab with GE parameters depending on its type;
- **Colors** – a tab with GE color information;
- **Position** – a tab with parameters describing position and orientation of the current GE in the SC of the current GO;
- **Material** – parameters defining inertia properties of the GE material (such as density). They are used for automatic calculation of inertia parameters of bodies.

3.6.8.2.1. Polyhedron

A polyhedron is used when a GE of irregular shape is needed. An UM polyhedron is a set of 3D vertices forming one or more polygons. The polygons may be drawn both in wired and surface modes.

To describe a polyhedron, the following parameters should be set (Figure 3.103).

- Coordinates of vertices. If only vertices are given, the result is a single polyline connecting the vertices. If several polygons are needed, the lower part of the form shown in (Figure 3.103) is to be filled.
- For each of the polygons, vertex indices separated by commas should be pointed; filled polygons should be marked with ☑.



1	0	0	ow_hk-ow
2	ow_delta2	0	ow_hk-ow
3	ow_delta2	ow_delta8	ow_hk-ow
4	-ow_delta	ow_delta8	ow_hk-ow
5	0	0	0

Залить/Полигон
☑ 13,2,6,14
☑ 2,3,7,6
☑ 3,4,8,7
☑ 4,13,14,8

Figure 3.103. Polyhedron

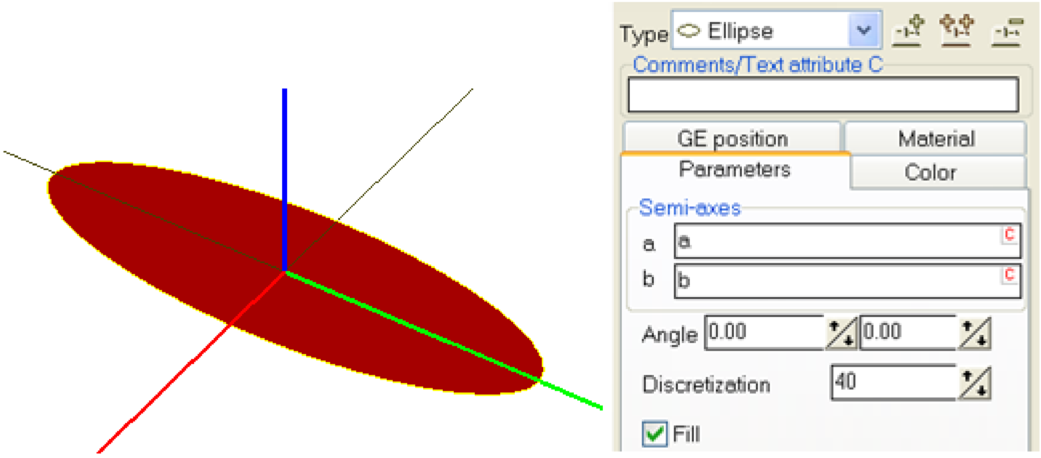
Example. Four vertices might set a tetrahedron

(0 0,0), (1,0,0), (0,1,0), (0,0,1)

by four polygons

(1,2,4), (4,2,3), (1,4,3), (1,3,2).

3.6.8.2.2. Ellipse



Type ○ Ellipse

a a

b b

Angle 0.00 0.00

Discretization 40

☑ Fill

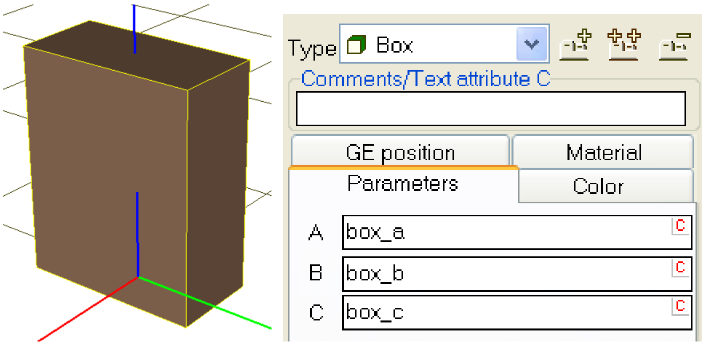
Figure 3.104. Example of ellipse

Parameters of an ellipse (Figure 3.102):

- **a, b** are the semi-axes;
- boundary values of an **angles** for a elliptic sector. The default values (0,0) correspond to a full ellipse;
- **discretization** is the number of points approximating the ellipse;
- **fill** option determines the element either as a surface or a line.

3.6.8.2.3. Box

All ribs of a *box* are parallel to the axes of the GE-fixed SC. The box parameters are



Type: Box

Comments/Text attribute C

GE position | Material | Parameters | Color

A: box_a

B: box_b

C: box_c

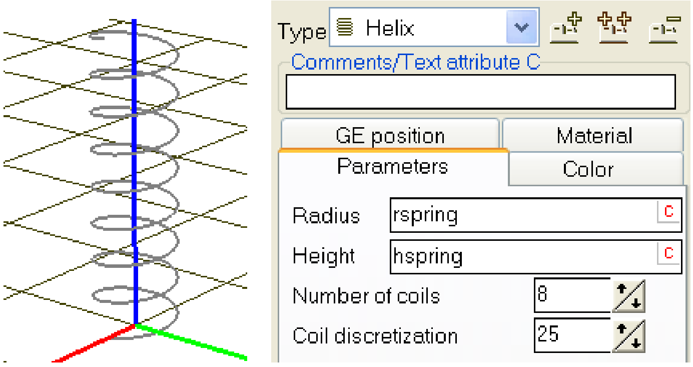
Figure 3.105. Example of box

Length, width, height **A, B, C** as constant symbolic expressions, Figure 3.105.

3.6.8.2.4. Spiral

This GE is used customarily as to draw elastic linear of bipolar force elements. The spiral parameters are (Figure 3.106):

- coil radius *r* (a constant symbolic expression);
- spring height *H* (a constant symbolic expression);
- the number of coils;
- coil discretization (number of points per coil).



Type Helix

Radius rspring

Height hspring

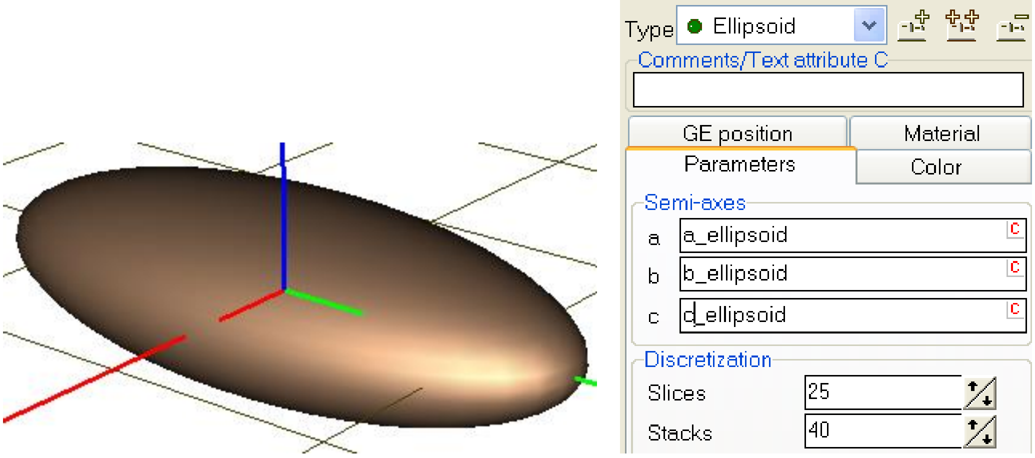
Number of coils 8

Coil discretization 25

Figure 3.106. Parameters of a spiral

The axis of the spiral coincides with the z-axis of the GE-fixed SC.

3.6.8.2.5. Ellipsoid



Type: Ellipsoid

Semi-axes

a: a_ellipsoid

b: b_ellipsoid

c: c_ellipsoid

Discretization

Slices: 25

Stacks: 40

Figure 3.107. Parameters of an ellipsoid

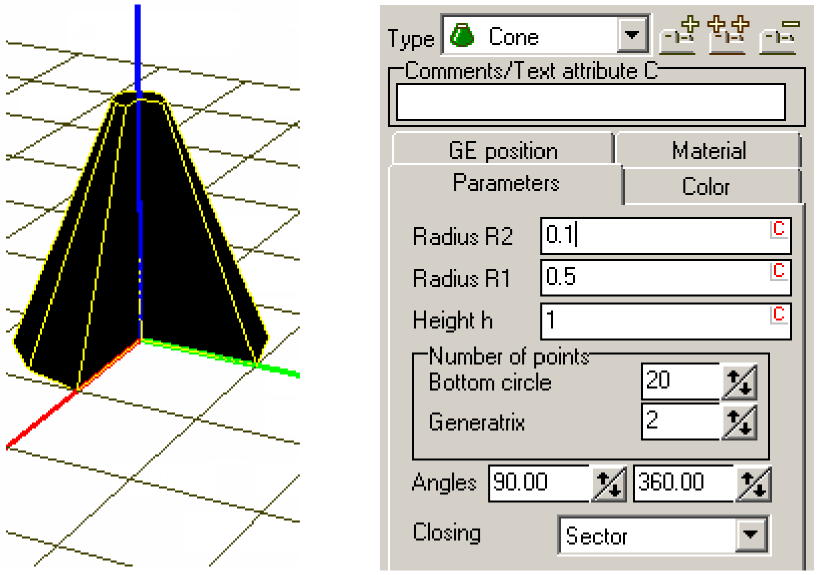
This GE creates an ellipsoid, in particularly a sphere (Figure 3.107).

The parameters of an ellipsoid are:

- **a, b, c** are the semi-axes;
- **Slices, Stacks** are the numbers characterizing the surface discretization.

3.6.8.2.6. Cone

The cone parameters are shown in Figure 3.108.



Type: Cone

Comments/Text attribute C

Radius R2: 0.1

Radius R1: 0.5

Height h: 1

Number of points

Bottom circle: 20

Generatrix: 2

Angles: 90.00 360.00

Closing: Sector

Figure 3.108. Cone

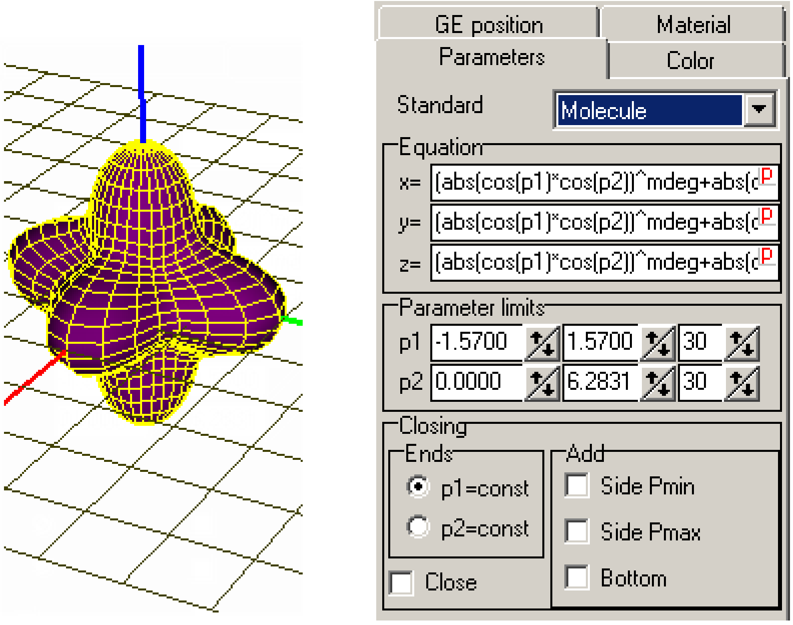
The GE allows getting such images as cylinder, cone and truncated cone, as well as a part of these surfaces.

For description of the conic surface the user should set the following parameters:

- **R1, R2** are the radii of the upper and bottom circle-bases;
- **h** is the height of cone;
- **Angles** are the boundary values (in degrees) that form a plate angle at the Z-axis delimiting the conic surface. The angles are counted from the X-axis counterclockwise. If both angles are zeros, the full cone (cylinder) is generated;
- **Number of points** on the circle bases and the generating line, which define a discretization of the conic surface;
- **Closing** is the switch with three positions *(None)*, *Sector*, *Segment*. It defines how the conical surface should be closed.

3.6.8.2.7. Parametrical GE

The GE provides the wide possibilities to describe the complex surface by analytic expressions (Figure 3.109). Description of both 3D curves and surfaces is allowed. The user should describe them as arbitrary functions of a single (*p1* or *p2*) or the two parameters (*p1* and *p2*).



Standard: Molecule

Equation

x= (abs(cos(p1)*cos(p2))^mdeg+abs(c

y= (abs(cos(p1)*cos(p2))^mdeg+abs(c

z= (abs(cos(p1)*cos(p2))^mdeg+abs(c

Parameter limits

p1 -1.5700 1.5700 30

p2 0.0000 6.2831 30

Closing

Ends

● p1=const

○ p2=const

□ Close

Add

□ Side Pmin

□ Side Pmax

□ Bottom

Figure 3.109. Parametrical GE

The parameters of the parametrical GE are

- the drop-down menu **Standard** contains a set of parametrical elements:

Plane	Ellipsoid	Ring
Torus	Cone	Paraboloid
Spring	Horn	Molecule (Figure 3.109)
Smooth cube	Gear	Elliptic gear

- the **Equation** group contains analytical expressions describing the dependence of the Cartesian coordinates X, Y and Z on a parameter *p1* or *p2* for a 3D curve, or *p1* and *p2* for a 3D surface;
- the **Parameter limits** group sets the minimal and the maximal values of the parameters, and the number of points inside the interval to approximate the surface; for example, the values *p1* from –1.4 to +1.2, *p2* from –0.2 to 4.8 modify the surface above as shown in Figure 3.110.

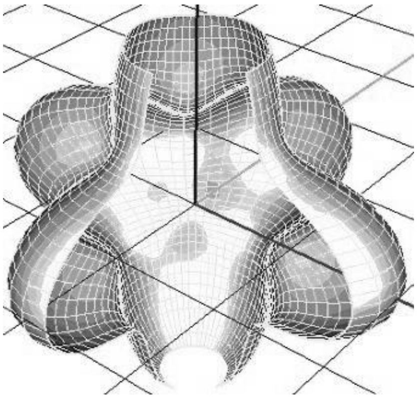


Figure 3.110. Influence of the parameters p1, p2 on the GE shape

- **Closing** is a possibility to complete the surface to the closed one by adding lids; since the parameters *p1* and *p2* are equivalent, so a lid can be treated either as the *p1=const* or as the *p2=const* surfaces (Figure 3.111); for this purpose the corresponding switch is intended (Figure 3.109).

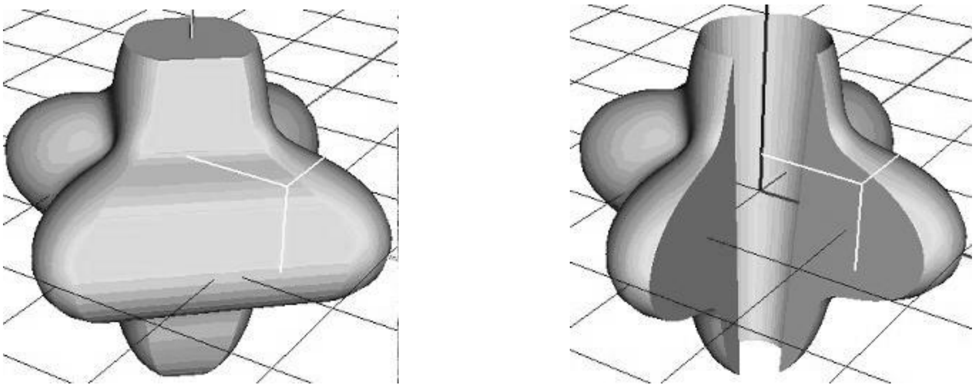
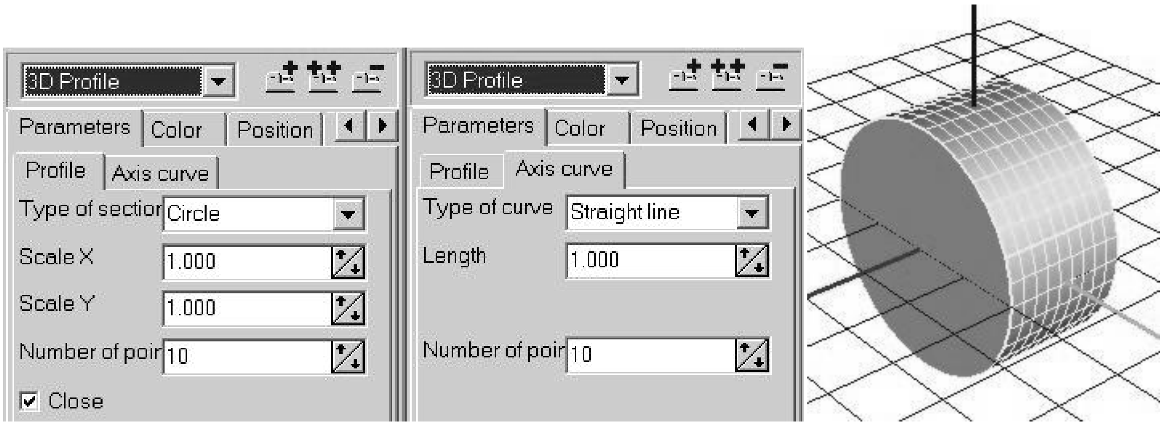


Figure 3.111. Different ways of making the parametric GE closed

3.6.8.2.8. Profiled GE

This GE has many possibilities. It defines a surface formed by moving a certain *profile* curve along another *axis curve*. There is a number of ways to build the curves.



Type of sectior Circle
Scale X 1.000
Scale Y 1.000
Number of poir 10
☑ Close

Type of curve Straight line
Length 1.000
Number of poir 10

Figure 3.112. Profiled GE: a cylinder

An example of the profiled GE is shown in Figure 3.112.

Parameters of the *profile* are:

- **Type of section** – possible values are:
  - *Circle*: semi-axes are **Scale X, Scale Y**;
  - *Curve 2D*: a section given by a number of points, holes are allowed, too;
  - *Spline 3D*: a set of consecutive sections given by points;
  - *Expression*: a section given by analytical formulas.
- **Scale X, Scale Y** – scales in X and Y axes;
- **Number of points** to approximate the section;
- **Close** – a switch for automatic adding lids to make the surface closed.

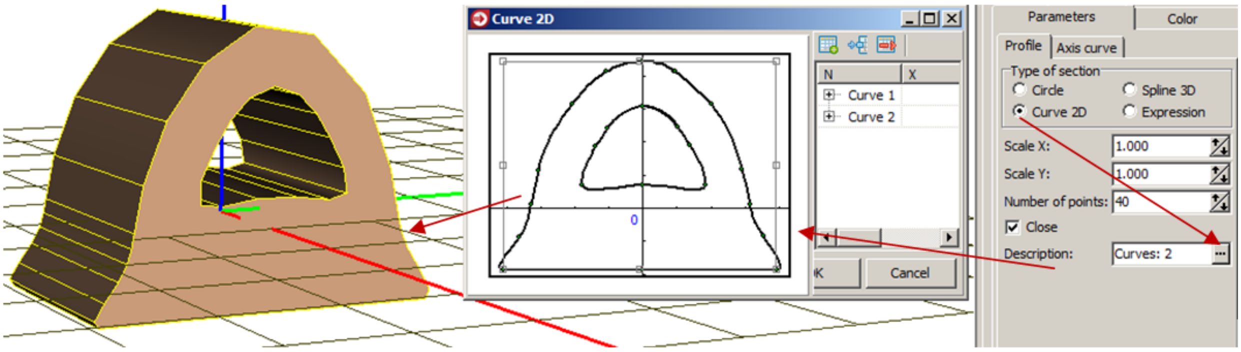
Parameters of the *axis curve* are:

- **Type of curve** – possible values are:
  - Straight line;
  - Circle;
  - *Curve*: given by points;
  - *Expression*: given by analytical formulas.
- **Length** – the length of the axial line (in case of straight axis curve);
- **Number of points** to approximate the axial curve.

Using various combinations «type of profile curve – type of axis curve», it is possible to get different shapes of GE.

Examples of profiled elements

After choosing the profile type Curve 2D or Curve 3D, a new input parameter Description appears (Figure 3.113)



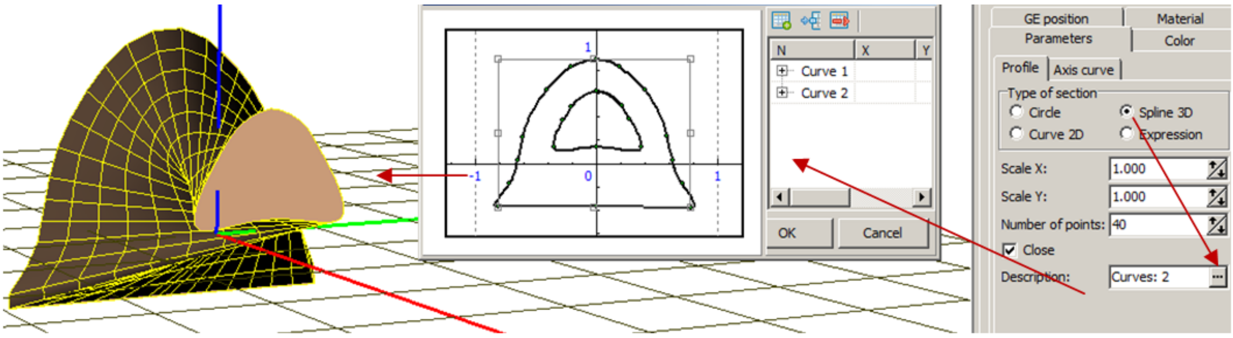
Type of section
○ Circle ○ Spline 3D
● Curve 2D ○ Expression
Scale X: 1.000
Scale Y: 1.000
Number of points: 40
☑ Close
Description: Curves: 2

Figure 3.113. Profiled GE: a Curve 2D section

Clicking mouse button on ..., the user can open a window of the curve editor (see Sect. 3.6.8.6) and input the section.

If several curves are presented with the Curve 2D type then the profile section will be multiple connected (Figure 3.113).

In case of the Curve 3D section type, several curves are treated as a set of consecutive sections along the axis curve (Figure 3.114).

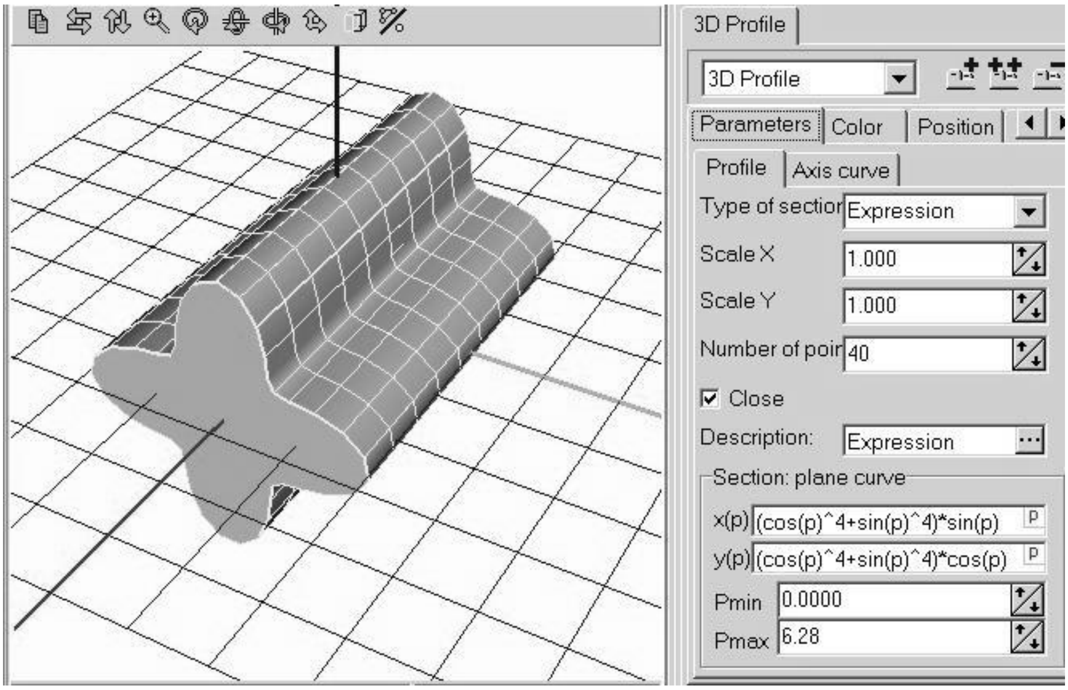


Type of section
○ Circle ● Spline 3D
○ Curve 2D ○ Expression
Scale X: 1.000
Scale Y: 1.000
Number of points: 40
☑ Close
Description: Curves: 2

Figure 3.114. Profiled GE: a Curve 3D section

After choosing the **Expression** type of profile section, a new parameter group appears (Figure 3.115):

- **x(p), y(p)** are the parametric equations of the 2D profile curve depending on **p**;
- **Pmin, Pmax** are the **p** interval boundaries.



3D Profile

Type of sectior: Expression

Scale X: 1.000

Scale Y: 1.000

Number of poir: 40

☑ Close

Description: Expression

x(p) (cos(p)^4+sin(p)^4)*sin(p)

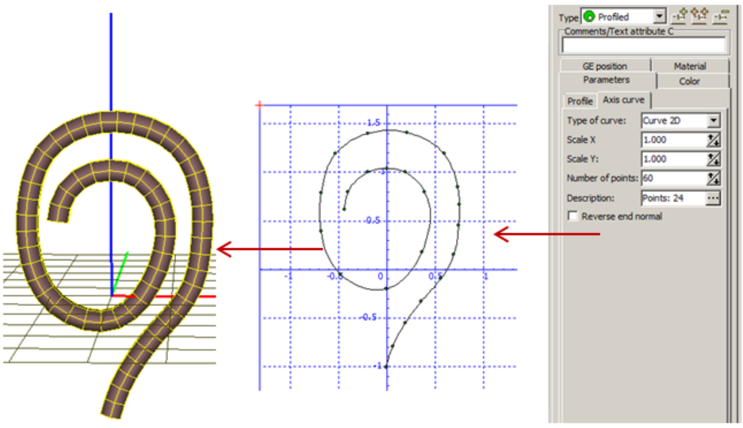
y(p) (cos(p)^4+sin(p)^4)*cos(p)

Pmin 0.0000

Pmax 6.28

Figure 3.115. Profiled GE: section given by formulas

Analogously, the axis curve can be described by points (Figure 3.116).



Type: Profiled

Type of curve: Curve 2D

Scale X: 1.000

Scale Y: 1.000

Number of points: 60

Description: Points: 24

☐ Reverse end normal

Figure 3.116. Profiled GE: a 2D axis curve given by points

Finally consider the image of a wheelset, which is created with the profiled elements, Figure 3.117.

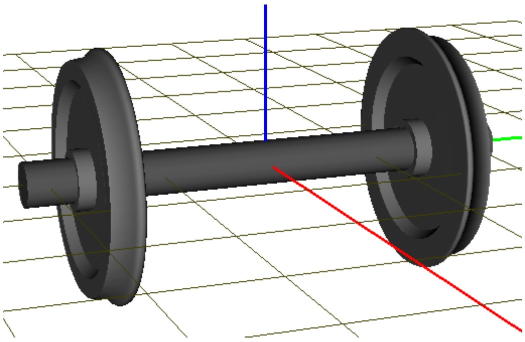


Figure 3.117. Image of a railway wheelset

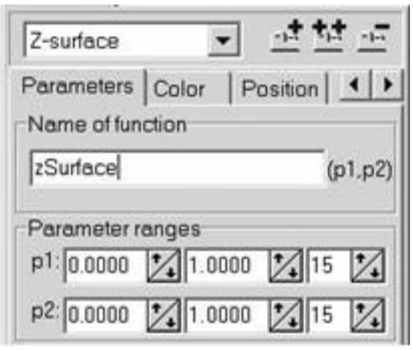
Remark. Computation of inertia parameters is not supported for a profiled element.

3.6.8.2.9. Z-surface

A GE of this kind is intended for programming graphical image of an arbitrary surface in the control file. The surface should be described as

$$z = f(x, y),$$

where *x* and *y* coordinates correspond to parameters *p1*, *p2*.



Z-surface

Name of function

zSurface (p1,p2)

Parameter ranges

p1: 0.0000 1.0000 15

p2: 0.0000 1.0000 15

Figure 3.118. Z-surface parameters

The GE description includes

- **name of function**;
- **parameter ranges** – limits of changing both *p1* and *p2* parameters and numbers of pointes to approximate the surface.

When generating equations of motion, a template for *zSurface* function is included in the control file (for example, as Pascal code):

```
function ZGraphicElementFunctions (
  _index, _isubs : integer;
  _p1, _p2 : real_ ) : real_;
begin
  _ := _PzAll[SubIndx[_isubs]];
  case _index of
    0 : begin
      { Function zSurface }
      Result := 0;
    end;
  end;
end;
```

For each function of type Z-surface introduced for representation of graphical images, an operator *Result := 0* is inserted, and the user should replace it with a proper calculation.

GE of type Z-surface is used to represent images with the help of complex implicit functional expressions, including time-dependent ones. For example, the Z-surface was used to get a traveling wave (Figure 3.119).

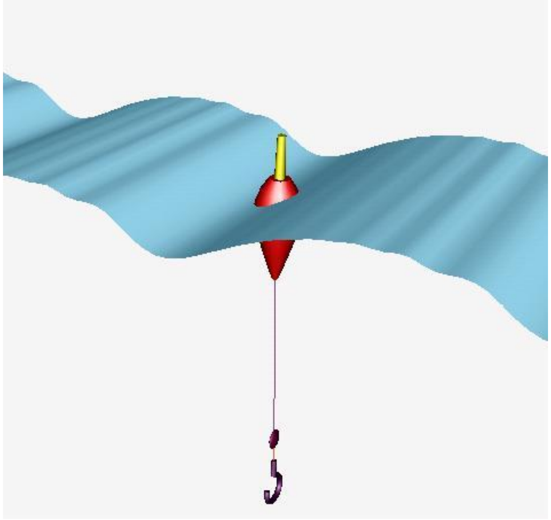


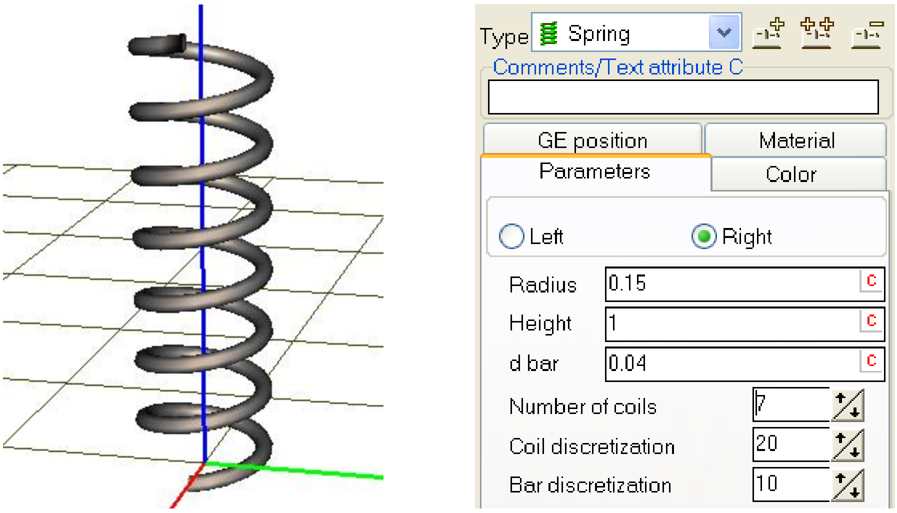
Figure 3.119. Z-surface

More detailed information concerning this GE is discussed in the manual for programming in UM environment.

Remark 1. Since the real description of the GE is located in the control file and is accessible in the UM Simulation program only, so in the input program the element is showed as a rectangle of sizes defined by change limits of parameters *p1*, *p2*.

Remark 2. Use of this element requires generation of equations of motion in the symbolic form, Sect. 3.9.

3.6.8.2.10. Spring



Type Spring

Comments/Text attribute C

○ Left ● Right

Radius	0.15
Height	1
d bar	0.04
Number of coils	7
Coil discretization	20
Bar discretization	10

Figure 3.120. Example of a spring

A **Spring** GE (Figure 3.120) has been developed for images of cylindrical springs, Figure 3.121. The list of parameters of the element is similar to that for the **Spiral** element, Sect. 3.6.8.2.4. Additional parameters are

- type of spring **Left/Right**,
- bar diameter,
- bar discretization.

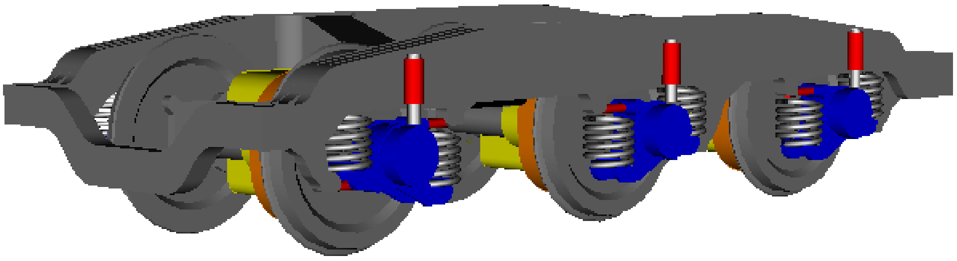
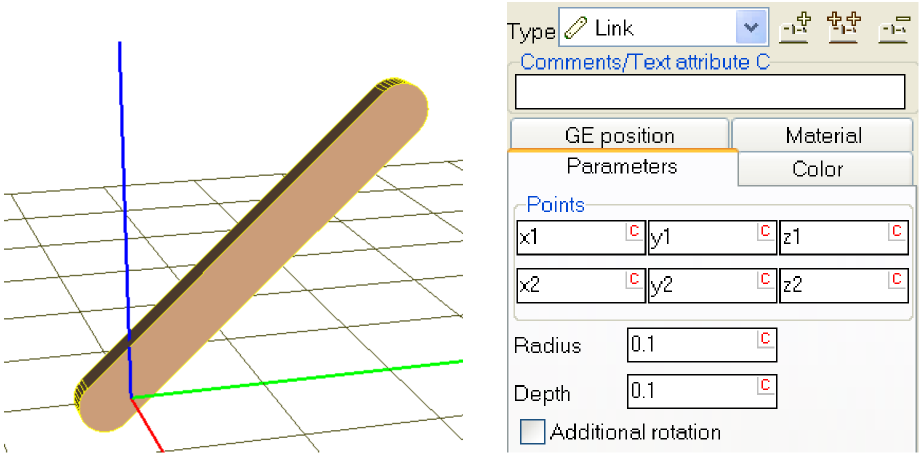


Figure 3.121. Use of GE ‘Spring’ in the model of a locomotive bogie

3.6.8.2.11. Link



Type: Link

Comments/Text attribute C

GE position | Material | Parameters | Color

Points

x1	y1	z1
x2	y2	z2

Radius 0.1

Depth 0.1

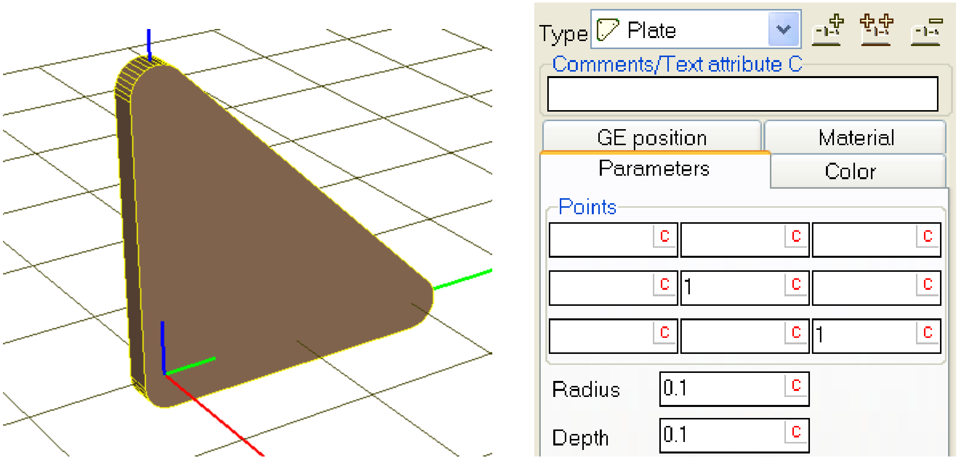
□ Additional rotation

Figure 3.122. Parameters of ‘Link’ element

The **Link** element is a rectangular plate with rounded corners, Figure 3.122. The position and sizes of the plate are the following:

- coordinates of centers of round parts in the **Points** group;
- **radius** of the rounding which is the half of the plate width;
- **depth** of the plate;
- the **Additional rotation** key rotates the plate on 90 degrees about its longitudinal axis.

3.6.8.2.12. Plate



Type: Plate

Comments/Text attribute C

GE position | Material | Parameters | Color

Points

	1	
		1

Radius 0.1

Depth 0.1

Figure 3.123. Plate parameters

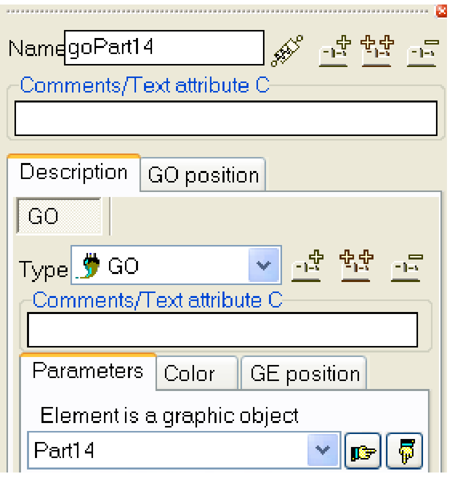
The **Plate** element is a triangle plate with rounded corners, Figure 3.123. The position and sizes of the plate are the following:

- coordinates of centers of round parts in the **Points** group;
- **radius** of rounding;
- **depth** of the plate;

3.6.8.2.13. GO as a graphic element

This type of graphical element is a reference to a previously crated GO in the list, or GE-GO. The element is recommended in the following cases:

- an image contains several equal parts,
- by merging several GO into one image,
- by import data from CAD programs; in particular, by merging parts into one body, Sect. 3.10.3.



Name goPart14

Type GO

Element is a graphic object

Part14

Figure 3.124. Example of a graphic element as a reference to GO

The reference to a GO is selected from the drop-down list, Figure 3.124.

The ☞ button is used to access the referenced GO.

The referenced GO can be directly included in the current GO by the ☟ button.

This type of element allows the user to create a many time repeating group of graphic elements as a separate GO and insert it several times into another GO. After that, the correction of included GO leads to modification of all the GE-GO.

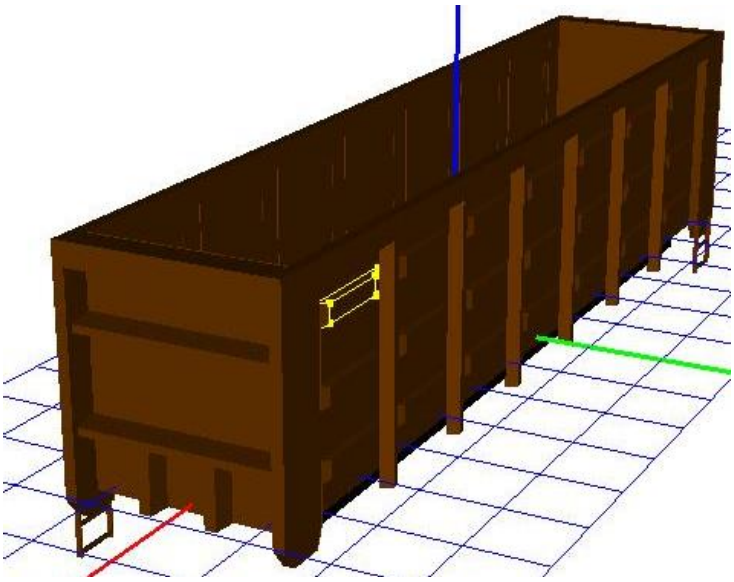
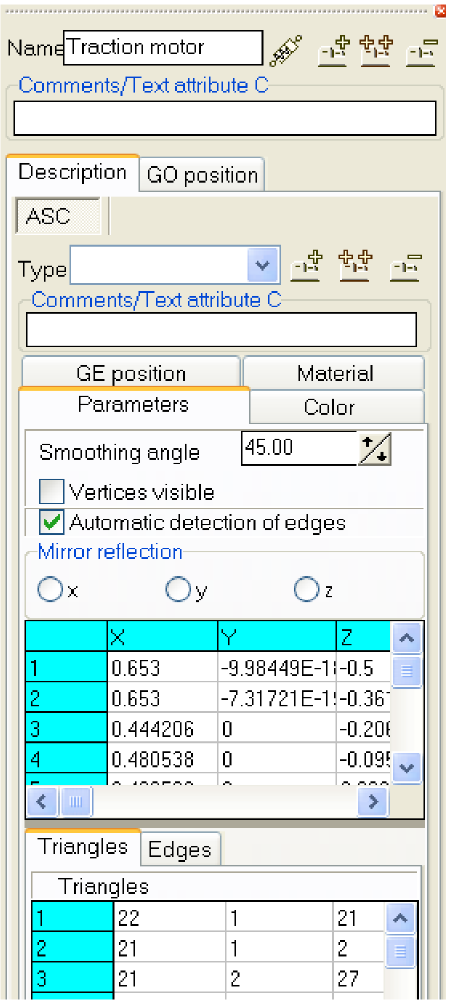


Figure 3.125. Graphic image of a fright couch body

For example, consider an image of a fright couch body. It includes a lot of repeating parts (stiffening ribs) one of which is selected in the picture (Figure 3.125). It should be created a GO corresponding to a separate rib and then this image is dozens of times included in the image of the body as GE-GO and positioned by a proper way. If the image of the rib must be modified (e.g. its height must be reduced), the only GO of the rib should be corrected, and all the ribs included in the image of the coach body are changed automatically.

3.6.8.2.14. Graphic element ASC imported from CAD program



Name Traction motor

Type

Smoothing angle 45.00

□ Vertices visible

☑ Automatic detection of edges

○ x ○ y ○ z

	X	Y	Z
1	0.653	-9.98449E-1	-0.5
2	0.653	-7.31721E-1	-0.36
3	0.444206	0	-0.20
4	0.480538	0	-0.09

Triangles

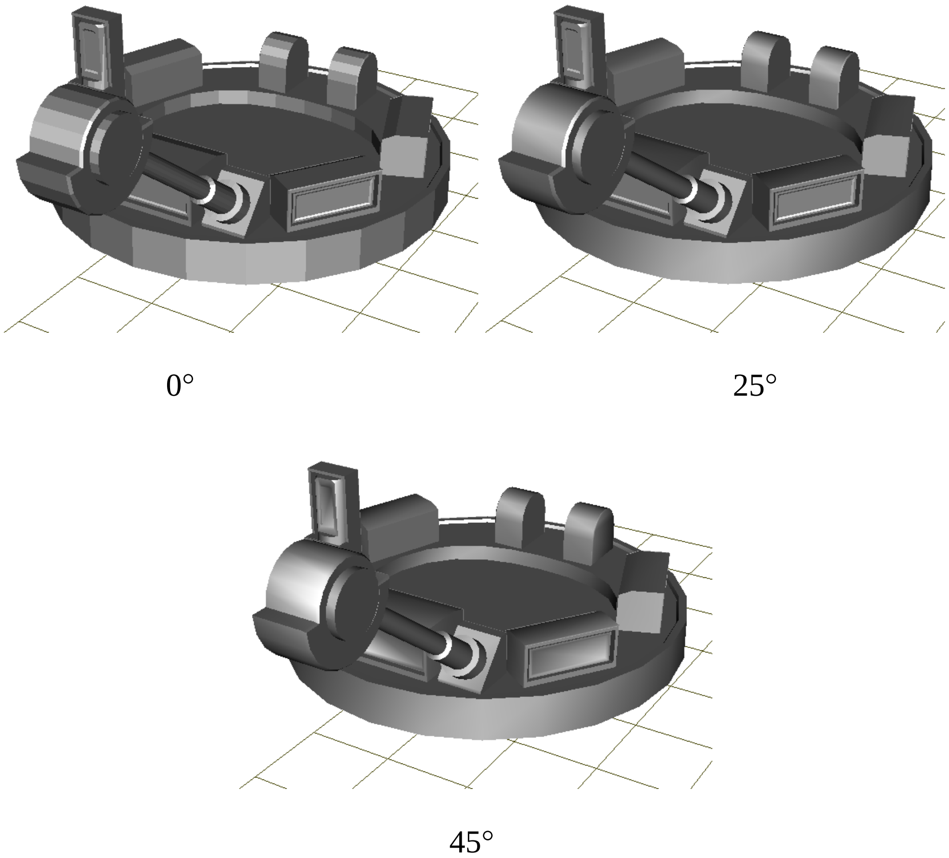
1	22	1	21
2	21	1	2
3	21	2	27

Figure 3.126. Example of an ASC element

Graphic elements of the **ASC** type correspond to images imported from CAD programs, Sect. 3.10. The element contains a set of vertices and a set of triangles specified by indices of vertices, Figure 3.126. Neither vertex coordinates nor triangles can be modified.

The user can change the following data and parameters.

- **Smoothing angle.** The angle is used for an automatic smoothing of the surface; a common side of two neighbor triangles is considered as an edge (and not smoothed!) if the angle between the triangles is greater than the given value; in particular, for zero value of the smoothing angle, no smoothing take place, Figure 3.127.



0°

25°

45°

Figure 3.127. Smoothing of surface for different values of angle parameter

- The **Vertices visible** key makes vertices visible/invisible in the animation window. If vertices are visible, a click by the mouse on the vertex image allows the user to find the vertex coordinates in the list.
- The **Automatic detection of edges** key calls a procedure for evaluation of edges, if the edges are not imported from CAD program, Figure 3.128.

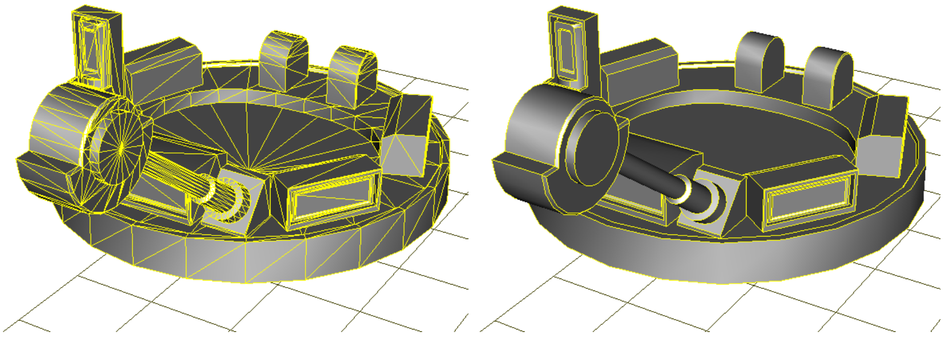
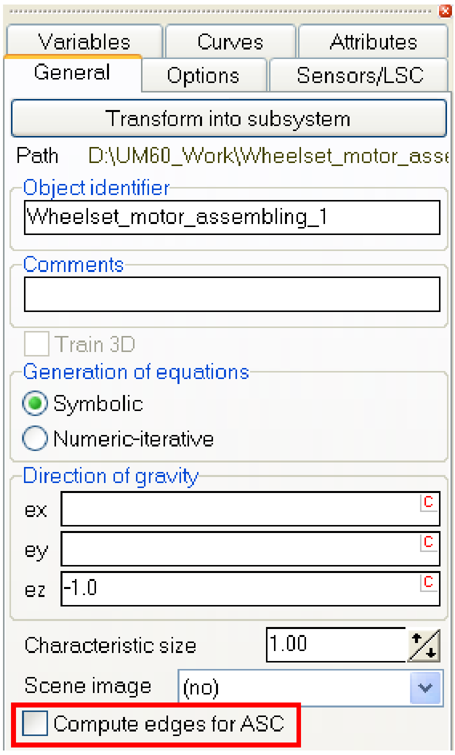


Figure 3.128. Edges before and after automatic computation

Remark. If a model includes a big number of ASC elements, edges for all of them can be detected by one operation. To do it, open the **Object** tab in the inspector and check the key Compute edges for ASC, Figure 3.129.



Path D:\UM60_Work\Wheelset_motor_asse

Object identifier
Wheelset_motor_assembling_1

Comments

□ Train 3D

Generation of equations
● Symbolic
○ Numeric-iterative

Direction of gravity
ex
ey
ez -1.0

Characteristic size 1.00

Scene image (no)

□ Compute edges for ASC

Figure 3.129. Key for automatic computation of edges for all of the ASC elements

3.6.8.3. GE colors

Colors of a GE are set by parameters on the **Color** tab, Figure 3.130.

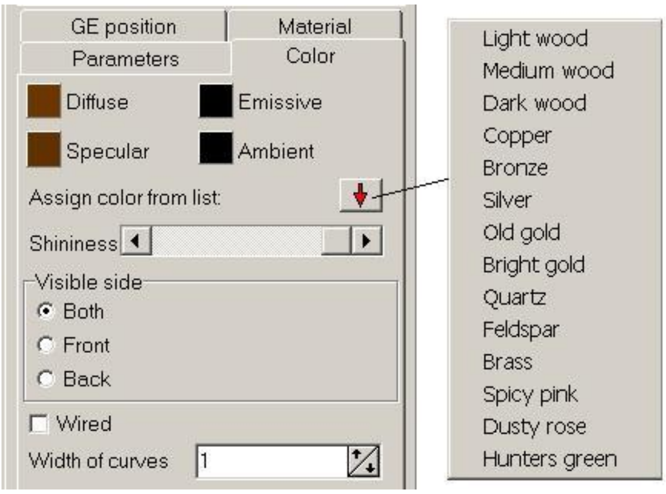


Figure 3.130. Colors of GE

There are several kinds of color parameters:

- **Diffuse** (color of material);
- **Specular** (color of the reflected light); black color means no reflection;
- **Emissive** (the object shines with this color); it is off if the color is black;
- **Ambient** (usually not used).

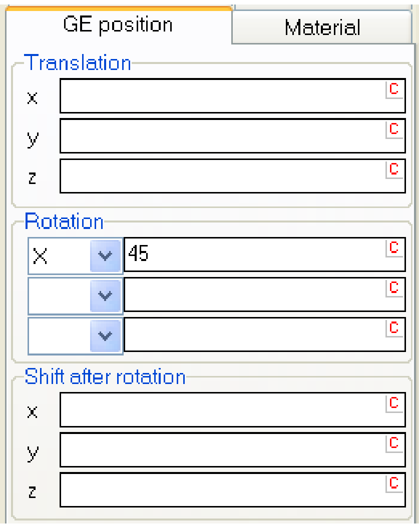
To choose each of the colors, click mouse button on the corresponding color rectangle.

Use the ↓ button to select one of the standard color sets.

The parameter *Shininess* defines size of the reflected light's spot.

Check the **Wired** option to set the element into the wire mode; simultaneously the **Width of curves** parameter set the line width.

3.6.8.4. Position and Orientation of GE

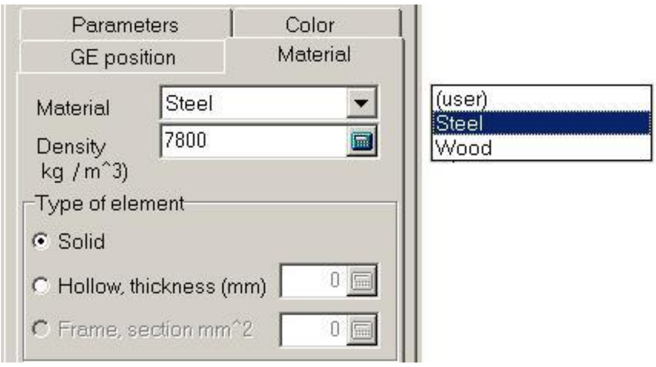


Rotation: X 45

Figure 3.131. Position and orientation of GE

Each GE is described in its own system of coordinates SC GE, which can be positioned in SC GO in an arbitrary way, see Sect. 3.6.4.

3.6.8.5. Inertia parameters of GE



Material: Steel

Density kg / m^3): 7800

Type of element

● Solid

○ Hollow, thickness (mm) 0

○ Frame, section mm^2 0

Figure 3.132. Inertia parameters of GE material

Inertia parameters of bodies (mass, tensor of inertia, coordinates of center of mass) can be computed automatically according to the body image. A necessary condition for that is filling the material properties for graphic elements (at least for one of them). On the **Material** tab (Figure 3.132) set

- **Density**
- **Type of graphic element** (solid, hollow) or frame (for wire element, Sect.3.6.8.3)
- A thickness and a section square should be set for a hollow and a wire GE

Warning. Automatic computing of inertia parameters may lead to wrong results if separate graphic elements interSect. The intersected volumes are taken into account several times. For example, the mass of the body is computed according to the formula

$$m = \sum m_i$$

where m_i is the mass of a separate GE calculated independently on possible intersections.

The computation of inertia parameters is not supported for a profiled element, Sect. 3.6.8.2.8.

3.6.8.6. Curve editor

The window of the curve editor is shown in Figure 3.133. See Sect. 3.5.4 for additional information about this tool.

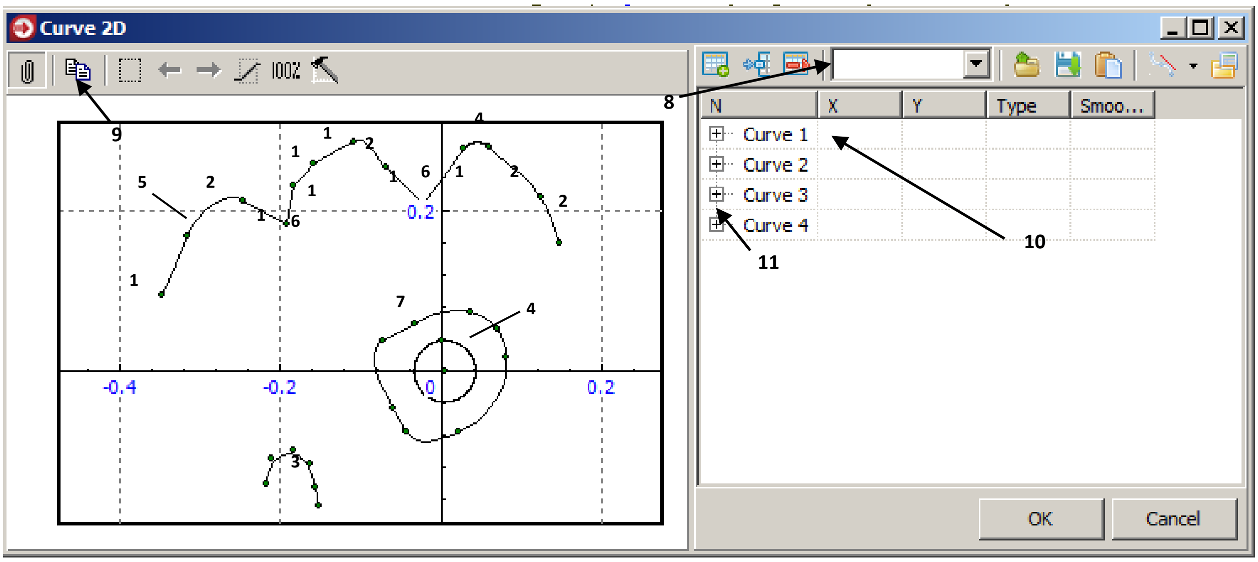


Figure 3.133. Curve editor

Main elements of the editor are marked in Figure 3.133 by indices:

- graphical primitives:

1 – straight lines;

2 – cubic splines;

3 – B-splines (special kind of splines);

4 – circle (arc);

- points (vertices):

5 – a point of smooth conjugation of primitives;

6 – a point of non-smooth (sharp) conjugation of primitives;

- continuous curves (continuums):

1-5-2-1-6-1-1-2-1-6-1-4-2-2, and also 3 – non-closed curves;

4, 7 – closed curves;

- other controls and possibilities of the editor:

8 – drop-down list of types of primitives;

9 – panel of control buttons;

10 – list of curves;

11 – open the list of point coordinates of the current (selected) curve.

To add points (vertices), double click by the left mouse button in the desired position.

To copy a curve:

- select it using mouse pointer;
- put it to the Windows clipboard;
- insert from the clipboard.

Exchange with the clipboard is performed using a text format. So, the user can type coordinates of vertices as two columns of numbers in any text editor, and insert them into the curve editor.

Pop-up menus are used for various actions with primitives and points of curves. It appears after clicking the right mouse button over the corresponding element.

For example, clicking over the free field of the curve editor results in appearing a pop-up menu shown in Figure 3.134.

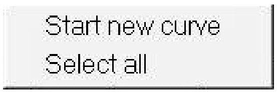


Figure 3.134. Curve editor main pop-up menu

The menu has two items:

- **Start new curve** – the next point being input will be the start point of the new curve (continuum);
- **Select all** – is used to select all curves in the editor.

Clicking right mouse button over any vertex results in appearing a pop-up menu for a point (Figure 3.135).

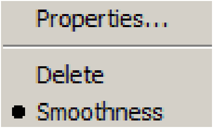


Figure 3.135. Pop-up menu for a point

Actions allowed for a selected point are:

- **Properties...** – opens a point property dialog box intended for changing the point coordinates;
- **Delete** – removes the selected point;
- **Smooth** – turns on/off the smooth conjugation of primitives.

A pop-up menu for a primitive appears after clicking any primitive (line, spline or circle); see Figure 3.136.

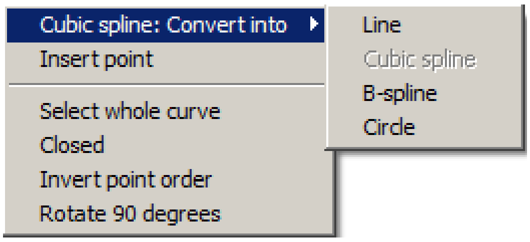


Figure 3.136. Pop-up menu for primitive

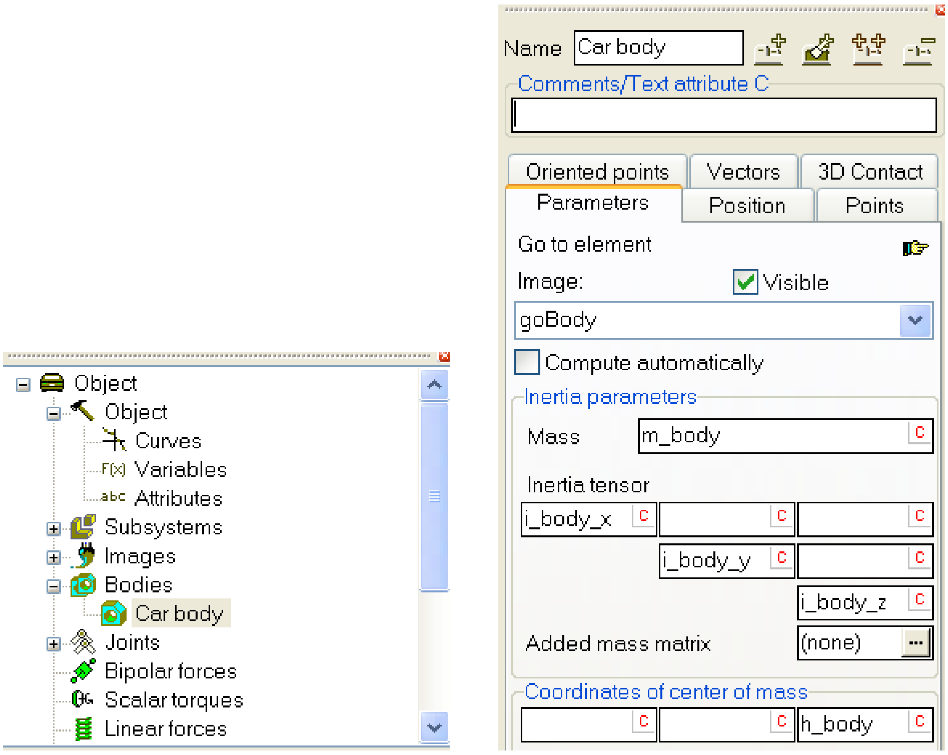
Actions allowed for a primitive are:

- **Convert into** – changes the type of the selected primitive;

- **Insert point** – a new point will have the coordinates defined by the current position of the mouse pointer (double clicking the left mouse button has the same effect);
- **Delete** – removes the selected primitive;
- **Properties...** – calls a primitive property dialog box;
- **Select whole curve** – selects the curve (continuum) containing the selected primitive;
- **Closed** – a tag for closing the curve containing the selected primitive;
- **Reverse order** – for inverting the ordering of points;
- **Rotate 90** – turns the current continuum through 90 degrees counterclockwise.

3.6.9. Describing rigid bodies

The *Bodies* item of the object element tree (Figure 3.137 left, Sect. 3.5.1.1) is used to access the tool of creation and modification of the list of bodies and their parameters. Alternative ways are the Ctrl+Alt+B hot key or clicking by the mouse on the corresponding body image in the whole object mode of the animation window (Sect.3.5.1.2.2). The right picture in Figure 3.137 shows the inspector with parameters of a body.



Name: Car body

Image: goBody ☑ Visible

□ Compute automatically

Mass: m_body

Inertia tensor: i_body_x, i_body_y, i_body_z

Added mass matrix: (none)

Coordinates of center of mass: h_body

Figure 3.137. *Bodies* item of the object element tree and body parameters

3.6.9.1. Image and visualization of a body. Body-fixed SC

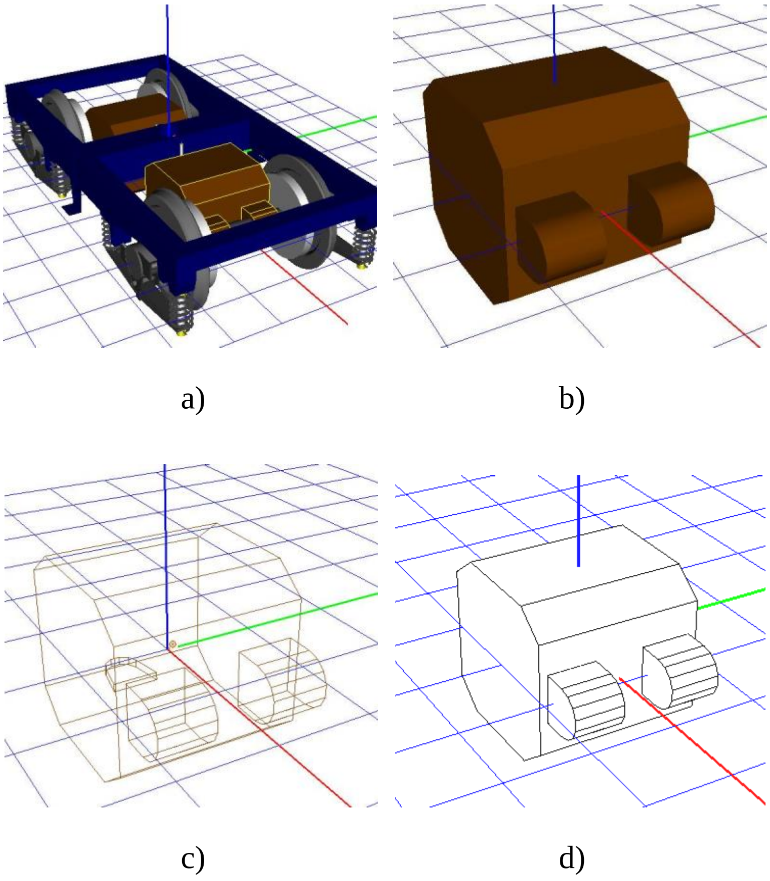
3.6.9.1.1. Assignment of an image to body

The *Image* pull-down list contains the list of available graphical objects, Sect.3.6.8. Click the ☞ button to access the assigned GO parameters or to create a new GO, which will be assigned to the body.

The SC of the assigned GO coincides with the body-fixed SC. That is why the image moves when the body changes its position and orientation.

3.6.9.1.2. Modes of animation window and body image

Use the single element mode of the animation window (Sect.3.5.1.2.2) to see the body-fixed SC and the body image.



a) b)

c) d)

Figure 3.138. Visualization of an active body in different modes of the animation window

Figure 3.138 shows visualization of a *motor* body in different modes of the animation window:

1. *Full object mode*. The button must be in the 'down' state. The active body is drawn as selected, axes of the SC0 are drown.
2. *Single element mode*. The button must be in the 'up' state. The window contains the body image in the body-fixed SC which axes are drawn. Modes of drawing

- shaded mode, button ,
- wired mode, button ,
- contour mode, button ,
- shaded mode with edges, button .

Important remark. A body is drawn in the full object mode of the animation window, if there exists a path from the body to the Base0 through the joints (Chapter 2, Sect. Compendency of systems and the definition of a joint). For example, if body2 is connected to body1 and body1 is connected to Base0 by means of two rotational joints, both body1 and body2 are visible in the full object mode. If the joint between body1 and Base0 is removed, both bodies disappear. If the joint between body2 and body1 is removed, body1 disappears.

3.6.9.1.3. Modes of body image

A body can be made temporarily wires or invisible independently on the state of the animation window.

3.6.9.1.3.1. Invisibility of body

In particular, the invisibility mode of a body is used by merging a number of parts into a body for models imported from CAD, see Chapter 9. The mode can be set by the following ways:

- the Visible key in the inspector, Figure 3.137; the same key make the body visible;
- the **Make invisible "name of body"** command of the pop-up menu of the animation window, Figure 3.139; to call the menu, move the mouse cursor to the body image and click the right mouse button; this method is available only if the button for visual operations is in the 'down' state;

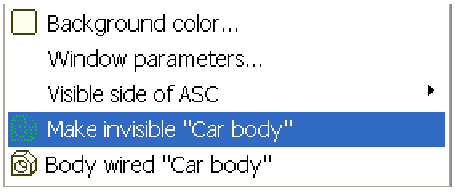
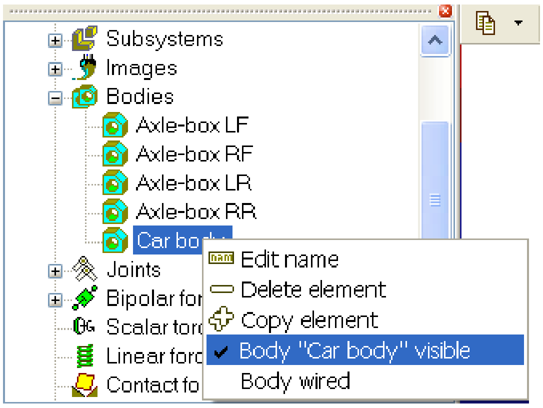


Figure 3.139. Fragment of pop-up menu of animation window

- the **Body "Name of body" visible** command of the pop-up menu of the tree of elements, Figure 3.140; to call the menu, move the mouse cursor to the name of the body in the tree and click the right mouse button; the same command is used to make the body visible;

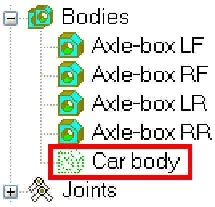


Subsystems
Images
Bodies
Axle-box LF
Axle-box RF
Axle-box LR
Axle-box RR
Car body
Joints
Bipolar for
Scalar forc
Linear forc
Contact fo

Edit name
Delete element
Copy element
✓ Body "Car body" visible
Body wired

Figure 3.140. Pop-up menu of tree of elements

The invisible body is marked in the element tree by the icon , Figure 3.141.



Bodies
Axle-box LF
Axle-box RF
Axle-box LR
Axle-box RR
Car body
Joints

Figure 3.141. Icon of invisible body

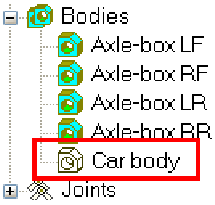
Remark. The invisibility mode does not affect the image of the body in the simulation module, where an alternative method for changing visibility is used.

3.6.9.1.3.2. Wired body mode

The wired mode can be set by the following ways:

- the **Body wired "name of body"** command of the pop-up menu of the animation window, Figure 3.139; to call the menu, move the mouse cursor to the body image and click the right mouse button; this method is available only if the button for visual operations is in the 'down' state;
- the **Body wired** command of the pop-up menu of the tree of elements, Figure 3.140; to call the menu, move the mouse cursor to the name of the body in the tree and click the right mouse button; the same command is used to cancel the mode;

The invisible body is marked in the element tree by the icon , Figure 3.142.



Bodies
Axle-box LF
Axle-box RF
Axle-box LR
Axle-box RR
Car body
Joints

Figure 3.142. Icon of wired body in element tree

3.6.9.1.4. Reassignment of body image in simulation program

The body image can be reassigned in the simulation module, Figure 3.143:

- move cursor to the body image,
- open the pop-up menu by the right mouse button,
- select the **Assign GO [Name of body] | [Name of GO]** command, Figure 3.144.

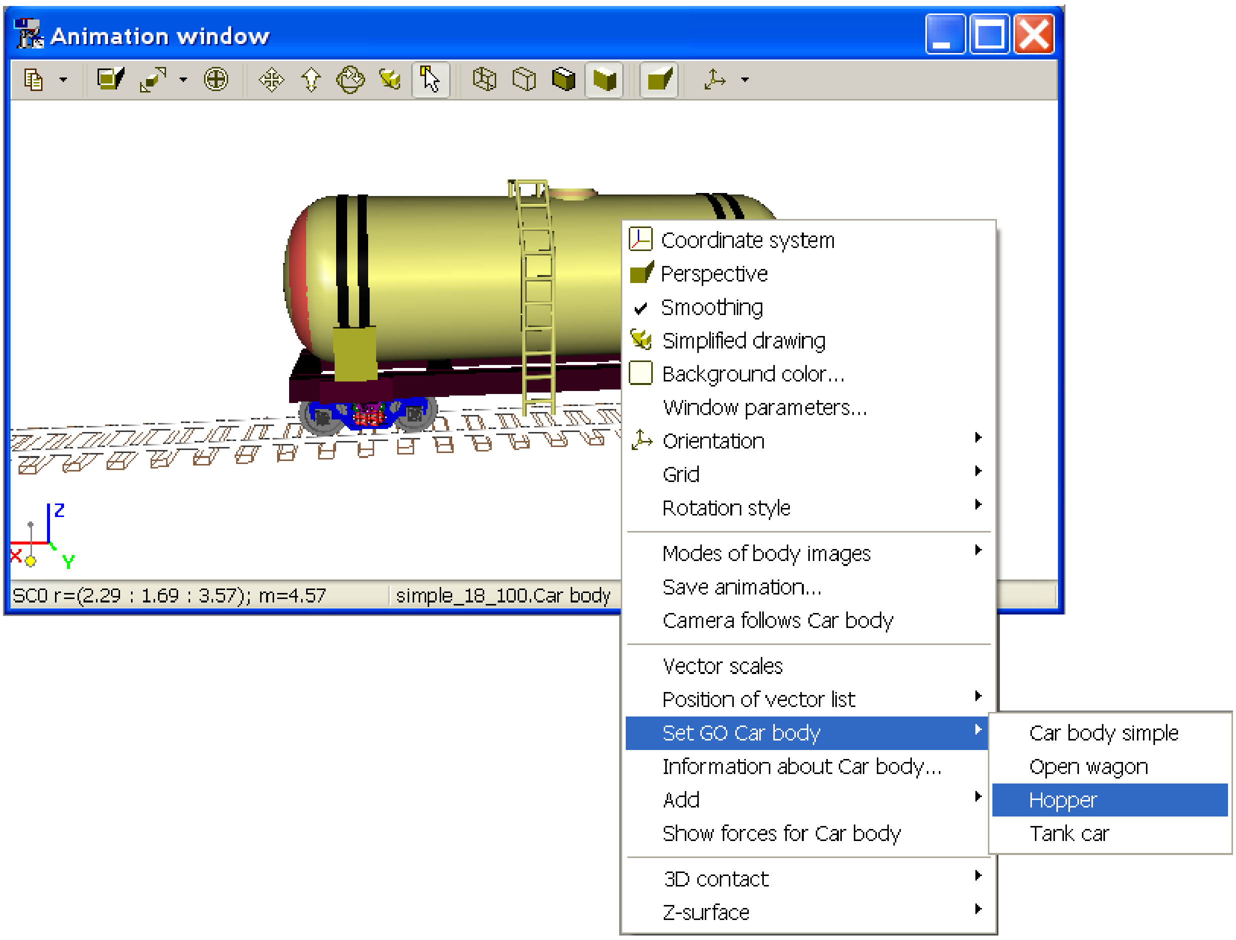


Figure 3.143. Reassignment of graphic object to the body ‘Car body’

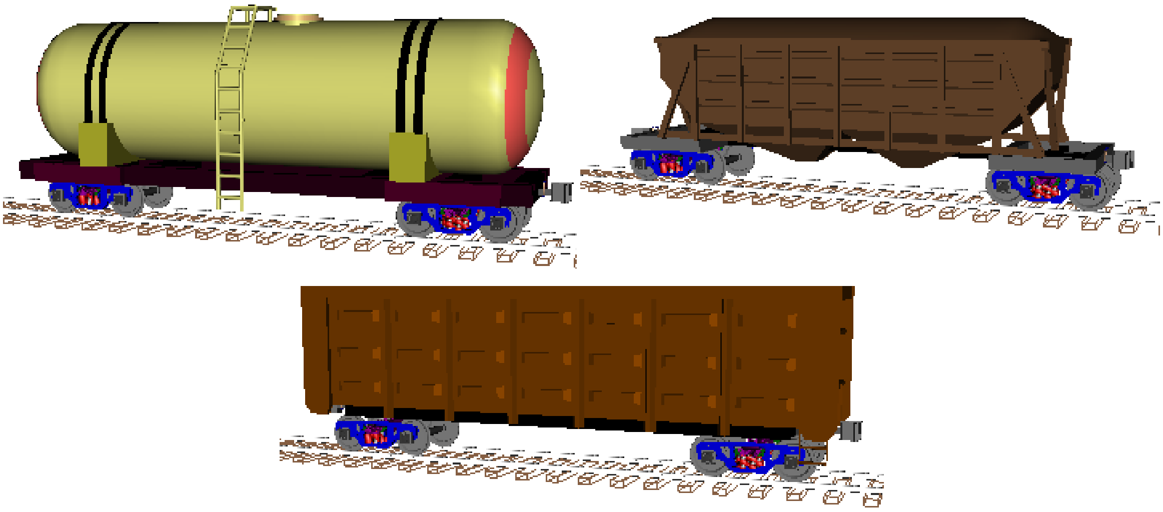


Figure 3.144. Model of a freight car with reassigned imaged of the car body

Reassignment of the images is stored in the configuration file of the model *.icf.

3.6.9.2. Inertia parameters

Inertia parameters of a body are:

- mass;
- symmetric inertia tensor;
- coordinates of center of mass in the body-fixed SC;
- symmetric added mass matrix.

The icon ⊗ marks the position of the center of mass in the animation window.

Important **remark and warning**. Moments of inertia and elements of the added mass matrix should be calculated in SC, which origin coincides with the center of mass and axes are parallel to those of the body-fixed SC.

There exist two modes for setting the inertia parameters except the added masses. Use the **Automatic calculation** key to select the necessary one, Figure 3.137.

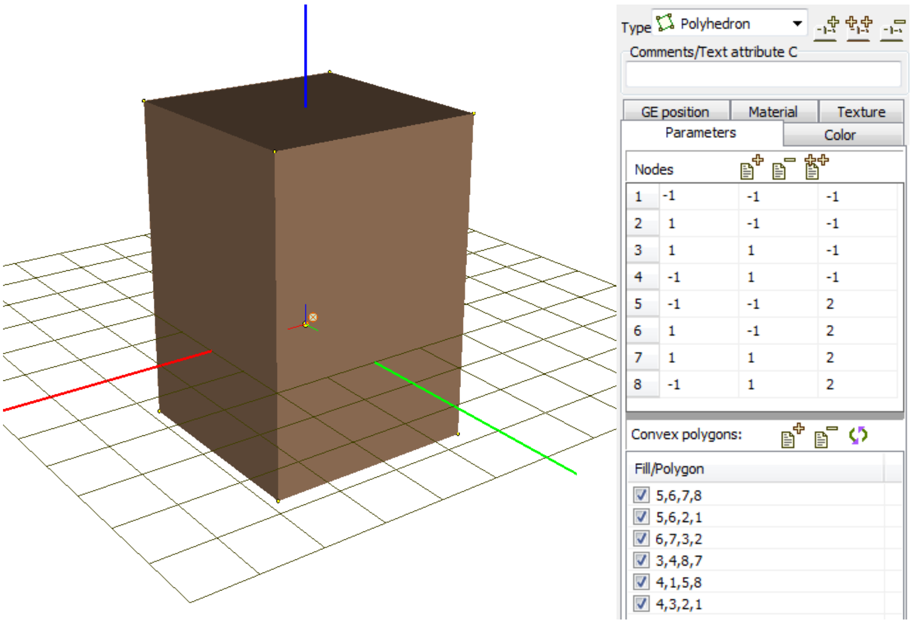
User’s defined inertia parameters

All parameter edit boxes are enabled in this mode. The parameters might be *constant symbolic expressions* (Sect. 3.5.2.4.5).

Automatic calculation of inertia parameters

The parameters are calculated automatically according to the body image (Sect. 3.6.8.5). Parameter edit boxes are not available for the user.

Note. In this mode inertia parameters are calculated automatically for nearly all graphical objects; however, the problems may occur when using polyhedrons (Sect. 3.6.8.2.1). For correct calculation, it is required that, when defining faces of the polyhedrons, the order of vertices were set counterclockwise if seen from outer side of the object. Otherwise, the automatically computed inertia parameters might be incorrect. For example, in definition of a user-defined parallelepiped shown in Figure 3.145 there is wrong set of vertices for the second (vertices 5,6,2,1) and the third (vertices 6,7,3,2) face since this order of vertices corresponds to clockwise order. As a consequence, the center of mass is calculated incorrectly, see the figure. If the user changes the order of vertices to 1,2,6,5 and 2,3,7,6 respectively, the center of mass (and other inertia parameters) are computed correctly, see Figure 3.146.



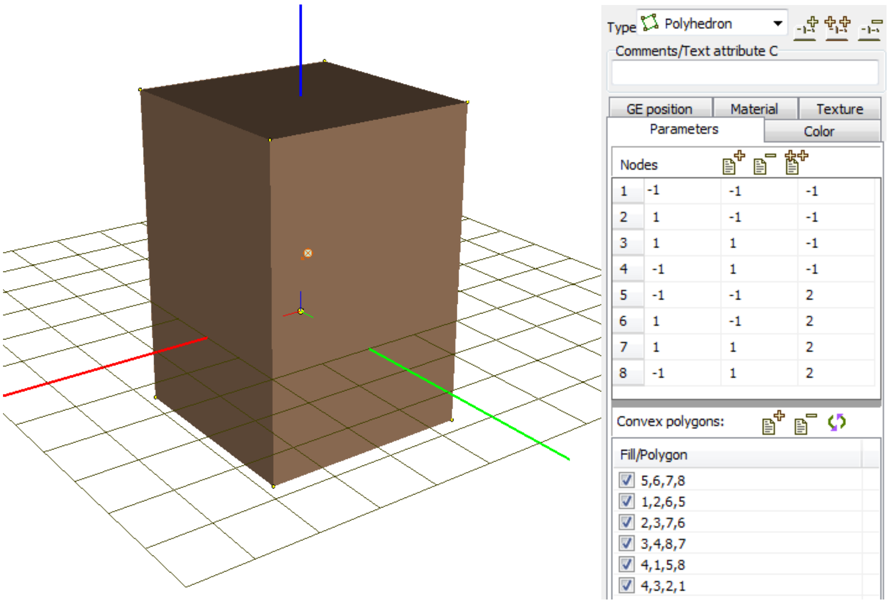
Nodes			
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	2
6	1	-1	2
7	1	1	2
8	-1	1	2

Convex polygons:

Fill/Polygon

☑ 5,6,7,8
☑ 5,6,2,1
☑ 6,7,3,2
☑ 3,4,8,7
☑ 4,1,5,8
☑ 4,3,2,1

Figure 3.145. Incorrect enumeration of vertices of a polyhedron when automatic computing inertia parameters



Nodes			
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	2
6	1	-1	2
7	1	1	2
8	-1	1	2

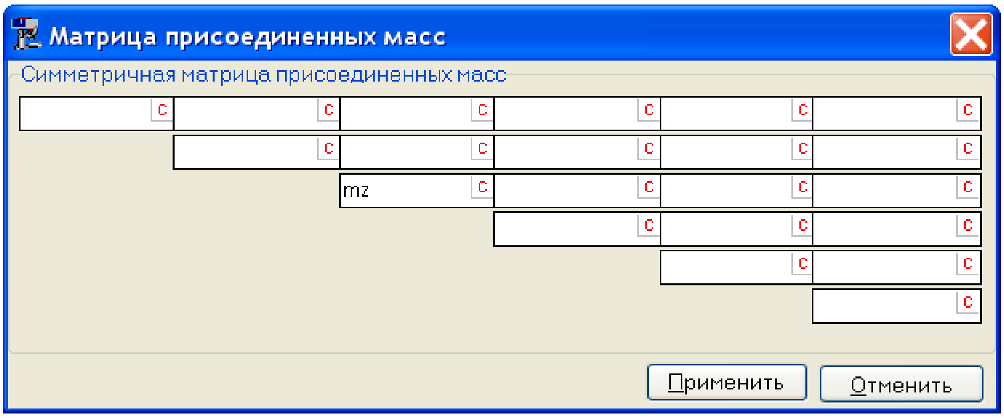
Convex polygons:

Fill/Polygon

☑ 5,6,7,8
☑ 1,2,6,5
☑ 2,3,7,6
☑ 3,4,8,7
☑ 4,1,5,8
☑ 4,3,2,1

Figure 3.146. Correct enumeration of vertices of a polyhedron when automatic computing inertia parameters

Added mass matrix


Матрица присоединенных масс
Симметричная матрица присоединенных масс
Применить
Отменить

		mz			

Figure 3.147. Window for elements of added mass matrix

The added mass matrix mainly used for simulation of bodies moving in a fluid[1] as well as in the case of railway wheelsets. Let M be the 6x6 inertia matrix of a rigid body specified in the body-fixed SC

$$M = \begin{pmatrix} mI_3 & 0 \\ 0 & J \end{pmatrix},$$

where m is the mass of the body; J is the matrix of the inertia tensor relative to the center of mass, I_3 is the 3x3 identity matrix. If we denote A the 6x6 added mass matrix, which element are set in Figure 3.147, the equations of motion of the body look like

$$(M + A)W + k = G,$$

$$W = \begin{pmatrix} a \\ \varepsilon \end{pmatrix}, \; k = \begin{pmatrix} 0 \\ \tilde{\omega} J \omega \end{pmatrix}, \; G = \begin{pmatrix} F \\ T \end{pmatrix},$$

Here a, ε, ω are the acceleration of the body center of mass, the angular acceleration and the angular velocity of the body; F,T are the principal vector and torque of the applied and reaction forces reduced to the center of mass. All vectors in this equation are resolved in the body-fixed SC.

Remarks.

- Automatic calculation of inertia parameters is enabled after assigning a GO to the body. The corresponding GO must include at least one GE with the specified material which density is not zero.
- Inertia parameters are recalculated every time when the GO is modified, in particular when identifiers parameterizing the GO are changed.

[1] Newman, John Nicholas (1977). *Marine hydrodynamics*. Cambridge, Massachusetts: MIT Press.

3.6.9.3. Internal (hidden) body joint

Position and motion of a body relative to SC0 can be defined by an **internal or hidden joint**. Type of joint is “6 d.o.f”, Sect. 3.6.11.6. The following tools are available for creating a body with an internal joint

- the button in the inspector, Figure 3.137,
- the **Add body with internal joint** command of the popup menu of the element tree, Figure 3.148.

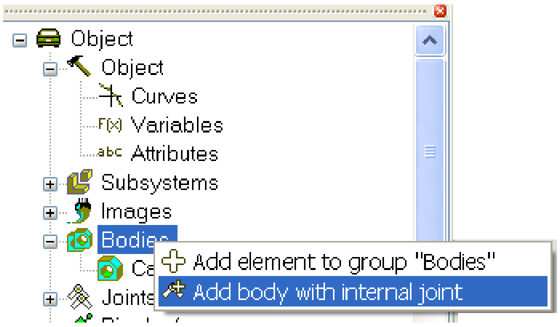


Figure 3.148. Adding a body with internal joint by popup menu command in element tree

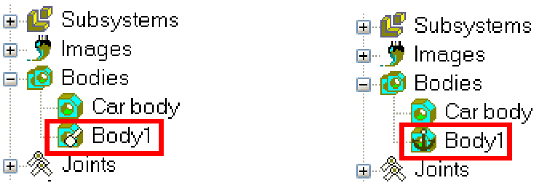
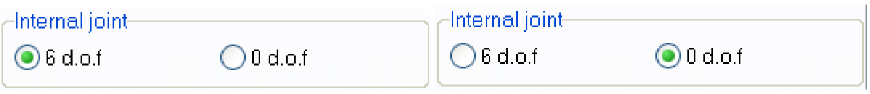


Figure 3.149. Icon of body with internal joint having six (left) and zero (right) degrees of freedom

The joint is referred as an internal one because it is not visually presented in the list of joints of the model. Consider some features of the internal joint.

1. An internal joint can have either six or zero degrees of freedom, Figure 3.149. In the first case, the joint specifies the position of the free body relative to the SC0. It the second case, the joint fixes the body to the SC0 and the body-fixed SC coincides with SC0. The number of degrees of freedom is set by the user, Figure 3.150.



Internal joint
● 6 d.o.f ○ 0 d.o.f

Internal joint
○ 6 d.o.f ● 0 d.o.f

Figure 3.150. Number of degrees of freedom of an internal joint

Example of using internal joint with six degrees of freedom: fictitious body, Sect. 3.6.3.3.3.
Example of using internal joint with zero degrees of freedom: body Ground, Sect. 3.6.9.7.

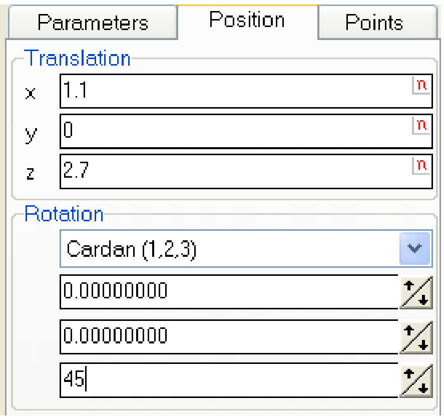
2. In case of an internal joint with six d.o.f., three Cartesian coordinates set location of the origin of the body-fixed SC relative to SC0, and three orientation angles specify the orienta-

tion of the body-fixed SC. The sequence of the rotation angles is (1,2,3), Sect. 3.6.11.6. SC of the body coincides with SC0 for zero values of coordinates.

3. A large weight is assigned to the internal joint, Sect. 3.6.11.2. If the user connects this body with SC0 by a usual joint either directly or through a chain of bodies, the internal joint is cut and obtains the status of the *deleted* joint. It is important to know that **cut joints with six degrees of freedom do not affect the motion process and ignored**, Sect. 3.6.11.4.2. If the joint connecting the body with the SC0 is removed, the internal joint is restored.

4. Values of coordinates of internal joint with six d.o.f. can be changed on the **Position** tab of the inspector, Sect. 3.6.9.4, as well as on the **Coordinates** tab, Sect. 3.5.2.3.4.

3.6.9.4. Position tab



Parameters | Position | Points

Translation

x: 1.1

y: 0

z: 2.7

Rotation

Cardan (1,2,3)

0.00000000

0.00000000

45

Figure 3.151. Position of body relative to SC0

The **Position** tab reflects the current position of the body-fixed SC relative to the SC0, Figure 3.151.

Values of coordinates can be changed on this tab if the body has a non-deleted internal joint with six d.o.f., Sect. 3.6.9.3. Otherwise, the coordinates are given for information only and cannot be changed here.

Remark. To see the actual position of the body relative to SC0, set the full object mode in the animation window by the button on the tool panel, Sect. 3.5.1.2.2.

3.6.9.5. Rapid access to elements connected with body

Use the ☞ button (**Go to element**, Figure 3.152) for rapid access to elements connected with the current body:

- body image,
- joints,
- force elements,
- contact manifold.

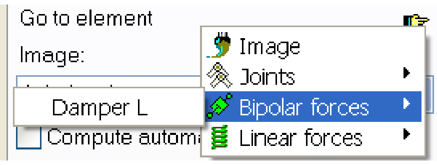


Figure 3.152. List of elements connected with the body

3.6.9.6. Connection points

Connection points are assigned to the body. They are used by

- operations with visual components;
- describing joints and force elements (visual assignment of bodies, joint and attachments points);
- assignment of bodies as well as the corresponding joint and attachment points to external joint and force elements (for subsystem technique only).

There exist three types of connection points:

- general
- oriented (local body-fixed SC)
- vectors

The following parameters should be set for each of the points

- coordinates in the body-fixed SC (coordinates are parameterized);
- comments simplifying operations with points (auxiliary parameter);
- orientation of point-fixed SC relative to the body-fixed SC (for oriented point); the orientation is set by three rotations.

An icon □ marks a connection point in the animation window (Figure 3.153). A point-fixed SC are additionally drawn for oriented points, and a red line section is drawn for a vector along its direction.

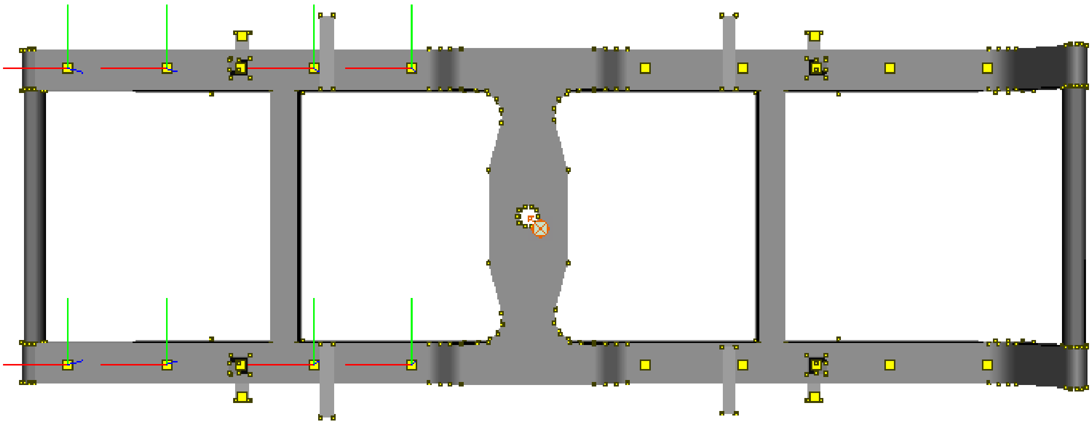
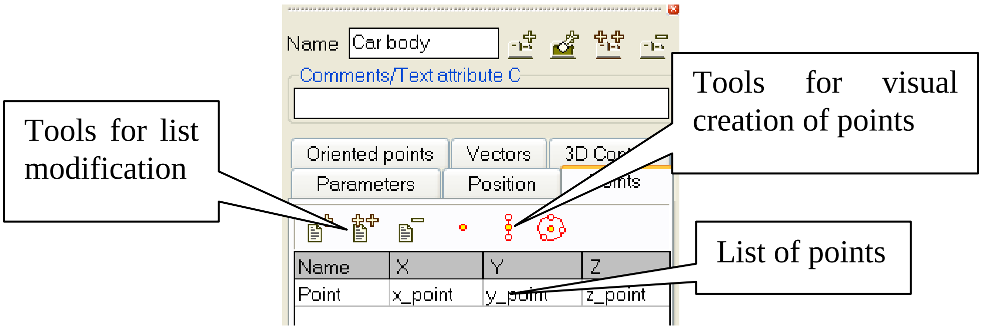


Figure 3.153. Frame with general and oriented connection points

Note. Connection points are used to assign their positions and/or orientation to object elements (joints and forces). At the same time the connection between the element and the points is not kept. This means that if the user changes the point parameters after its usage for description of an element, the corresponding changes are not transferred to the element automatically.

3.6.9.6.1. Adding general connection points



Name Car body

Comments/Text attribute C

Name	X	Y	Z
Point	x_point	y_point	z_point

Figure 3.154. List of general connection points

The **Points** tab is used to create or modify the list of points, Figure 3.154.

To add a point, the user should set its coordinates in the body-fixed SC. Here are instructions how to do this.

1. Use the (add) or the (copy) button. Set point coordinates as constant symbolic expressions.
2. Click the button, select by the mouse a point on the body image, and click the mouse button again. The selected point is added automatically.

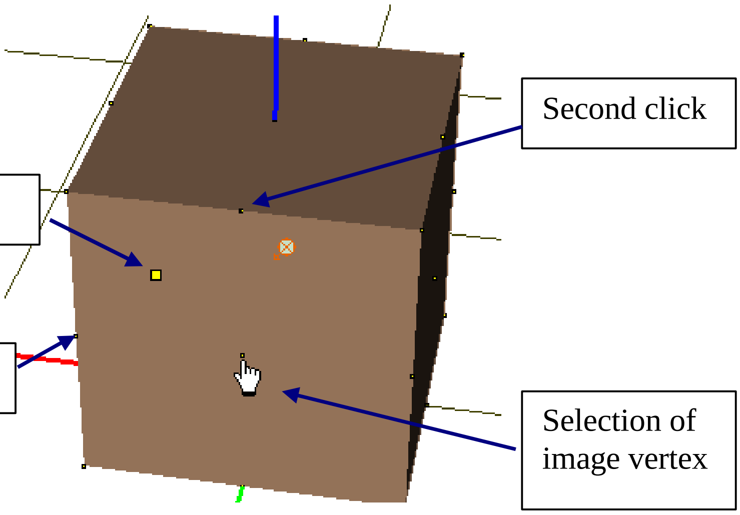


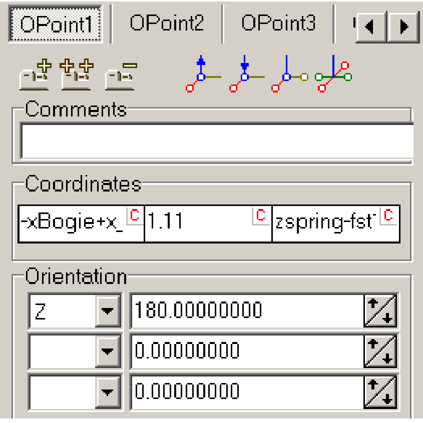
Figure 3.155. Adding a point as a middle point of a section

3. The button allows the user to get a point as a middle point of a section selected by the mouse, Figure 3.155.
4. The button adds a point which is the center of a circle passing through three points selected by the mouse.

Use vertices on the body image for exact positioning of points. Near the vertex the mouse cursor is changed to . After visual adding, the point can be corrected or removed by the button.

3.6.9.6.2. Adding oriented connection points

Use the **Oriented points** tab to add oriented connection points (a local system of coordinates, LSC), Figure 3.156. Its description includes coordinates of the points and the orientation of axes of the LSC in the body-fixed SC. The orientation is set as a sequence of three successive elementary rotations.



Comments

Coordinates

-xBogie+x_ | 1.11 | zspring-fst

Orientation

Z 180.00000000

0.00000000

0.00000000

Figure 3.156. List of oriented points

Here five methods for adding oriented points (LSC) are described.

1. Use the (add) or the (copy) button. Set point coordinates as constant symbolic expressions. Set rotation axes and angles in *degrees* to define the orientation.

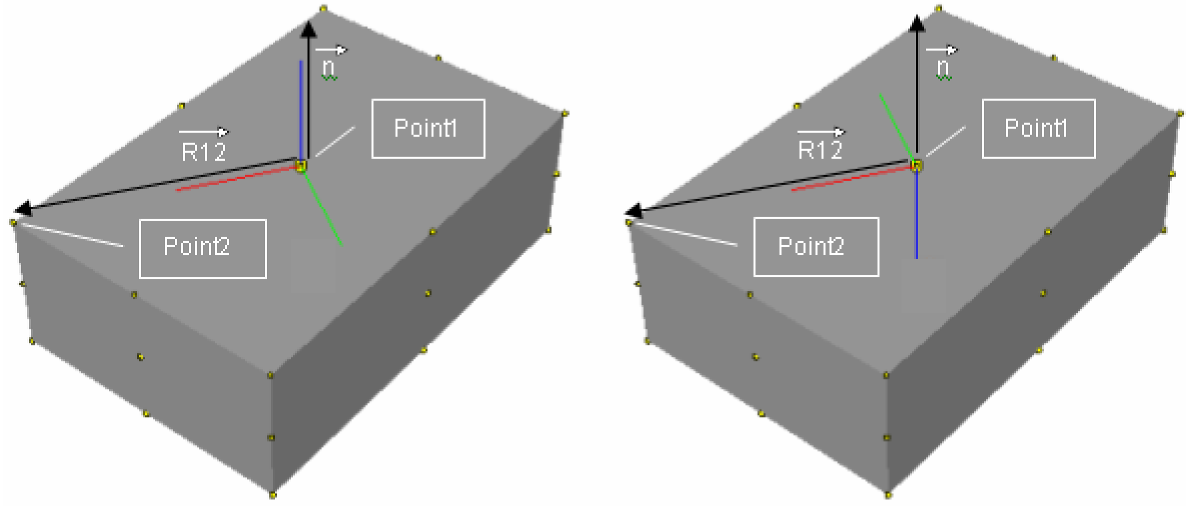
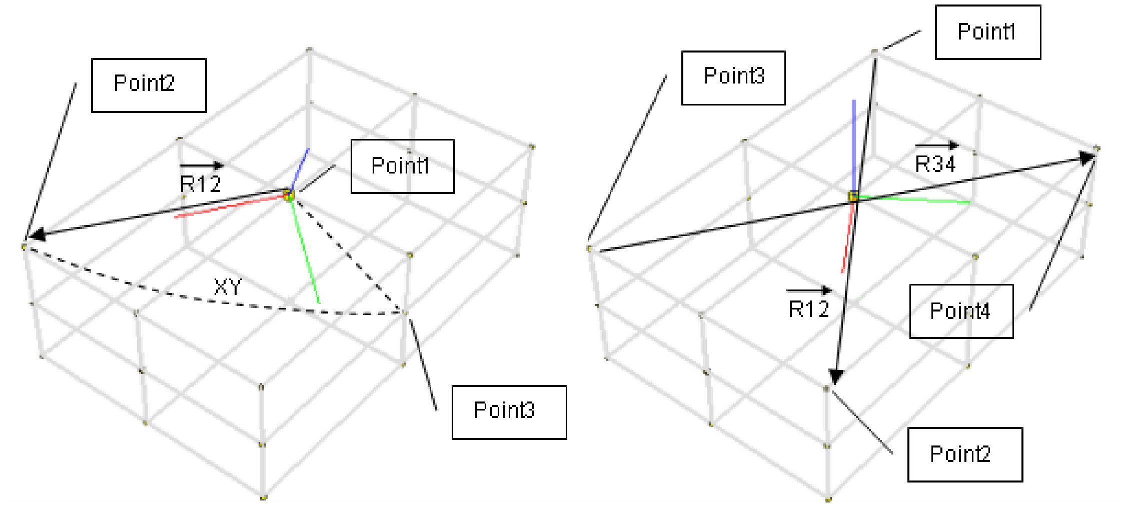


Figure 3.157. Adding a LSC by a normal and a point

2. Visual adding by a normal and a point.
   - Click the button.
   - Select a vector or a point on a plane for the origin and the Z-axis of the LSC.
   - Select a point in the XZ-plane.
3. Visual adding by an opposite normal and a point.

- Click the button.
- Select a vector or a point on a plane for the origin and the Z-axis of the LSC. The Z-axis is directed opposite to the vector or inside the plane surface.
- Select a point on the XZ-plane.



Visual adding a LSC by tree and by four points

4. Visual adding by three points.

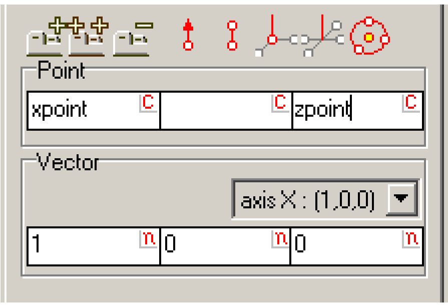
- Click the button.
- Select a point for the origin of the LSC.
- Select a point on the X-axis.
- Select a point in the XY-plane.

5. Visual adding by four points.

- Click the button.
- Select two points. The middle point of the section is the origin of the LSC, and the vector $\mathbf{r}_{12}$ connecting the points set the X-axis.
- Select the 3rd and the 4th points. The cross product $\mathbf{r}_{12} \times \mathbf{r}_{34}$ sets the Z-axis.

3.6.9.6.3. Adding vectors

Use the **Vectors** tab to add a vector. Its description includes coordinates of the points and components defining the direction of the unit vector in the body-fixed SC. The vectors are used for visual correction and adding joints and some special force elements.



Point

xpoint | | zpoint

Vector

axis X : (1,0,0)

1 | 0 | 0

Figure 3.158. List of vectors

Here five methods for adding vectors are described.

1. Use the (add) or the (copy) button. Set point coordinates as constant symbolic expressions. Set numeric values for the vector components.
2. Visual adding by a normal and a point.
   - Click the button.
   - Select a point on a plane for the vector, which is the external normal to the plane.
3. Visual adding by two points.
   - Click the button.
   - Select a point for the vector point.
   - Select the second point for the vector direction.
4. Visual adding by three points.
   - Click the button.
   - Repeat actions for setting the LSC by three points from the previous section. The vector corresponds to the Z-axis.
5. Visual adding by four points.
   - Click the button.
   - Repeat actions for setting the LSC by four points from the previous section. The vector corresponds to the Z-axis.
6. Visual adding as a normal to the center of a circle passing through 3 points.
   - Click the button.
   - Select three points lying on a circle. The resulting vector begins at the circle center. The direction of the vector is perpendicular to the circle plane according to the right-hand screw rule according to sequence of selected points.

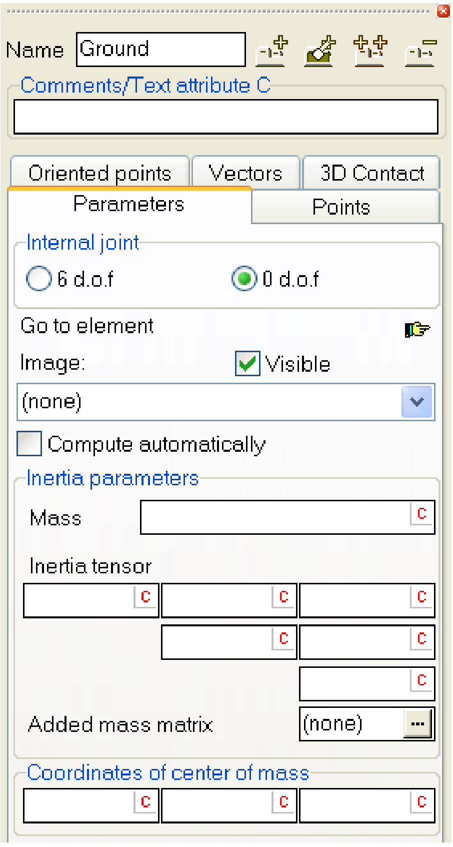
3.6.9.7. Body «Ground»

Ground is a body, which could be automatically added to the model by its creating, if the corresponding option is checked, Sect. 3.3.1.

The ground body has zero inertia parameters. It is rigidly connected by an internal joint to SC0, i.e. the internal joint has zero degrees of freedom, Sect. 3.6.9.3. Body-fixed SC coincides with SC0. In fact, according to the listed properties of the Ground, this body is equivalent to the SC0, and all bodies connected to the Ground by joints and force elements are connected to SC0.

As opposed to SC0, connection points can be assigned to the Ground (Sect. 3.6.9.6) and used by visual assignment of the points to force elements and joints.

Another reason for introduction of the Ground consists in the use of this body by conversion MSC.ADAMS models, Sect. 3.11.


Name Ground

Internal joint
○ 6 d.o.f ● 0 d.o.f

Image: ☑ Visible
(none)

□ Compute automatically

Added mass matrix (none)

Figure 3.159. Parameters of body «Ground»

3.6.9.8. 3D Contact

There is a possibility to assign a contact manifold described as a graphical object (see Sect. 3.6.8) to a rigid body. All bodies that have such a contact manifold will interact between each other during simulation dynamics of a mechanical system in **UM Simulation**. Parameters of a contact interaction as well as turning on/off contact between pairs of bodies are available in **UM Simulation**.

3D Contact supports parameterization of graphical objects that are used as contact manifolds. Firstly parameterized graphical objects for contact manifolds you can consider various configuration of contacting bodies in quite wide range simply changing corresponding parameters without remaking the graphical object itself. Parameterization of graphical object may be effectively used, for example, for searching the optimal shape of the friction wedge for so-called three-piece bogie for freight cars, Figure 3.160 below.

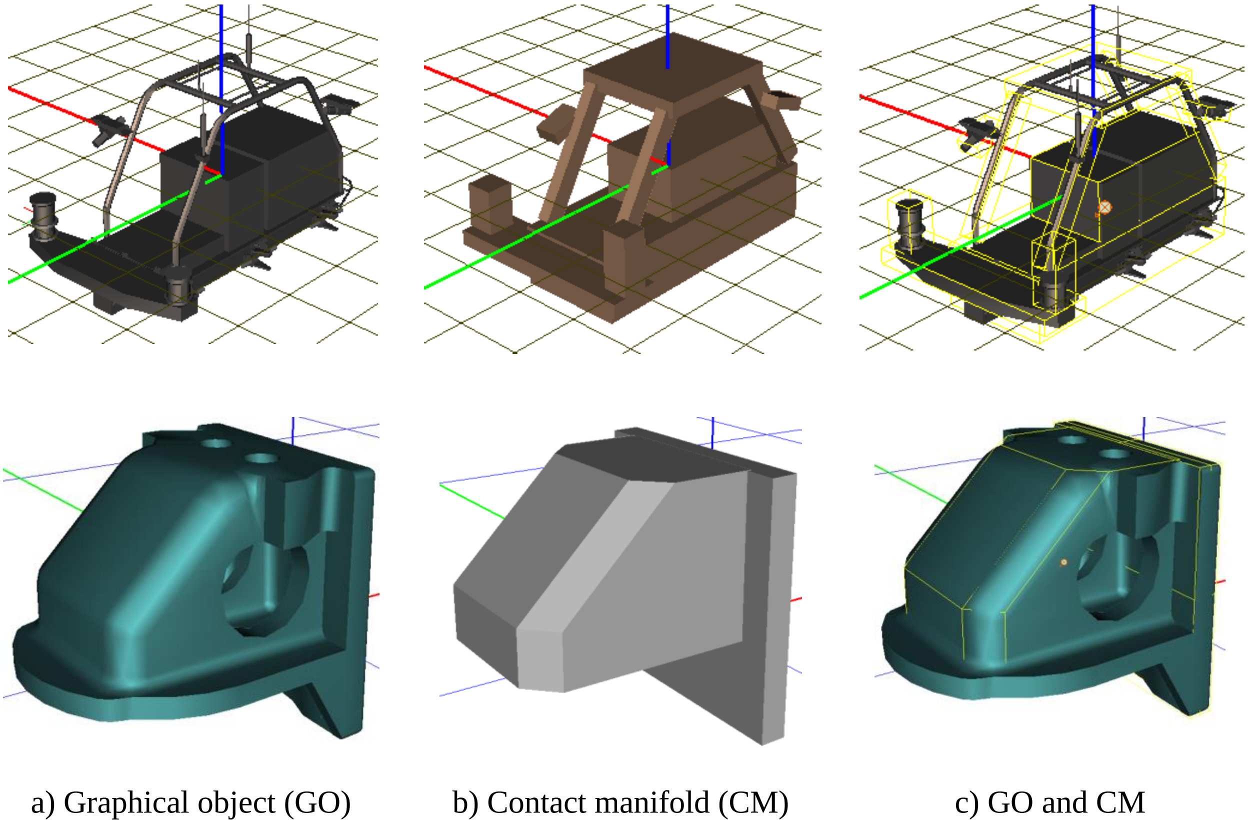
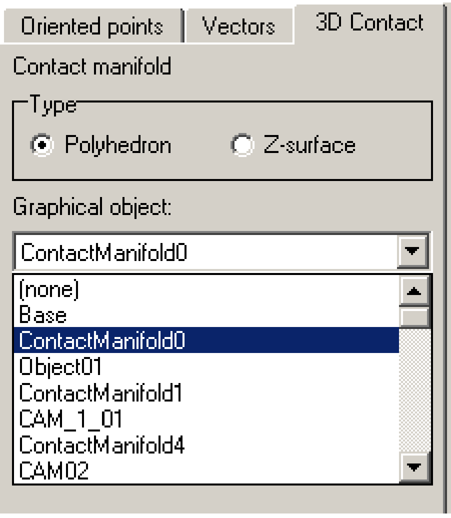


Figure 3.160. Graphical object and contact manifold for a body of wheeled robot

A graphical object that is assigned as a contact manifold for a body may differ from a graphical for the body. It is absolutely not necessary that it should be the same graphical objects. It is recommended to use simplified contact manifolds for decreasing the CPU efforts for simulation.

The contact manifold for the rigid body is selected from prepared in advance graphical objects that are available in the drop down list. Coordinate system of the contact manifold coincides with the coordinate system of the body. In the **Single element mode** (see Sect. 3.5.1.2.2. Modes of animation window, page 3-23) the contact manifold is drawn in yellow lines, see Figure 3.161.

There is no possibility to assign a contact manifold for *Base0* body. If it is necessary you should create an extra rigid body with 0 d.o.f. and assign the contact manifold of *Base0* with this extra body.



Contact manifold

Type

● Polyhedron ○ Z-surface

Graphical object:

ContactManifold0

(none)
Base
ContactManifold0
Object01
ContactManifold1
CAM_1_01
ContactManifold4
CAM02

Figure 3.161. 3D Contact parameters between pair of bodies

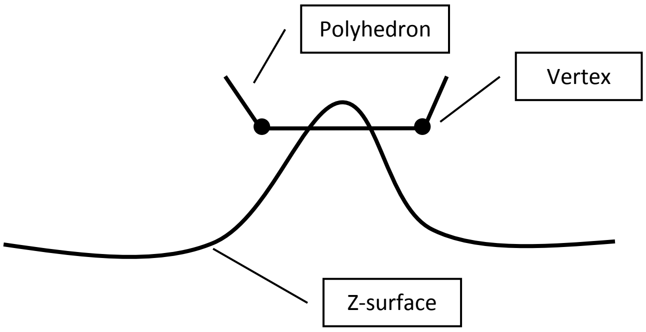
Types of contact manifold

Polyhedron. In general case use **Polyhedron** type for describing the contact interaction between bodies. Graphical objects that are chosen as contact manifolds should consist of graphical elements of the following types: **Box, Polyhedron** and **ASC**. Graphical elements of **Polyhedron** and **ASC** types should be convex and closed.

Z-surface. This type of contact manifold is used to describe contact between all bodies and the ground. Restrictions of convexity and closure are not applied for graphical objects for contact manifold of **Z-surface** type. Some comments concerning preparing a graphical object for using as a contact manifold of Z-surface type are given in Sect. 3.6.12.6.8. Points | Sphere | Circle - Z surface contact, page 3-219.

3D Contact simulation between **Polyhedrons** and **Z-surface** is based on using *Points – Z-surface* contact force that is described in Sect. 3.6.12.6.8. That is why it is also possible to describe contact interaction between polyhedrons and Z-surface with the help of *Points – Z-surface* contact force. The only difference between both variants is that **3D Contact** generates all contact forces of *Points – Z*-surface type automatically without necessity to create the forces for each body and the Z-surface manually. At the same time both variants are identical from point of view of results of simulation of dynamics of the system.

Treating the *Points – Z-surface* contact forces in **3D Contact** does not consider the interaction of edges and Z-surfaces, so contact forces will not appear in the case which is depicted in the Figure below.



Note 1. CPU efforts for simulation of *near contact* are proportional to square of count of faces and edges. To accelerate simulation process it is recommended to simplify contact manifolds 1 as far as possible for solving the particular problem.

Note 2. 3D Contact model as any mathematical model is just an approximation of real physical processes that take place between two bodies contacting bodies. Certainly the model has confined area of effective applications as well as there are cases where the model describes real processes incorrectly. 3D Contact model is rather a fast algorithm that is suitable for simulation of some models based on multibody system dynamics approach but surely is not suitable for detailed analysis of contact problem (contact stresses, deformations, wear, plastic effects and so on).

Comparison of the contact models.

Universal Mechanism software has to approaches to simulate contact interaction between rigid bodies. The first approach supposes using contact forces of “point-plane” and other types and the second approach is based on 3D Contact described above.

3D Contact is based on using “point-plane” contact forces and in this sense can be considered as an algorithm that detecting interpenetration of rigid bodies arranges contact points and identifies nearest faces as contact planes. At the same time calculation of interpenetration of contact manifolds is rather time-consuming operation that takes quite many CPU efforts. That is why simulating models with contacts it necessary to understand clearly advantages and disadvantages of both approaches. Detailed overview of contact forces see in Sect. 3.6.12.6. *Input of contact force elements*, page 3-205.

3D Contact often makes simulation of contact interaction more easy-to-use, intuitive and more suitable for parameterization, as well as widens contact interaction for “edge-edge” penetration case, see Chapter 2, Sect. *3D contact*. To simulate “edge-edge” penetration with the help of contact forces of “point-plane” type it needs to create contact points on each edge with quite small distance between them that significantly increases CPU efforts that neglect the only benefit in comparison with 3D Contact.

Please note that 3D Contact widens possibilities of simulation of contact interactions but needs extra computational efforts. At the same time approach based on using contact forces generally faster but not so universal as 3D Contact one.

Examples.

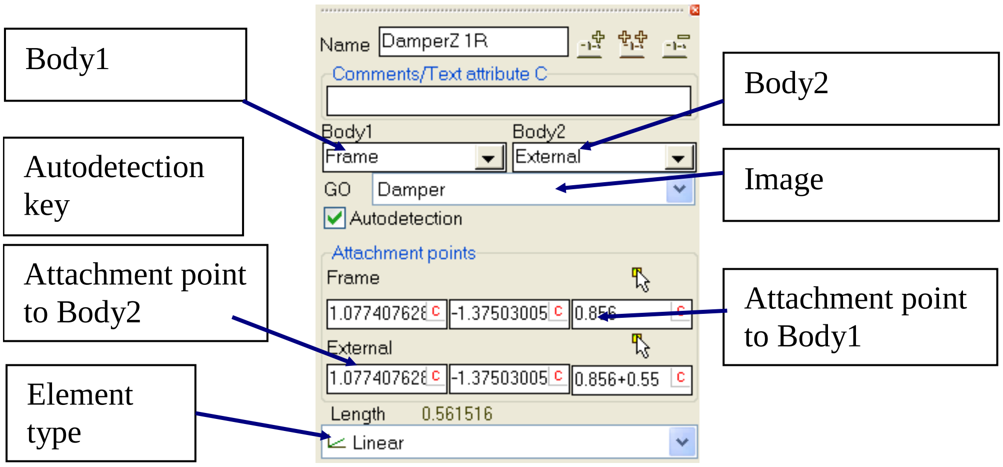
Please find some examples of using 3D Contact in the following models:

- {UM Data}\SAMPLES\Misc\Clockwork;
- {UM Data}\SAMPLES\Misc\DominoDay;

- {UM Data}\SAMPLES\Misc\Earthquake;
- {UM Data}\SAMPLES\Misc\FallingFigures;
- {UM Data}\SAMPLES\Rail vehicles\WedgeTest3DContact;
- {UM Data}\SAMPLES\Robots\krt_200;
- {UM Data}\SAMPLES\Robots\Manipulator.

3.6.10. Joints and force elements: some features of description

Joint and force elements have some parameters, which entering is quite analogous, e.g. a pair of bodies, attachment points or images should be assigned for the most of elements of these types. The standard interface is used for entering these parameters, for instance the corresponding interface for a bipolar force element looks like in Figure 3.162.


Body1
Autodetection key
Attachment point to Body2
Element type
Body2
Image
Attachment point to Body1
Comments/Text attribute C
Attachment points

Name DamperZ 1R

Body1 Frame

Body2 External

GO Damper

☑ Autodetection

Frame 1.07740762ε -1.37503005 0.856

External 1.07740762ε -1.37503005 0.856+0.55

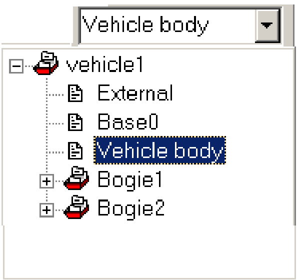
Length 0.561516

Linear

Figure 3.162. Main element of the interface

3.6.10.1. Assignment of bodies

A pair of bodies should be assigned to each joint or force element else the object description is considered as incomplete. One of the bodies is the first body (Body 1 in Figure 3.162), and another is the second body (Body 2). Use two drop-down lists to set the bodies (Figure 3.163).



Vehicle body

- vehicle1
  - External
  - Base0
  - Vehicle body
  - Bogie1
  - Bogie2

Figure 3.163. Drop-down list of bodies

The lists contain all bodies included in the object as well as the base body (Base0, SC0) and an *external* body (the second list only). The *Base0* is used for attachment of a body to a fixed point (to the base). The *External* body is used to make the element *external*, Sect. 3.6.3.3.1.

The first and the second bodies must be different.

3.6.10.2. Type of element

A type must be chosen with the help of a drop-down list (Figure 3.162, Figure 3.164). Changing the type leads to deleting of previous element description.

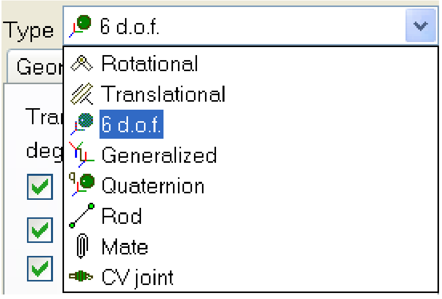


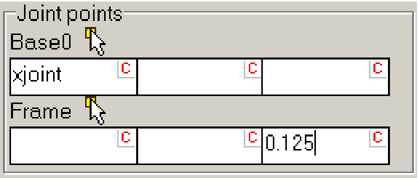
Figure 3.164. List of joint types

3.6.10.3. Attachment points

Attachment points have different notations for elements of different types ('joint points', 'attachment point' etc.). Anyway two points should be assigned for each element. One of the points belongs to the first body, another point – to the second one. The coordinates of the point should be entered in SC of the corresponding body as constant symbolic expressions.

Some elements require assignment of local system of coordinates (LSC) instead of points (bushing, generalized linear element, 6 d.o.f. joint, etc.) or vectors (rotational and translational joints, gearing, etc.).

3.6.10.4. Visual assignment of bodies and attachment points



xjoint		
		0.125

Figure 3.165. Joint points

To assign visually a body and the corresponding attachment point, vector or LSC click one of the buttons and select a point, an oriented point or a vector, which should be preliminarily described (see Sect. 3.6.9.6. Connection points).

Note 1. The type of the point should be chosen depending on the element. For example, description of a generalized linear force element requires both oriented points and general point. The description of a bipolar force element requires two general points.

Note 2. If the element requires coordinates of a point only, all types of connection points can be used. If the element requires a vector, an oriented point can be used (Z-axis of the LSC is used as the vector).

3.6.10.5. Autodetection

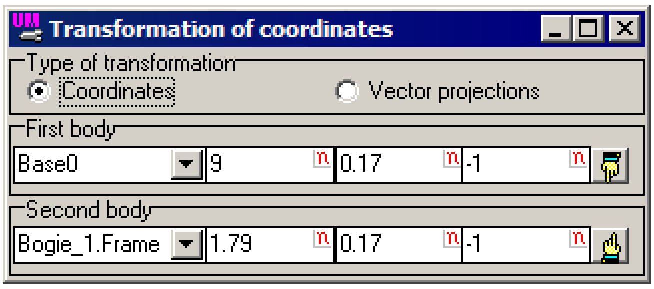
The autodetection mode can be set for the most of the force elements: bipolar force element, scalar torque, generalized linear force element, bushing, points-plane contact end so on. If the autodetection mode is assigned, the coordinates of attachment point or LSC of the second body are assigned in SC of the first body *by zero values of all joint coordinates*.

In particular, the autodetection mode is frequently used by external element disrobed in subsystems, Sect. 3.6.3.3.1.

3.6.10.6. Transformation of coordinates

Attachment points of force elements are set in the body-fixed SC, therefore, a tool for transformation of coordinates of points into different SC could be useful. To call the corresponding tool use the **Tools | Transformation of coordinates** ore use the hot key *Alt+T* (Figure 3.166).

Choose two bodies with the help of drop-down lists and set coordinates of a point in SC of one of the bodies (where the coordinates are known). After that click either the ☟ button (if the coordinates are known for the first body) or the ☝ button if the coordinates are known for the second body), and the coordinates are computed is SC of another body. Analogously can be computed projections of a vector on the axes of different SC.



Transformation of coordinates

Type of transformation: ● Coordinates ○ Vector projections

First body: Base0 | 9 | 0.17 | -1

Second body: Bogie_1.Frame | 1.79 | 0.17 | -1

Figure 3.166. Example of transformation of coordinates

3.6.11. Input of joints

3.6.11.1. Visualization of joints

In the single element mode of the animation window: a pair of bodies (kinematical pair) connected by the joint is drawn. A GO may be assigned for a *rod constraint* only.

In the full object mode of the animation window: the bodies of the kinematical pair are drawn as *active*. Optionally an icon marks a position of the joint point and visualizes its degrees of freedom (Sect. 3.5.1.2.1. Visualization of object elements). Click near the active region of the icon to call the description of the joint in the inspector (Sect. 3.5.1.2.1).

If the joint is cut (except rod, mate and CV joints), the second body is drawn twice in the full object mode: in the positions determined by adjusted and the cut joints (Sect. 3.6.11.1). It is recommended to change coordinate values in such a way that the both images of the body were close. The exact values of the coordinates are calculated in the Simulation program.

Note. By default some information about the joint is hidden. Use the ⯯ button to make it visible.

3.6.11.2. Weight of joint

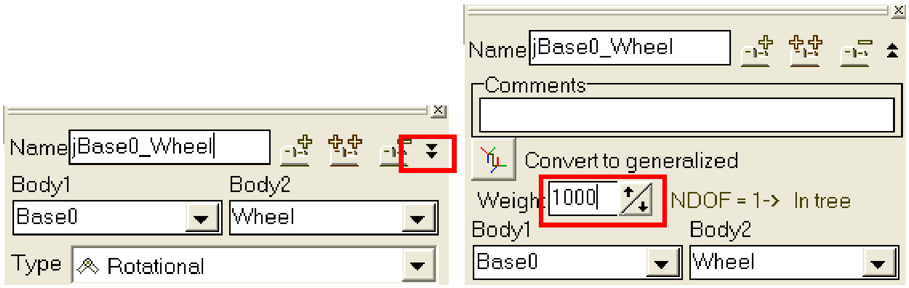


Figure 3.167. Setting joint’s weight

A weight coefficient can be assigned to each of the joints. This parameter is used when the object has closed kinematical loops (Chapter 2, Sect. *System graph. Closed kinematical loops*). If a large weight (e.g. 1000) is assigned to a joint in a closed loop, it will be cut.

Note. By default the weight information is hidden. Use the ⯯ button to make it visible, Figure 3.167.

3.6.11.3. Conversion of joint type

Joint’s types

- rotational
- translational
- six degrees of freedom (6 d.o.f.)

can be converted to the *generalized* joint type.

For example, a 6 d.o.f. joint can be converted to the generalized type to introduce joint forces for some degrees of freedom. A rotational or a translational joint is recommended to be transformed to the generalized joint type, e.g. to add degrees of freedom or to parameterize inclination of the joint axis.

To convert the joint type

- use the ⏬ button to open additional joint information, Figure 3.167.
- click the button to make the conversion.

Example. User's manual, Chapter 7, Sect. *Joint type conversion. Parameterization of axis inclination*.

3.6.11.4. Status of joint

Irrespective of a type, the joint can have a definite status, which must be known to the user for professional acquisition of the modeling methods:

- joint in tree,
- cut joint,
- removed joint,
- joint-fixation,
- joint with negative direction.

Consider definitions of these notions in more details. It is recommended to become familiar preliminary with Chapter 2, Sect. *Connectivity of systems and definition of a joint, System graph. Closed kinematical loops*.

3.6.11.4.1. Joint in tree and cut joint. Closure of cut joints

A joint is in **tree** if it is included in the object tree otherwise it is **cut**. Thereby, the cut joint is not used for computation of a body positions relative to SC0 with joint coordinates. In the mathematical sense, the cut joint adds a number of constraint equations to the equations of motion. Coordinates of joints included in the tree build the set of coordinates of the object relative to which equations of motion are generated, Sect. 3.5.2.3.4.

There exist two types of cut joints.

1. Joints – constraints, which do not have joint coordinates:
   - rigid rod, Sect. 3.6.11.9;
   - mate, Sect. 3.6.11.10;
   - convel joint, Sect. 3.6.11.11.

These joints introduce restrictions on relative motion of a pair of bodies. The joints of this type are always cut.

2. Joint with a set of local joint coordinates included in a closed kinematic loop, and cut by the program after automatic analysis of the object graph:
   - rotational;
   - translational;
   - joint with six degrees of freedom;
   - generalized joint;

- quaternion joint;
- internal body joint, Sect. 3.6.9.3.

The user can have an influence on the selection of cut joint in a closed loop by the joint weight, Sect. 3.6.11.2.

The following methods are available for verifying the joint status:

- icon of a cut joint in the element tree is stroked, Figure 3.168;
- status ‘cut’ of a joint is presented in the section of additional joint parameters; the section is available by the ⇊ button, Figure 3.169;
- on the Indices tab of the object inspector, Figure 3.170;
- joint coordinates of a cut joint are marked in the list of coordinates in the simulation module, Figure 3.171.

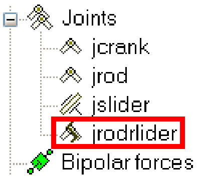
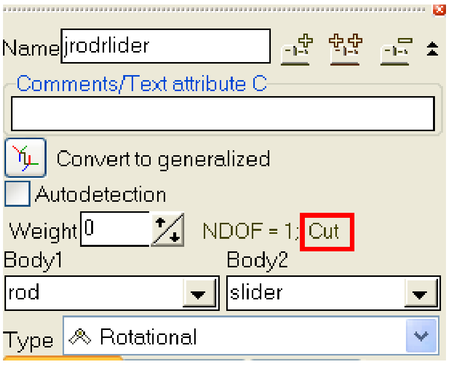


Figure 3.168. Icon of a cut joint in the element tree



Name jrodrlider

Comments/Text attribute C

Convert to generalized

□ Autodetection

Weight 0 NDOF = 1; Cut

Body1 rod

Body2 slider

Type Rotational

Figure 3.169. Status ‘cut’ of a joint in the section of additional joint parameters

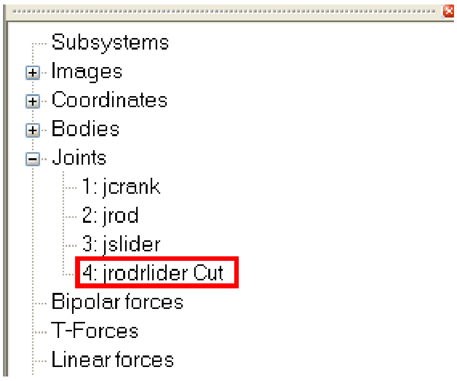
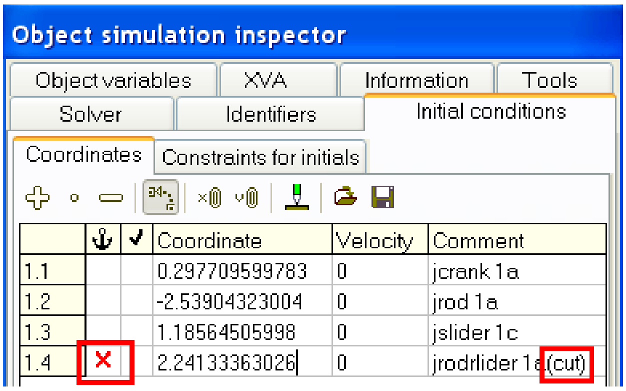


Figure 3.170. Status ‘cut’ of a joint on the Indices tab



	⚓	✓	Coordinate	Velocity	Comment
1.1			0.297709599783	0	jcrank 1a
1.2			-2.53904323004	0	jrod 1a
1.3			1.18564505998	0	jslider 1c
1.4	×		2.24133363026	0	jrodrlider 1a(cut)

Figure 3.171. Marker for coordinates in a cut joint

Closure of cut joints

Closure of cut joints is the process of solving constraint equations by the program, Figure 3.172. To close the cut joints, the button on the tool panel of the animation window or the buttons ✥ ∘ ▭ on the Coordinates tab of the object inspector are used.

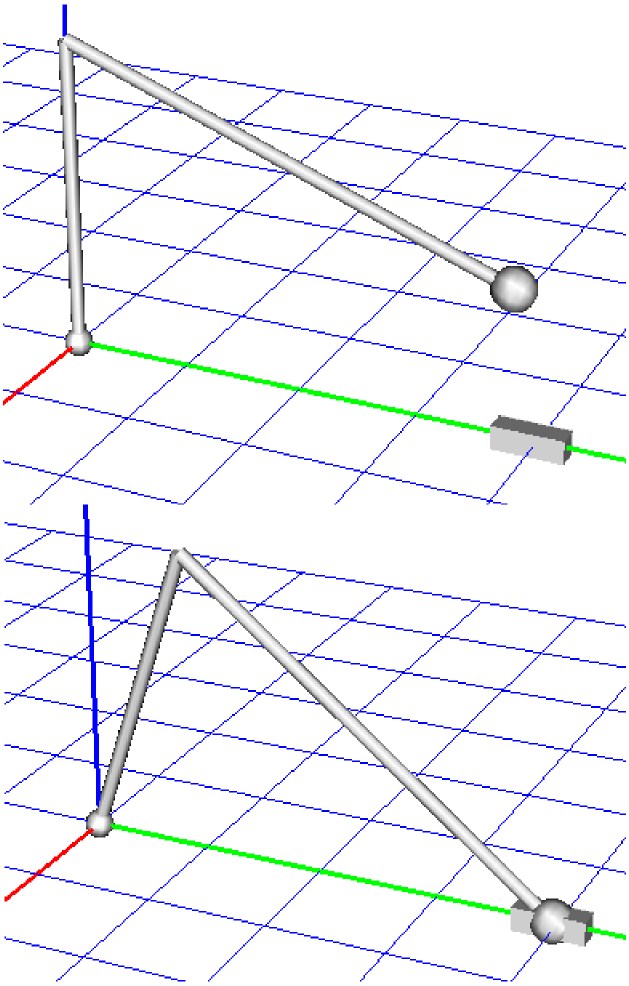


Figure 3.172. Crank-rod mechanism with open and closed cut joint

Remark. Solving constraint equations is a difficult numeric procedure, which does not always succeed, see Chapter 2, Sect. *Theoretical foundations for solving constraint equations*. If the equations are not solved in the input model and user cannot detect the cause, it is recommended to do it in the simulation module where advanced algorithms are used.

3.6.11.4.2. Removed joint

The **removed** status is assigned to the *cut joint with 6 degrees of freedom*, if the joint does not contain description of joint forces. Such the joint does not restrict the relative motion of connected bodies and does not affect the kinematic and dynamic properties of the model. In fact, the joint is not deleted from the model but it is ignored by the program in all computations.

A fictitious body gives an example of using such joints, Sect. 3.6.3.3.3.

If the joint is not an internal body joint (Sect. 3.6.9.3), its status ‘removed’ can be verified by the following ways:

- icon of the deleted joint in the element tree is crossed, Figure 3.173;
- status ‘removed’ is shown in the section of additional joint parameters; the section is available by the ⏷ button, Figure 3.174;
- on the **Indices** tab of the object inspector, Figure 3.175;
- coordinates of the deleted joint are not included in the *list of model coordinates*.

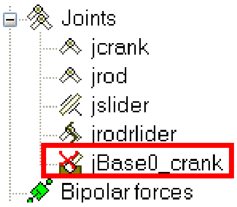
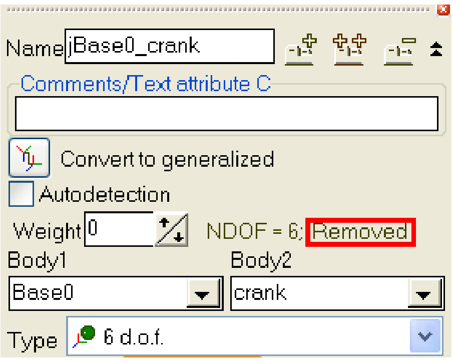


Figure 3.173. Icon for removed joint in the tree of elements


Name jBase0_crank

Comments/Text attribute C

Convert to generalized

□ Autodetection

Weight 0 NDOF = 6; Removed

Body1 Base0

Body2 crank

Type 6 d.o.f.

Figure 3.174. Status ‘removed’ in the section of additional joint parameters

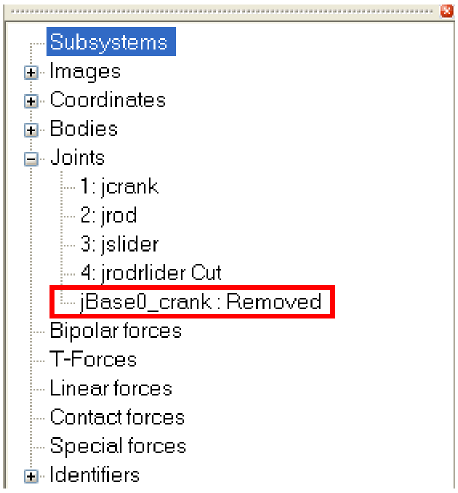


Figure 3.175. Status 'removed' the in Indices tab of the object inspector

3.6.11.4.3. Fixation joint

Fixation joint is the joint with zero degrees of freedom of the following types:

- generalized joint, see Sect. 3.6.3.3.3 as an example;
- 6 d.o.f. joint, see Sect. 3.6.9.7, 3.6.9.3 as an example.

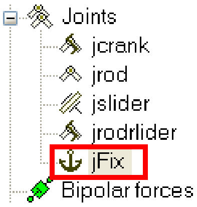


Figure 3.176. Icon of a fixation joint in the element tree

The joint is used to connect rigidly a pair of bodies.

3.6.11.4.4. Joint with negative direction

The notion 'joint with negative direction' is related to the evaluation of a body position relative to SC0 by a chain of joints.

After automatic cutting closed kinematic loops, a single path to each of the bodies from SC0 (body Base0) through a set of joints can be found, if the connectivity condition is satisfied for the model, Chapter 2, Sect. *Connectivity of systems and definition of a joint*. According to UM rule, each of the joints in this path specifies the *position of the second body in kinematic pair relative to the first one*. So, there exist joints of two types.

1. A joint has a **positive direction** if the first body in the kinematic pair lies in the path closer to SC0 than the second one, in particular if the *first body lies in the path from SC0 to the second body*. For instance, if the first body in the joint description is Base0 and the second body is Body1, the joint has the positive direction.
2. A joint has a **negative direction** if *the second body in the path is closer to Base0 than the first one*. For instance if the first body in the joint description is Body1 and the second body is Base0, the joint has the negative direction.

Consider an additional example of a chain of three bodies in the sequence Base0, Body1, Body2. The joint connecting Body1 and Body2 is positive if Body1 is assigned as the first one and negative in the opposite case.

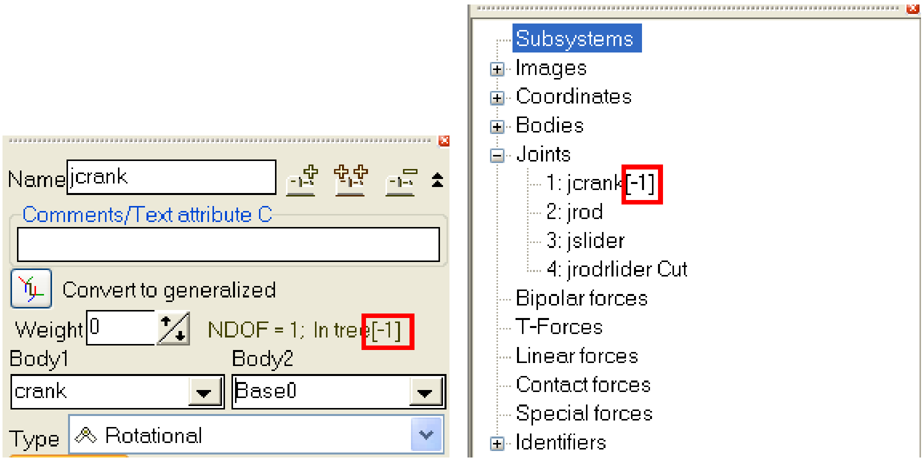
There exists a number of causes related to internal realization of kinematics of multibody systems, according to which **it is not recommended to introduce joints with negative directions**. Lack of logic in the case of a negative joint is clear from the following example. Consider a rotational joint connecting Body1 (the first body) with Base0. The joint coordinate in this case describe rotation of SC0 relative to Body1, and logically more correct is the opposite case.

The negative direction of a joint is marked as [-1] in the section of additional joint parameters, available by the ▼ button, as well as in the Indices tab of the object inspector, Figure 3.177.

To escape joints with negative direction, it is recommended to follow the simple rules for sequence of description of joints described in Sect. 3.6.1.

Remark 1. Direction is not introduced for cut joints.

Remark 2. Direction of joints can be changed if the user exchanges bodies in discretion of the joints and corrects coordinates of joint points.

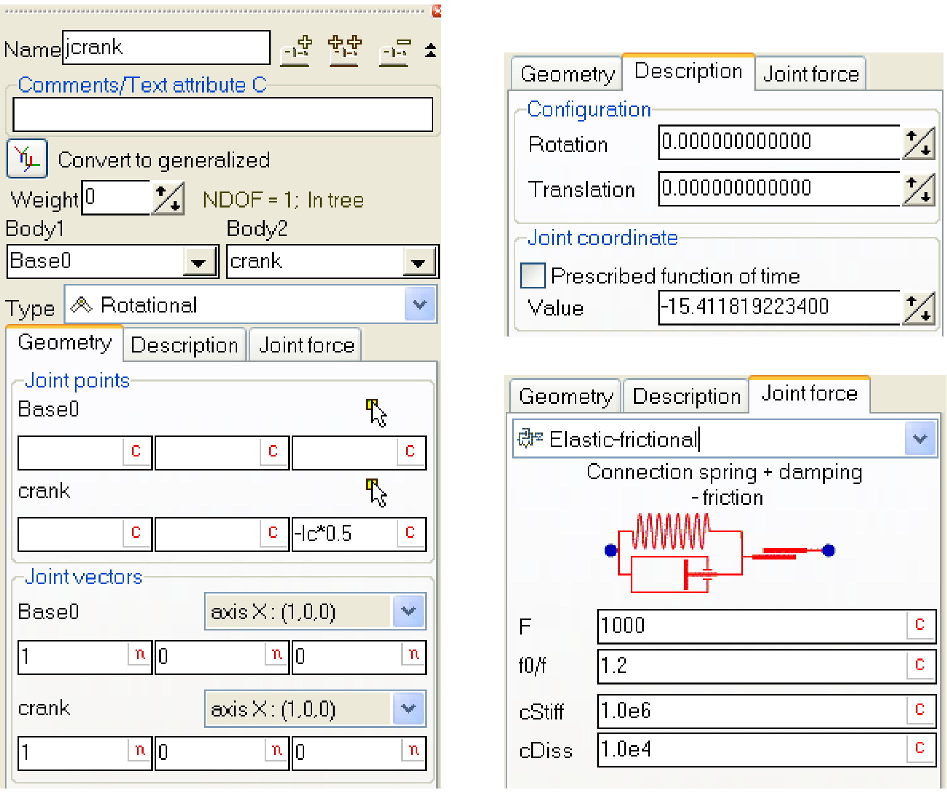


Name: jcrank
Comments/Text attribute C
Convert to generalized
Weight: 0
NDOF = 1; In tree[-1]
Body1: crank
Body2: Base0
Type: Rotational

Subsystems
- Images
- Coordinates
- Bodies
- Joints
  - 1: jcrank[-1]
  - 2: jrod
  - 3: jslider
  - 4: jrodrlider Cut
- Bipolar forces
- T-Forces
- Linear forces
- Contact forces
- Special forces
- Identifiers

Figure 3.177. Indication of joint with negative direction

3.6.11.5. Input of rotational and translational joints



Name: jcrank

Comments/Text attribute C

Convert to generalized

Weight: 0 NDOF = 1; In tree

Body1: Base0 Body2: crank

Type: Rotational

Geometry | Description | Joint force

Joint points

Base0: (empty) | (empty) | (empty)

crank: (empty) | (empty) | -lc*0.5

Joint vectors

Base0: axis X : (1,0,0) — 1 | 0 | 0

crank: axis X : (1,0,0) — 1 | 0 | 0

Geometry | Description | Joint force

Configuration

Rotation: 0.000000000000

Translation: 0.000000000000

Joint coordinate

□ Prescribed function of time

Value: -15.411819223400

Geometry | Description | Joint force

Elastic-frictional

Connection spring + damping - friction

F: 1000

f0/f: 1.2

cStiff: 1.0e6

cDiss: 1.0e4

Figure 3.178. Parameters of a rotational joint

Notions of translational and rotational joints are discussed in Chapter 2, Sect. *Translational and rotational joints*.

The following parameters should be entered in addition to bodies and joint points (Sect. 3.6.10.1, 3.6.10.3):

- projections of the joint vector on axes of two body-fixed SC (the **Geometry** tab, Figure 3.178); the vector cannot be zero;
- additional shift and rotation (the **Description** tab, the **Configuration** group); the parameters are optional.

If the bodies connected by the joint are in the tree (visible in the full object mode of the animation window, set the button in the ‘down’ state to verify this), the buttons can be used for visual entering both the joint points and the joint vectors. A joint vectors can be obtained from connection points of *vector* and *oriented point* types (see Sect. 3.6.10.4. Visual assignment of bodies and attachment points). Thus, selection of a vector or an oriented point allows the user to assign simultaneously a body, a joint point and a joint vector. If the oriented point is selected, the vector is directed along the Z axis of LSC, specified by the oriented point.

Further description of joint depends on type of the joint coordinate. The coordinate can be a degree of freedom or a time function.

Coordinate is a degree of freedom

All parameters are optional:

- *value* of the coordinate – the box is usually used to verify the correctness of the joint description: stepwise changing of the value leads to motion of bodies;
- type of the joint force/torque (Chapter 2, Sect. *Joint forces and torques*) can be chosen from the drop-down list on the *Joint force* tab, Figure 3.179, Sect. 3.6.12.2.

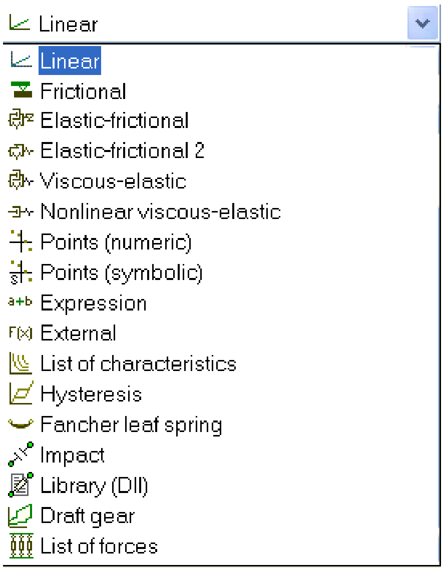
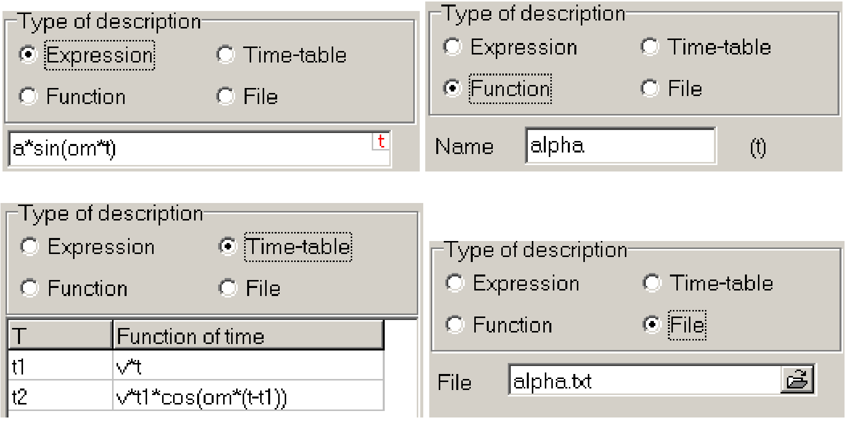


Figure 3.179. Types of joint force

After the type has been chosen, the boxes for force parameters appear.

Coordinate is a time function

Use the *Prescribed time function* check box to set this type of the coordinate.



Type of description

● Expression ○ Time-table

○ Function ○ File

a*sin(om*t)

Type of description

○ Expression ○ Time-table

● Function ○ File

Name alpha (t)

Type of description

○ Expression ● Time-table

○ Function ○ File

T	Function of time
t1	v*t
t2	v*t1*cos(om*(t-t1))

Type of description

○ Expression ○ Time-table

○ Function ● File

File alpha.txt

Figure 3.180. Time function

The time function can be set as

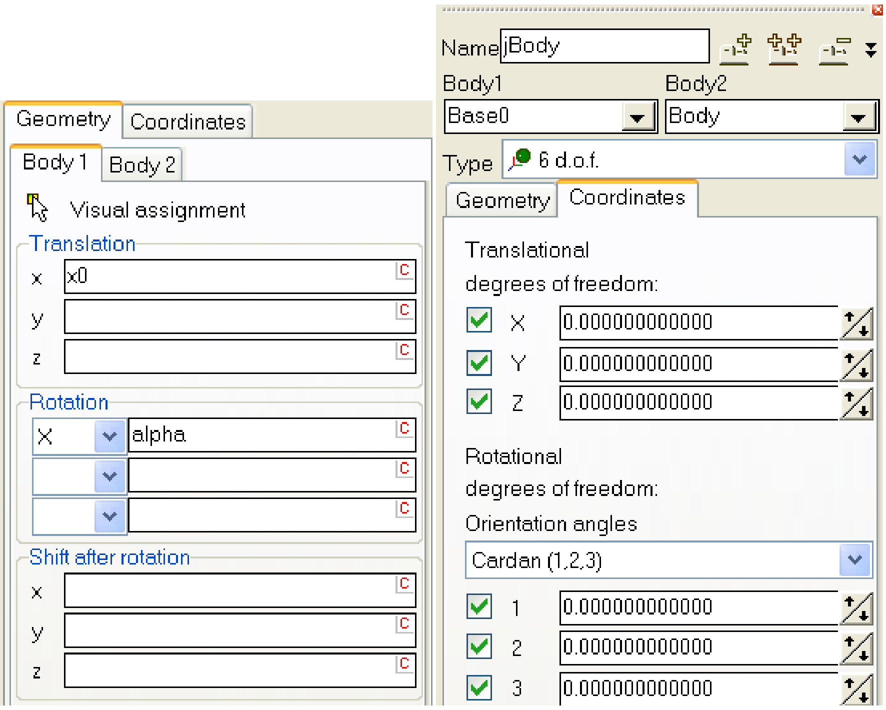
- an explicit expression (Sect. 3.5.2.4.6)
- an external function (Sect. 3.5.2.4.10)
- a time table (Sect. 3.5.2.4.12)
- a function from a text file (Sect. 3.5.2.4.11)

Remark. Both rotational and translational joints is recommended to be transformed to the generalized joint type, e.g. to add degrees of freedom or to parameterize inclination of the joint axis, Sect. 3.6.11.3.

Example. User's manual, Chapter 7, Sect. *Joint type conversion. Parameterization of axis inclination*.

3.6.11.6. Input of 6 d.o.f. joint

Detailed description of the joint can be found in Sect. Chapter 2, Sect. Six d.o.f. joint.



Translation: x: x0; y: ; z:

Rotation: X alpha

Name: jBody

Body1: Base0

Body2: Body

Type: 6 d.o.f.

☑ X 0.000000000000

☑ Y 0.000000000000

☑ Z 0.000000000000

Orientation angles: Cardan (1,2,3)

☑ 1 0.000000000000

☑ 2 0.000000000000

☑ 3 0.000000000000

Figure 3.181. Description of 6 d.o.f. joint

In addition to connected bodies (Sect. 3.6.10.1), the user should set positions of local joint system of coordinates SC1A and SC2B relative to SC1 and SC2 in the **Geometry** tab. For this purpose the sheets Body1 and Body2 are used, Figure 3.181 left. The standard interface is used for parametric setting the SC positions, see Sect. 3.6.4.

The following parameters should be entered to specify the joint coordinates (Figure 3.181, right):

- type of orientation angles;
- "turned on" degrees of freedom (check the presented degrees of freedom and turn off the others); the three upper rows correspond to translational d.o.f. (Figure 3.181).

Use the buttons ↑↓ in the edit boxes to set initial values as well as to verify the correctness of the joint description.

Remark 1. Cut joints with **six** d.o.f. are ignored in the Simulation program.

Remark 2. 6 d.o.f. joint can be converted to the generalized type to introduce joint forces for some degrees of freedom, Sect. 3.6.11.3.

Example. An example of a joint used for setting six degrees of freedom of a box relative to SC0 according to data in Figure 3.181 left for *x0*=1m, *alpha*=30 degrees is shown in Figure 3.182. The box is shown in the position corresponding to zero values of joint coordinates. Features of the joint are the parametric description of both the SC1A origin (shift along the X-axis *x0*) and angle of rotation about the X axis (*alpha*). As a result two translational and two rotational degrees are specified along inclines axes Y and Z.

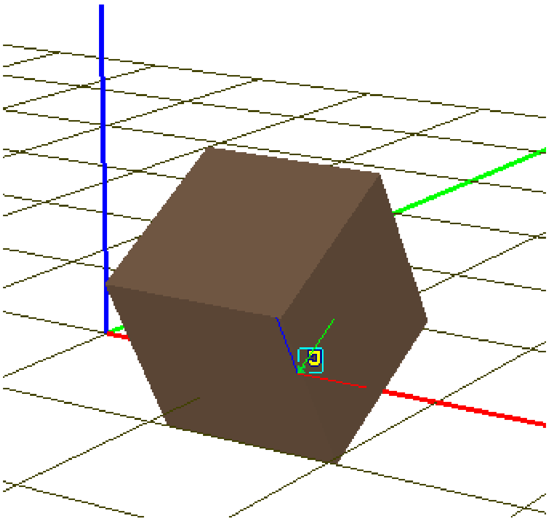
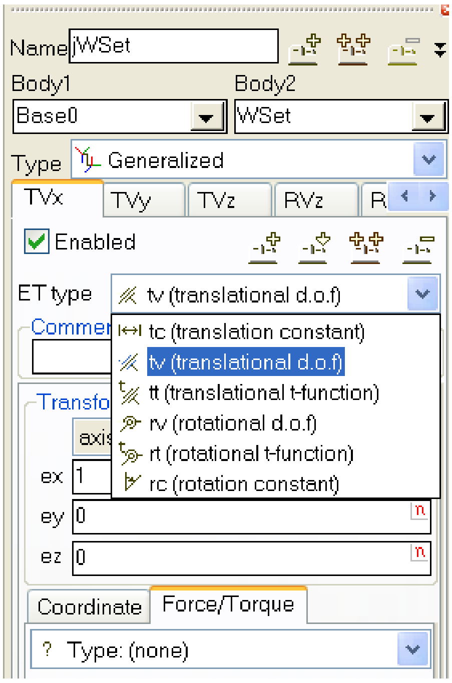


Figure 3.182. Example of a joint with six degrees of freedom

3.6.11.7. Input of joint of generalized type

The joint description consists of a sequence of elementary transformations (ET), Chapter 2, Sect. Generalized joint.



Name jWSet

Body1 Base0

Body2 WSet

Type Generalized

☑ Enabled

ET type tv (translational d.o.f)

ex 1

ey 0

ez 0

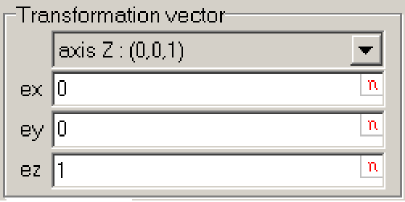
? Type: (none)

Figure 3.183. Generalized joint

The list of ET is created with the help of the standard interface. To specify a type of ET use the pull-down menu (Figure 3.183). Edit boxes for ET parameters appear after the choice of the type.

The **Enabled** key allows excluding a transformation.

A unit **transformation vector** should be entered for all ET types except tc (the vector cannot be zero).



Transformation vector

axis Z : (0,0,1)

ex 0

ey 0

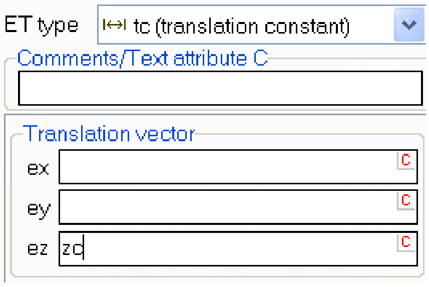
ez 1

Figure 3.184. Unit vector of a transformation

Use the pull-down list to set a standard value of the vector.

Cut joints with 6 d.o.f. are ignored in the Simulation program if no joint forces are presented.

3.6.11.7.1. Elementary transformation tc



ET type: tc (translation constant)

Comments/Text attribute C

Translation vector

ex

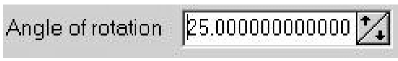
ey

ez: zc

Figure 3.185. ET: translation constant

Enter a shift vector, which components are constant symbolic expressions.

3.6.11.7.2. Elementary transformation rc



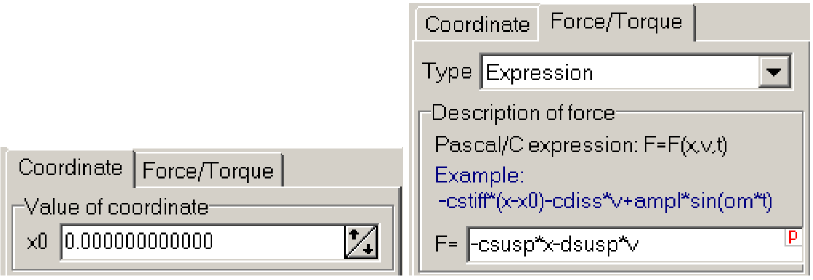
Angle of rotation: 25.000000000000

Figure 3.186. ET: constant angle rotation

Angle of rotation (in degrees) must be entered in addition to the transformation vector.

Remark. Use the *rt* type to set the constant rotation angle as an identifier or constant expression.

3.6.11.7.3. Elementary transformations tv, rv



Value of coordinate

x0: 0.000000000000

Type: Expression

Description of force

Pascal/C expression: F=F(x,v,t)

Example: -cstiff*(x-x0)-cdiss*v+ampl*sin(om*t)

F= -csusp*x-dsusp*v

Figure 3.187. ET: rotational and translational degree of freedom

The following parameters can be optionally set in addition to the transformation vector

- Numeric (initial) *value* of the joint *coordinate* (the angle coordinate is entered in degrees); use the buttons in the edit box to get the animation of motion corresponding to changing the coordinate.
- Mathematical model of a joint force/torque. Choose the force type from the list

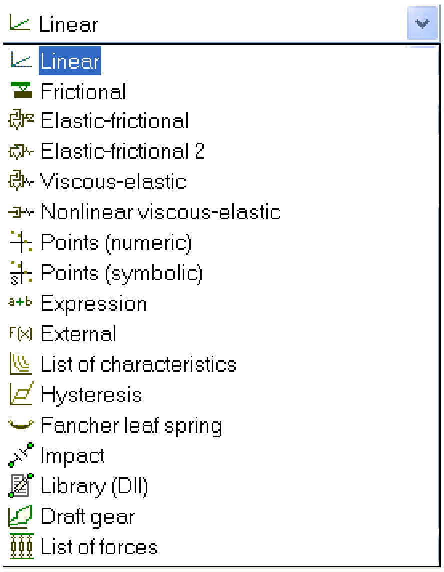
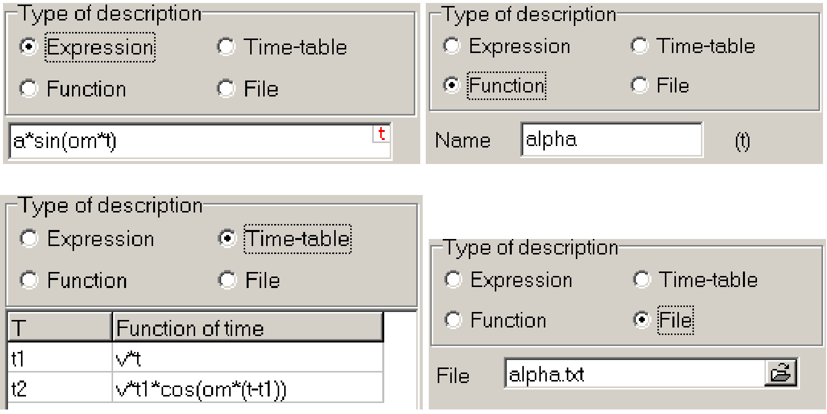


Figure 3.188. Types of joint force

After the type has been chosen, the boxes for force parameters appear.

3.6.11.7.4. Elementary transformations tt, rt



Type of description
● Expression ○ Time-table
○ Function ○ File
a*sin(om*t)

Type of description
○ Expression ○ Time-table
● Function ○ File
Name alpha (t)

Type of description
○ Expression ● Time-table
○ Function ○ File

T	Function of time
t1	v*t
t2	v*t1*cos(om*(t-t1))

Type of description
○ Expression ○ Time-table
○ Function ● File
File alpha.txt

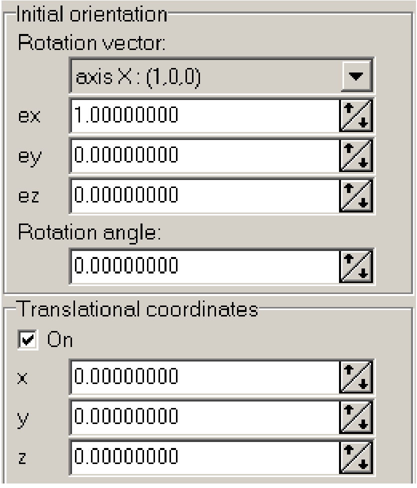
Figure 3.189. Time function

In addition to transformation vector enter a time function for the coordinate as

- an explicit expression (Sect. 3.5.2.4.6)
- an external function (Sect. 3.5.2.4.10)
- a time table (Sect. 3.5.2.4.12)
- a function from a text file (Sect. 3.5.2.4.11)

Remark. Use the type *rt* for entering a rotation with a constant angle, which can be set as an identifier or an explicit expression. In this case chose the *Expression* type of description and enter an expression, which does not depend on time t (Figure 3.189, left).

3.6.11.8. Input of quaternion joint



Initial orientation

Rotation vector:

axis X : (1,0,0)

ex 1.00000000

ey 0.00000000

ez 0.00000000

Rotation angle:

0.00000000

Translational coordinates

☑ On

x 0.00000000

y 0.00000000

z 0.00000000

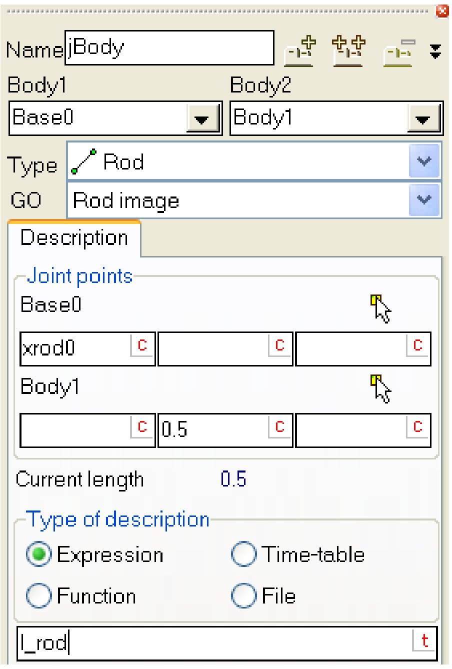
Figure 3.190. Parameters of a quaternion joint

A detailed description of a quaternion joint can be found in Chapter 2, Sect. *Quaternion joint*. The following data can be set in addition to bodies and joint points (Sect. 3.6.9.7):

- Initial orientation of the second body relative to the first one (angle is entered in degrees);
- Translational coordinates can be turned off to obtain a spherical joint;

Remark. Cut quaternion joints with 6 d.o.f. are ignored in the Simulation program.

3.6.11.9. Input of rod constraint



Name jBody

Body1 Base0

Body2 Body1

Type Rod

GO Rod image

Description

Joint points

Base0

xrod0 | |

Body1

| 0.5 |

Current length 0.5

Type of description

● Expression ○ Time-table

○ Function ○ File

l_rod

Figure 3.191. Rod joint parameters

A detailed description of a rod joint can be found in Chapter 2, Sect. *Weightless rod constraint*. In addition to the bodies and attachment points (Sect. 3.6.9.7), the length of the rod should be entered. The length can be either constant of a time function.

Set a length as

- an explicit expression (Sect. 3.5.2.4.6)
- an external function (Sect. 3.5.2.4.10)
- a time table (Sect. 3.5.2.4.12)
- a function from a text file (Sect. 3.5.2.4.11)

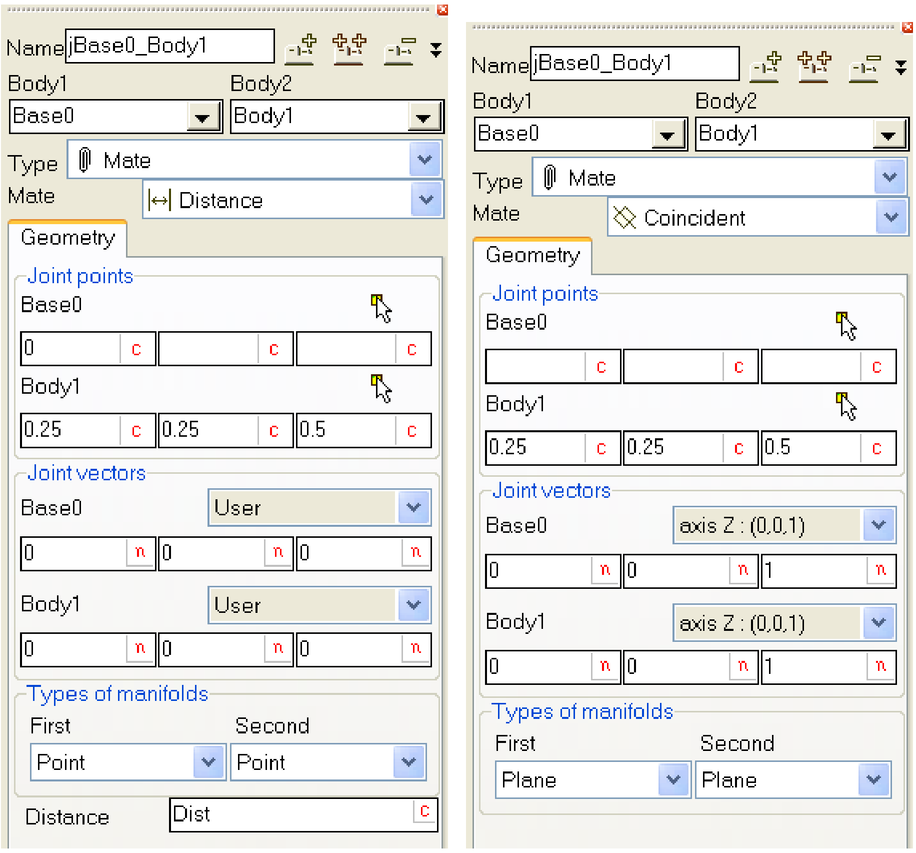
As a rule, a graphic object is assigned to the rod (Sect. 3.6.6).

The current distance between the rod attachment points is presented in the *Current length* box. Use this parameter to verify the correctness of the length description.

Remark. The rod is a constraint. It does not introduce coordinates but restricts relative position of connected bodies. Exact calculation of positions in this case can be done by the button in the top of animation window or in the Simulation program. That is why the current length of the rod in the Input program can differ from the real length entered by the user until the constraint computation is done.

3.6.11.10. Input of mates

Definition of mates can be found in Chapter 2, Sect. *Mates*.



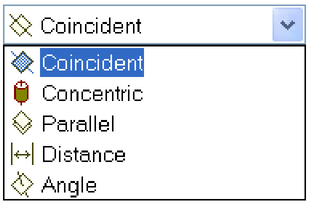
Name jBase0_Body1
Body1 Base0
Body2 Body1
Type Mate
Mate Distance
Geometry
Joint points
Base0: 0 c, c, c
Body1: 0.25 c, 0.25 c, 0.5 c
Joint vectors
Base0: User
0 n, 0 n, 0 n
Body1: User
0 n, 0 n, 0 n
Types of manifolds
First: Point
Second: Point
Distance: Dist c

Name jBase0_Body1
Body1 Base0
Body2 Body1
Type Mate
Mate Coincident
Geometry
Joint points
Base0: c, c, c
Body1: 0.25 c, 0.25 c, 0.5 c
Joint vectors
Base0: axis Z : (0,0,1)
0 n, 0 n, 1 n
Body1: axis Z : (0,0,1)
0 n, 0 n, 1 n
Types of manifolds
First: Plane
Second: Plane

Figure 3.192. Examples of mates

The following parameters describe a mate, Figure 3.192.

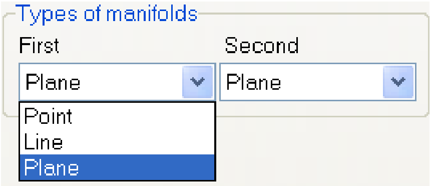
- **Mate type**, Figure 3.193.



Coincident

Figure 3.193. Mate types

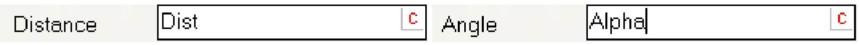
- **Types** of the first and second **manifolds**,



Types of manifolds
First: Plane
Second: Plane

Figure 3.194. Manifolds

- Coordinates of joint points for the first and second body in the SC of the corresponding body. The points specify
  - coordinates of the manifold of the *point* type;
  - coordinates of the initial point of the *vector* manifold;
  - coordinates of any point on the *plane* manifold.
- If the manifold is a vector or a plane, projections of the vector or the normal to the plane must be defined in the group **Joint vectors**. The joint vector is ignored in the case of a point manifold.
- Mates **Distance** and **Angle** require an additional parameter: the value of the distance or the angle between manifolds, Figure 3.195.



Distance: Dist

Angle: Alpha

Figure 3.195. Distance and angle parameters

The standard components Mates in Figure 3.196 are used for a visual assignment of bodies, joint points and vectors. The following steps are necessary:

- connection points or vectors should be preliminary assigned to the bodies;

- to create a mate, click on the button with the visual component and follow the instructions.

To assign a body, a joint point and a vector, it is enough to select by the mouse a *connection point* of a vector type or an oriented point, Sect. 3.6.9.6.

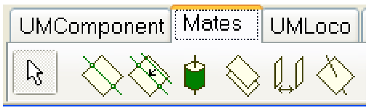


Figure 3.196. Standard mate components

Remark. If mates are presented in a model, the *Park Parallel* numeric method cannot be used for simulation.

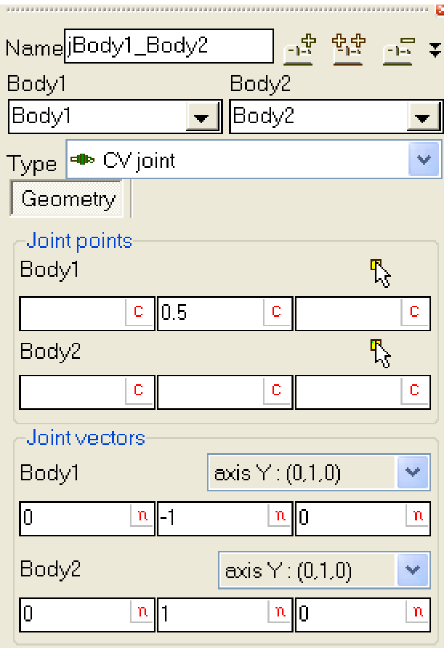
Simple simulation examples with mates are available in the directory
{Path to UM}\Samples\Library\Mates

3.6.11.11. Input of convel (CV) joint

See Chapter 2, Sect. *Convel joint* for a definition of the joint.

The following parameters specify the joint, Figure 3.197:

- coordinates of joint points in the body-fixed CS.
- projections of joint vectors in the body-fixed CS.



Name jBody1_Body2

Body1: Body1

Body2: Body2

Type: CV joint

Geometry

Joint points

Body1: | 0.5 |

Body2: | |

Joint vectors

Body1: axis Y : (0,1,0)

0 | -1 | 0

Body2: axis Y : (0,1,0)

0 | 1 | 0

Figure 3.197. Example of a convel joint

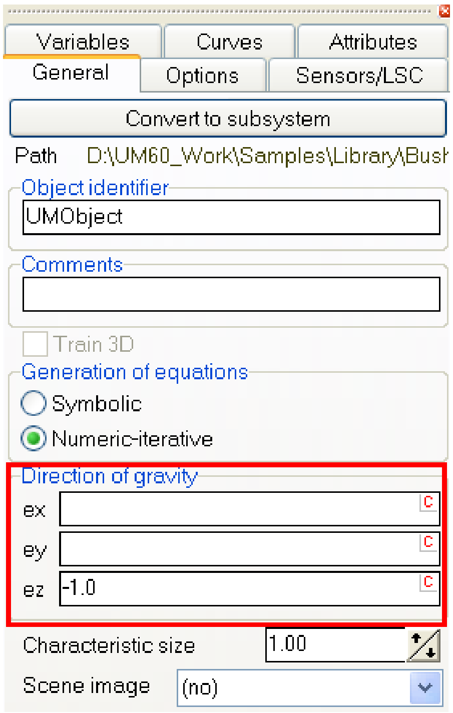
Remark. If a convel joint is presented in a model, the *Park Parallel* numeric method cannot be used for simulation.

Example. See Chapter 7, Sect. *Convel joint*.

3.6.12. Input of force elements

3.6.12.1. Input of gravity

Use the **Object | General** tab of the inspector for setting the gravity force, Figure 3.198, Sect. 3.5.2.1. Here both the direction of the gravity force and the gravity acceleration can be changed.



Path D:\UM60_Work\Samples\Library\Bush

Object identifier: UMObject

Comments:

□ Train 3D

Generation of equations
○ Symbolic
● Numeric-iterative

Direction of gravity
ex
ey
ez -1.0

Characteristic size 1.00

Scene image (no)

Figure 3.198. Direction of gravity

3.6.12.2. Description of scalar force and torque

Mathematical models of scalar forces are described in Chapter 2, Sect. *Types of scalar forces*.

The scalar forces are used by descriptions of the following types of force elements:

- joint force, Sect. 3.6.11.5, 3.6.11.7.3,
- scalar torque, Sect. 3.6.12.4,
- bipolar force element, Sect. 3.6.12.3,
- generalized bushing, Sect. 3.6.12.8.6.2,
- axial force for combined friction, Sect. 3.6.12.8.7.5.

Below we consider parameters of scalar forces of different types, Figure 3.199.

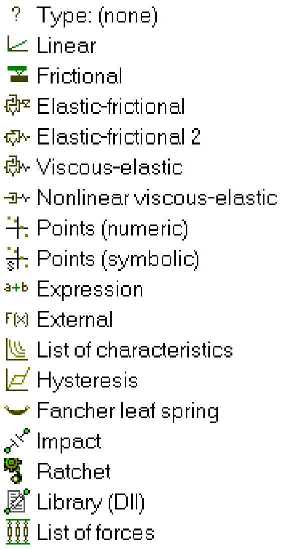
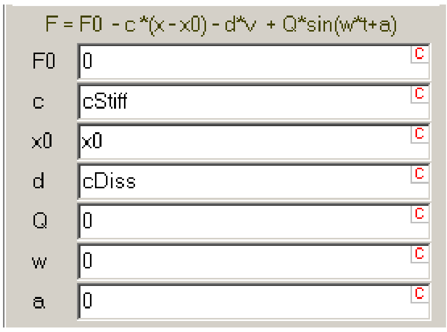


Figure 3.199. Types of scalar force

3.6.12.2.1. Linear force element

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces | Linear force*.



F = F0 - c*(x-x0) - d*v + Q*sin(w*t+a)

F0	0
c	cStiff
x0	x0
d	cDiss
Q	0
w	0
a	0

Figure 3.200. Parameters of linear force element describing a linear viscous-elastic interaction

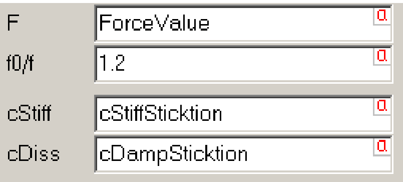
The boxes in the window (Figure 3.200) correspond to the following parameters of the element:

- **F0** is constant component of the force;
- **cStiff** is stiffness constant;
- **cDiss** is damping constant;
- **x0** is the coordinate for zero value of the elastic component;
- **Q, w, a** is amplitude, frequency (rad/s) and initial phase (rad) of the harmonic excitation.

All the parameters are constant symbolic expressions, Sect. 3.5.2.4.5.

3.6.12.2.2. Friction and elastic-frictional elements

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces | Friction force and Types of scalar forces | Elastic-friction force*.



F	ForceValue
f0/f	1.2
cStiff	cStiffSticktion
cDiss	cDampSticktion

Figure 3.201. Parameters of friction element

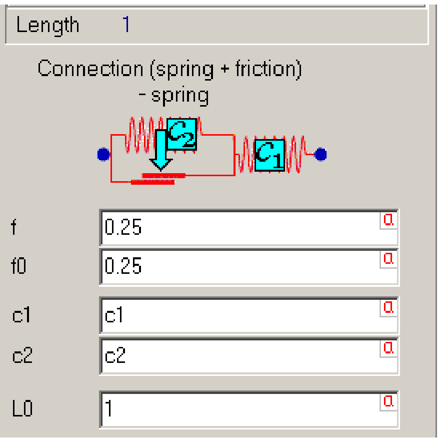
The following parameters should be specified:

- friction force value *F*;
- static/dynamic coefficient of friction ratio *f0/f*;
- stiffness at sticking *cStiff*.
- damping constant at sticking *cDiss*.

All parameters are constant symbolic expressions, Sect. 3.5.2.4.5.

3.6.12.2.3. Elastic-frictional element 2

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces* | *Elastic-friction force 2*.



Length 1

Connection (spring + friction)
- spring

f	0.25
f0	0.25
c1	c1
c2	c2
L0	1

Figure 3.202. Parameters of elastic-friction element 2

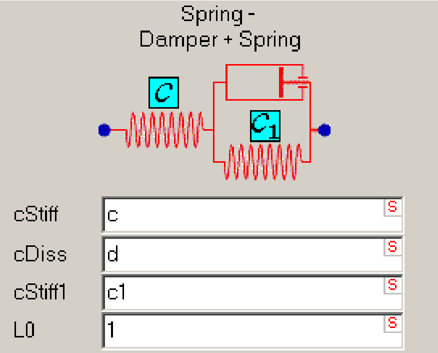
The following parameters should be specified:

- Dynamic coefficient of friction *f*;
- Static coefficient of friction *f*0 (usually a bit greater than the dynamic one);
- Stiffness of the first spring *c*1;
- Stiffness of the second spring *c*2;
- Element length in the unloaded state.

All parameters are constant symbolic expressions, Sect. 3.5.2.4.5.

3.6.12.2.4. Viscous-elastic element

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces | Stiffness and damping in series and in parallel*.



cStiff c

cDiss d

cStiff1 c1

L0 1

Figure 3.203. Parameters of viscous-elastic element

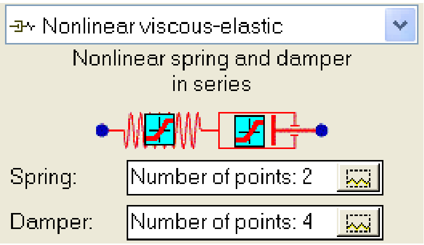
The following parameters should be specified:

- Stiffness *cStiff* in series (N/m), *c* in figure;
- Damping constant *cDiss* (Ns/m);
- Stiffness *cStiff* 1 in parallel (N/m), c_1 in figure (can be zero);
- Length of unloaded element L0 (ignored if c_1=0)

Parameters are specified as constant symbolic expressions.

3.6.12.2.5. Nonlinear viscous-elastic element

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces | Nonlinear spring and damper in series*.



Nonlinear viscous-elastic

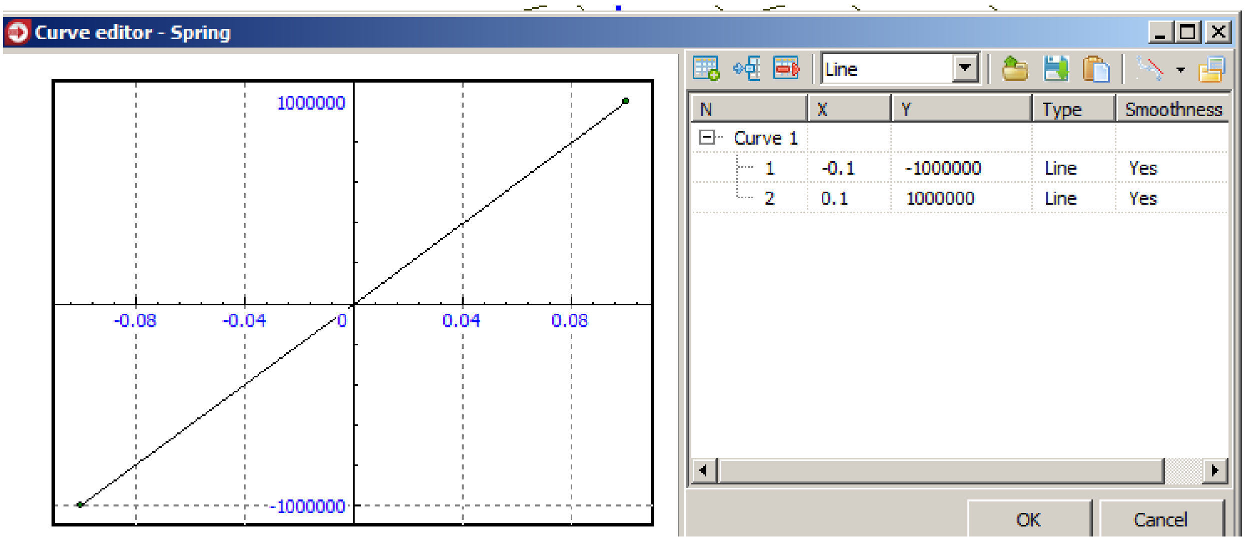
Nonlinear spring and damper in series

Spring: Number of points: 2

Damper: Number of points: 4

Figure 3.204. Element parameters

Description of this element includes two plots: spring force vs. deflection as well as damper force vs. velocity. Use the buttons to call the curve editor for defining point on plots (Figure 3.205, Figure 3.206).

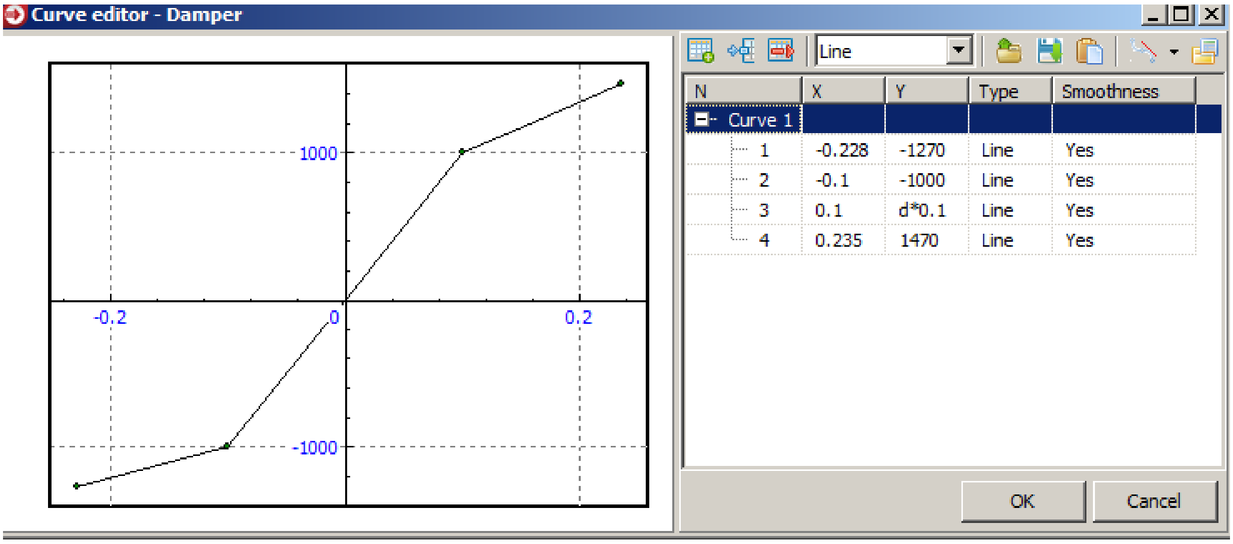


Curve editor - Spring

Line

N	X	Y	Type	Smoothness
Curve 1				
1	-0.1	-1000000	Line	Yes
2	0.1	1000000	Line	Yes

Figure 3.205. Parameterized model of linear spring



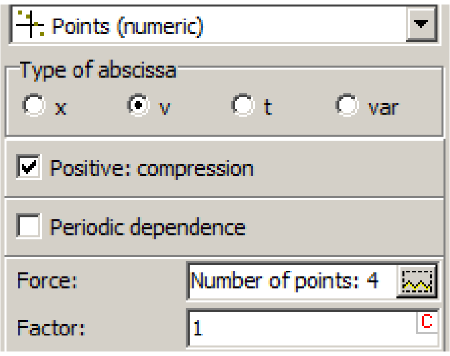
N	X	Y	Type	Smoothness
Curve 1				
1	-0.228	-1270	Line	Yes
2	-0.1	-1000	Line	Yes
3	0.1	d*0.1	Line	Yes
4	0.235	1470	Line	Yes

Figure 3.206. Parameterized model of nonlinear damper

Parameterized expressions can be used as coordinate values. Spline or B-spline can be applied for plot smoothing.

3.6.12.2.6. Points (numbers)

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces* | *Points model*.



Points (numeric)

Type of abscissa

○ x ● v ○ t ○ var

☑ Positive: compression

□ Periodic dependence

Force: Number of points: 4

Factor: 1

Figure 3.207. Parameters of Points (numbers) element

Curve editor is used to set a plot of the force by a set of points (Sect. 3.5.3, Figure 3.208), use the button to call the editor.

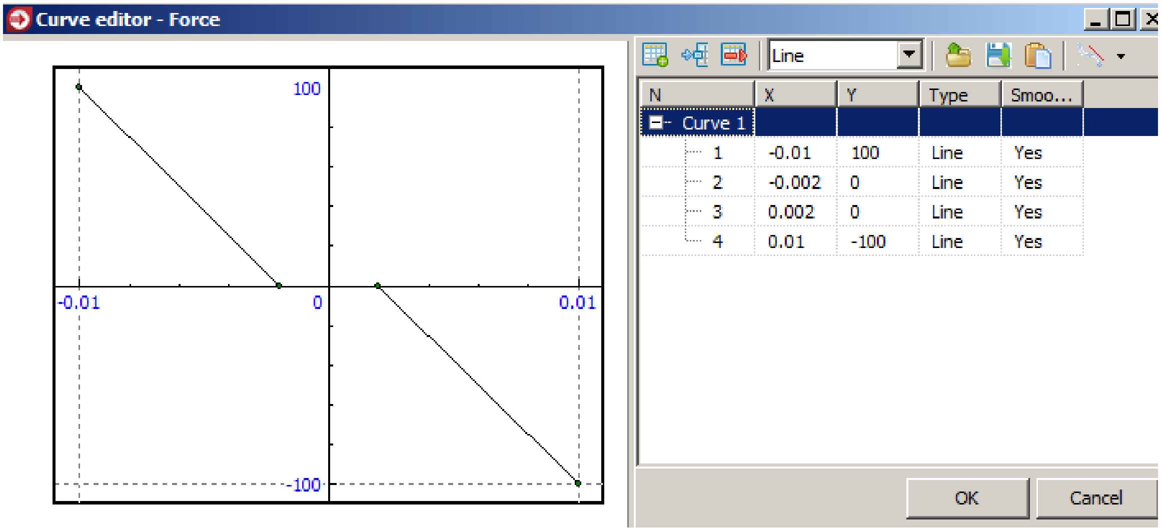
The force can depend on the

- coordinate *x*,
- velocity *v*,
- time *t*,
- variable defined by the user, Sect. 3.5.2.4.8.

Use the *Type of abscissa* group to select a necessary dependence.

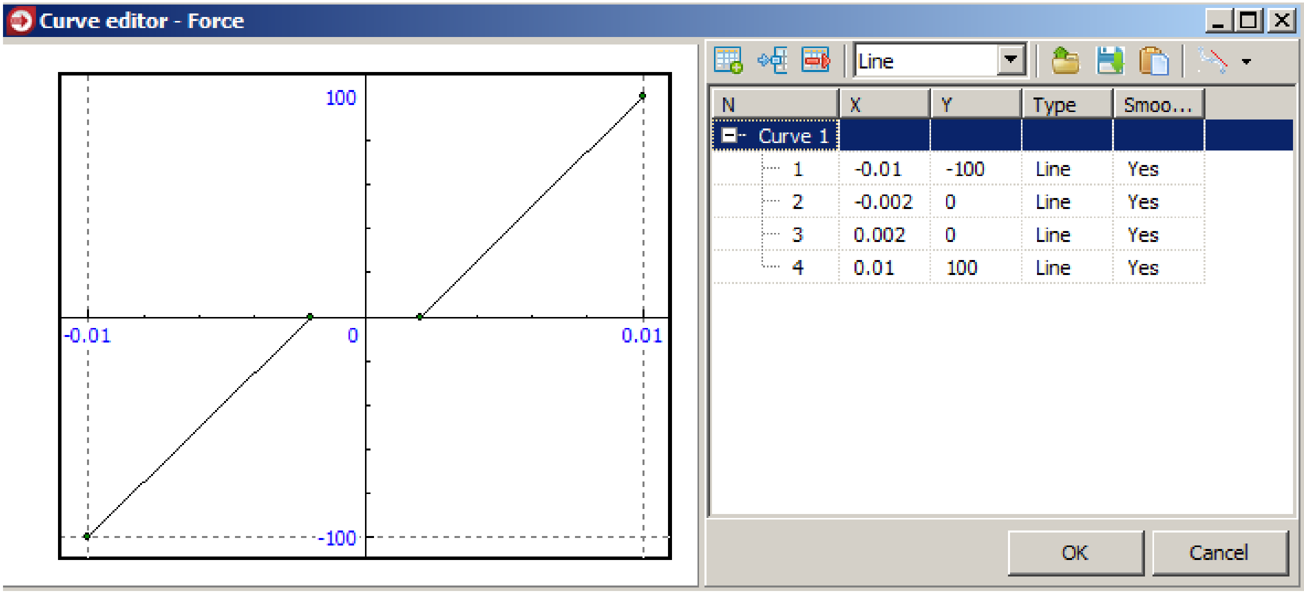
The *Factor* parameter changes scale of ordinate values. Zero value of the factor, in fact, removes the element. Usually the multiplier is set by an identifier.

3.6.12.2.6.1. Force dependence on x, v, t



N	X	Y	Type	Smoo...
Curve 1				
1	-0.01	100	Line	Yes
2	-0.002	0	Line	Yes
3	0.002	0	Line	Yes
4	0.01	-100	Line	Yes

a)



N	X	Y	Type	Smoo...
Curve 1				
1	-0.01	-100	Line	Yes
2	-0.002	0	Line	Yes
3	0.002	0	Line	Yes
4	0.01	100	Line	Yes

b)

Figure 3.208. Elastic bipolar force element with 4 mm gap; option “**Positive compression**” is off (a) and on (b)

The **Compression positive** option is used for choice of the positive abscissa value on the force law plot. If the option is *not checked* (the default value), abscissa **increases** with the growth of the length (coordinate); in this case the force usually decreases with the growth of abscissa. If the option is checked, on the contrary, abscissa **decreases** with the growth of the length (coordinate), and the force usually **increases** with the growth of abscissa.

To insert data from a text file use the clipboard (Sect. 3.5.4.7).

Abscissa matching

Abscissa matching is often used in the case of a bipolar force element, when the force depends on the element length. Matching means that abscissa value on the plot must correspond to a definite length L of the element, which value should be set in the **L** edit box, Figure 3.207. Often L value is the element length for zero values of coordinates. Two methods can be used for assignment of abscissa value to the element length according to the **Type of abscissa matching** group, Figure 3.207. If the **X-value** option is selected, the abscissa value corresponding to L is directly set in the **X(L)/F(L)** edit box. Often this value is zero, X(L)=0. If the **F-value** option is selected, the force value corresponding to the length L is set, and program compute the corresponding abscissa value automatically. It is clear that in the last case the user must ensure the existence and uniqueness of solution X(F).

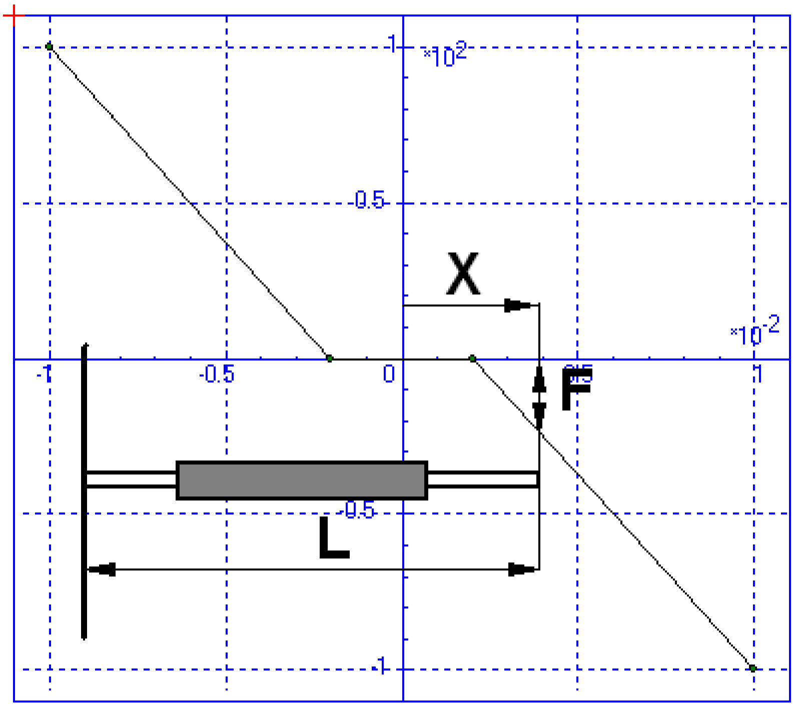


Figure 3.209. To the notion of abscissa matching.
The **Compression positive** option is not checked.

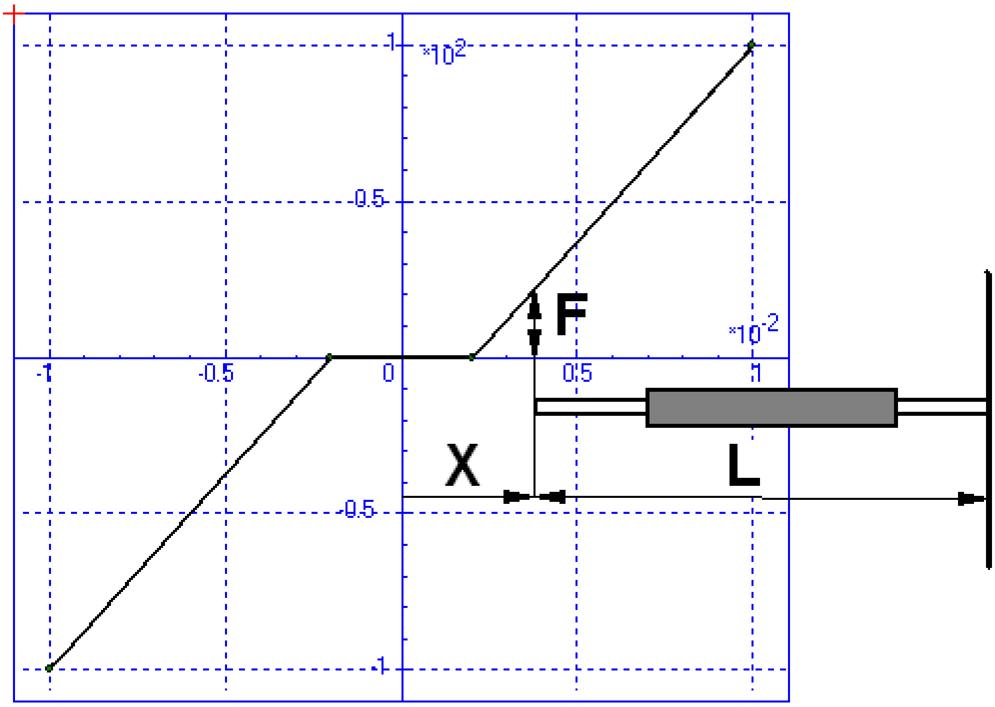
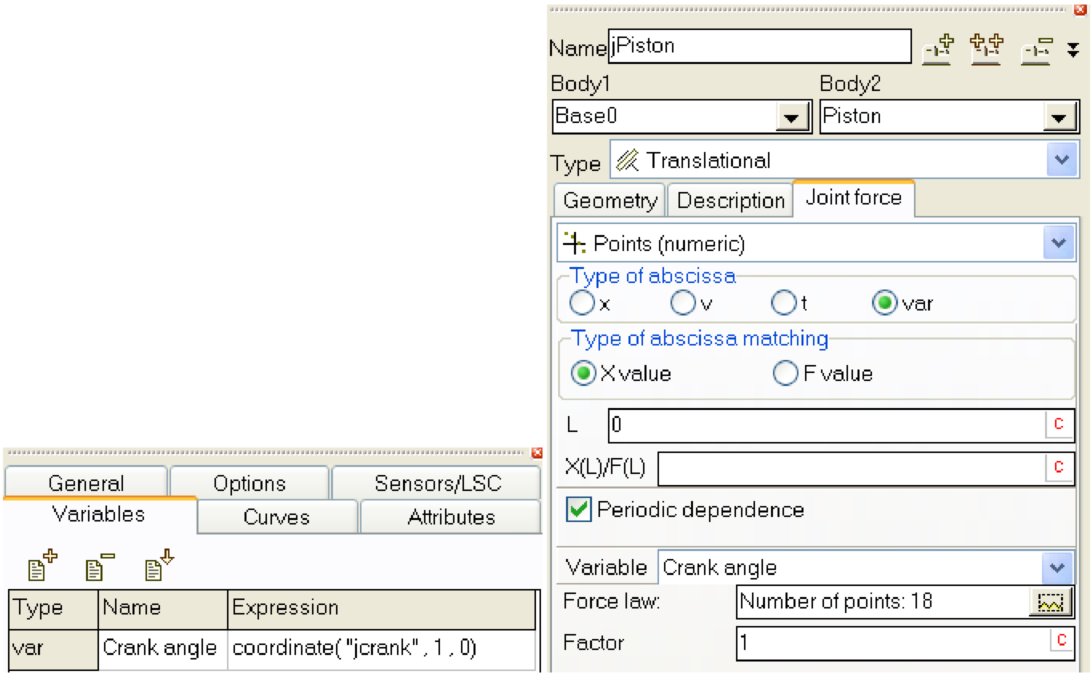


Figure 3.210. To the notion of abscissa matching.
The **Compression positive** option is checked.

3.6.12.2.6.2. Indicator diagram

Dependence of a force on a variable created by the user is used mainly for modeling periodic indicator diagrams; the **Periodic dependence** key must be checked. In the indicator diagram the force does not depend on the current coordinate *x*, but on angle of rotation of another link. For instance, *x* is the joint coordinate of the engine piston, and the force depends on the angle of rotation of a crank, Figure 3.211.



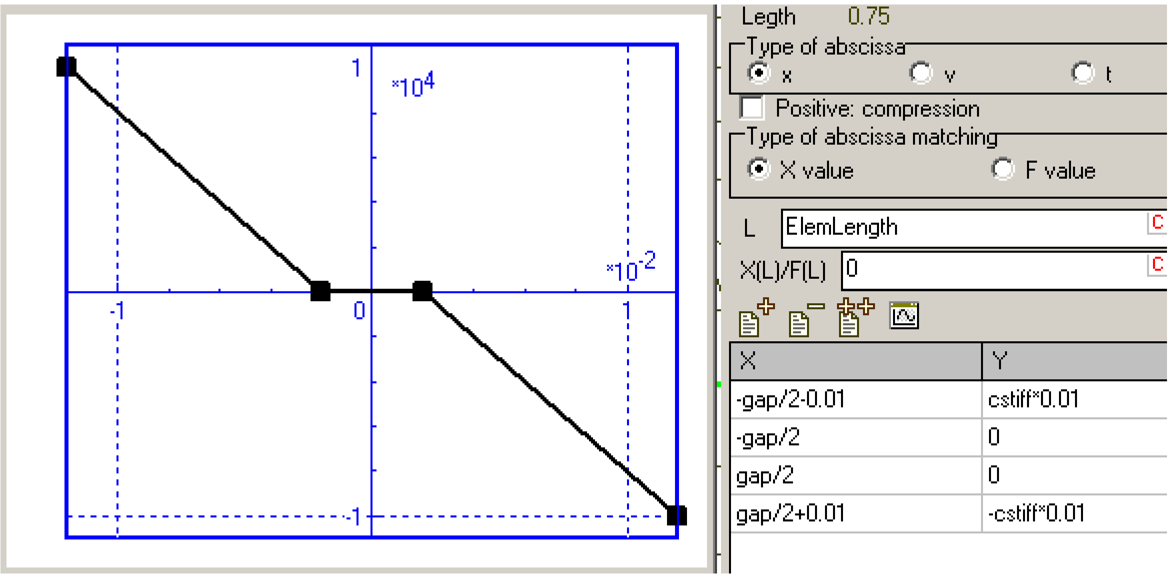
Type	Name	Expression
var	Crank angle	coordinate("jcrank" , 1 , 0)

Name: jPiston
Body1: Base0
Body2: Piston
Type: Translational
Points (numeric)
Type of abscissa: ○ x ○ v ○ t ● var
Type of abscissa matching: ● X value ○ F value
L: 0
X(L)/F(L):
☑ Periodic dependence
Variable: Crank angle
Force law: Number of points: 18
Factor: 1

Figure 3.211. Example of an indicator diagram

3.6.12.2.7. Points (expressions)

General information about force element of this type can be found in Chapter 2, Sect. *Types of scalar forces | Points model*.



Legth 0.75

Type of abscissa ● x ○ v ○ t

□ Positive: compression

Type of abscissa matching ● X value ○ F value

L ElemLength

X(L)/F(L) 0

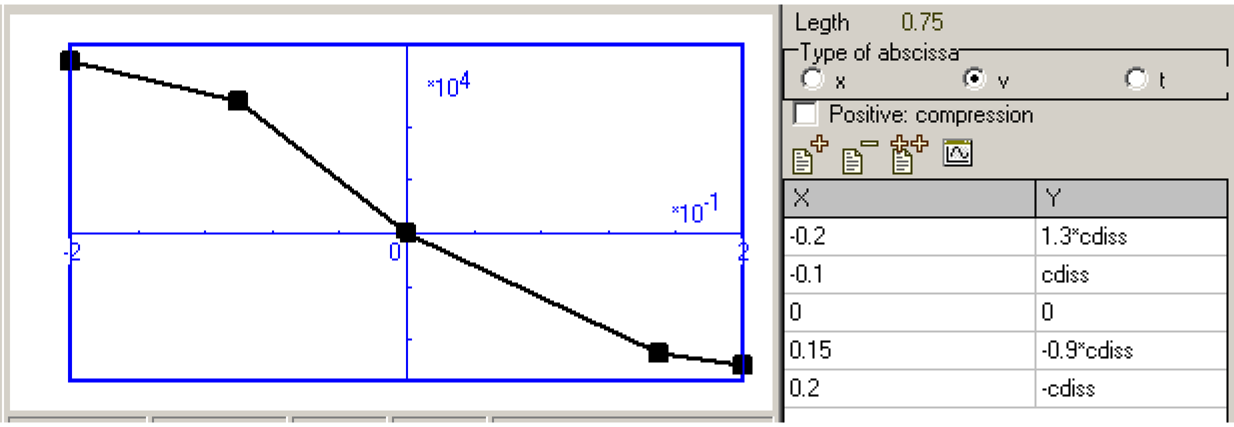
X	Y
-gap/2-0.01	cstiff*0.01
-gap/2	0
gap/2	0
gap/2+0.01	-cstiff*0.01

Figure 3.212. Elastic bipolar force element with a gap, which value is set by the **gap** identifier

The force element is similar to the previous one, but both abscissa and ordinate coordinates of points can be set by expressions.

Use the buttons to add, copy or delete a selected point. The button is used for preview the function in a graphical window.

In the figures of this section we consider a description of the same function as in the previous one but the length of element, clearance, and stiffness are parameterized. Parameterization allows changing these parameters in the simulation.



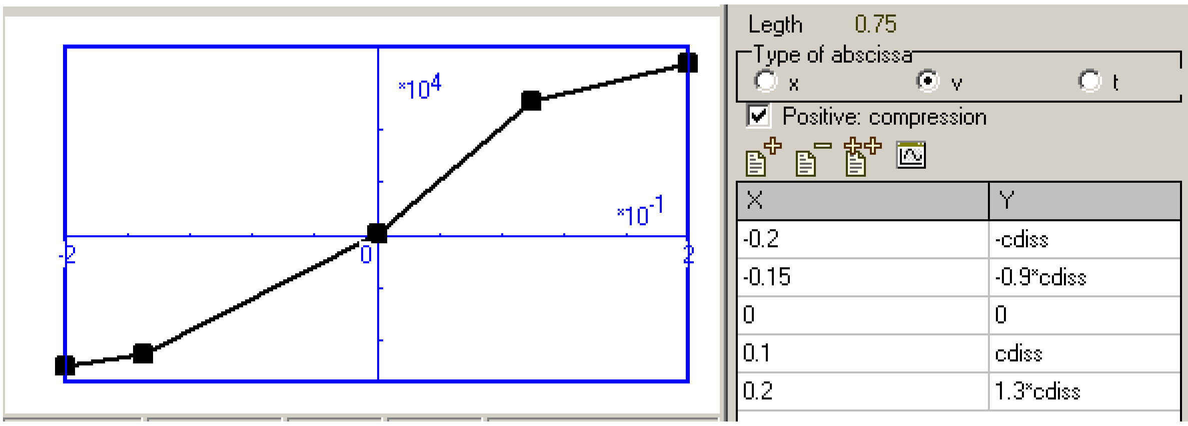
Legth 0.75

Type of abscissa ○ x ● v ○ t

□ Positive: compression

X	Y
-0.2	1.3*cdiss
-0.1	cdiss
0	0
0.15	-0.9*cdiss
0.2	-cdiss

Figure 3.213. Nonlinear damper. The **Compression positive** option is not checked.



Legth 0.75

Type of abscissa
○ x ● v ○ t

☑ Positive: compression

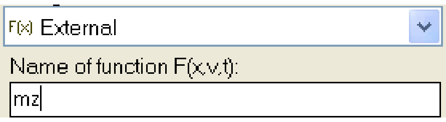
X	Y
-0.2	-cdiss
-0.15	-0.9*cdiss
0	0
0.1	cdiss
0.2	1.3*cdiss

Figure 3.214. Nonlinear damper. The Compression positive option is checked.

3.6.12.2.8. External function

The name of the function must be entered. A template of the function will be automatically inserted in the control file for programming by the user, Sect. 3.5.2.4.10. One function can be used in description of several force elements.

Remark. Using of external function requires the generation of equations of motion in symbolic form, Sect. 3.9.2.



F(x) External

Name of function F(x,v,t):

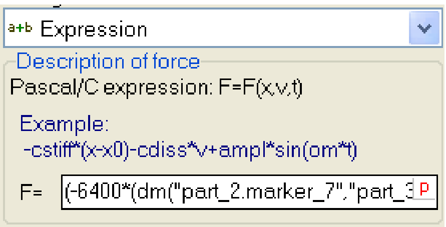
mz

Figure 3.215. Example of external function

3.6.12.2.9. Expression

The force model is described by an explicit expression including

- identifiers,
- x, v, t – length of the element, joint coordinate or angle of rotation (depending on the force type), its time derivative and time,
- kinematic functions for positions and velocities, Sect. 3.5.2.4.7.
- standard functions, Sect. 3.5.2.4.3.



Pascal/C expression: F=F(x,v,t)

Example:
-cstiff*(x-x0)-cdiss*v+ampl*sin(om*t)

F= (-6400*(dm("part_2.marker_7","part_3

Example: a combined linear spring/damper force

-cstiff(x-x0)-cdiss*v*

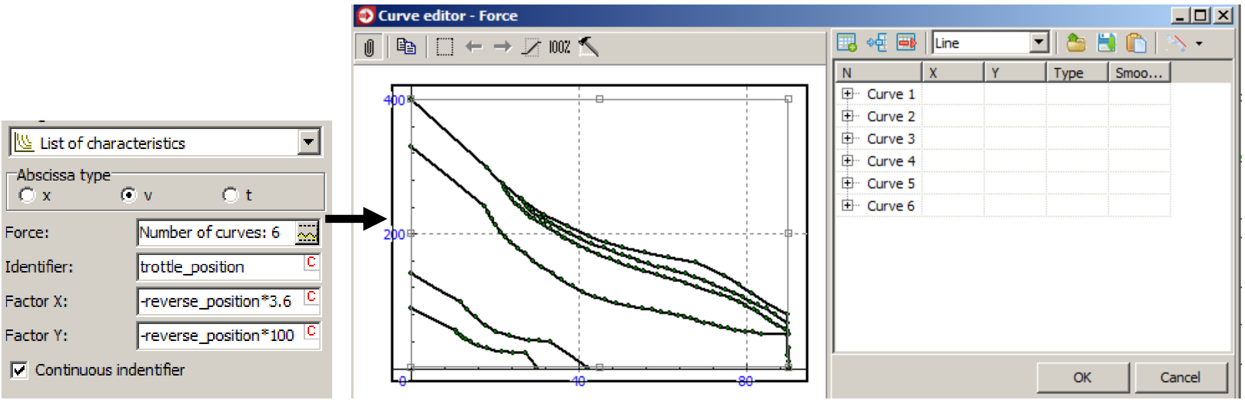
where *cstiff*, *cdiss* are the spring and damper constants, *x0* is the length/coordinate of the unloaded element.

3.6.12.2.10. List of characteristics

Description of the force is located in Chapter 2, Sect. *Types of scalar forces | List of characteristics*.

Sequence of the element description:

- set **Abscissa type** (coordinate, velocity of time);
- open the curve editor by the button and create a list of curves; to start a new curve, double click by the mouse on a big enough distance from other curves and confirm creation of a new curve; please note that curves are numbered in the input order, drag the curve name by the mouse to change in position in the list;
- enter a **Curve identifier**, which value select a characteristic for computation of the force;
- if necessary, specify the scale factors kx, ky for the abscissa and ordinate values;
- set type of the curve identifier: **either discrete or continuous** – checked in the last case.



List of characteristics

Abscissa type
○ x ● v ○ t

Force: Number of curves: 6

Identifier: trottle_position

Factor X: -reverse_position*3.6

Factor Y: -reverse_position*100

☑ Continuous indentifier

N	X	Y	Type	Smoo...
Curve 1				
Curve 2				
Curve 3				
Curve 4				
Curve 5				
Curve 6				

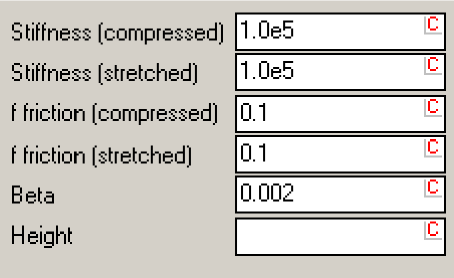
Figure 3.216. Example of a traction force model as a list of characteristics

3.6.12.2.11. Fancher leaf spring

The mathematical model of the element can be found in Chapter 2, Sect. *Fancher leaf spring*.

Force model parameters:
Stiffness (compressed) – the spring vertical stiffness in the compressed state, c;
Stiffness (stretched) – the spring vertical stiffness in the stretched state, c;
f friction (compressed) – the value of friction coefficient in the compressed state, f ;
f friction (stretched) – the value of friction coefficient in the stretched state, f ;
Beta - the exponential suspension parameter, β ;
Height – the height of the spring in the unloaded state x_0.

All the parameters are constant symbolic expressions.



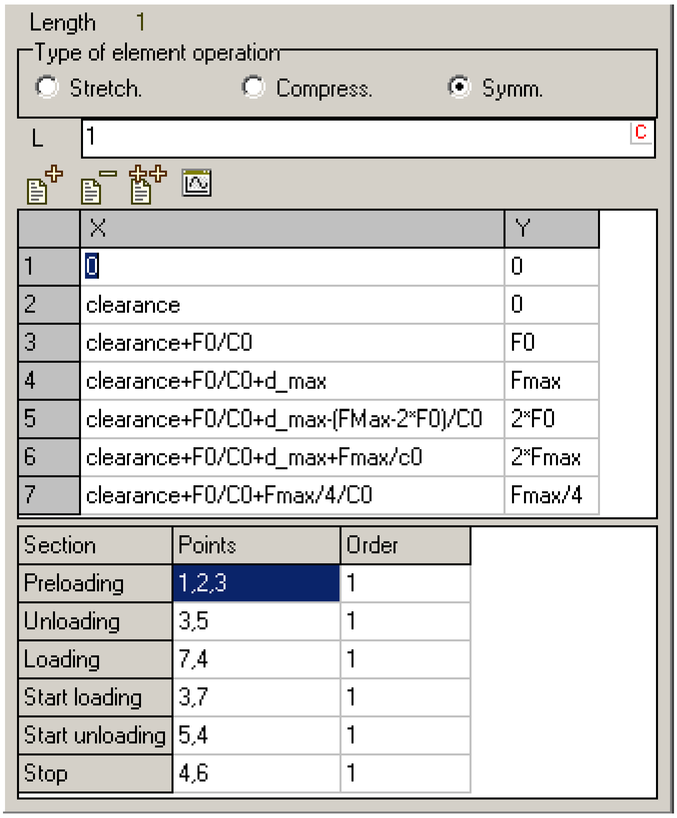
Stiffness (compressed)	1.0e5
Stiffness (stretched)	1.0e5
f friction (compressed)	0.1
f friction (stretched)	0.1
Beta	0.002
Height	

Figure 3.217. Parameters of leaf spring

Remark. The user should remember that bipolar force elements degenerate at zero length. The lengths of the Fancher elements in the model of the leaf spring must be at least two times greater than the maximal dynamic shortening the element even if the real prototype has a less height.

3.6.12.2.12. Hysteresis

Mathematical model of the force is described in Chapter 2, Sect. *Types of scalar forces / Hysteresis*.



Length 1

Type of element operation

○ Stretch. ○ Compress. ● Symm.

L 1

	X	Y
1	0	0
2	clearance	0
3	clearance+F0/C0	F0
4	clearance+F0/C0+d_max	Fmax
5	clearance+F0/C0+d_max-(FMax-2*F0)/C0	2*F0
6	clearance+F0/C0+d_max+Fmax/c0	2*Fmax
7	clearance+F0/C0+Fmax/4/C0	Fmax/4

Section	Points	Order
Preloading	1,2,3	1
Unloading	3,5	1
Loading	7,4	1
Start loading	3,7	1
Start unloading	5,4	1
Stop	4,6	1

Figure 3.218. Example of hysteresis data input window

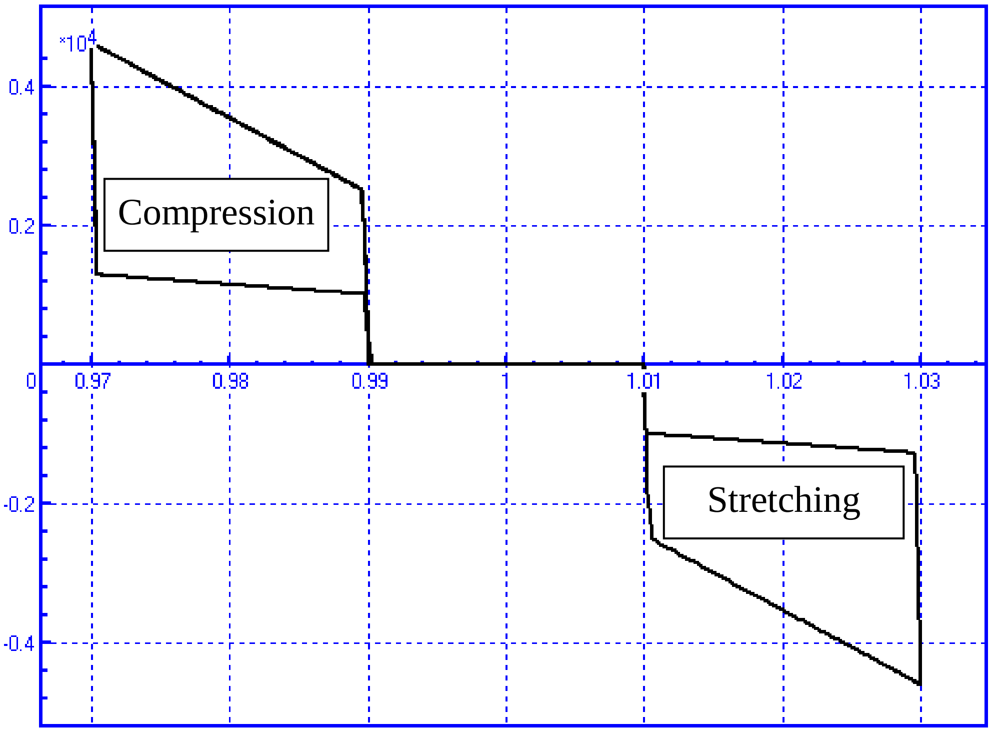


Figure 3.219. Example of hysteretic element with symmetric operation relative to length 1 m

The **Type of element operation** group:

- **stretch(ing)** is the element works only if coordinate value is greater than the value in the **L** box;
- **compress(ion)** is the element works only if coordinate value is less than the value in the **L** box;
- **symm(etric)** is the element works symmetrically both by stretching and by compression.

The **L** box contains the element length/coordinate value corresponding to the zero value of abscissa of points. The buttons of operations with the list of points – add a point, delete selected point, copy selected point, and draw data in the graphic window.

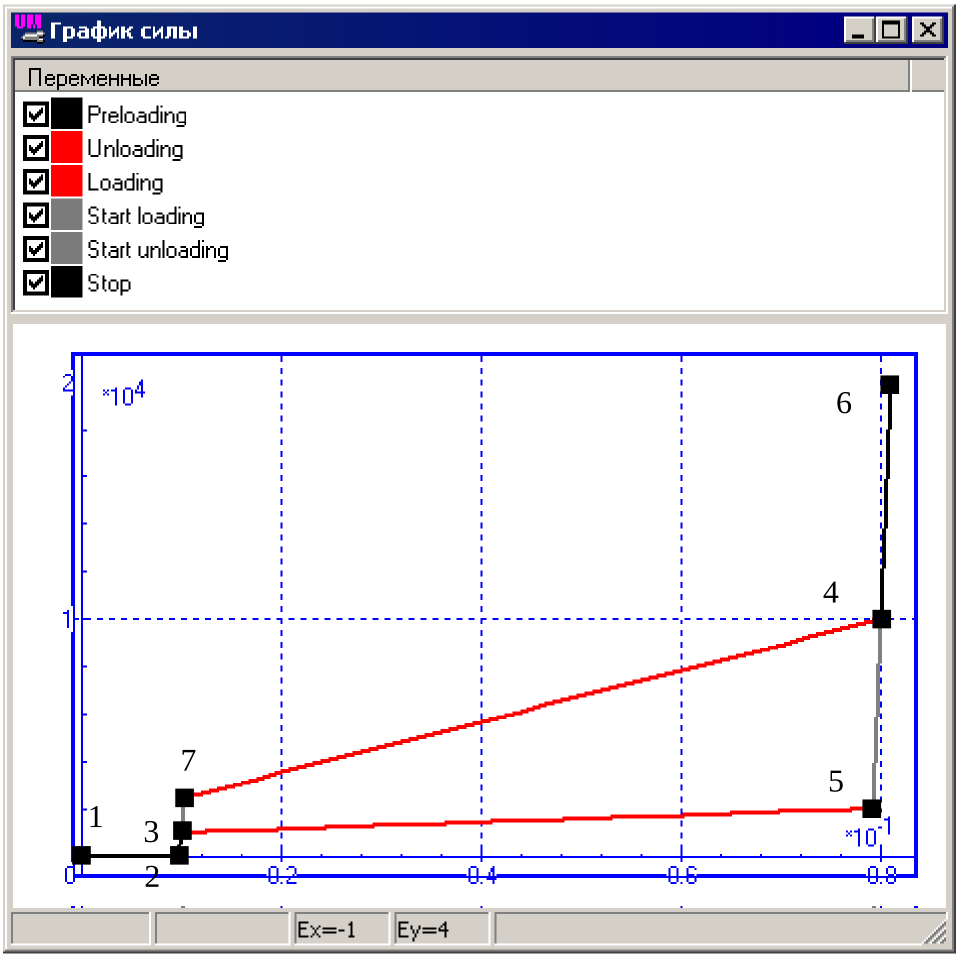


График силы

Переменные

☑ ■ Preloading
☑ 🟥 Unloading
☑ 🟥 Loading
☑ ■ Start loading
☑ ■ Start unloading
☑ ■ Stop

Ex=-1 Ey=4

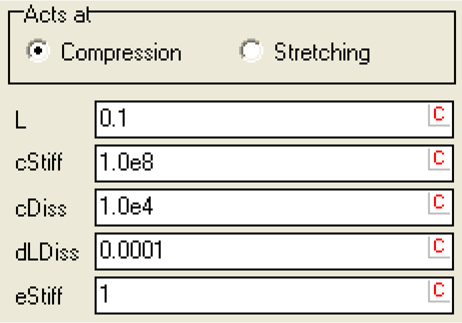
Figure 3.220. Graphic window with hysteresis data

Coordinates of points can be parameterized.

Indices of points starting with 1 separated by commas for hysteretic curves are entered in the lowest table of the data window. Orders of interpolation polynomials are set in the last column of the table (order 1 corresponds to a polyline).

More details about hysteretic element can be found in Chapter 2, Sect. *Types of scalar forces / Hysteresis*.

3.6.12.2.13. Impact (bump stop)



Acts at

● Compression ○ Stretching

L 0.1

cStiff 1.0e8

cDiss 1.0e4

dLDiss 0.0001

eStiff 1

Figure 3.221. Parameters of an impact element

Mathematical model of the force element of this kind is described in Chapter 2, Sect. *Impact*.

The model has the following parameters:

L is the length of the element at zero clearance when force element starts to work, l;

cStiff is a stiffness coefficient in a contact, c;

cDiss is a damping coefficient in a contact, d;

dLDiss is the contact deflection where damping coefficient reaches its maximal value **cDiss**, Δ_d;

eStiff is the force curve exponent, is not used in the current version of UM software, assumed to be 1.

All the parameters are constant symbolic expressions.

This force element can works as a border for compression and stretching modes. Let us consider the compression case. While the length of the force element more or equal to **L** than force is zero. As soon as the length of the element becomes less than **L**, the viscoelastic force starts to act. In the case of the stretching mode the force starts to act if the length of the element exceeds **L**.

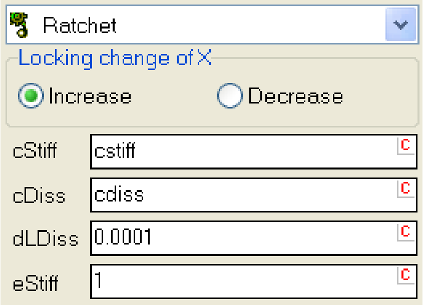
If force element of *Impact* type acts as a joint force, the **L** parameter should be considered as a joint coordinate. Thus, introducing two forces of this type (one for stretching, another one for compression) as joint forces in a joint, we can define limits for the joint coordinate.

Dimensions of parameters for bipolar and joint forces are given in the table below.

Parameters	Dimension	
	Bipolar of joint force along translational degree of freedom	Joint force along rotational degree of freedom
L	m	rad
cStiff	N/m	Nm/rad

cDiss	Ns/m	Nms/rad
dLDiss	m	rad

3.6.12.2.14. Ratchet



Ratchet

Locking change of X

● Increase ○ Decrease

cStiff: cstiff

cDiss: cdiss

dLDiss: 0.0001

eStiff: 1

Figure 3.222. Parameters of a ratchet

Mathematical model of the force element of this kind is described in Chapter 2, Sect. *Ratchet*.

The model has the following parameters:

cStiff is a stiffness coefficient while locking, *c*;

cDiss is a damping coefficient while locking, *d*;

dLDiss is the contact deflection where damping coefficient reaches its maximal value **cDiss**, Δ_d;

eStiff is the force curve exponent, is not used in the current version of UM software, assumed to be 1.

All the parameters are constant symbolic expressions.

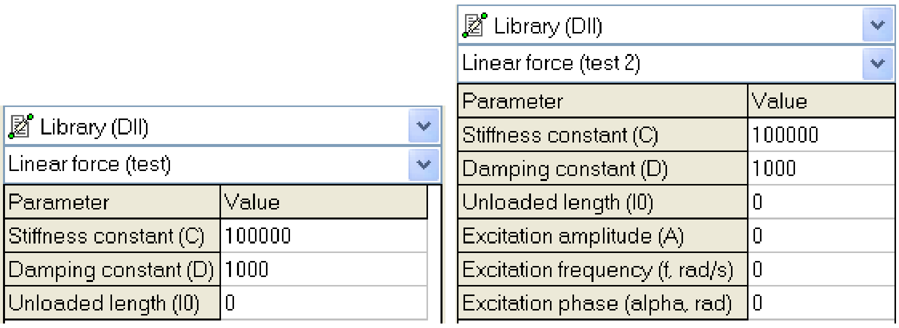
Units of the parameters depend on the unit of the x coordinate, Sect. 3.6.12.2.13.

Models:

{UM Data}\SAMPLES\LIBRARY\Ratchet;

{UM Data}\SAMPLES\LIBRARY\ChainGear.

3.6.12.2.15. Library DLL



a)

Library (Dll)

Linear force (test)

Parameter	Value
Stiffness constant (C)	100000
Damping constant (D)	1000
Unloaded length (l0)	0

b)

Library (Dll)

Linear force (test 2)

Parameter	Value
Stiffness constant (C)	100000
Damping constant (D)	1000
Unloaded length (l0)	0
Excitation amplitude (A)	0
Excitation frequency (f, rad/s)	0
Excitation phase (alpha, rad)	0

Figure 3.223. Examples of scalar forces described with dynamic libraries

The force realizes an access to the models and parameters of scalar forces described in dynamic linked libraries and located in the directory **{UM Data}\lib\bfrc**.

See Chapter 2, Sect. Scalar force of *Library (DLL)* type as well as Chapter 5, Sect. Scalar force of *Library (DLL)* type

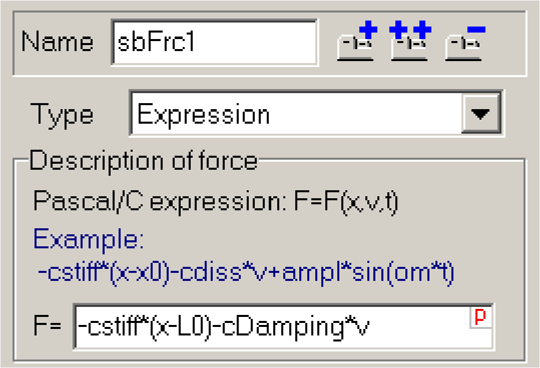
Examples of Delphi projects:

{UM Data}\SAMPLES\LIBRARY\DLL\BfrcSample, Figure 3.223a,

{UM Data}\SAMPLES\LIBRARY\DLL\BfrcSample1, Figure 3.223b.

Parameters of force elements are constant symbolic expressions, Sect. 3.5.2.4.5.

3.6.12.2.16. List of forces



Name sbFrc1

Type Expression

Description of force

Pascal/C expression: F=F(x,v,t)

Example:
-cstiff*(x-x0)-cdiss*v+ampl*sin(om*t)

F= -cstiff*(x-L0)-cDamping*v

Figure 3.224. Example of a list of two forces

This type of force creates an arbitrary set of forces of the above types, which work in parallel. Use the buttons to add, copy or delete a separate force.

3.6.12.3. Input of bipolar force elements

Definition of a bipolar force element can be found in Chapter 2, Sect. *Bipolar forces*.

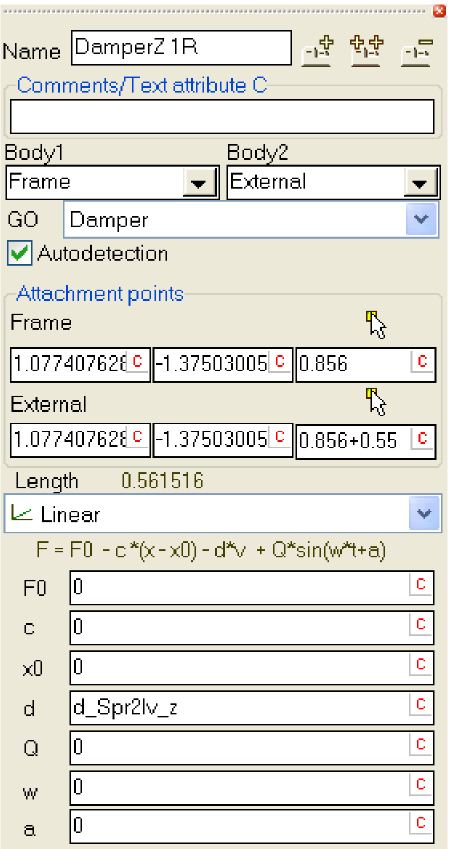
General parameters of a bipolar element are:

- Connected bodies;
- Attachment points (constant symbolic expression);
- Type (use the drop-down list).

Other parameters of the element depend on its type and should be entered in boxes, which appear after choice of the type. Some features of description of the force as an explicit expression can be found in Sect. 3.5.2.4.6, as an external function – in Sect. 3.5.2.4.10.

Mathematical model of a bipolar force includes often the element length (the distance between the attachment points). Use the current *Length* parameter to verify the correctness of description of the element.

A GO is usually assigned to the bipolar force (Sect. 3.6.6).



Name: DamperZ 1R

Comments/Text attribute C

Body1: Frame

Body2: External

GO: Damper

☑ Autodetection

Attachment points

Frame: 1.07740762ε | -1.37503005 | 0.856

External: 1.07740762ε | -1.37503005 | 0.856+0.55

Length 0.561516

Linear

F = F0 - c*(x-x0) - d*v + Q*sin(w*t+a)

F0: 0

c: 0

x0: 0

d: d_Spr2lv_z

Q: 0

w: 0

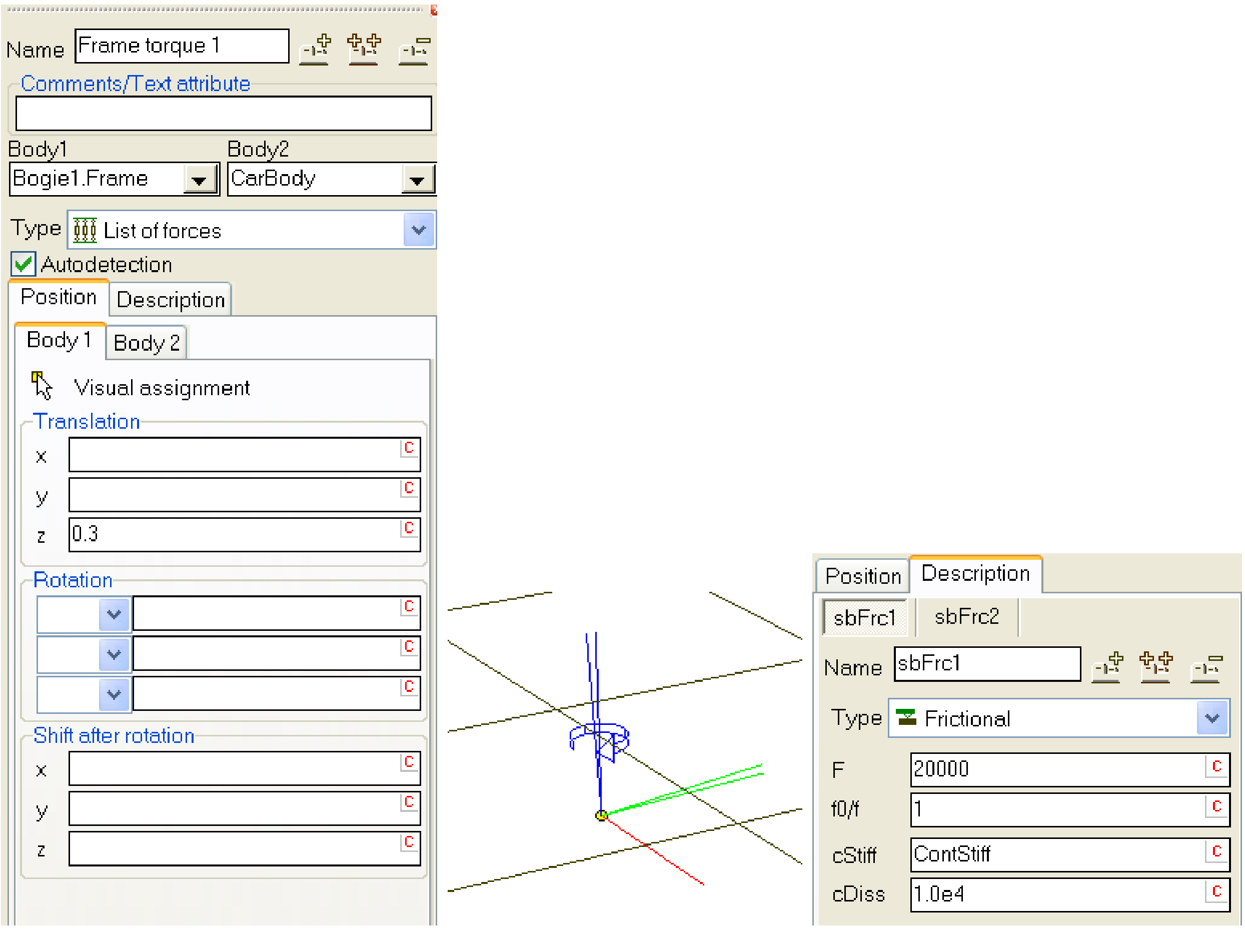
a: 0

Figure 3.225. Bipolar force element

3.6.12.4. Input of scalar torque force element

Mathematic model of the element is described in Chapter 2, Sect. *Scalar torque*. Description of a scalar torques includes the following steps.

- Choice of interacting bodies.
- Setting additional local coordinate systems SCA1 and SCB2, Figure 3.226, left. These systems of coordinates can be assigned visually by the mouse using preliminary created oriented connection points for the necessary bodies, Sect. 3.6.9.6.2. Systems of coordinates are drawn in animation window like in Figure 3.226, middle. If the **Autodetection** mode is selected, position of SCB2 is computed by the program automatically: SCB2 coincides with SCA1 for *zero values* of all coordinates.
- Selection of the torque type from the standard list of scalar forces, see Sect. 3.6.12.2 for more details.



Name Frame torque 1

Body1 Bogie1.Frame

Body2 CarBody

Type List of forces

☑ Autodetection

z 0.3

Name sbFrc1

Type Frictional

F 20000

f0/f 1

cStiff ContStiff

cDiss 1.0e4

Figure 3.226. Scalar torque description

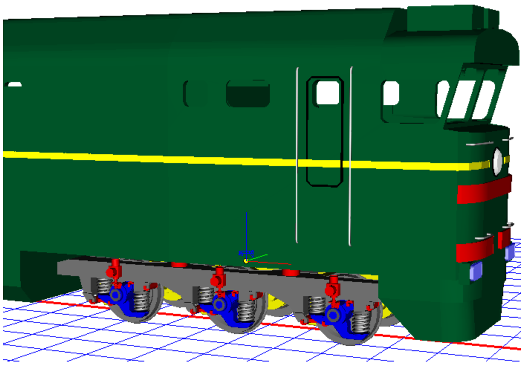
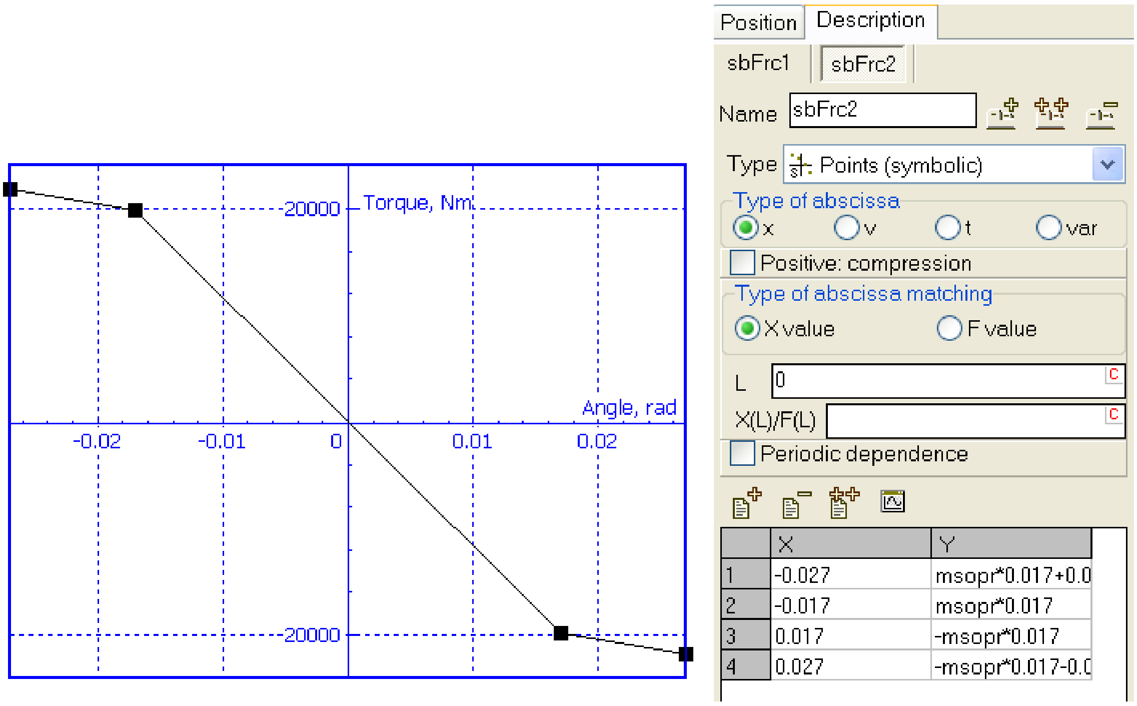


Figure 3.227. Example of scalar torque



sbFrc1 sbFrc2

Name sbFrc2

Type Points (symbolic)

Type of abscissa
● x ○ v ○ t ○ var

□ Positive: compression

Type of abscissa matching
● X value ○ F value

L 0

X(L)/F(L)

□ Periodic dependence

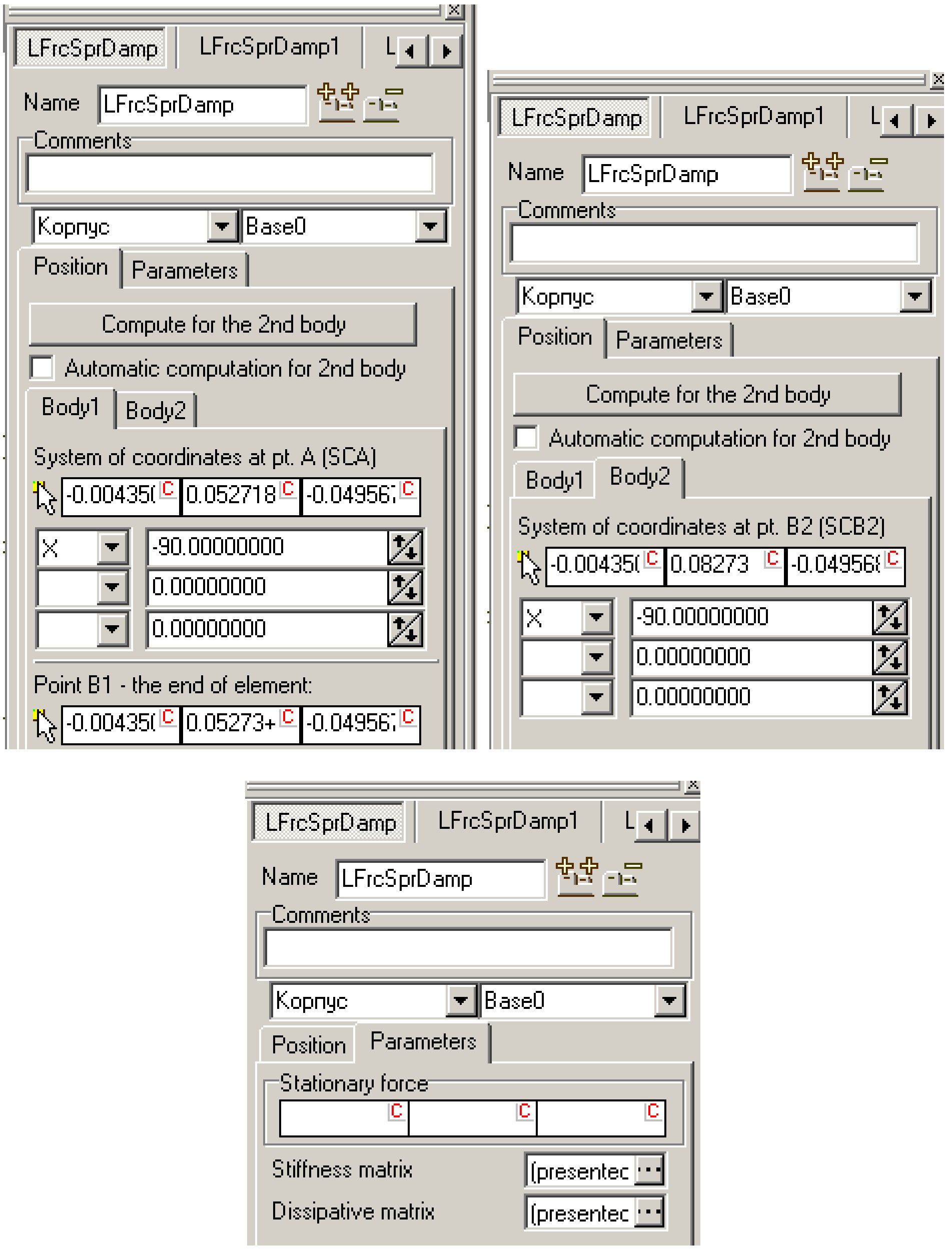
	X	Y
1	-0.027	msopr*0.017+0.0
2	-0.017	msopr*0.017
3	0.017	-msopr*0.017
4	0.027	-msopr*0.017-0.0

Figure 3.228. Nonlinear elastic torque component

Example. Consider an example of usage of the scalar torque. In a model of a locomotive, a scalar torque appears when a car body turns relative to the bogie frame about the vertical axis, Figure 3.227. The torque includes two components. The first one is a friction torque with the magnitude 2000Nm. The second one is a nonlinear elastic torque, which plot is shown in Figure 3.228, left. In this case the torque type is ‘List of forces’. The list of forces includes two elements: a frictional (Figure 3.226, right) and ‘Points symbolic’, Figure 3.228, right.

3.6.12.5. Input of generalized linear force elements

3.6.12.5.1. General information on generalized linear force elements



Name: LFrcSprDamp

Кopпyc | Base0

□ Automatic computation for 2nd body

System of coordinates at pt. A (SCA): -0.00435(| 0.052718 | -0.04956;

X | -90.00000000
 | 0.00000000
 | 0.00000000

Point B1 - the end of element: -0.00435(| 0.05273+ | -0.04956;

System of coordinates at pt. B2 (SCB2): -0.00435(| 0.08273 | -0.04956(

X | -90.00000000
 | 0.00000000
 | 0.00000000

Stiffness matrix: (presentec

Dissipative matrix: (presentec

Figure 3.229. Parameters of Linear force element

Mathematic model of the element is described in Chapter 2, Sect. *Generalized linear force element*.

Examples of description and/or usage:

- Chapter 7, Sect. *Models of Springs*
- model {UM Data}\SAMPLES\Rai_Vehicles\Manchester_Benchmarks\Vehicle1 (module UM Loco is required);
- model {UM Data}\SAMPLES\Rail_Vehicles\wedgetest.

General parameters of a linear element are:

- Adjusted bodies;
- Geometric parameters on the **Position** tab;
- Elements of stiffness and damping matrix (use the button in the corresponding box).
- Stationary force value

A GO is usually assigned to the linear elastic force (Sect. 3.6.6).

Coordinates of attachment points as well as elements of the stiffness/damping matrix are parameterized.

3.6.12.5.2. Some features of description of elastic element

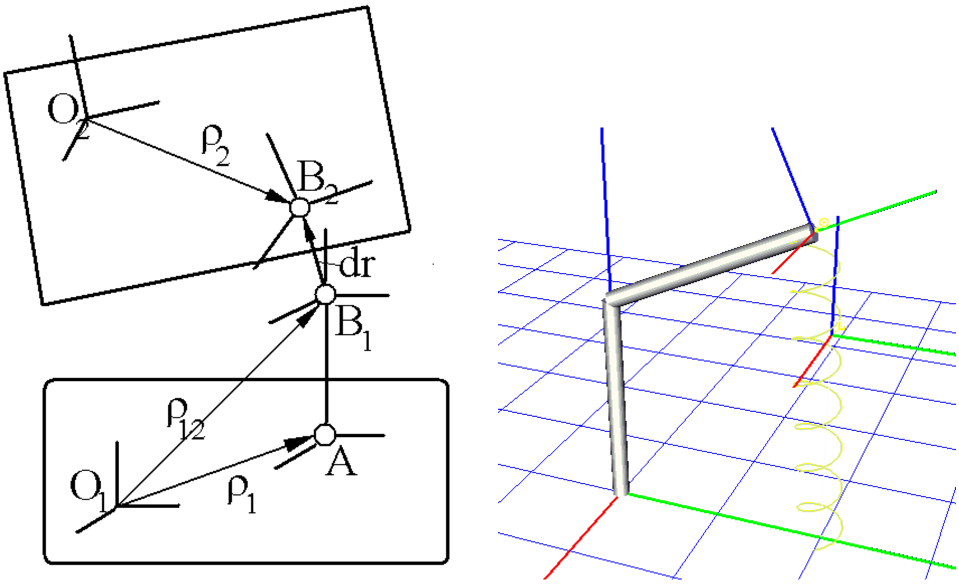


Figure 3.230. Systems of coordinates related to linear force

Some details relevant to the elastic force element can be found in Chapter 2, Sect. *Generalized linear force element*. There can be found notations used in figures as well.

The following parameters should be entered for the first body attached to the element:

- Coordinates of points A, B_1 in SC of the first body;

For the second body (if the option *Automatic computation for 2nd* body is off):

- Coordinates of point B_2 in SC of the second body;
- Orientation of SC connected with point B_2 relative to the SC of the second body (use up to three rotations). If the orientation does not set, the SCB_2 coincides with SC of the second body.

These data are not necessary for the second body if the *Automatic computation for 2nd* body option in on. This option is used exclusively if the object is described in such a way that points B1 and B2 coincide at *zero values of model coordinates*. This case is quite usual for models of railway vehicles.

If you click the *Compute for the second body button*, UM computes coordinates of point B_1 in SC of the 2nd body and inserts these values as coordinates of point B_2 (even if the *Automatic computation for 2nd* body is off).

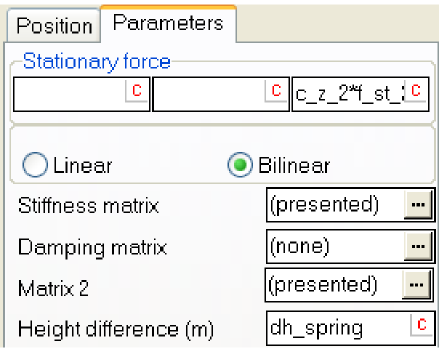
The stationary value of the force acting on the *second* body can be entered, too. The force is resolved in SC of the *first* body.

Coordinates of attachment points, components of stationary force, and elements of the element matrix are parameterized.

Remarks:

1. Points A, B1, B2 as well as systems of coordinates attached to them are visualized in the single element mode (Sect. 3.5.1.2.2). The icon marks point B_1.
2. These points are visualized in the whole object mode if the corresponding option is chosen (Sect. 3.5.1.2.2). Click the icon to call the window with element parameters.
3. Use general and oriented *connection points* (Sect. 3.6.9.6) to set points A, B_1, B_2 together with the attached SC. The connection points must be preliminary added to the corresponding body. The buttons start the mode of visual selection of connection points.

3.6.12.5.3. Bilinear force element



Stationary force: | | c_z_2*f_st_

○ Linear ● Bilinear

Stiffness matrix (presented)

Damping matrix (none)

Matrix 2 (presented)

Height difference (m) dh_spring

Figure 3.231. Parameters of bilinear force element

A bilinear force element consists of two springs. *The second spring is shorter than the first one*.

Additional parameters are related to the second spring

- stiffness matrix of the shorter second spring **Matrix 2**,
- **difference** in the heights of the springs in meters.

The shorter spring begins to act when the compression of the first spring including the static deflection is greater than the height difference. Otherwise, the force from the second spring is zero.

3.6.12.6. Input of contact force elements

Use the **Contact forces** tab for description of the list of contact forces (Chapter 2, Sect. *Contact forces*).

Types of a contact force element:

- **Points-Plane;**
- **Point-Curve;**
- **Circle-Cylinder (with curved axis);**
- **Sphere-Plane;**
- **Circle-Plane;**
- **Sphere-Sphere;**
- **Points-Z surface;**
- **Circle-Z surface;**
- **Sphere-Z surface.**

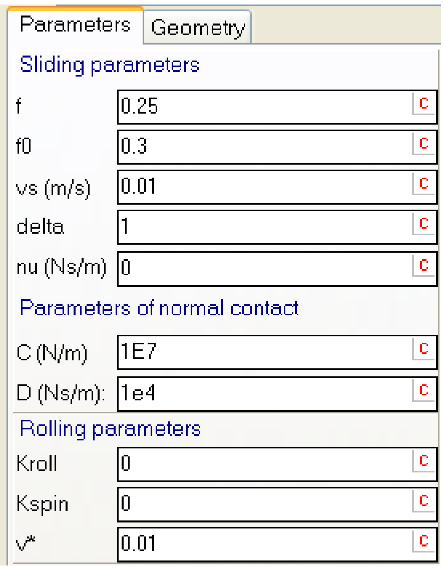
Remark. Changing the type deletes all previous description of the element.

The first interacting body always contains the first type of contact manifold in the name of the contact. For instance, the first body contains *points* and the second one – a *plane* in the case of the **Points-Plane** contact.

There exist two tabs for description of an element:

- The **Parameters** tab contains **some** general contact parameters such as static and dynamic friction coefficient, contact stiffness and damping coefficient etc.;
- The **Geometry** tab contains **parameters** depending of the element type.

3.6.12.6.1. Coefficient of friction, stiffness, damping at contact



Sliding parameters

f	0.25
f0	0.3
vs (m/s)	0.01
delta	1
nu (Ns/m)	0

Parameters of normal contact

C (N/m)	1E7
D (Ns/m):	1e4

Rolling parameters

Kroll	0
Kspin	0
v*	0.01

Figure 3.232. Parameters of contact friction

Model of friction is considered in Chapter 2, Sect. *Contact forces* | *Dependence of coefficient of friction on sliding velocity*.

Independently on the type of contact interaction, the coefficient of friction is specified on the **Parameter** tab:

- **f** is the coefficient of friction for infinite sliding velocity f_∞;
- **f0** is the coefficient of friction for zero sliding velocity f_0 (static coefficient of friction)
- **vs** is the Stribeck velocity;
- **delta** is the empirical exponent depending of materials, $\delta \in [0.5, 1]$;
- **nu** is the viscous friction constant;

Additional parameters relates to the rolling contact (not used for contacts of points):

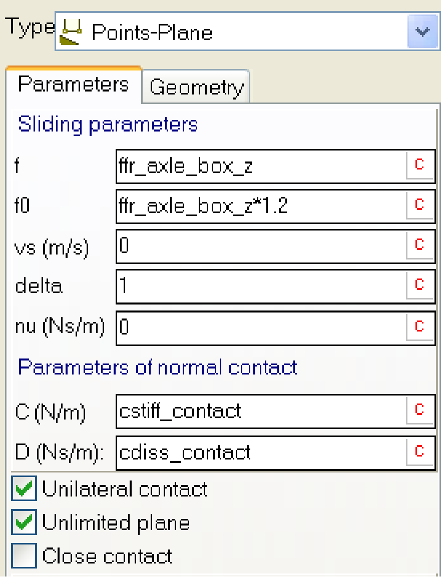
- **Kroll** (m) is the rolling coefficient of friction;
- **Kspin** (m) is the coefficient of pivoting friction;
- **v*** is the regulatory sliding velocity (see Charter 2, Sect. *Contact forces* | *Other types of contact forces*).

Two constants are used for computation of the normal force in dependence on penetration distance and velocity

- C is the stiffness constants;
- D is the damping constants.

Detailed information about rational choice of the contact stiffness and damping constants can be found in Chapter 2, Sect. *Methodology of choice of contact parameters*.

3.6.12.6.2. Points-Plane contact



Type: Points-Plane

Parameters | Geometry

Sliding parameters

f	ffr_axle_box_z
f0	ffr_axle_box_z*1.2
vs (m/s)	0
delta	1
nu (Ns/m)	0

Parameters of normal contact

C (N/m)	cstiff_contact
D (Ns/m):	cdiss_contact

☑ Unilateral contact

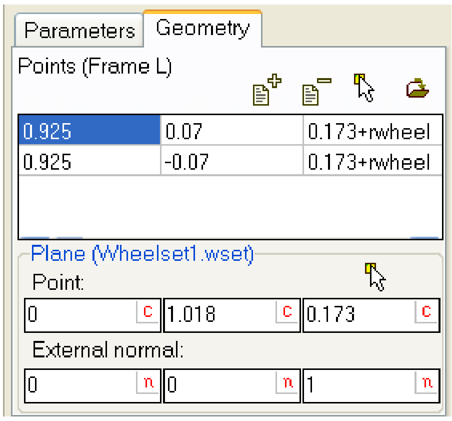
☑ Unlimited plane

☐ Close contact

Figure 3.233. Parameters of 'points-plane' contact force

Contact points belong to the first body, and the plane to the second one.

3.6.12.6.2.1. Contact geometry



Parameters | Geometry

Points (Frame L)

0.925	0.07	0.173+rwheel
0.925	-0.07	0.173+rwheel

Plane (Wheelset1.wset)

Point:

0	1.018	0.173

External normal:

0	0	1

Figure 3.234. Geometry of points-plane contact

Parameters on the Geometry tab describe points and plane data, Figure 3.234.

The plane belongs to the second body; it is described by the following parameters:

- a **point** is any point on the contact plane specified by its coordinates in the SC of the second body;
- **external normal** is the vector of the normal to the plane; in the case of an unilateral contact, the vector in directed outside of the contact.

In the single element mode of the animation window (Sect. 3.5.1.2.2), the normal vector is to verify the correctness of its direction.

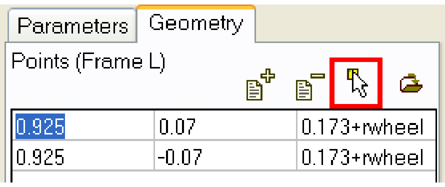
There is three methods for entering contact points:

- by keyboard on the **Geometry** tab, the buttons ,
- visually by the body image, the button ,
- from a text file, the button .

Visual input of contact points

The visual input is available if the mode of element selection by the mouse as available, i.e. if the button on the tool panel of the animation window is in the 'down' state.

- It is recommended to define for the body a set of connection point corresponding to the contact points, Sect. 3.6.9.6.



0.925	0.07	0.173+rwheel
0.925	-0.07	0.173+rwheel

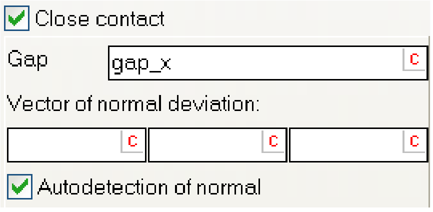
Figure 3.235. This button starts visual mode

- Start the mode of visual selection of points by the button on the inspector, Figure 3.235.
- Use the body image in the animation window and the left mouse button to select the contact points and their adding to the list in the **Geometry** tab. Any number of contact points can be selected both with the help of the preliminary created connection points, and by a simple click on the body image.
- Click again the button to break the mode.

Input contact points from a test file

A set of contact points can be read from a test file by the button in the inspector. The file must contain three columns of x,y,z coordinates of contact points.

3.6.12.6.2.2. Close contact


☑ Close contact

Gap: gap_x

Vector of normal deviation:

☑ Autodetection of normal

Figure 3.236. Parameters of close contact

Check the **Close contact** key for autodetection of the plane parameters such as the plane normal and point, Figure 3.236.

In the autodetection mode, the first contact point as considered as the point on the plane. The autodetection of the normal requires

- at least three contact points; the plane passes through the first three contact points for zero values of object coordinates, and the normal direction is detected by the right-hand screw rule; change the sequence of points in the list to get an opposite direction of the normal;
- checked the key **Autodetection of normal**.

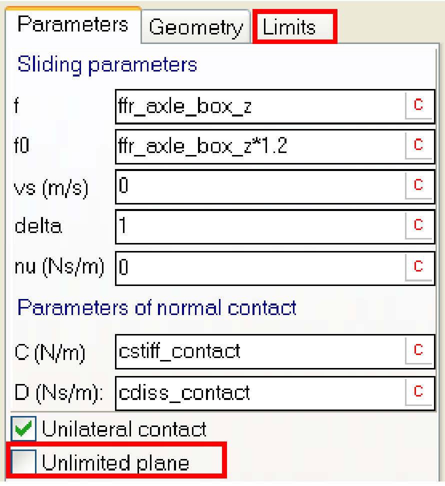
In the close contact mode, a gap and a deviation of the normal from the detected position can be specified for zero coordinates of the object.

3.6.12.6.2.3. Unilateral and bilateral contact

In the **unilateral** mode, the contact force appears if the contact point penetrates the contact plane in the direction opposite to the plane normal, otherwise the force is zero.

In the **bilateral** mode, the force appears by deviation of the point in both directions on a distance, which is greater than the **gap** specified by the user. Therefore, if the gap is non-zero, in fact, two contact planes are presented with the distance between them equal to the double gap. In a particular case, the gap can be zero.

3.6.12.6.2.4. Limited contact plane



Sliding parameters

f	ffr_axle_box_z
f0	ffr_axle_box_z*1.2
vs (m/s)	0
delta	1
nu (Ns/m)	0

Parameters of normal contact

C (N/m)	cstiff_contact
D (Ns/m):	cdiss_contact

☑ Unilateral contact

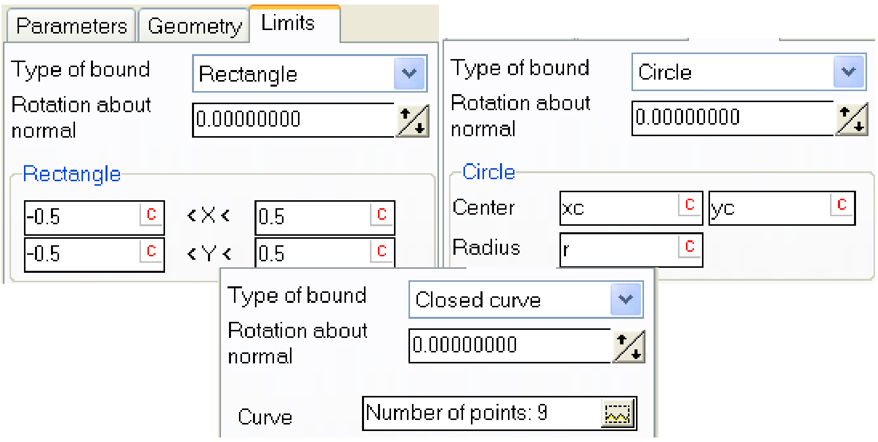
□ Unlimited plane

Figure 3.237. Mode of limited contact plane

The checked Unlimited plane key states that the contact plane has no limits. Otherwise, an additional tab Limits appears (Figure 3.237), which allows the user to describe the geometry of the contact area on the plane by its bounds as

- a rectangle,
- a circle,
- a curve specified by a set of points with a possible smoothing by splines and circle arcs.

The bounds of the contact area are drawn in the animation window in the single element modem Sect. 3.5.1.2.2, which allows the user to control the correctness of data. If necessary, the curve can be turned in the plane on a definite angle in degrees according to the value in the **Rotation about normal** box.



Type of bound: Rectangle

Rotation about normal: 0.00000000

Rectangle

-0.5 < X < 0.5

-0.5 < Y < 0.5

Type of bound: Circle

Rotation about normal: 0.00000000

Circle

Center: xc, yc

Radius: r

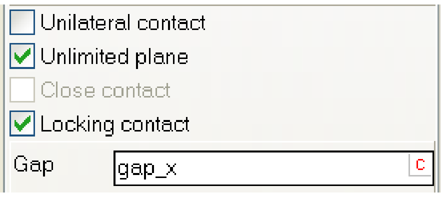
Type of bound: Closed curve

Rotation about normal: 0.00000000

Curve: Number of points: 9

Figure 3.238. Different bounds of the contact area

3.6.12.6.2.5. Locking contact



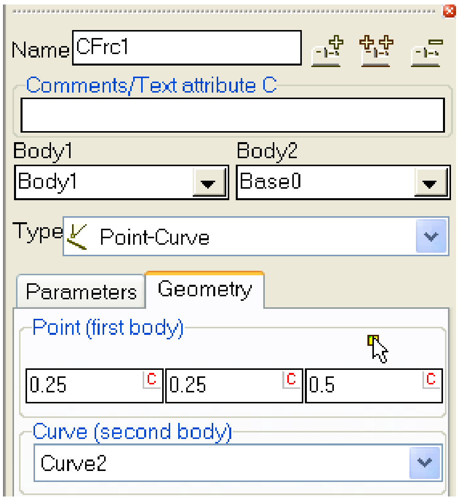
□ Unilateral contact
☑ Unlimited plane
□ Close contact
☑ Locking contact
Gap gap_x

Figure 3.239. The locking mode of bilateral contact

In the case of a bilateral contact, a **locking contact mode** is available, Figure 3.239. At the begin of the simulation the contact force is zero if the contact point is located in the positive part of the half-space defined by the contact plane and the normal to it, like in the case of an *unilateral contact*. By the first penetration of the contact point into the plane, the contact become a bilateral one with a possible gap between the contact planes, i.e. the contact is locked up to the end of the simulation.

The most frequent example of use: a collision of rail vehicles with the following locking the automatic couplers.

3.6.12.6.3. Point-Curve contact



Name CFrc1

Comments/Text attribute C

Body1: Body1

Body2: Base0

Type: Point-Curve

Point (first body): 0.25 | 0.25 | 0.5

Curve (second body): Curve2

Figure 3.240. Parameters of Point-Curve contact

The mathematical model of the curve-plane interaction is described in Chapter 2, Sect. *Force elements | Contact forces| Point-Curve contact*.

Body with the contact points must be assigned as the *first* one, and the curve as connected with the *second* body.

The following parameters are set on the **Geometry** tab, Figure 3.240:

- coordinates of the contact point, which can be assigned visually by the ⇱ button; the corresponding connection point must be preliminary assigned to the body, Sect. 3.6.9.6;
- preliminary created curve from the list of 3D curves, Sect. 3.6.7.

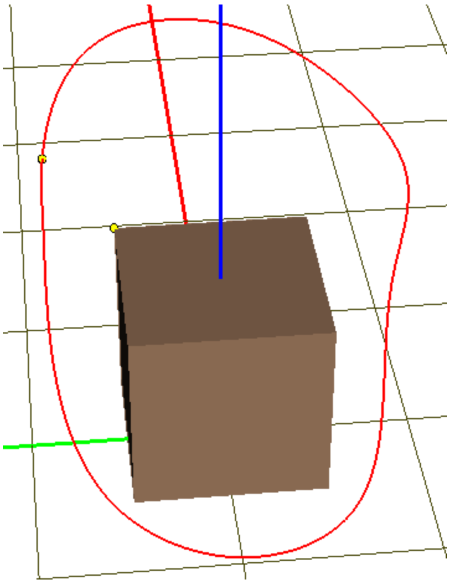
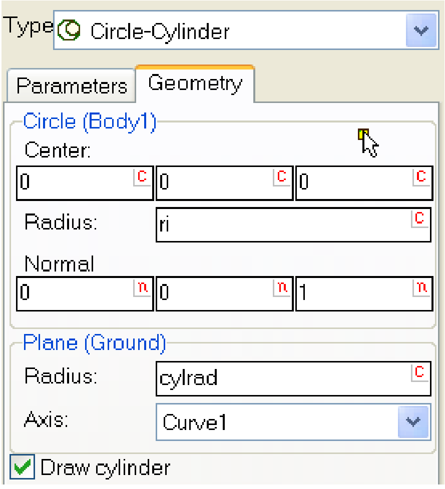


Figure 3.241. Curve in the animation window

The contact curve is drawn automatically in the animation window, Figure 3.241. In the single element mode of the animation window, the contact point as well as the nearest point to the contact on the curve are drawn.

Remark. It is not necessary that the contact point lies on the curve in input module. It must be done in the simulation module by a special tool during assignment of initial coordinates.

3.6.12.6.4. Contact of circle with cylinder (curved axis)



Type: Circle-Cylinder

Circle (Body1)

Center: 0 0 0

Radius: ri

Normal: 0 0 1

Plane (Ground)

Radius: cylrad

Axis: Curve1

☑ Draw cylinder

Figure 3.242. Parameters of Circle-Cylinder contact

The mathematical model of the curve-plane interaction is described in Chapter 2, Sect. *Force elements* | *Contact forces*| *Contact of circle with cylinder with curved axis*.

Body with the circle must be assigned as the first one, and the cylinder as connected with the *second* body.

The following parameters are set on the **Geometry** tab, Figure 3.240:

- coordinates of the circle center point in SC of the first body, which can be assigned visually by the ↖ button; the corresponding connection point must be preliminary assigned to the body, Sect. 3.6.9.6;
- radius of the circle and the normal to the circle plane;
- radius of the cylinder, which must be greater than the circle radius;
- preliminary created curve from the list of 3D curves, Sect. 3.6.7;

After assignment of parameters, the cylinder is automatically drawn in the animation window, if the **Draw cylinder** option is checked, Figure 3.243. In the single element mode of the animation window, the circle and the normal are drawn as well.

The automatically generated image of the cylinder consists of a series of circles perpendicular to the axis. Centers of the circles lie of the axis curve. Number of circles is specified by the user as the **Number of points in plot** parameter, Figure 3.244.

If contacts of several circles with the same cylinder are modeled, the **Draw cylinder** key is recommended to be unchecked to avoid the multiple drawing the cylinder.

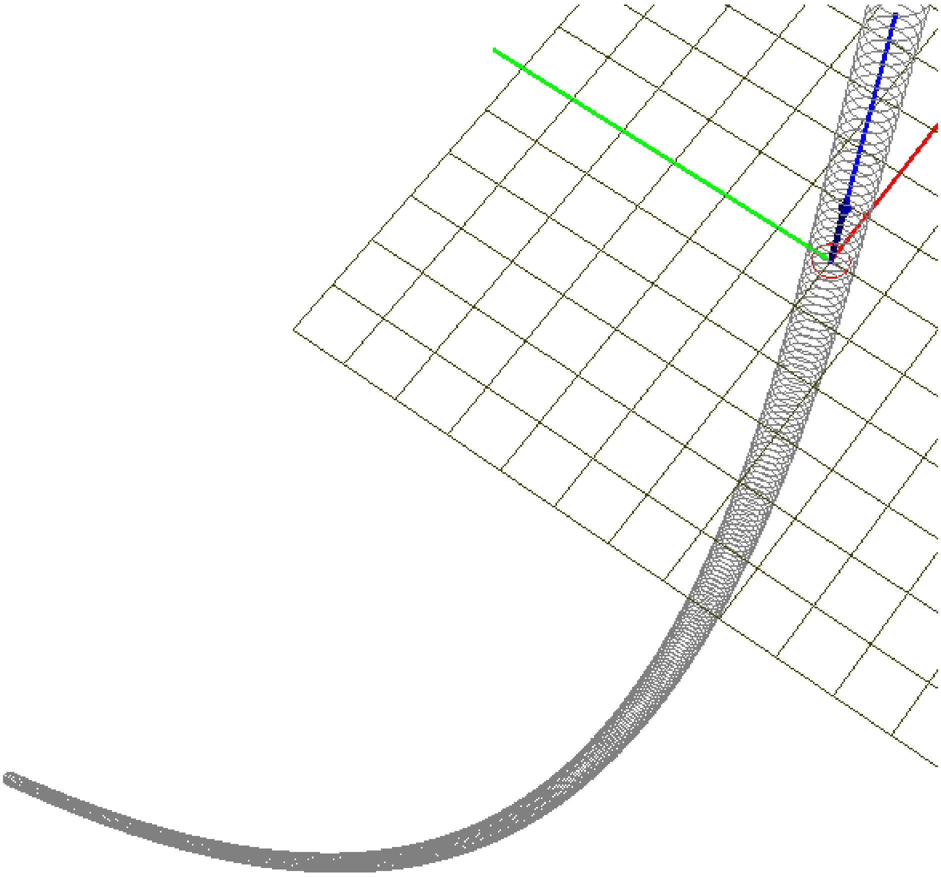
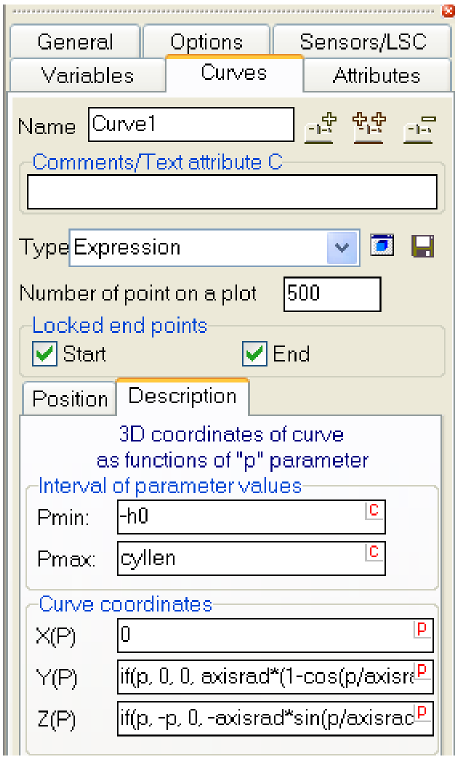


Figure 3.243. Cylinder image



Name Curve1

Comments/Text attribute C

Type Expression

Number of point on a plot 500

Locked end points

☑ Start ☑ End

3D coordinates of curve as functions of "p" parameter

Interval of parameter values

Pmin: -h0

Pmax: cyllen

Curve coordinates

X(P) 0

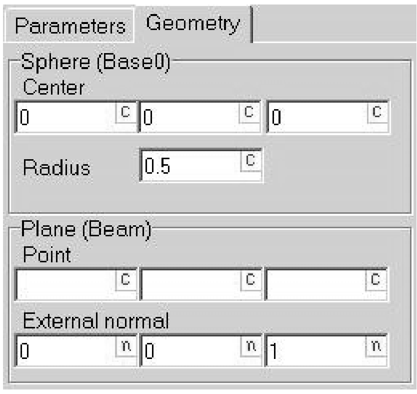
Y(P) if(p, 0, 0, axisrad*(1-cos(p/axisra

Z(P) if(p, -p, 0, -axisrad*sin(p/axisrac

Figure 3.244. Description of the curve, which is the axis of cylinder in Figure 3.243

Model: {UM Data}\SAMPLES\LIBRARY\CylCircle.

3.6.12.6.5. Sphere-Plane contact



Sphere (Base0)

Center: 0, 0, 0

Radius: 0.5

Plane (Beam)

Point

External normal: 0, 0, 1

Figure 3.245. Geometric parameters of sphere-plane contact

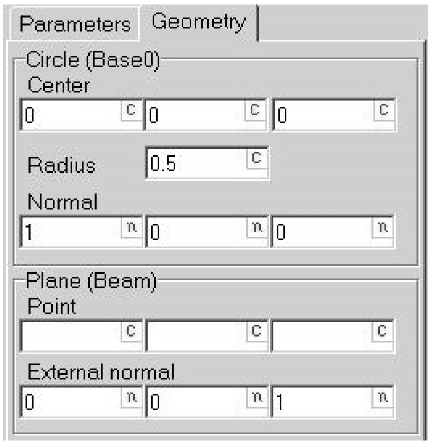
The contact sphere corresponds to the first body. The sphere is described by

- **Center** (a point in SC of the first body);
- **Radius**.

These data are parameterized.

The plane parameters are described in Sect. 3.6.12.6.2.1.

3.6.12.6.6. Circle-Plane contact



Circle (Base0)

Center: 0, 0, 0

Radius: 0.5

Normal: 1, 0, 0

Plane (Beam)

Point: , ,

External normal: 0, 0, 1

Figure 3.246. Geometric parameters of circle-plane contact

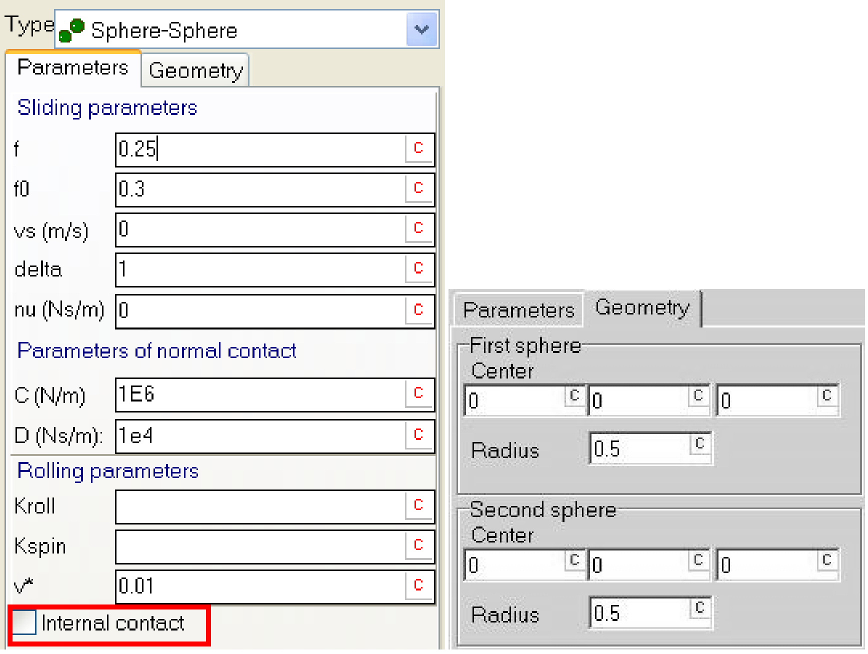
The contact circle belongs to the first body. The circle is described by

- **Center** (a point in SC of the first body);
- **Radius**;
- **Normal** to the circle plane (in SC of the first body).

The plane parameters are described in Sect. 3.6.12.6.2.1.

All the data except the normals are parameterized.

3.6.12.6.7. Sphere-Sphere contact



Type: Sphere-Sphere

Sliding parameters

f	0.25
f0	0.3
vs (m/s)	0
delta	1
nu (Ns/m)	0

Parameters of normal contact

C (N/m)	1E6
D (Ns/m):	1e4

Rolling parameters

Kroll	
Kspin	
v*	0.01

□ Internal contact

First sphere

Center: 0, 0, 0

Radius: 0.5

Second sphere

Center: 0, 0, 0

Radius: 0.5

Figure 3.247. Parameters of sphere-sphere contact

The element description contains parameters for two body-fixed spheres

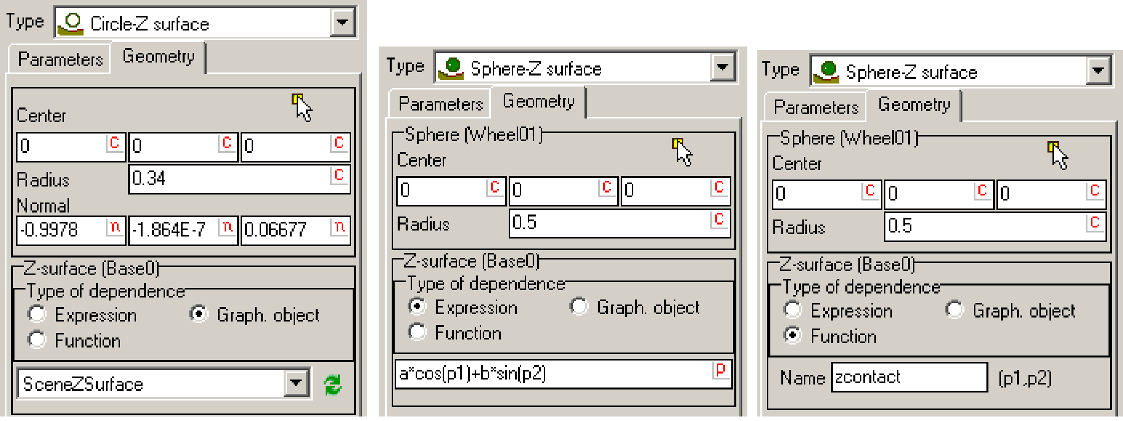
- **Center** (a point in SC of the corresponding body);
- **Radius**.

These data are parameterized.

There exist two types of the contact, Figure 3.247:

- one sphere is **outside** the another one (the internal contact key is not checked)
- one sphere is **inside** the another one (the internal contact key is checked)

3.6.12.6.8. Points | Sphere | Circle - Z surface contact



Type: Circle-Z surface
Parameters | Geometry
Center: 0, 0, 0
Radius: 0.34
Normal: -0.9978, -1.864E-7, 0.06677
Z-surface (Base0)
Type of dependence: ○ Expression ● Graph. object ○ Function
SceneZSurface

Type: Sphere-Z surface
Parameters | Geometry
Sphere (Wheel01)
Center: 0, 0, 0
Radius: 0.5
Z-surface (Base0)
Type of dependence: ● Expression ○ Graph. object ○ Function
a*cos(p1)+b*sin(p2)

Type: Sphere-Z surface
Parameters | Geometry
Sphere (Wheel01)
Center: 0, 0, 0
Radius: 0.5
Z-surface (Base0)
Type of dependence: ○ Expression ○ Graph. object ● Function
Name: zcontact (p1,p2)

Figure 3.248. Parameters of contacts with a Z surface

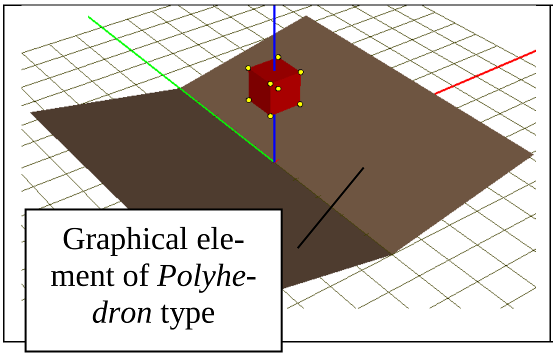
The contact points / sphere / circle belong to the first body.

A **points-surface** contact element includes any number of contact points like in Sect. 3.6.12.6.2.1. No bilateral, locking a close contact modes are available for this type of interaction.

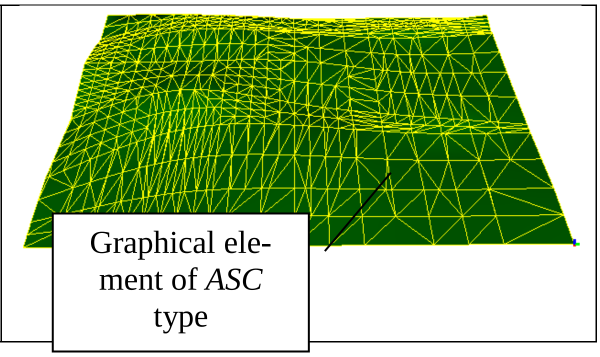
Sphere is described with the help of its center and radius. Circle is described with the help of its center, radius and normal to its plane.

A **Z-surface** belongs to the second body. The surface function corresponds to the second body and can be described by

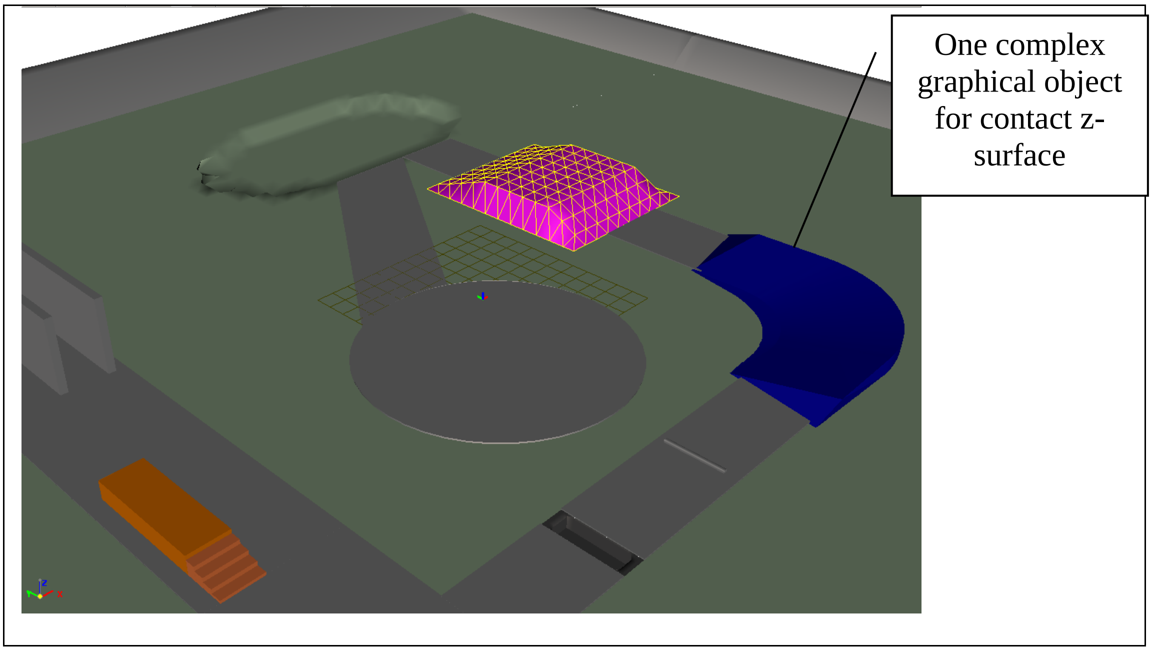
- An explicit **expression** *z = z(p1, p2),* $p_1, p_2 \in (-\infty,+\infty)$;
- An external **function** *z = z(p1, p2),* which should be programmed in the *Control file* (Sect. 3.5.2.4.10).
- A **Graph**[ical] **object** which is selected in the drop down list of graphical objects in the model.



Graphical element of *Polyhedron* type



Graphical element of *ASC* type



One complex graphical object for contact z-surface

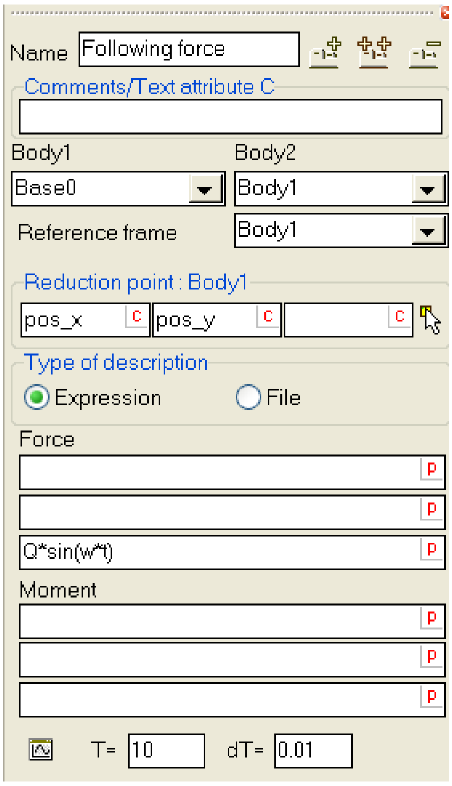
Figure 3.249. Examples of graphical objects for z-surface

Please note the following things using z-surface as a **Graph**[ical] **object**.

- Selected graphical object can include one or several graphical elements of Polyhedron, **ASC**[2], **Box, Ellipsoid, Cone, Z-surface**.
- In the case if there are several possible z-coordinates for any (*x, y*) point the biggest *z*-coordinate will be considered.
- It is desirable that graphical object be smooth enough. Using essentially non-smooth surfaces might lead to incorrect results of simulation.

[2] Graphical objects of *ASC* type are created automatically during model import from external CAD-programs

3.6.12.7. T-forces



Name: Following force

Comments/Text attribute C

Body1: Base0

Body2: Body1

Reference frame: Body1

Reduction point : Body1

pos_x pos_y

Type of description

● Expression ○ File

Force

Q*sin(w*t)

Moment

T= 10 dT= 0.01

Figure 3.250. Example of a harmonic following force

Use the **T-Force** tab to enter a set of force elements of T-type. An element is described by

- A pair of interacting bodies;
- A reference body for the force/moment components;
- A point the force is applied to (a point of the second body in the SC of this body); the button allows the visual setting the second body as well as the point of application using connection points (Sect. 3.6.9.6. Connection points).
- Type of the force description Expression or File

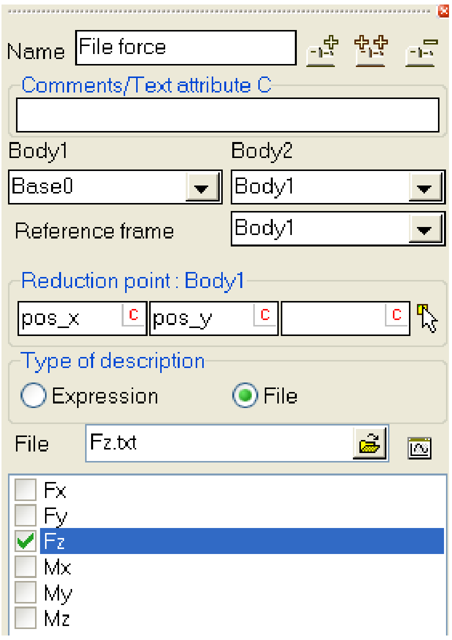
Remark. To describe a following force, set Base0 as the first body, and the reference body coinciding with the second one, figure 3.250.

3.6.12.7.1. Expression type of T-Force

In this case, the force and moment components are symbolic expressions including identifiers, standards functions of time *t* as well as kinematic functions.

Example. Description of a following harmonic force directed along Z-axis of the body1-fixed SC and applied to the point (*pos_x*, *pos_y*, *0*) is shown in Figure 3.250.

3.6.12.7.2. File type of T-Force



Name: File force

Comments/Text attribute C

Body1: Base0

Body2: Body1

Reference frame: Body1

Reduction point : Body1

pos_x | pos_y

Type of description

○ Expression ● File

File: Fz.txt

□ Fx

□ Fy

☑ Fz

□ Mx

□ My

□ Mz

Figure 3.251. File description of T-force

The force components can be described by a text file. The file format is the following: the first column contains values of time in seconds. Other columns contain optionally forces and torques in the sequence Fx, Fy, Fz, Mx, My, Mz in N and Nm. Any components can be omitted. For instance, the file can contain three columns t, Fz, Mx.

The button 📂 is used for selection of the file. After the selection, the file is copied automatically in the model directory. The user must check the available components in the inspector. For instance, the file in Figure 3.241 contains the component Fz of the force.

During the simulation, a linear interpolation is used for evaluation of the force and moment component values.

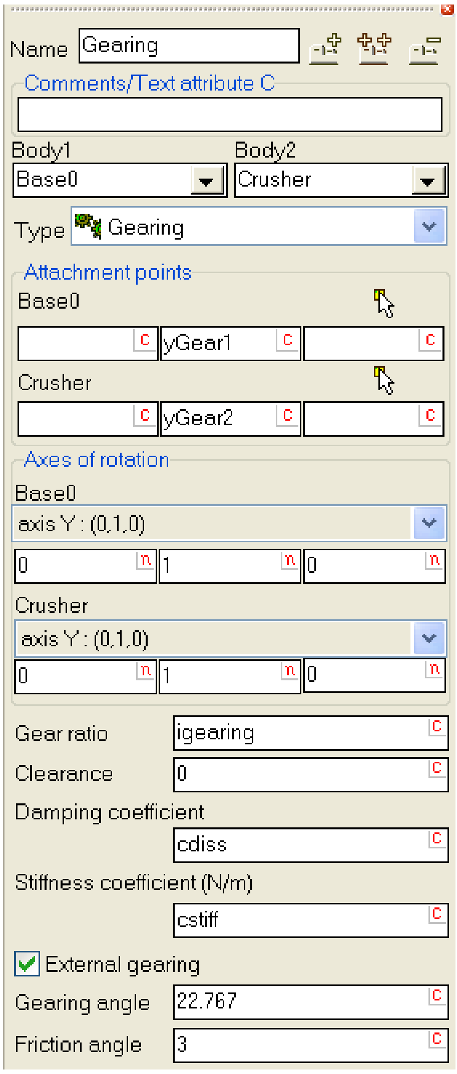
The 📈 button shows the plots of non-zero components of the force in a graphic window.

3.6.12.8. Special forces

The following types of forces are considered in this section:

- Gearing
- Chain gear
- Cam
- Spring
- Rack and pinion
- Bushing
- Combined friction

3.6.12.8.1. Gearing



Name: Gearing

Comments/Text attribute C

Body1: Base0

Body2: Crusher

Type: Gearing

Attachment points

Base0

	yGear1	

Crusher

	yGear2	

Axes of rotation

Base0

axis Y : (0,1,0)

0	1	0

Crusher

axis Y : (0,1,0)

0	1	0

Gear ratio: igearing

Clearance: 0

Damping coefficient: cdiss

Stiffness coefficient (N/m): cstiff

☑ External gearing

Gearing angle: 22.767

Friction angle: 3

Figure 3.252. Gearing parameters

A gearing is described by (Figure 3.252)

- **Attachment points** (centers of gears in SC of the corresponding bodies);
- **Gear axes** (unit vectors in SC on the bodies);

- Gearing parameters: **gear ratio, clearance** (optionally), **stiffness** and **damping** coefficients of tangential contact of teeth.

Check the **external/internal** option for plane gearing.

All the data except the gear axes are parameterized.

Gear axes and gear circle are visualized in the single element mode (Sect. 3.5.1.2.2), Figure 3.253.

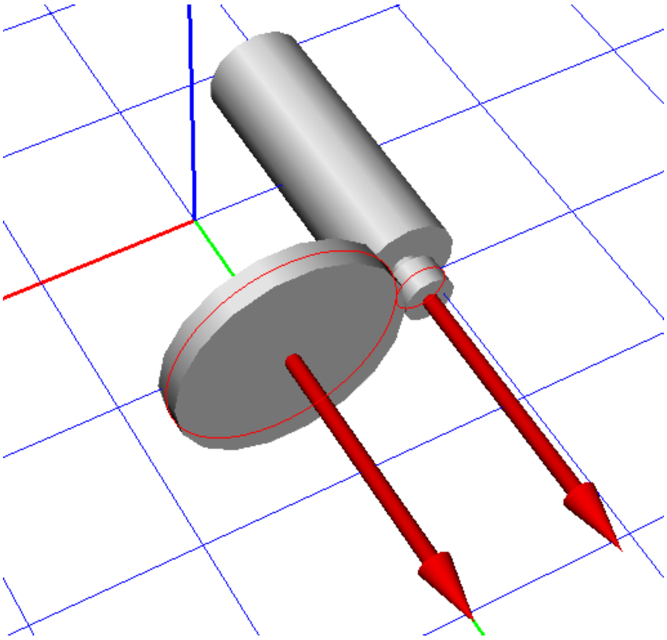
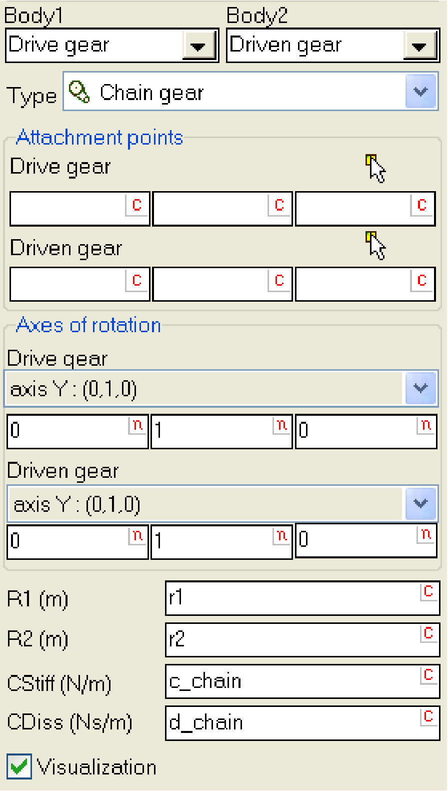


Figure 3.253. Visualization of gearing

Models:
{UM Data}\SAMPLES\LIBRARY\Gears;
{UM Data}\SAMPLES\TUTORIAL\Crusher.

3.6.12.8.2. Chain gear



Body1: Drive gear
Body2: Driven gear
Type: Chain gear

Attachment points
Drive gear
Driven gear

Axes of rotation
Drive gear: axis Y : (0,1,0)
0 | 1 | 0
Driven gear: axis Y : (0,1,0)
0 | 1 | 0

R1 (m): r1
R2 (m): r2
CStiff (N/m): c_chain
CDiss (Ns/m): d_chain
☑ Visualization

Figure 3.254. Chain gear parameters

The chain gear is described by (Figure 3.254)

- **Attachment points** (centers of gears in SC of the corresponding bodies);
- **Gear axes** (unit vectors in SC on the bodies); the vectors must be either parallel or nearly parallel;
- **Gear radii**
- **Stiffness** and **damping** coefficients of the chain by stretching;

Use the **Visualization** option for automatic drawing the chain in the animation window, Figure 3.255.

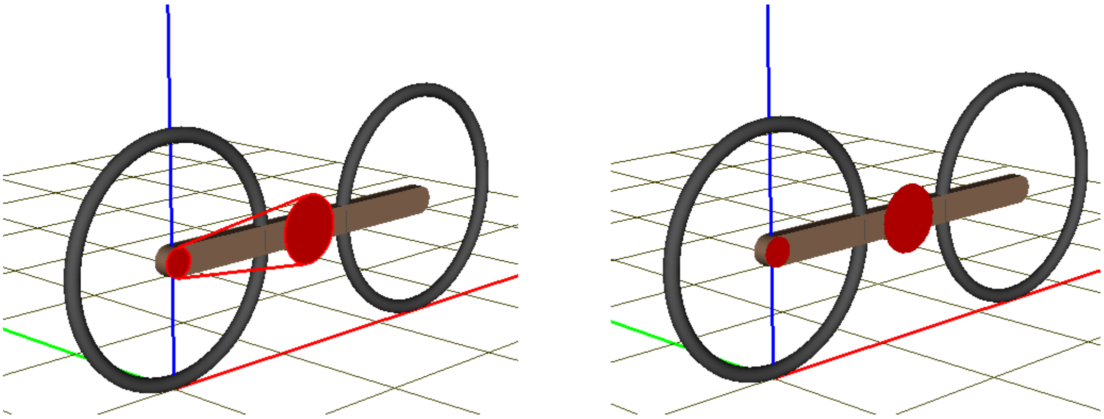


Figure 3.255. Model with (left) and without drawing the chain

All the data except the gear axes are parameterized.

Gear axes, circles and chain are visualized in the single element mode (Sect. 3.5.1.2.2), Figure 3.256.

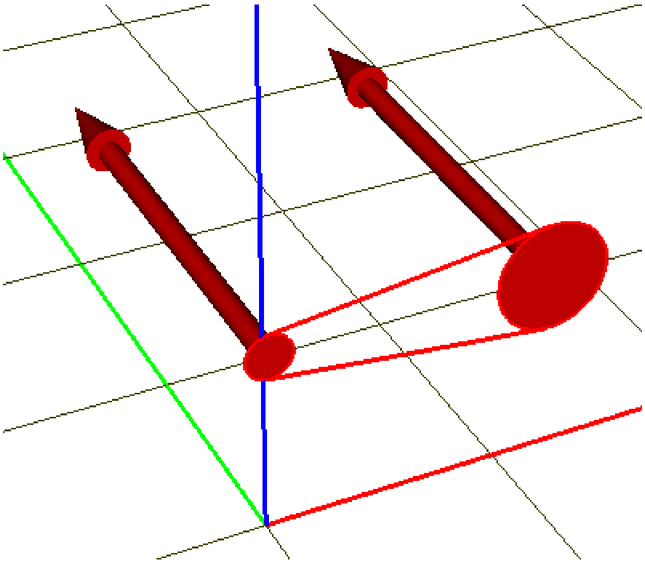


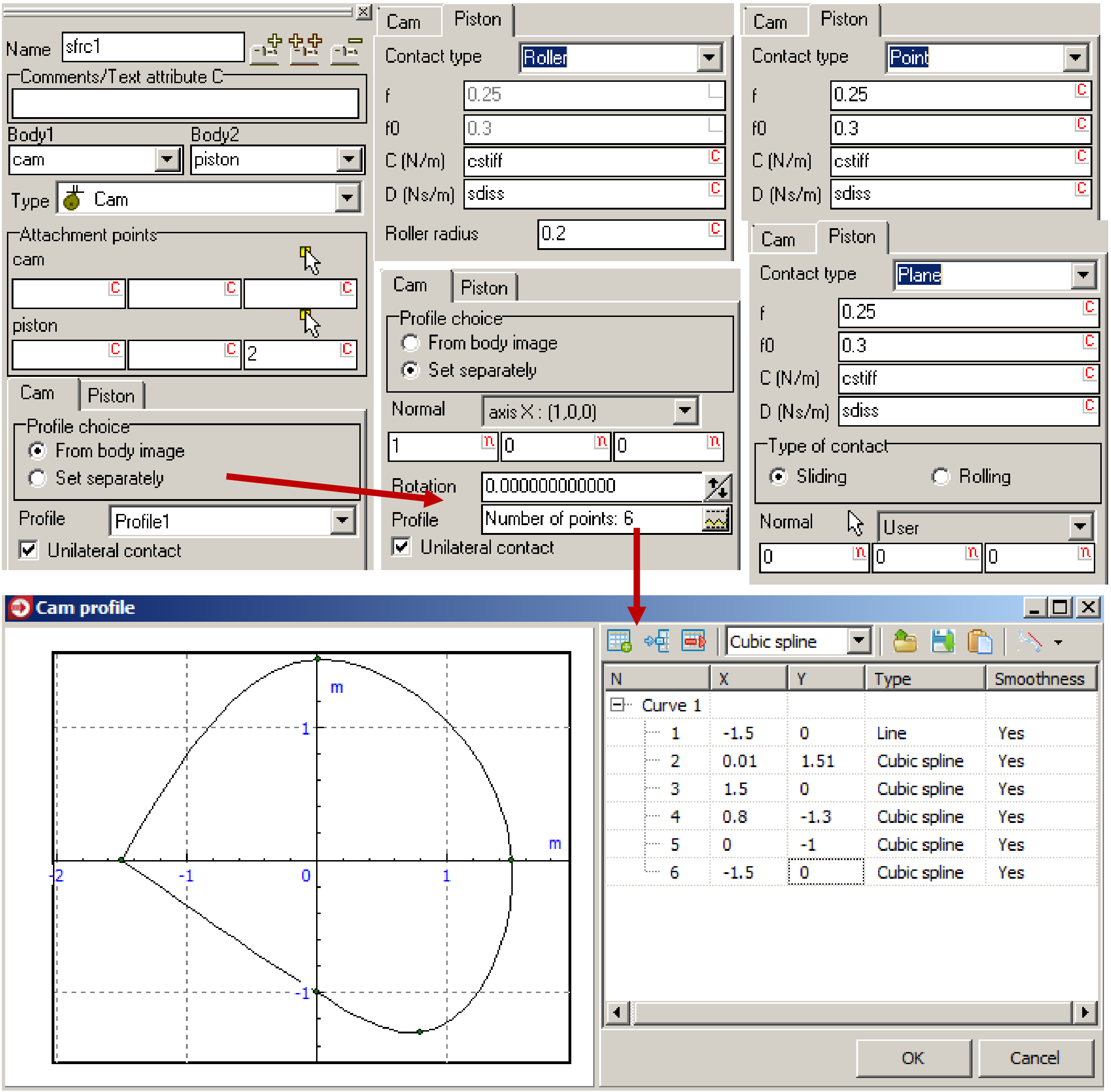
Figure 3.256. Visualization of chain gear

Model (Figure 3.255): {UM Data}\SAMPLES\LIBRARY\ChainGear.

3.6.12.8.3. Cam

A plane cam connection is realized as a variant of a contact interaction of two bodies, see Chapter 2, *Special forces/Cam*.

Example of using: model {UM Data}\SAMPLES\Mechanisms\cams.



Name: sfrc1
Comments/Text attribute C
Body1: cam
Body2: piston
Type: Cam
Attachment points
cam: (empty) (empty) (empty)
piston: (empty) (empty) 2

Cam | Piston
Profile choice: ● From body image ○ Set separately
Profile: Profile1
☑ Unilateral contact

Cam | Piston
Profile choice: ○ From body image ● Set separately
Normal: axis X : (1,0,0)
1 0 0
Rotation: 0.000000000000
Profile: Number of points: 6
☑ Unilateral contact

Cam | Piston
Contact type: Roller
f: 0.25
f0: 0.3
C (N/m): cstiff
D (Ns/m): sdiss
Roller radius: 0.2

Cam | Piston
Contact type: Point
f: 0.25
f0: 0.3
C (N/m): cstiff
D (Ns/m): sdiss

Cam | Piston
Contact type: Plane
f: 0.25
f0: 0.3
C (N/m): cstiff
D (Ns/m): sdiss
Type of contact: ● Sliding ○ Rolling
Normal: User
0 0 0

Cam profile

Cubic spline

N	X	Y	Type	Smoothness
Curve 1				
1	-1.5	0	Line	Yes
2	0.01	1.51	Cubic spline	Yes
3	1.5	0	Cubic spline	Yes
4	0.8	-1.3	Cubic spline	Yes
5	0	-1	Cubic spline	Yes
6	-1.5	0	Cubic spline	Yes

Figure 3.257. Cam parameters

The *cam-piston* parameters are presented in the Figure 3.257. The mathematical model of such the interaction is similar to the contact interaction described in Chapter 2, Sect. *Contact forces*. To create a cam element, select the **Special forces** item in the element tree and add a new element. Set the type of the special force as **Cam**. As a result the inspector shows some boxes for the parameters of the cam (Body 1) and for the piston or the link (Body 2). The user should enter the following parameters:

- **Characteristic points** in SC of each body. The first point is a point of a cam profile plane (if the profile is not imported from the body image). The second one is a point of contact

(contact type **Point**), the center point of a roller (contact type **Roller**) or a point on a contact plane (contact type **Plane**);

- **Profile of the cam** can be chosen as one of the graphic elements in the body 1 image (**From body image**) or as a planar closed curve created with the **Curve editor (Set separately)**. Use the **Unilateral contact** flag to choose either unilateral or bilateral type of the contact;
- Set the **point, normal** to the profile plane and **angle** of rotation about the normal to define the location of the **separately** defined profile in the SC of body1;
- Cam profile can be chosen as one of the graphic elements in the body 1 if the image contains one of the following GE:
  - cone, if the top and the bottom radii are equal;
  - ellipse with equal semi-axes (circle);
  - element of the profiled type (*Curve* 2D profile type, axis should be a straight line), see. Sect. 3.6.8.2.8.
- **Piston parameters**. Here the user can set dynamic and static coefficients of friction (except the **Roller** contact type), coefficients of contact stiffness and damping as well as the radius of the roller (contact type **Roller**).
- Additional parameters for the Plane contact type
  - external **normal** to the piston plane;
  - type of contact **Sliding/Rolling**; the **Rolling** contact type, in particular, allows modeling the rolling of non-circular wheels on a plane.

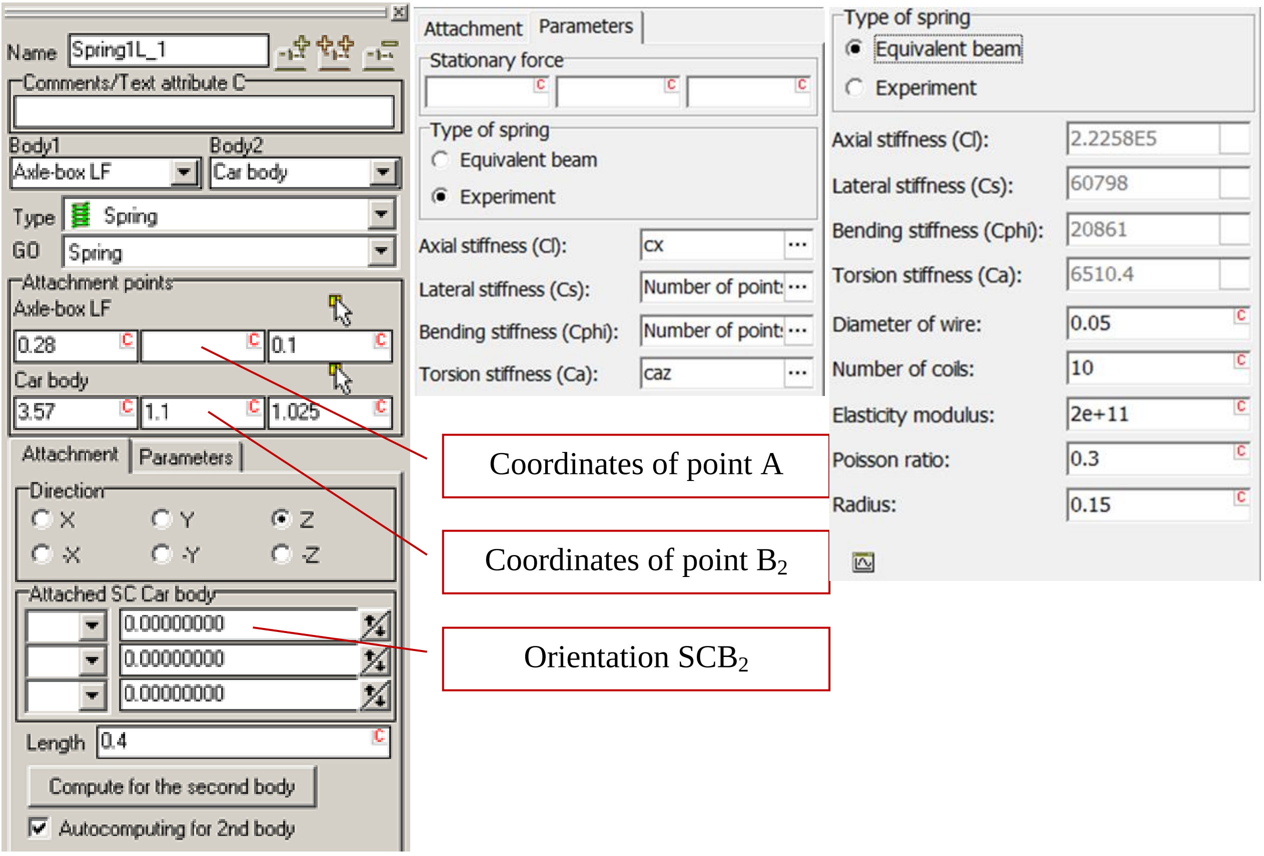
All of the data except the normals and points on the cam profile points can be parameterized.

3.6.12.8.4. Spring

The mathematical model of the element is described in Chapter 2, Sect. *Special forces/Spring, Generalized linear force element*.

Examples of description and/or usage:

- Chapter 7. Sect. *Models of Spring*;
- {UM Data}\SAMPLES\Rail_Vehicles\ac4 (requires UM Loco).



Name: Spring1L_1

Comments/Text attribute C:

Body1: Axle-box LF

Body2: Car body

Type: Spring

GO: Spring

Attachment points

Axle-box LF: 0.28 | | 0.1

Car body: 3.57 | 1.1 | 1.025

Attachment | Parameters

Direction: ○ X ○ Y ● Z ○ -X ○ -Y ○ -Z

Attached SC Car body: 0.00000000 | 0.00000000 | 0.00000000

Length: 0.4

Compute for the second body

☑ Autocomputing for 2nd body

Attachment | Parameters

Stationary force

Type of spring: ○ Equivalent beam ● Experiment

Axial stiffness (Cl): cx

Lateral stiffness (Cs): Number of point

Bending stiffness (Cphi): Number of point

Torsion stiffness (Ca): caz

Type of spring: ● Equivalent beam ○ Experiment

Axial stiffness (Cl): 2.2258E5

Lateral stiffness (Cs): 60798

Bending stiffness (Cphi): 20861

Torsion stiffness (Ca): 6510.4

Diameter of wire: 0.05

Number of coils: 10

Elasticity modulus: 2e+11

Poisson ratio: 0.3

Radius: 0.15

Figure 3.258. Spring parameters

Description of spring parameters similar to that for a generalized linear force element to a considerable extent, namely:

- coordinates of points A, B_2,
- orientation of SCB_2,
- usage of the *Compute for the second body* button and the *Autocomputing for 2nd body* option,
- stationary force.

It is supposed therefore that the user have already studied input of generalized linear force element, Sect. 3.6.12.5.

Here we consider some features of the spring description only.

1. Some equivalent information is entered instead of the point B1: direction of the spring axis in SC of the first body (radio group **Direction**) and the length of the spring under the static

load, which is set in the **Stationary force** group. If this force is zero, the length of free spring is set.

2. On the *Parameters* tab:

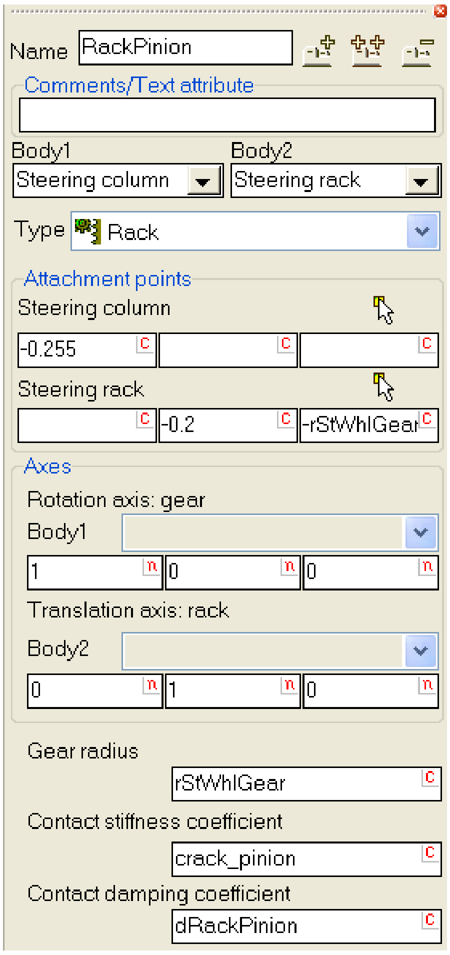
- Value of a stationary force, i.e. the force value for zero coordinate values and for the given spring length;
- Type of the spring description:
    - *Equivalent beam*: computation of stiffness parameters according to the theoretical formulas described in Chapter 2 for the given basic spring parameters (wire diameter, spring radius, number of active coils and so on); the 📈 button is used for getting plots of the shear and bending stiffness in dependence on the axial spring compression.
    - *Experiment*: spring stiffness are set as constant values, which can be parameterized as in figure above, or variable values in pointwise dependence on the spring axial compression; the ... button calls a curve editor for plot description.

The following stiffness parameters are specified:

- shear (lateral) stiffness **Cs**,
- longitudinal (axial) stiffness **Cl**,
- bending stiffness **Cphi**,
- torsion stiffness **Ca**.

3.6.12.8.5. Rack and pinion

Rack and pinion is a particular case of a gearing.



Name: RackPinion

Comments/Text attribute:

Body1: Steering column

Body2: Steering rack

Type: Rack

Attachment points

Steering column: -0.255 | |

Steering rack: | -0.2 | -rStWhlGear

Axes

Rotation axis: gear

Body1: 1 | 0 | 0

Translation axis: rack

Body2: 0 | 1 | 0

Gear radius: rStWhlGear

Contact stiffness coefficient: crack_pinion

Contact damping coefficient: dRackPinion

Figure 3.259. Rack and pinion parameters

The following parameters describe rack and pinion mechanism, Figure 3.259.

- Attachments points in SC of connecting bodies: center of pinion and point on the axis of the rack.
- Unit vectors along the pinion and rack axes (rotation and translation axes respectively).
- Pinion radius.
- Contact stiffness and damping parameters.

All parameters except unit vectors can be parameterized.

Example. Use of the rack and pinion force element in a car steering system is shown in Figure 3.260 (see the car model {UM Data}\SAMPLES\Automotive\Vaz21_09).

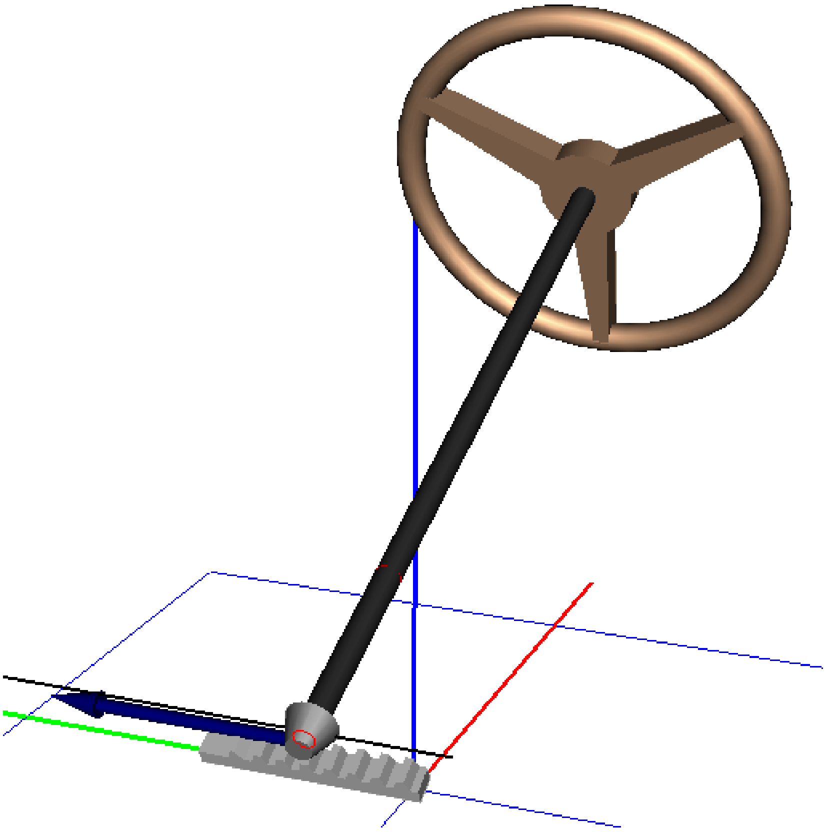


Figure 3.260. Rack and pinion in the car steering system

3.6.12.8.6. Bushing

The mathematical model of the element is described in Chapter 2, Sect. *Special forces*| *Bushings*.

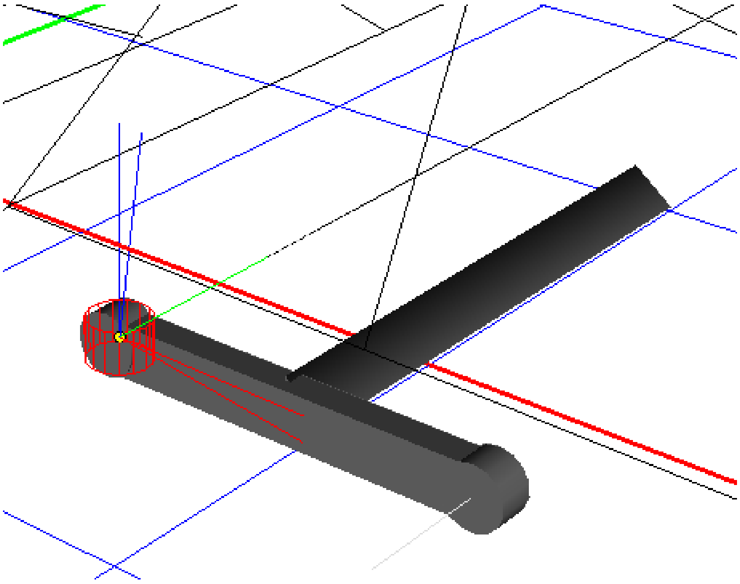


Figure 3.261. Example of a bushing: model of a car VAZ21_09 from directory Samples/Automotive

In the single element mode of the animation window, the bushing is drawn as a red wired cylinder. SCB1 and SCB2 are drawn as well, Figure 3.261.

To describe a bushing

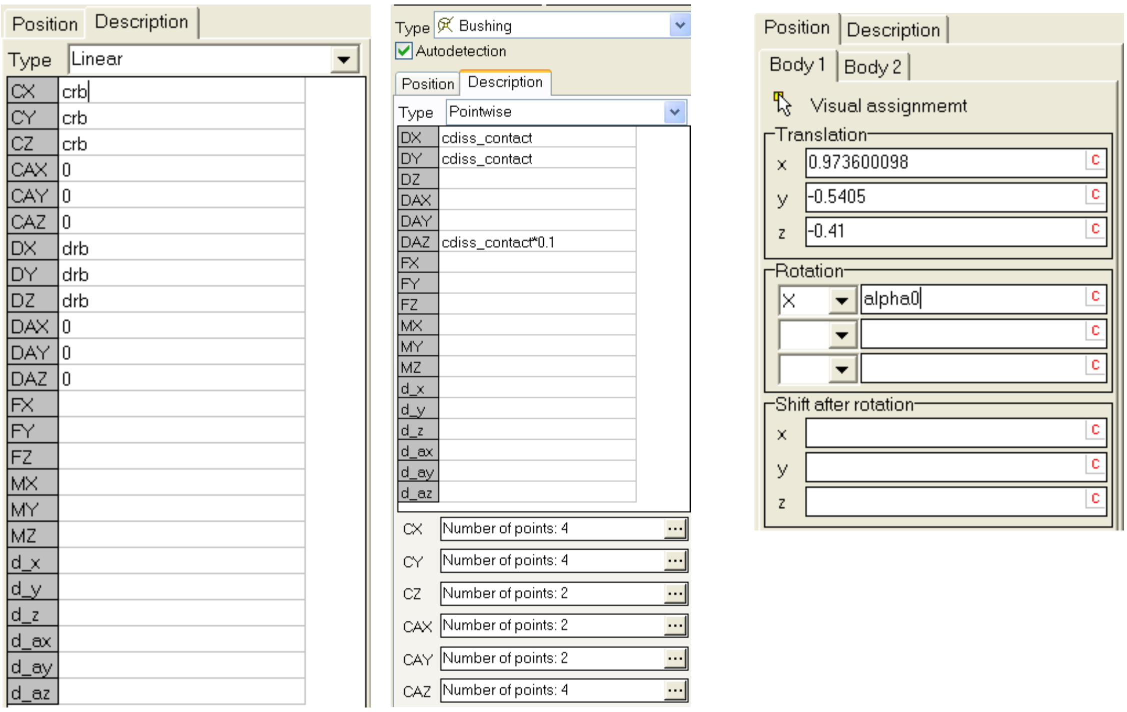
- set positions of SCB1 (Body 1) and SCB2 (Body 2) with a standard interface for specifying positions of local system of coordinates;
- select element type **Linear/Pointwise/Generalized**.

Remark. The **autodetection** mode is often used by description of bushings. At this mode, SCB2 is automatically detected by positions of SCB1 for *zero value of object coordinates*.

3.6.12.8.6.1. Parameters of linear and pointwise bushings

In case of linear bushing: enter stiffness and damping constants for shifts (CX, CY, CZ) and rotations (CAX, CAY, CAZ) relative to axes of CSB1.

In case of pointwise bushing: enter damping constants and nonlinear plots for force and torque components versus the corresponding displacements and rotations.



Position | Description

Type: Linear

CX	crb
CY	crb
CZ	crb
CAX	0
CAY	0
CAZ	0
DX	drb
DY	drb
DZ	drb
DAX	0
DAY	0
DAZ	0
FX	
FY	
FZ	
MX	
MY	
MZ	
d_x	
d_y	
d_z	
d_ax	
d_ay	
d_az	

a)

Type: Bushing

☑ Autodetection

Position | Description

Type: Pointwise

DX	cdiss_contact
DY	cdiss_contact
DZ	
DAX	
DAY	
DAZ	cdiss_contact*0.1
FX	
FY	
FZ	
MX	
MY	
MZ	
d_x	
d_y	
d_z	
d_ax	
d_ay	
d_az	

CX	Number of points: 4
CY	Number of points: 4
CZ	Number of points: 2
CAX	Number of points: 2
CAY	Number of points: 2
CAZ	Number of points: 4

b)

Position | Description

Body 1 | Body 2

Visual assignmemt

Translation

x	0.973600098
y	-0.5405
z	-0.41

Rotation

X	alpha0

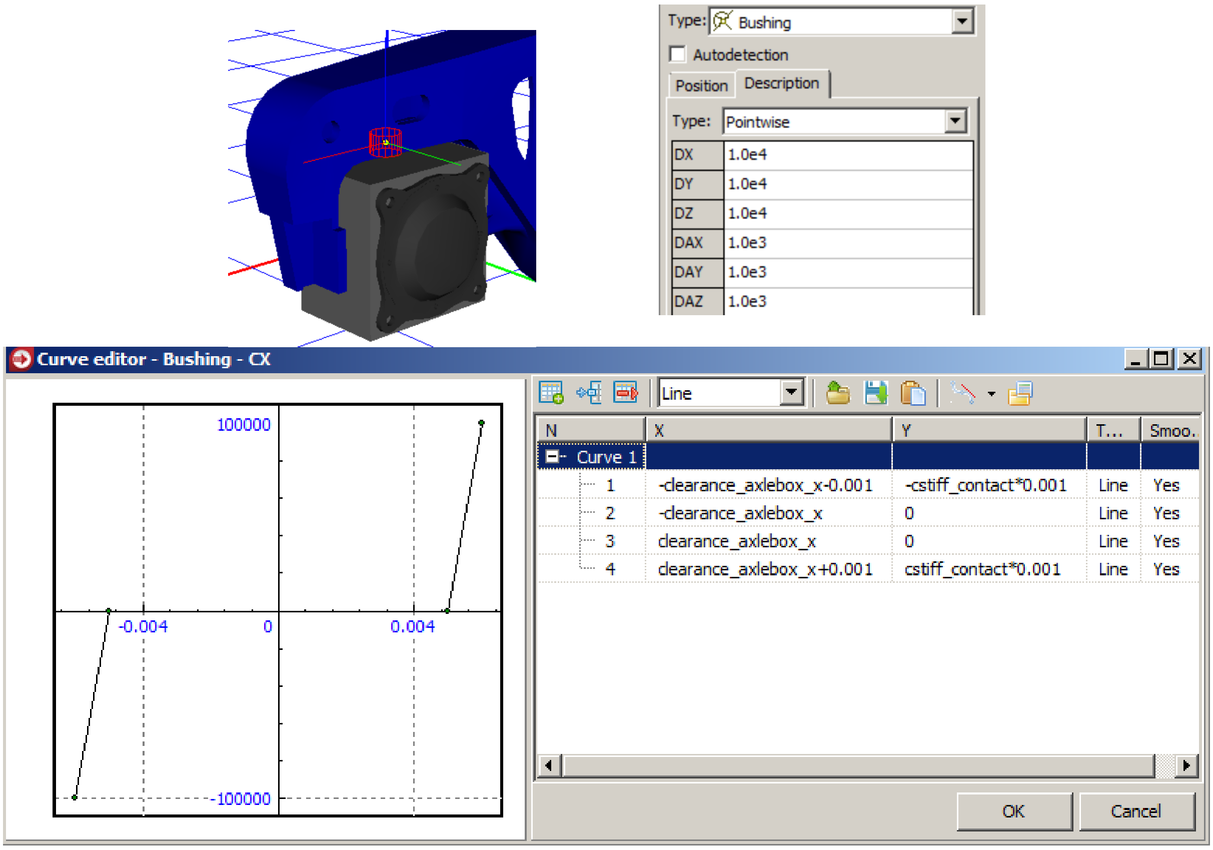
Shift after rotation

x	
y	
z	

c)

Figure 3.262. Linear bushing: compliant ball joint (a) Pointwise bushing. Autodetection mode is on (b). c) Position of SCB1

If necessary, static values of force and torque F_0, M_0 (FX, FY, FZ, MX, MY, MZ) and/or static offset for SCB2 $\Delta r_0,\ \Delta\pi_0$ (d_x, d_y, d_z, d_ax, d_ay, d_az) are specified.



Type: Bushing

□ Autodetection

Position | Description

Type: Pointwise

DX	1.0e4
DY	1.0e4
DZ	1.0e4
DAX	1.0e3
DAY	1.0e3
DAZ	1.0e3

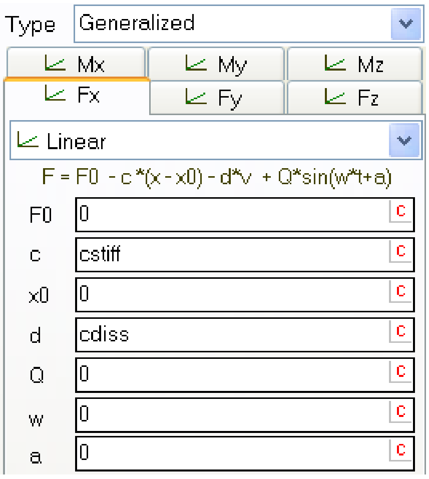
Curve editor - Bushing - CX

N	X	Y	T...	Smoo.
Curve 1				
1	-clearance_axlebox_x-0.001	-cstiff_contact*0.001	Line	Yes
2	-clearance_axlebox_x	0	Line	Yes
3	clearance_axlebox_x	0	Line	Yes
4	clearance_axlebox_x+0.001	cstiff_contact*0.001	Line	Yes

Figure 3.263. Example of a pointwise bushing, which is used for modeling support and gaps between a side frame and an axle-box in the model of a three-piece bogie of a freight car

Remark. The following agreement about signs is assumed by description of pointwise force and torque components. Positive value of elastic force/torque in a plot corresponds to positive value of displacement/rotation, see the above plot.

3.6.12.8.6.2. Description of generalized bushing



Type: Generalized

Linear

F = F0 - c*(x-x0) - d*v + Q*sin(w*t+a)

F0	0
c	cstiff
x0	0
d	cdiss
Q	0
w	0
a	0

Figure 3.264. Example of generalized bushing

In the case of a generalized bushing, the user specifies projections of force and torque in SCB1, Figure 3.264. Any type of scalar force can be assigned to the components, Sect. 3.6.12.

This property of the generalized bushing makes it one of the most powerful tools for description of force interactions in UM.

Models:

{UM Data}\SAMPLES\LIBRARY\Bushing\Bushing general;

{UM Data}\SAMPLES\LIBRARY\Bushing\Bushing general Ext.

3.6.12.8.7. Combined friction

The combined friction is a generalization of the point-plane contact. Full description and the mathematical model of this force element can be found in Chapter 2, Sect. Force elements | Special forces | Combined friction.

Examples of usage:

Directory {UM Data}\SAMPLES\LIBRARY\CombFriction, models:

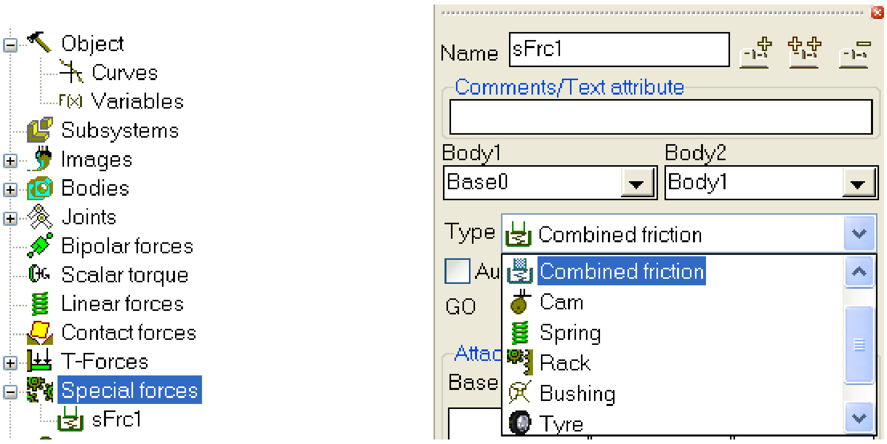
CF2D_without_fict;

CF2D_with_fict;

CF2D_with_fict_limit_fict;

CF2D_with_fict_limit_body2.

The following steps are required for development of the element and setting element parameters.



Name sFrc1

Comments/Text attribute

Body1 Base0

Body2 Body1

Type Combined friction

Creation of a new element of the “Combined friction” type

3.6.12.8.7.1. Creation of element

To create a combined friction force element

- open the list of special forces;
- add a new force;
- set its type as Combined friction.

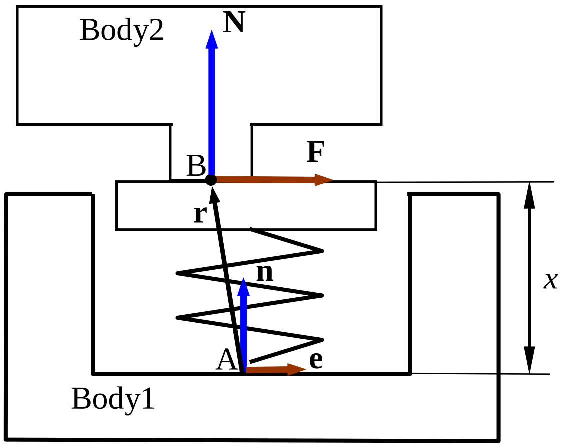
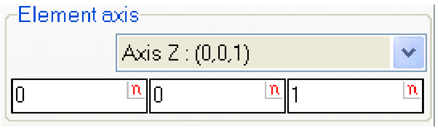


Figure 3.265. Scheme of the force element

3.6.12.8.7.2. Choice of a pair of bodies and setting the attachment point

Bodies and attachment points (point A for the first body and point B for the second one, Figure 3.265) are specified in a standard manner, see. Sect. 3.6.10.1, 3.6.10.3, 3.6.10.4.

3.6.12.8.7.3. Specifying element axis


Element axis

Axis Z : (0,0,1)

0 | 0 | 1

Figure 3.266. Element axis

Element axis is set in numerical form (Figure 3.266) by a unit vector **n** in SC of the first body, Figure 3.265.

3.6.12.8.7.4. Assignment of image

Image of the element is selected from the preliminary created list of graphic objects. Image of the element should be created according to the method described for the bipolar force element, Sect. 3.6.6.

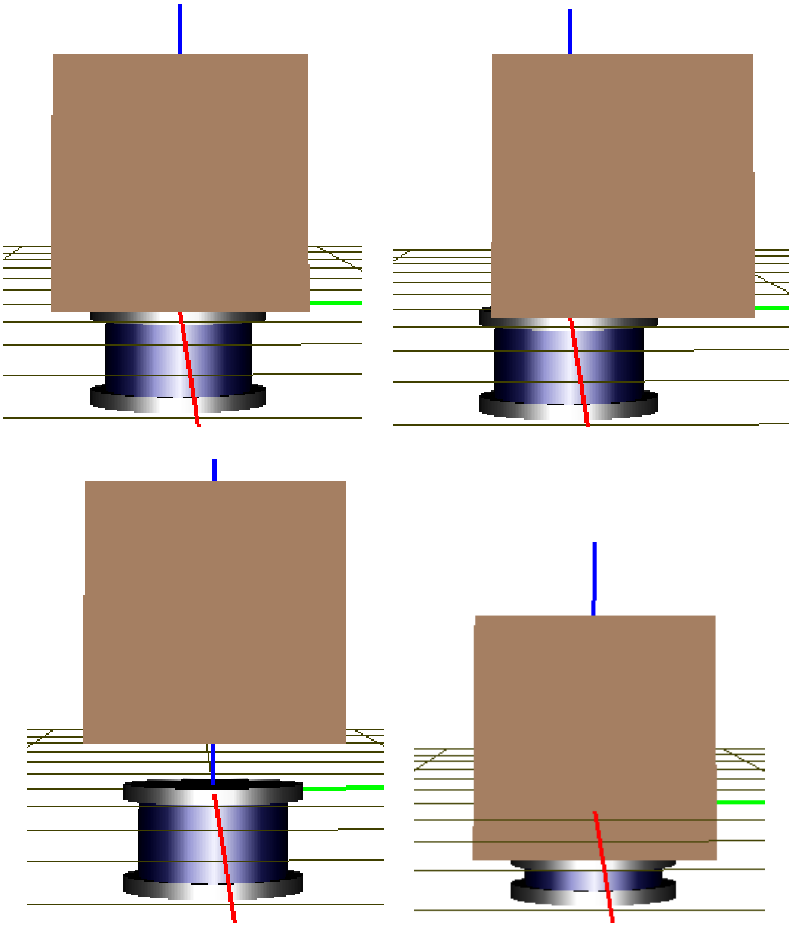
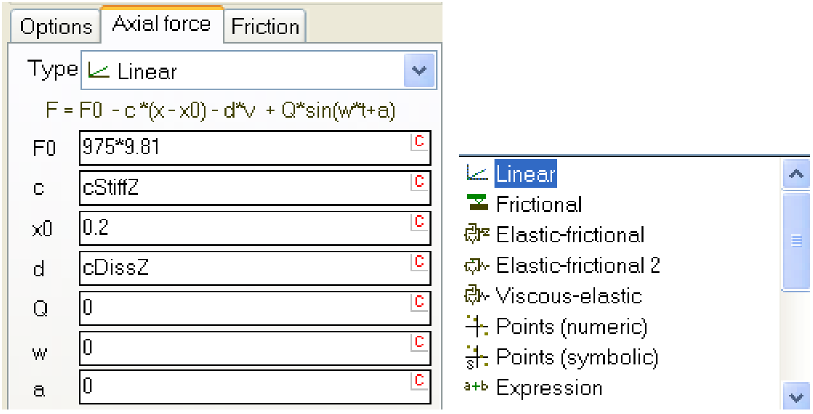


Figure 3.267. Image of combined friction force element by shifts of the second body relative to the first one

As opposed to the bipolar force element, the image in this case does not connect two points of Body1 and Body2. It is oriented along the element axis n, Figure 3.267. Moreover, in the case of a unilateral element, the user can specify a visual lifting of the second body from the element.

3.6.12.8.7.5. Setting axial force model



Type: Linear

F = F0 - c*(x-x0) - d*v + Q*sin(w*t+a)

F0: 975*9.81

c: cStiffZ

x0: 0.2

d: cDissZ

Q: 0

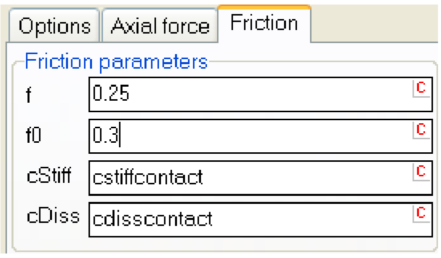
w: 0

a: 0

Figure 3.268. Setting axial force model and parameters

The model of an axial or a normal force *N* is described of the **Axial force** tab. The user should select one of the possible types of the scalar force description and set its parameters, Sect 3.6.12.1.

3.6.12.8.7.6. Friction force parameters



f 0.25

f0 0.3

cStiff cstiffcontact

cDiss cdisscontact

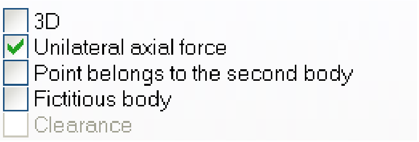
Figure 3.269. Friction force description

Parameters specifying the friction force should be set on the **Friction** tab. The user must set

- dynamic coefficient of friction (f);
- static coefficient of friction (f0);
- stiffness constant in the sticking mode (cStiff);
- damping constant in the sticking mode (cDiss).

See Chapter 2, Sect. Force elements | Contact forces | Points-Plane and Points-Z-surface types for additional information.

3.6.12.8.7.7. Setting force element modes



□ 3D

☑ Unilateral axial force

□ Point belongs to the second body

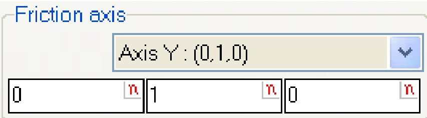
□ Fictitious body

□ Clearance

Figure 3.270. Element mode keys

Use the **Options** tab to set a number of keys specifying the force element modes.

- **3D**



Axis Y : (0,1,0)

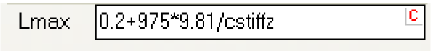
0 1 0

Figure 3.271. Friction axis in 2D mode

In the 3D mode, the friction force lies in the plane perpendicular to the element axis **n**. If 3D key is off, the element is in 2D mode, when the friction is directed along the fixed axis (axis of friction) relative to Body2. The friction axis is specified by a unit vector e, Figure 3.271.

Remark. Restrictions on shift of the fictitious body (FB) relative to Body1 of Body2 relative to the FB are not available in 3D model of the element.

- **Unilateral axial force**



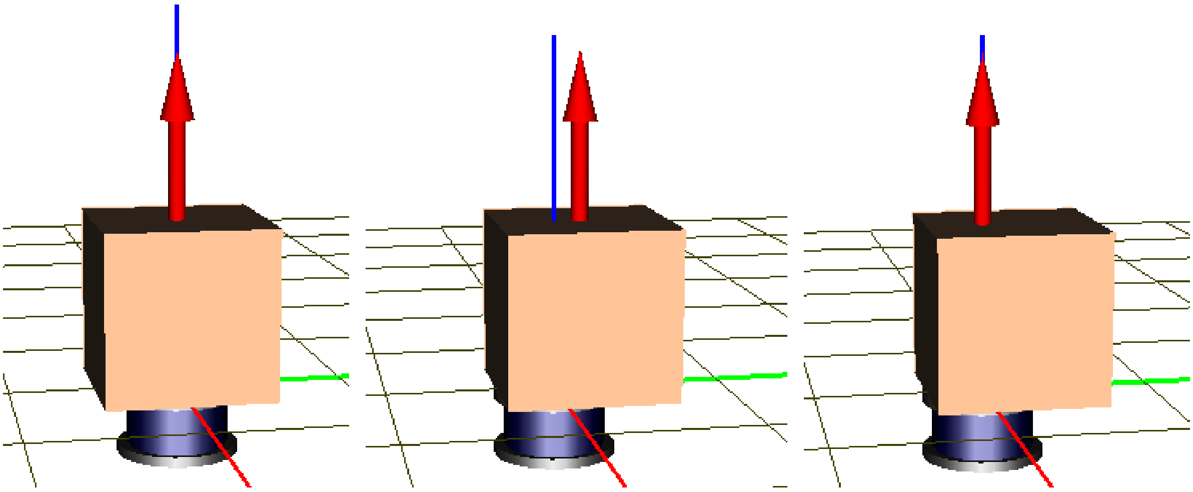
Lmax 0.2+975*9.81/cstiffz

Figure 3.272. Setting the maximal length of element

If this key is on, the unilateral contact takes place, i.e. forces vanish if the normal force is negative. The user can define the maximal length of the element in this mode. Note that this parameter influences on the element image only, and does not affect the force values. Namely, the element length in animation window will not stretched more than the specified value, Figure 3.267.

If the key is off, the element becomes a bilateral.

- **Point belongs to the second body**



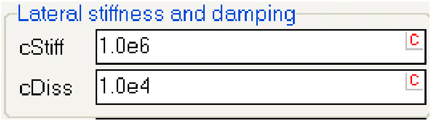
b) c)

Figure 3.273. Position of the normal force by shift of the second body. The contact point belongs to the first body (b) or the first body (c)

The contact point can belong either to Body2 (if the key is on) or to Body1 the key is off. This key does not affect the forces value, but it specifies the position of the normal force N by shifts of the body2 relative to Body1. If the contact point belongs to the first body, it is fixed relative to Body1 and directed along the axis **n**. If the point belongs to the second body, the force N moves in the lateral direction together with Body2, but still parallel to **n**.

- **Fictitious body**

If the key is on, the model includes a fictitious body (FB), which can move relative to the first body in lateral directions. The second body is in the contact with the FB.



Lateral stiffness and damping

cStiff 1.0e6

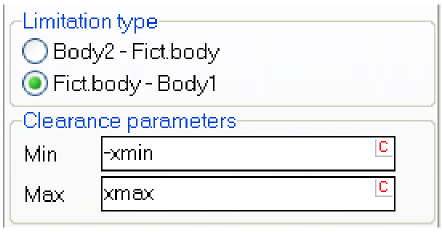
cDiss 1.0e4

Figure 3.274. Parameters of viscous-elastic connection of FB with Body1

A linear viscous-elastic force appears by lateral shifts of the FB relative to Body1, which constants should be specified by the user.

More 1information about the FB can be found Chapter 2, Sect. *Force elements | Special forces | Combined friction*.

- **Clearance**



Limitation type

○ Body2 - Fict.body

● Fict.body - Body1

Clearance parameters

Min -xmin

Max xmax

Figure 3.275. Parameters of clearance (gap)

This option is available in 2D mode of the element with the FB. If the key is on, the **Clearance** tab with additional parameters appears.

Clearance introduces a restriction on the lateral shift of the FB relative to Body1 or Body2 relative to the FB.

Limit values of the shift must be set in the **Min, Max** boxes. See Chapter 2, Sect. Force elements | Special forces | Combined friction for more details.

3.7. UM Components

3.7.1. Basic notions

UM components give an efficient tool for development of models. A component helps to add to a model a fully parameterized element or a group of elements with preliminary specified properties. During the visual adding a component, the user should define necessary geometric information with connection points like attachment points of a force element, Sect. 3.6.9.6.

The following elements and substructures can be converted into a component form:

- body with/without image
- joint with/without image
- force element with/without image
- images
- subsystem
- object

If an image is assigned to an UM element like a body or a force element, the element can be included in the component together with the image.

Two files are assigned to any UM component: a text file with description of the component in UM format, and a bitmap (*.bmp) file with the component icon.

The standard extensions for the component text files are

- Joints: *.jnt
- Bodies: *.bdy
- Bipolar force elements: *.bfc
- Images: *.img
- Subsystem (UM object): *.sbs
- Generalized linear force element s: *.lfrc
- Contact forces: *.cfrc
- General type forces: *.afrc
- Special forces: *.sfrc

Each UM component can be parameterized. The corresponding identifiers and their default values are included in the component description file.

An example of a file with component including element image

```
with const;
  cnst=(mbody, 100, "Mass");
  cnst=(ibodyx, 12, "Moment of inertia X");
  cnst=(ibodyy, 10, "Moment of inertia Y");
  cnst=(ibodyz, 15, "Moment of inertia z");
  cnst=(xcg, 0, "Center of gravity X");
  cnst=(ycg, 0, "Center of gravity Y");
```

```
   cnst=(zcg, 0.5, "Center of gravity Z");
   cnst=(a, 0.3, "Depth");
   cnst=(b, 0.4, "Width");
   cnst=(c, 0.7, "Height");

 with go;
   name="Body image";

   with ge; type= BAR;
   VisibleSide=vsFront;  material=6520998,6520998,0,0,10,1;

   width=1;
   discret=1;  box=a,b,c;
 with body1; name="Body"; igrobj=1;
   m=mbody; ixx=ibodyx; iyy=ibodyy; izz=ibodyz; rcx=xcg; rcy=ycg; rcz=zcg;
```

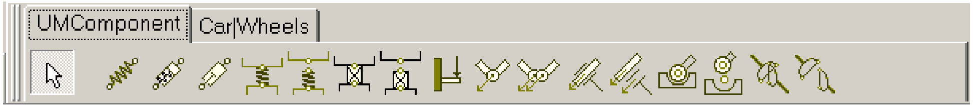


Figure 3.276. Component panel

A set of component can be grouped in a *component library*, which description is stored in a *.umc file. A tab on the tool panel corresponds to each of the *linked* component libraries. The library tabs or the list of component window are used for both visual and non-visual adding to the active object all elements included in the component.

The following component libraries are delivered with UM:

- UMComponents: a set of standard forces and joints;
- Mates: a set of standard mate components, Sect. 3.6.11.10;
- Car|Wheels standard component of the UM Automotive module if the module is available in the UM configuration, see the file 12_UM_Automotive.pdf;
- standard component of UM Loco module if the module is available in the UM configuration, see the files 08_UM_LOCO.pdf, 17_UM_TRAIN3D.pdf.

The user may create own libraries of components to create a database of elements.

The button is visible in the visual mode otherwise it is invisible. To switch between the modes

- ○ click the right mouse button on the tab with components;
- ○ use the **Visual design** menu command in the pop-up menu.

3.7.2. List of components

The list of components is used for adding visual components to the model, i.e. it duplicates the component tabs. The component list window is available by then **Tools | List of components** menu command.

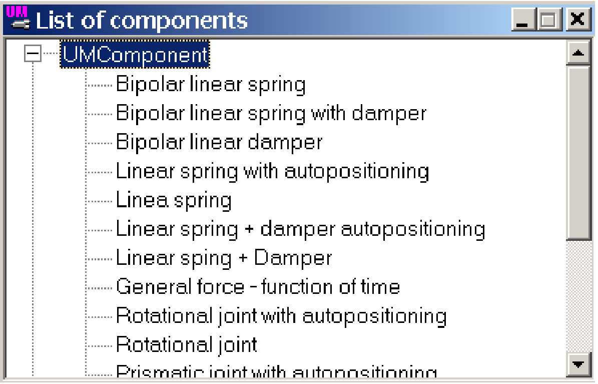

List of components
UMComponent
Bipolar linear spring
Bipolar linear spring with damper
Bipolar linear damper
Linear spring with autopositioning
Linea spring
Linear spring + damper autopositioning
Linear sping + Damper
General force - function of time
Rotational joint with autopositioning
Rotational joint
Prismatic joint with autopositioning

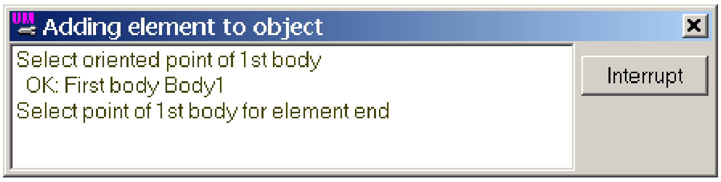
Figure 3.277. List of components

3.7.3. Adding a component in visual mode

Visual adding of components requires a preliminary description of connection points for bodies, Sect. 3.6.9.6. In this mode the components can be connected with the elements already presented in the object in a very simple and intuitively clear manner. This is the advantage of visual adding in comparison with the non-visual one.

To add a component in the visual mode

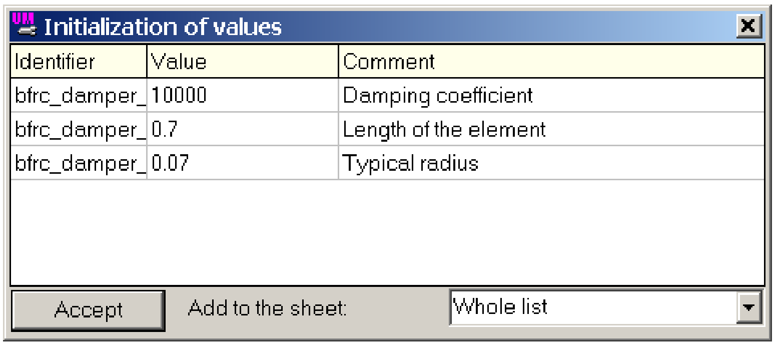
- Click by the left mouse button on the component button or double click on the component name in the component list window. The full object mode of the animation window is switched on automatically, Sect. 3.5.1.2.2, and connection points are visualized.



Select oriented point of 1st body
OK: First body Body1
Select point of 1st body for element end

Figure 3.278. Window with help comments to adding a component

- Follow instructions in the help window by selecting connection points or body images.



Identifier	Value	Comment
bfrc_damper_	10000	Damping coefficient
bfrc_damper_	0.7	Length of the element
bfrc_damper_	0.07	Typical radius

Add to the sheet: Whole list

Figure 3.279. List of identifiers of a component

- If necessary, set desired values to identifiers included in the component

To cancel the process of visual adding a component, click either the ↖ button on the component panel or the **Interrupt** button on the help window, Figure 3.278.

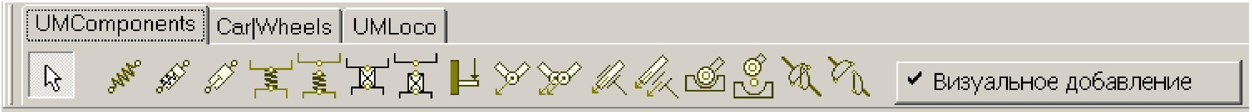
3.7.4. Mode of adding elements with and without autopositioning

The autopositioning function simplifies the process of visual adding elements. It is used if all bodies of the model are in the positions satisfying kinematic constraints. As a rule this is true for models converted from CAD assemblies. The advantages by the use of the autopositioning consist in simplification of setting a part of geometrical information to joints and force elements.

Example. By visual adding a spherical joint, two joint points should be specified in SC of a pair of connecting bodies. If these points coincide for the current positions of the bodies in the model, it is enough to select the joint point for the first body, and the coordinates of the second point are computed by the program. In no autopositioning is assigned to the visual component, both the points must be selected visually, i.e. they cannot coincide in the animation window.

3.7.5. Standard library of visual components UMComponents

3.7.5.1. List of standard components



✔ Визуальное добавление

Figure 3.280. Group of elements in the library of components

The library UMComponents includes the following elements:

• Force elements

- Bipolar linear spring
- Bipolar linear spring and damper
- Bipolar linear damper

- Linear spring

- Linear spring with autopositioning

- Generalized linear viscous-elastic element

- Generalized linear viscous-elastic element with autopositioning

- T-force

• Joints

- Rotational joint

- Rotational joint with autopositioning

- Translational joint

- Translational joint with autopositioning

- Hook joint

- Hook joint with autopositioning

- Spherical joint

- Spherical joint with autopositioning

3.7.5.2. Visual adding generalized linear elastic or viscous-elastic forces

Here we consider in details the process of visual adding a linear force element.

The generalized linear force element is an important tool for description of springs and viscous-elastic elements. We recommend to study the mathematical model of the element before start its usage because the model in quite not trivial, Sect. 3.6.12.5, Chapter 2, *Generalized force element*.

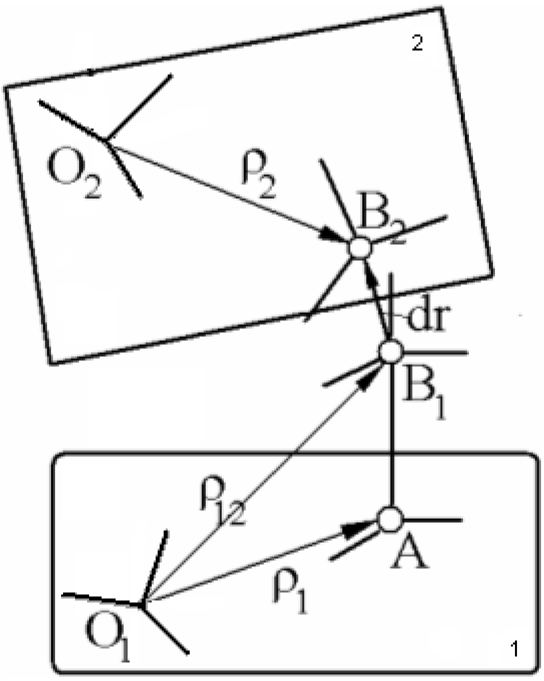


Figure 3.281. Systems of coordinates related to linear force elements

Description of geometric data for an elastic force element includes the following systems of coordinates (SC, Figure 3.281):

- SC1 - local SC of body 1 with origin O1
- SC2 - local SC of body 2 with origin O2
- SCA - fixed relative to body 1, begin of the linear force element, origin at A
- SCB1 - fixed relative to body 1, end of the linear force element in unloaded state or under the static load, origin at B1
- SCB2 - fixed relative to body 2, end of the linear force element in unloaded state or under the static load, origin at B2

3.7.5.2.1. Automatic positioning mode

The **linear spring with autopositioning** **or the Linear spring + damper with autopositioning** components.

This mode is usually used for dynamic objects created as a result of data conversion from the CAD programs.

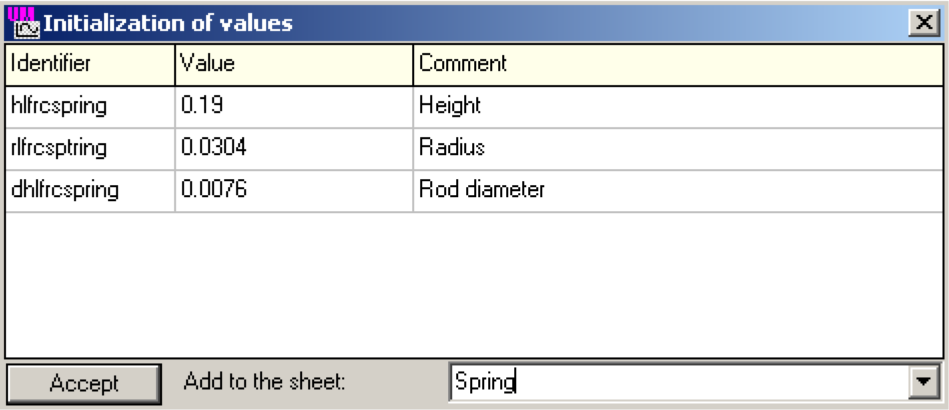
1. Add to the first body a connection point corresponding to point **A** or an oriented point for **SCA** (see Figure 3.281).
2. Add to the **first body** a connection point corresponding to point **B1** or optionally add to the **second body** a connection point corresponding to point **B2** or an oriented point for **SCB2** (see Figure 3.281)

3. Click one of the buttons on the component panel. The window with help comments appears. The window will include the following sequence of comments:

- "Select point/oriented point at 1st body" - select by the left mouse a point for the point A or an oriented point for **SCA**. If the selected point is not an oriented point, axes of **SCA** are set parallel to those of **SC1**.
- "Select element end point (first body) or point/oriented on second body".

If a point at the **second** body is selected, the point **B2** or **SCB2** is assigned. If the selected point is not an oriented point, axes of **SCB2** are set parallel to those of **SC2** (the body2 - fixed SC). Point **B1** is computed automatically coinciding with **B2** (see Figure 3.281) and the process of selection geometric parameters is over. Otherwise the next step is necessary:

- "Select the second body" - click by the mouse on the image of the second body. Point **B2** and **SCB2** are computed automatically coinciding with **B1, SCB1** (Note that axis of **SCB1** are parallel to **SCA**), see the Figure 3.281.



Identifier	Value	Comment
hlfrcspring	0.19	Height
rlfrcsptring	0.0304	Radius
dhlfrcspring	0.0076	Rod diameter

Figure 3.282. Identifiers parameterizing a force component

4. Correct names and values of identifiers, parameterizing the force element (Figure 3.282)

3.7.5.2.2. The User's mode

The **linear spring** or the **Linear spring + damper** components correspond to the linear force components without automatic positioning.

1. Add to the first body a connection point corresponding to point **A** or an oriented point for **SCA** (see Figure 3.281).
2. Add to the **first body** a connection point corresponding to point **B1**
3. Add to the **second body** a connection point corresponding to point **B2** or an oriented point for **SCB2** (see Figure 3.281)
4. Click one of the buttons on the component panel. The window with help comments appears, the window contains instructions and comments to the element adding process:

- "Select point/oriented point at 1st body" - select by the left mouse a point for the point A or an oriented point for **SCA**. If the selected point is not an oriented point, axes of SCA are set parallel to those of **SC1** (the body1 - fixed SC), see Figure 3.281.
- "Select point at 1st body for element end" - select a connection point for B1 (see Figure 3.281).
- "Select point/oriented point at 2nd body". On this step the point **B2** or **SCB2** is assigned. If the selected point is not an oriented point, axes of **SCB2** are set parallel to those of **SC2** (the body2 - fixed SC), see Figure 3.281.

5. Correct names and values of identifiers, parameterizing the force element

Remarks

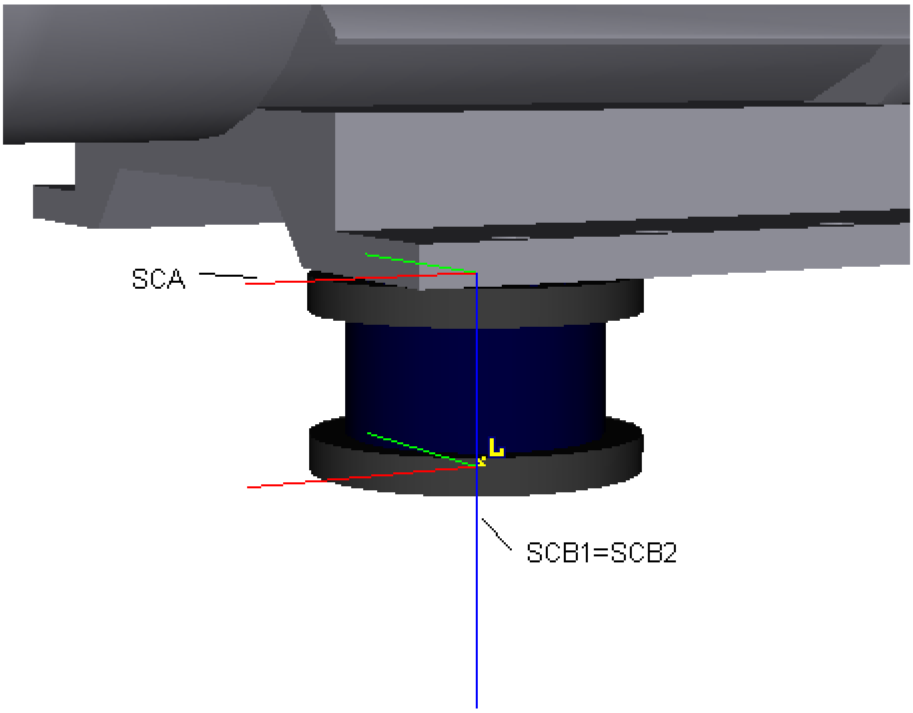


Figure 3.283. Visualization of SC of element

1. In the single element mode of the animation window, **SCA**, **SCB1** and **SCB2** are visualized; the origin of **SCB1** is marked by the ∟ icon, Figure 3.283. Images of **SCB1** and **SCB2** should usually coincide when the geometrical data of the element are correct.

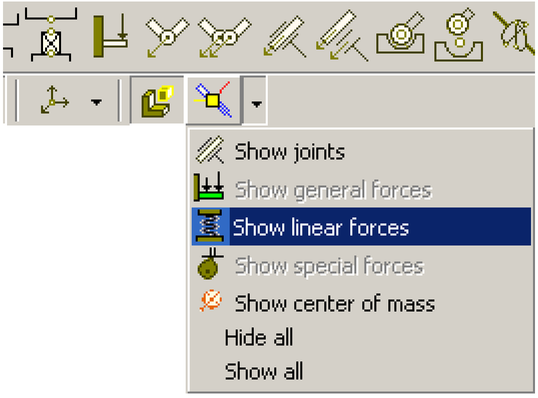


Figure 3.284. Selection of auxiliary drawings for linear force elements

2. In the single element mode of the animation window these systems of coordinates can be visualized if the corresponding option is on, Figure 3.284.

3.7.6. Development of libraries of visual components

3.7.6.1. Wizard of components

Wizard of components is designed for development and editing lists of components, Figure 3.285. Use the **Tools |Wizard of components…** menu command or the button to call the wizard.

The tool panel contains

- the name of the current tab (group) of the library (UMCommon in Figure 3.285),
- the button adds a new group of components of a library;
- the button deletes a group of components;
- the button opens a library file;
- the button saves the library to a file;
- the button saves the library with another name.

The Components group contains the tabs corresponding to the current group of components. The component edit panel contains parameters of an active component:

- **Name** of the component, which is used as the name of the corresponding tab,
- **Hint**, which is used as the component name in the list of components as a hint to the component button, Figure 3.280 ,
- **Path** to the file with component,
- **Button image** for the component panel, 32x32 pixels,
- **Automatic positioning** key, Sect. 3.7.4,
- the buttons and adds and removes components to the group.

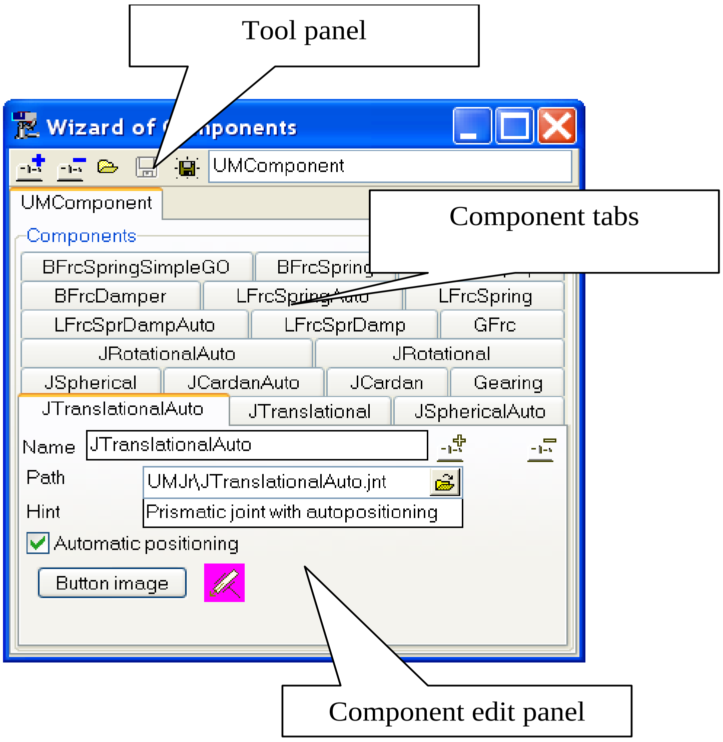


Figure 3.285. Wizard of components

3.7.6.2. Image of a component button

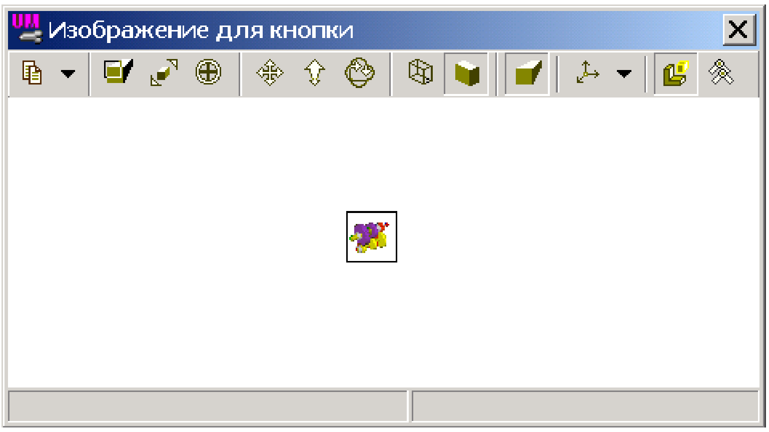


Figure 3.286. Animation window for creating an image for a button

A special animation window can be used for creation of an image for a component button, Figure 3.286. Use the **Tools | Button go** menu command to call the window. This function is available if an active object is available in **UM Input** program. Use the 📋 button to save the image into a file.

3.8. Saving object data

Use the **File | Save as… menu** command or the 💾 button to save the active object for the first time or to make a copy of the object (Sect. 3.4.1). Just here a name is assigned to the object.

To save a modified object use:

- the **File | Save** command of the main menu;
- the *Ctrl+S* hot key;
- the 💾 button.

Data are stored in the *input.dat* file located in the object directory.

3.9. Generation of equations of motion

Universal Mechanism supports two methods for automatic generation of equations of motion: *symbolic* and *numeric-iterative*. Let us consider them more detailed.

Before generation of equations, UM saves the modified object and verifies correctness of the object description.

If the object description contains errors or not full, the program opens the **Summary** tab of the object inspector. The summary contains a list of errors and warnings. Click a line with an error or a warning to go to the corresponding element of the object, Sect. 3.5.2.3.3.

Zero mass and moments of inertia are considered either as warnings or as errors. Use the General tab of the UM option window (Sect. 3.3.1) to change the status.

Symbolic method assumes generation equations of motion as source files in C or Pascal with posterior their compilation by one of the supported external compilers. As a result of compilation, the UMTask.dll appears. This library is used by **UM Simulation** program for numerical integration of equations of motion.

Use the **Tools | Generate equations** command of the main menu to generate and (optionally) compile equations of motion with the help of the built-in specialized computer-algebra system.

Numeric-iterative method assumes generation of equations of motion on each step of numerical integration directly in UM Simulation program. It does not requires compilation of equations with an external compiler.

Let us consider advantages and disadvantages of both methods.

In terms of CPU efforts, the symbolic method is faster. It provides decreasing CPU efforts up to 10-100% for complex (more than 10-20 degrees of freedom) models. For rather simple models CPU efforts for both methods are roughly the same. The symbolic method during generation of source code fulfills its optimization from the point of view of CPU-efforts.

On the other hand the symbolic method of generation of equations of motion expects any external compiler to be installed on the same computer. Universal Mechanism supports Borland Delphi, Borland C++ Builder, Microsoft Visual C++ as external compilers.

At the same time, the numeric-iterative method does not require explicit steps of generation and compilation of equations of motion and seems to be simpler in usage.

For beginner users it is recommended to use the numeric-iterative method of generation of equations of motion as simpler in usage. The symbolic method might be recommended for more experienced users, which work with more or less complex models.

3.9.1. Numeric-iterative method

To set *numeric-iterative* method of generation of equations of motion select the **Object** item in the tree of elements and then set **Generation of equations** to **Numeric-iterative** (see Inspector window in the right part of the constructor window).

3.9.2. Symbolic method

To set *symbolic* method of generation of equations of motion select the **Object** item in the tree of elements and then set **Generation of equations** to **Symbolic** (see Inspector window in the right part of the constructor window). See Sect. 3.3.2 for setup the external compiler.

Generation and compilation of equations of motions are performed within **UM Input** program.

Choice of an algorithm for generation the equations allows optimizing the number of floating-point operation in the equation codes.

The group **Language for output files** lets you to specify the program language for output files. You should select that language which compiler is installed on your computer.

The **Compile equations** checkbox presents an option for the user. If it is checked, the compilation will run right after the successful generation of equations (most often used).

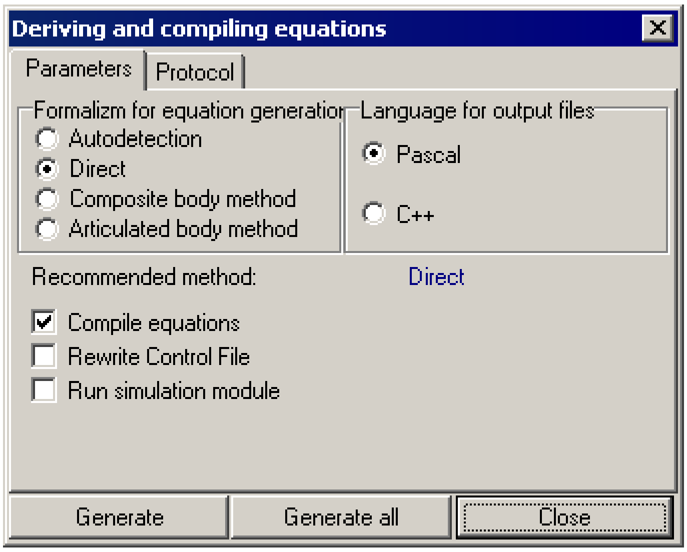
If the **Rewrite Control File** checkbox is checked, the new version of the Control File will replace the old one. The old Control File will be renamed as Cl[NameOfObject].old. If the box is not checked, the new file is created as Cl[NameOfObject].new. This is important if the object description contains external functions or/and the user write its own procedures in the Control file.

Run simulation module if on will start **UM Simulation** program with automatic loading the current model.

The **Generate** button starts the derivation of equations for the active object.

Use the **Generate all** button to derive equations for the object as well as for all external subsystems added to the object.

Use the **Object | Compile equations** or the *Ctrt-F9* hot key to compile the equations if the Control file has been modified but the equations have not been changed.



Formalizm for equation generatio

○ Autodetection
● Direct
○ Composite body method
○ Articulated body method

Language for output files

● Pascal
○ C++

Recommended method: Direct

☑ Compile equations
□ Rewrite Control File
□ Run simulation module

3.9.3. Compilation of equations of motion

If you have chosen *symbolic* method of generation of equations of motion you need to compile generated equations with the help of one of supported compilers. Universal Mechanism supports using Borland Delphi, Borland C++ Builder, Microsoft Visual C++ as external compilers.

To compile equations of motion use **Object/Compile equations** menu item or check **Compile equations** flag in the **Deriving and compile equations** dialog, see Sect. 3.9.2.

To setup external compiler paths select **Tools/Options** menu item. Your further actions depend on what external compiler you are going to use:

Delphi

1. Select **Paths | Delphi** tab.
2. Click **Search Delphi** button.

Borland C++ Builder, Microsoft Visual C++

1. Select **Paths | C++** tab.
2. Click one of the following buttons **Search Visual C** or **Search Borland C++ Builder** depending on which C compiler is installed on your PC.

If UM successfully detects external compiler all paths are set automatically. If not, you should set all paths manually.

3.10. Import data from CAD programs and formats

UM gives many tools for data import from CAD programs and formats. Three methods are available.

1. Use of **API** of a CAD program. In this case, the user opens an assembly in the corresponding CAD program and uses an UM converter for transformation of the assembly into an UM object. This approach is implemented for SolidWorks, Autodesk Inventor, ProE, Unigraphics NX, KOMPAS, see 9_UM_CAD_Interfaces.pdf file.
2. Use of an intermediate commercial viewer of CAD files CADLook for conversion of data from STEP (both AP203 and AP214), IGES, X_T (Parasolid), SAT formats, see Chapter 9 of the user's manual.
3. Direct reading 3ds, STL and wrl files. For instance, an assembly created in Catia v5 should be first exported in VRML (wrl file) or STL, and then this file is read by UM if the corresponding converter is available in the current configuration of UM.

The minimal UM Base configuration includes converter of 3ds files in the UCF format, Sect. 3.10.1, as well as reading UCF files. Other converters are included in UM configuration according to the license agreement.

3.10.1. UCF: UM CAD format

The text format ucf (**UM** **C**ad **F**ormat) has been developed as an intermediate format by conversion of models from CAD with the help of API (Chapter 9 of the user's manual) as well as by direct reading 3ds, stl, vrml files.

In some of the converters the use of the ucf format is hidden, and the user gets an UM model as a result of the import, see conversion of data from SolidWorks, Autodesk Inventor, and KOMPAS3D. In other cases, an ucf file is the result of data conversion, and the UM model is created after reading the file by the **Tools | Import from CAD | UM CAD file** menu command, or by the button, see Sect. 3.10.4 as an example.

3.10.2. UMD format for models imported from CAD

The UMD format of UM model is the basic format in UMLite and UM Express programs. In UM7.0 this format is used as an auxiliary one after conversion of an assembly from CAD.

The **input.umd** file is created simultaneously with the standard **input.dat** file by the first save of the imported model. In comparison with input.dat, the input.umd file contains additional information about the assembly tree, including parts, their names, inertia parameters, images. Taking into account this information, the user can automatically merge parts to bodies with automatic recomputation of the body inertia parameters and image. See Chapter 9, Sect. *Model processing after conversion* for more details. As a rule, the file is not used after finishing the model modification.

3.10.3. Model processing after import from CAD

As a rule, a completion of the model after import from CAD is necessary. The specific model modification include the following tools and steps:

- joining of parts into bodies, see Chapter 9, Sect. *Model processing after conversion*; this operation is available is the user reads the model by the **File | Open *.umd**.
- sometimes evaluation of inertia parameters is necessary; it is recommended to use the UM tool for automatic computation of inertia parameters on body image, Sect. 3.6.9.2;
- change of colors and positions;
- automatic detection of edges if the edges are not imported from CAD, i.e. in the most of cases, Sect. 0.

3.10.4. Reading 3ds files

UM supports import of images developed in 3D Studio. The converter reads a 3ds file and converts is to UCF format, Sect. 3.10.1.

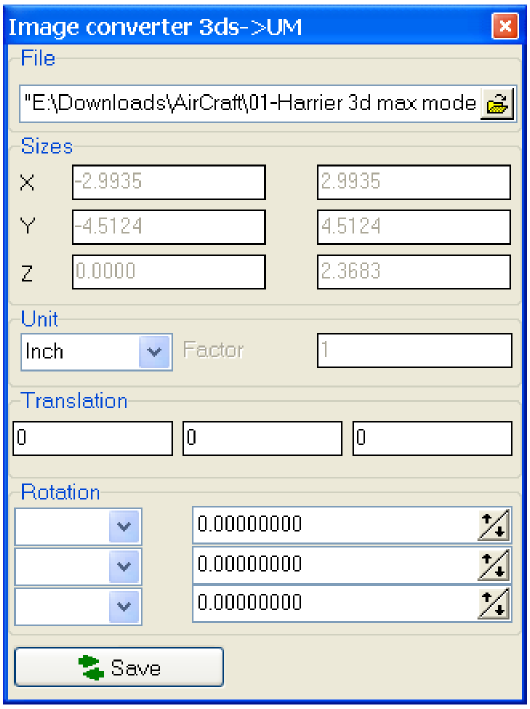


Image converter 3ds->UM

File

"E:\Downloads\AirCraft\01-Harrier 3d max mode

Sizes

X	-2.9935	2.9935
Y	-4.5124	4.5124
Z	0.0000	2.3683

Unit

Inch Factor 1

Translation

0 0 0

Rotation

0.00000000

0.00000000

0.00000000

Figure 3.287. Converter 3ds files

Consider steps of the conversion process.

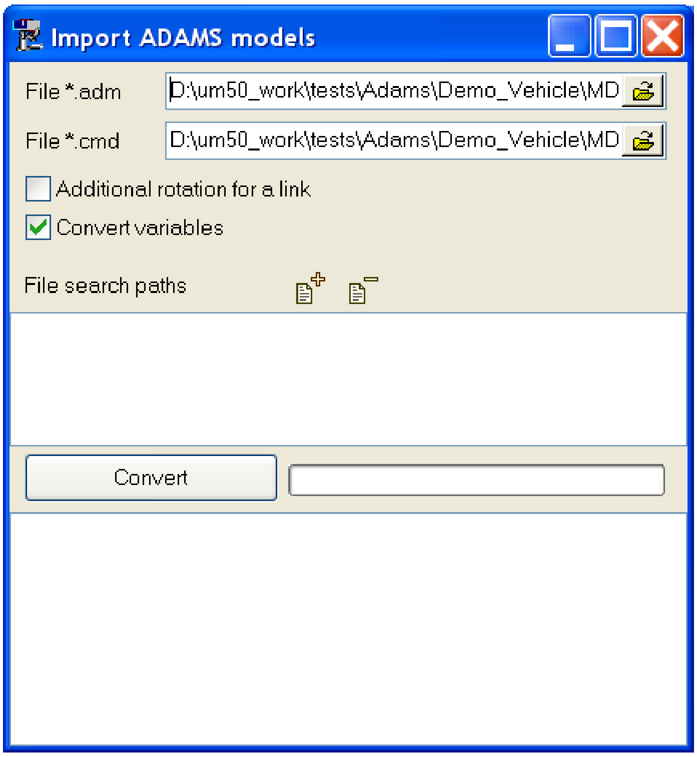
1. Open the converter by the **Tools | Import from CAD | 3ds** menu command or by the button on the tool panel, Figure 3.287.
2. Select a 3ds file by the button. The model **sizes** are computed automatically. The sizes include the minimal and maximal value of coordinates along each of the axes.
3. According to sizes data, a length unit must be selected with the drop-down list. If necessary, a special unit factor can be assigned by the user.
4. A shift and rotation can be assigned to the model, if necessary.

5. Save results of the conversion into an UCF file.
6. Read the UCF file by the **Tools | Import from CAD | UM CAD file** or by the button.
7. If necessary, change the unit, shift and rotation and repeat the conversion.

3.11. Import of MSC.ADAMS models

To convert MSC.ADAMS models in UM format, the following steps are necessary.

1. Load the model in MSC.ADAMS.
2. Use the **File | Export** menu command to save the model with the **File type**: ADAMS/View Command File (*.cmd).
3. Use the **File | Export** menu command to save the model with the **File type**: ADAMS/Solver Data Set (*.adm). The option **Write default values** must be active.
   File must have the same name and must be located in the same directory.
4. Load the UMInput program.
5. Open the converter window by the **File | Import from MSC.ADAMS** menu command, Figure 3.288.



File *.adm D:\um50_work\tests\Adams\Demo_Vehicle\MD

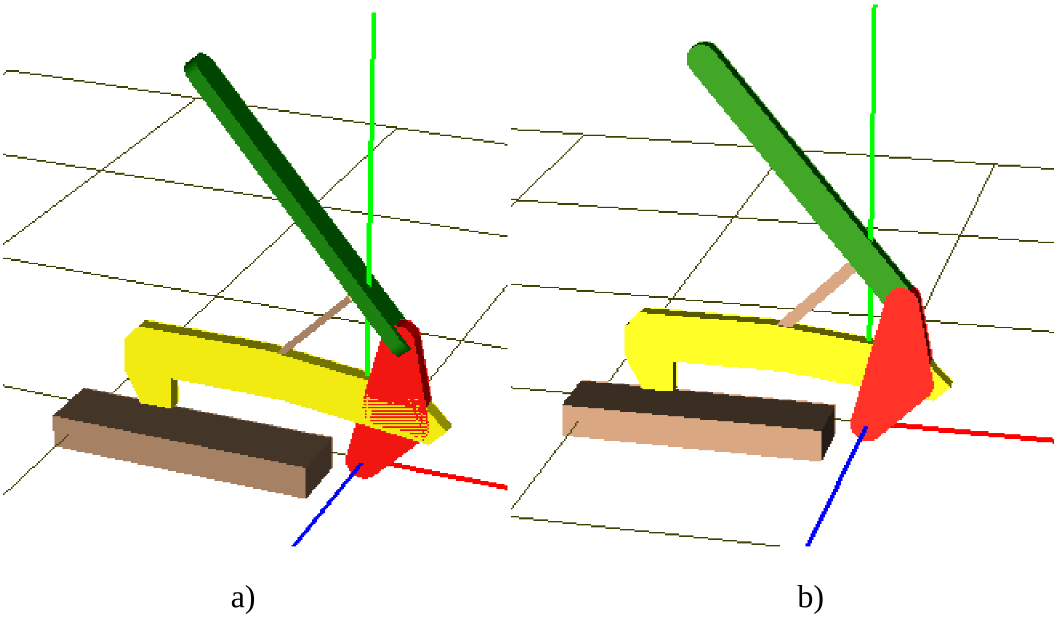
File *.cmd D:\um50_work\tests\Adams\Demo_Vehicle\MD

□ Additional rotation for a link

☑ Convert variables

Figure 3.288. Converter window

6. Select cmd and adm files with the model by the 📂 button.
7. Run the conversion process by the **Convert** button.
8. Verify the model. If some files with images are not found, add paths to the list and repeat the process.
9. If the elements of the “Link” have wrong orientation after the conversion, Figure 3.289a, check the **Additional rotation for a link** option and repeat the conversion, Figure 3.289b).



a) b)

Figure 3.289. Additional rotation for a link

A box in the bottom of the window contains information about faults in the conversion results.

List of convertible elements:

- bodies, their names, images and inertia parameters;
- markers are converted into oriented connection points, Sect. 3.6.9.6.2;
- joints of basic types;
- main force elements (ADAMS Solver notions): bushing, sforce, springdamper, ptcv (point-curve follower);
- curves;
- variables if the **Convert variables** option is checked, Figure 3.288.

Remark 1. There are some limitations in conversion of elements. For instance, the conversion of User functions as well as some kinematic functions and images is not supported. Contact interaction by images is not supported. If some units are different from SI (length - meter, time - second, mass - kilogram), incorrect conversion of some expressions is possible.

Remark 2. Conversion of rail vehicles leads to wrong results because of ADAMS/Rail errors by export data to ADAMS/View and ADAMS/Solver formats.

Experience in use of the model converter proves that even complex models developed in MSC.ADAMS can be successfully transformed to UM format, Figure 3.290, Figure 3.291.

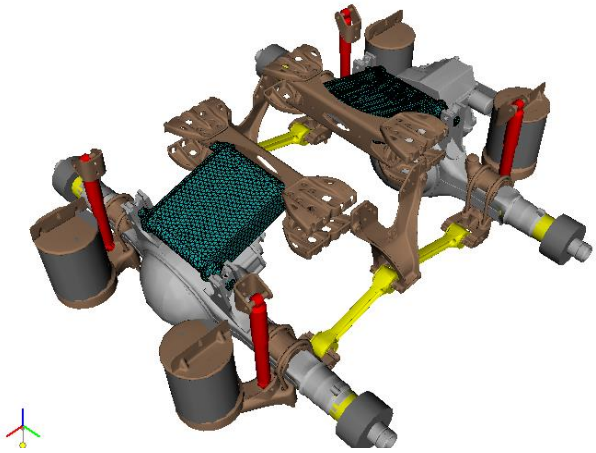


Figure 3.290. Model of a rear truck suspension

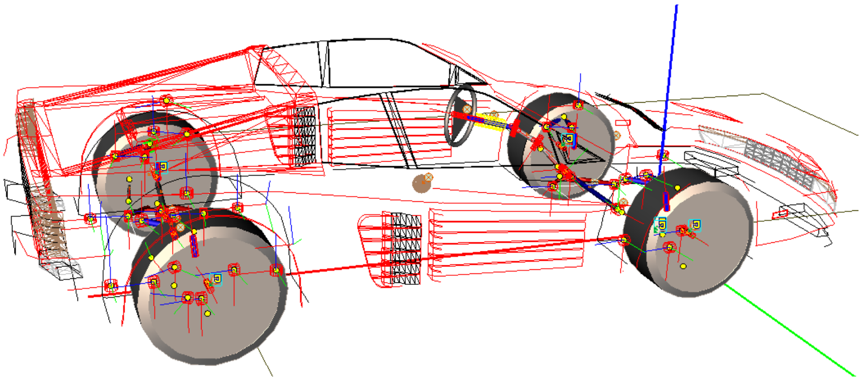


Figure 3.291. Car model