

# sonar

## sonar in a nutshell

program functionality

Version 2.0

Status : for general use

Author : Fritz Leibundgut, L&G Software

© L&G Software 2017



# Inhalt

Wie funktioniert ‚sonar‘ ? .....	5
Leitfaden >>> Modell erzeugen .....	6
Leitfaden >>> Modell-Einstellungen .....	7
Vorgehensweise zur Erzeugung von Primitivkörpern .....	8
Primitivkörper erzeugen .....	9
Interaktive Definition von Primitivkörpern (Primitives) .....	10
Komplexe Körper .....	11
Wie kann man viele gleiche und geordnete Objekte erzeugen? .....	12
Wie erzeugt man Antriebe? .....	13
weiche Materialien .....	14
Zahnrad, Kettenrad .....	15
Plane, Blache, Segel, Folie .....	16
Rotationskörper .....	17
Datenimport / Überblick .....	18
Primitivkörper verbinden (Links) .....	19
Transparente Objekte .....	20
Materialeigenschaften? .....	21
Oberflächenhärte (Objekte) .....	22
Simulationszyklus schematisch .....	23
Simulation >>> Welche Daten können gewonnen werden? .....	24
Die Genauigkeit von Simulationen .....	25
Schwingungsanalyse .....	26
Datensicherung .....	27
Objekt Zustand (physikalisch) .....	28
Modell & Simulation dokumentieren .....	29
Zeitschritt .....	30
Reibung .....	31
Kräfte .....	32
Zwangsbewegungen .....	33
Interaktion (Überblick) .....	34
Interaktions-Kontrolle .....	35
Interaktionskontrolle (Strategie) .....	36
Interaktionskontrolle (ACTIVE, PASSIVE, NO INTERACTION) .....	37
Interaktion (Ablauf) .....	38
Interaktion (Experiment) .....	39
arcball function – moving the model in 3D .....	40
snap function – exact object positioning .....	41
Kontext Menu .....	42
Die Dreifingerregel .....	43
Objekt Variablen .....	44
Weshalb wir das ‚cm-g-µs‘ – Einheitensystem benutzen .....	45
Der Schwanz wedelt mit dem Hund (the tail wags the dog) .....	46
Wie funktioniert ‚Undo‘ und ‚Redo‘ ? .....	47
Mit welchen Dialogen können Links bearbeitet werden ? .....	48
Grenzen der Modellierung (Stand 2017) .....	49

Die in diesem Buch enthaltenen Informationen sind mit keinerlei Verpflichtungen oder Garantien verbunden. Programmentwickler und Autoren übernehmen folglich keine Verantwortung oder Haftung für Folgen jeglicher Art die aus der Benutzung der Software oder des Begleitmaterials entstehen.

Die sonar Software ist Eigentum von L&G Software und als solche einschliesslich aller Teile urheberrechtlich geschützt. Jede Verwendung ausserhalb der engen Grenzen des Urheberrechts bedarf der ausdrücklichen Zustimmung von L&G Software. Dies gilt insbesondere für das Vervielfältigen, Übersetzen und Weiterverwenden des Inhalts oder Teilen davon in elektronischen Datenverarbeitungssystemen. Die Verwendung des sonar Dokumentationsmaterials und das Weiterkopieren in seiner ursprünglichen, unveränderten und vollständigen Form zum Zweck der Benutzung der sonar Software ist allerdings uneingeschränkt erlaubt.

Alle in diesem Buch verwendeten Warenzeichen oder Produktnamen sind Eigentum der jeweiligen Besitzer.

©L&G Software, CH-8352 Elsau, Switzerland

## Vorwort

‘sonar in a Nutshell / Program Functionality’ ist eine Sammlung von Einzelblättern. Verschiedene Merkmale der sonar Software werden in diesem Sinne jeweils auf einer einzigen Blattseite in kompakter Form beschrieben. Diese Einzelblätter haben die Funktion von Kurzanleitungen und können auch als Nachschlagewerk benutzt werden. Für weitergehende Fragestellungen, über die auf den Einzelblättern nicht erschöpfend Auskunft gegeben wird, ist der Benutzer an die betreffenden Manuals der Software verwiesen.

Die Reihe ‚sonar in a Nutshell’ besteht im Moment aus zwei Dokumenten. Das erste Dokument trägt die Bezeichnung ‚sonar in a Nutshell / Dialogs and Tools’.

L&G Software  
Fritz Leibundgut  
Im Sommer 2017

## Hinweis

In diesem Dokument werden gewisse Begriffe z.T. in der deutschen als auch in ihrer englischen Form verwendet, ohne dass jedes Mal mit Sonderzeichen darauf hingewiesen wird. So wird z.B. der Begriff „Primitivkörper“ und die englische Übersetzung „Primitive“ beliebig nebeneinander verwendet. Es wird also gar nicht erst versucht eine konsequent einheitliche deutsche Sprache einzuhalten.

# Wie funktioniert ‚sonar‘?

‚sonar‘ ist ein Mehrkörperdynamik-Simulationsprogramm mit dem Ziel, bei der Entwicklung anspruchsvoller Modelle die Modellierungs- und Simulations-Zeit zu verkürzen und gleichzeitig universell für jede Art von Modell einsetzbar zu sein. Um dieses Ziel zu erreichen, wurde ein pragmatischer Weg beschritten. Dazu betrachten wir die drei folgenden Punkte näher:

- 1. Modell-Erzeugung und –Aufbau mit Primitivkörpern
- 2. Die Verbindung von Primitivkörpern mit sog. Links
- 3. Die Interaktion von frei beweglichen Objekten

## 1. Modellaufbau : .....

Es steht uns eine Kollektion sog. geometrischer Primitivkörper und ein paar weitere geom. Grundfunktionen zur Erzeugung von allg. extruded contours, rotated contours, Freiformflächen, usw. zur Verfügung. Im Moment handelt es sich um insgesamt 18 Primitives. Alle Primitives sind Starrkörper.

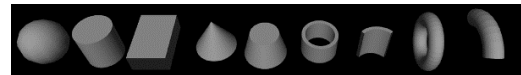


Abb. ein paar Primitives

Ganz ähnlich wie wir im CAD-Bereich die sog. CSG-Methode (Constructive Solid Geometry) verwenden um aus Primitives durch boolesche Kombination komplexere Objekte zu erzeugen, bauen wir in sonar durch die Addition von Primitives komplexere Körper. Umgekehrt werden aber bei Bedarf auch Objekte in eine grössere Anzahl von kleineren Primitives unterteilt, so wie dies bei Finite- Elemente- Programmen mit inneren Netzstrukturen geschieht. Wesentlich ist an diesem Punkt, dass alle Primitives immer elastisch verbunden werden. In diesem Sinne wird z.B. ein biegsamer Drahring aus 32 oder 64 Primitives vom Typ Zylinder konstruiert.

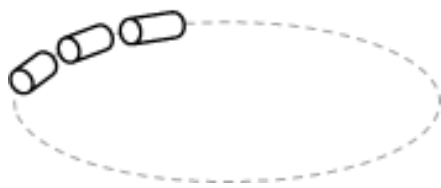


Abb. ein Drahring aus Zylindern

## 2. Links: .....

Jeder Link ist eine geeignete Kombination mehreren Feder- Dämpfungs-Elementen für die verschiedenen Beanspruchungsarten wie Tension, Bending und Torsion. Zusammen bilden diese Elemente das was wir gemeinhin als Stiffness-Matrix bezeichnen. Im Weiteren sind jedem Link ein Material- und ein Breakup-Modell zugeordnet. Diese Modelle steuern das Stress-Strain Verhalten der Verbindung. Ein Link kontrolliert folglich das gesamte elastisch-plastische Verhalten einer Objektverbindung.

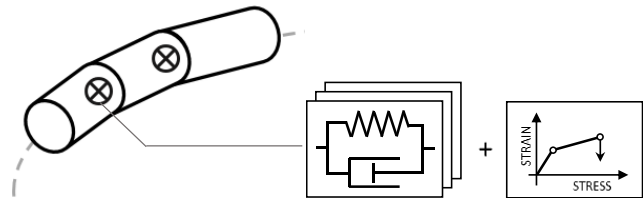


Abb. Die Zylinder werden mit Links verbunden

## 3. Interaktion: .....

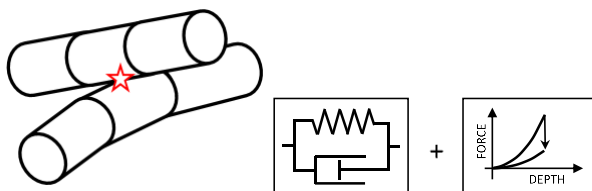


Abb. Die Interaktion der Primitives

Als rein expliziter Code werden alle Kontakte zwischen den Objekten mit abstossenden Kräften als Funktion der Eindringtiefe berechnet. Zu diesem Zweck werden in jedem Zeitschritt jeweils im Zentrum eines Kontaktes ein temporäres Feder-Dämpfungselement und ein Objekt-spezifisches Stoss-Modell installiert. Zusammen berechnen diese Funktionen den gesamten Stossvorgang in der Grössenordnung von hundert Zeitschritten. Dabei wird auch die Hysterese für die Berechnung der absorbierten Energie und die Reibung zwischen den Objekten berücksichtigt.

Da ein sonar Modell letztlich konsequent aus Primitivkörpern aufgebaut wird, existiert für jede mögliche Kombination von Objektbegegnung eine eigene Interaktionsfunktion welche für die gesamte Berechnung des Objektkontaktes verantwortlich ist. Mit  $n=18$  Primitives ergibt das eine Bibliothek von  $n*(n+1)/2 = 171$  Funktionen. Diese Funktionen-Matrix bildet so gesehen den Kernel der sonar Software. Natürlich kann ein Objekt gleichzeitig beliebig viele Kontakte eingehen welche nach dem Superpositionsprinzip alle bilateral gelöst werden. Der gesamte beschriebene Lösungsansatz eignet sich hervorragend für Multiprocessing-Berechnungen auf Mehrkernrechnern oder GPU-cards.

	○	□	⊞	.....
○	★	★	★	.....
□		★	★	.....
⊞			★	.....
...				.....

Abb. Interaction Function Matrix

# Leitfaden >>> Modell erzeugen

## Zweck:

- Welche Schritte sind in sonar-LAB der Reihe nach notwendig um zu einem vollständigen Simulationsmodell zu gelangen?
- Je nach Aufgabe und Komplexität des Modells können ggf. einzelne Schritte übersprungen werden.

## Vorbereitungen:

- Skizze des Modells erstellen
- Überlegungen anstellen, wie die einzelnen Objekte des realen Modells aus ‚Primitives‘ aufgebaut werden könnten
- Darüber nachdenken, wie Rotationsachsen und Verbindungen zwischen den Objekten ausgelegt werden sollen
- Soll das Modell manuell erzeugt oder (ev. teilweise) ein Makro geschrieben werden?
- Soll das Modell in mehrere Module unterteilt und später zusammengefügt werden?

## Objekte erzeugen:

- Wo möglich können vorgefertigte Makros verwendet werden
- Primitives erzeugen, drehen und positionieren (grafisch interaktiv, per Dialog, per Makro)
- wo notwendig, Primitives über Rohdaten erzeugen
- Primitives zu Objekten zusammenfügen (verlinken)
- Objekte zu Baugruppen zusammenfügen (ev. verlinken)

## Objekt Eigenschaften einstellen:

- Masse und Trägheitsmoment der Objekte einstellen
- Oberflächenhärte der Objekte einstellen (interaction constant)
- Verbindungs-Steifigkeit einstellen (Link-constant)
- Interaktions Regeln der Objekte einstellen (active, passive, no interaction)
- Bilaterale Interaktionsregeln, Interaction Rules by Name
- ev. Objekte räumlich fixieren oder ihre Bewegungsfreiheit einschränken
- Das Aussehen der Objekte einstellen (Farbe, Transparenz)

## Kräfte, Antriebe setzen:

- Gravitation
- Reibung (global, unilateral, bilateral)
- Externe Kräfte und Momente (Object Preferences, Formulas, Point Curves, sonar script)
- ev. Kontrollsysteme erzeugen

## Simulations-Vorbereitungen :

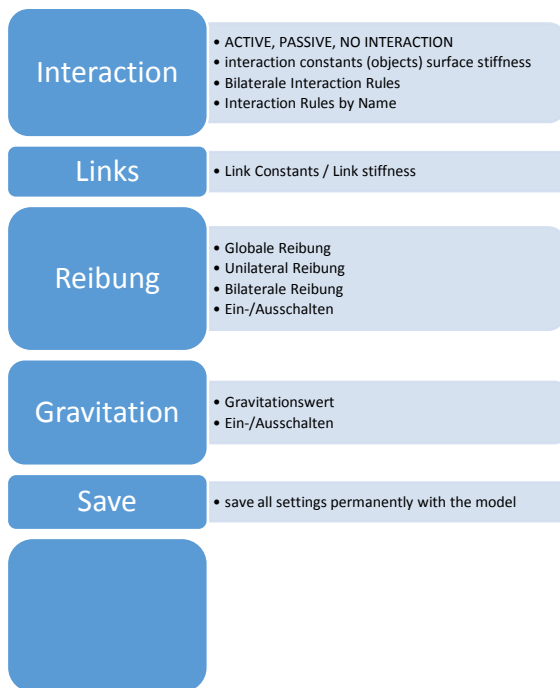
- Anfangswerte für Objekte setzen (Geschwindigkeit, Winkelgeschwindigkeit)
- Sicherungskopien erstellen

## Leitfaden >>> Modell-Einstellungen

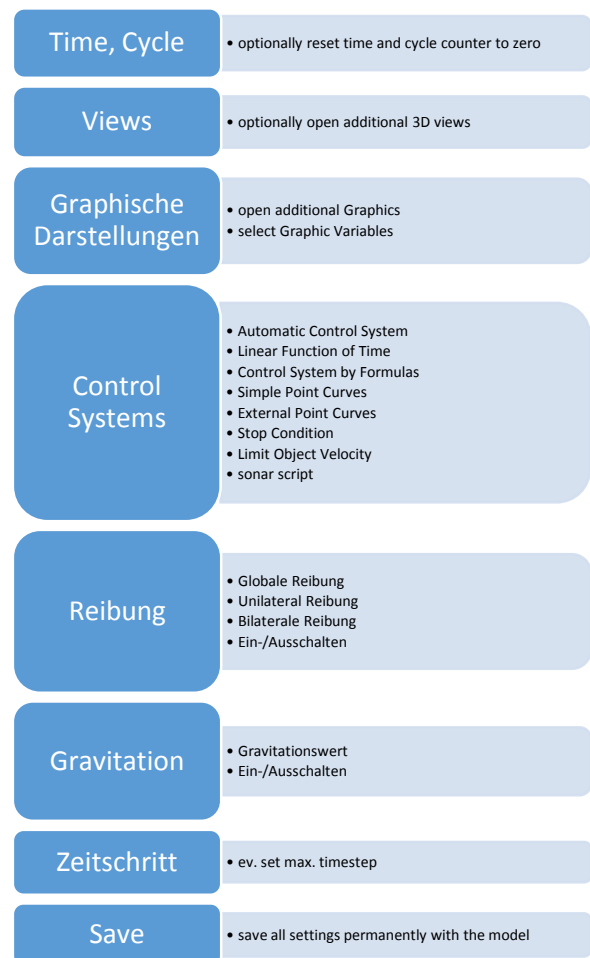
Zweck :

- Dieses Blatt beschäftigt sich mit der Frage, an was man denken muss, wenn man ein komplettes sonar-Modell erstellen will ?
- Die beiden abgebildeten Listen zählen alle Einstellungen auf die mit dem betreffenden Modell gespeichert werden.
- Diese Einstellungen müssen folglich mit jedem Modell nur einmal durchgearbeitet bzw. überprüft werden.
- Auf der linken Seite sind die Punkte für sonar-LAB abgebildet, rechts diejenigen für das Simulationsprogramm sonar-SIM
- Wird mit dem betr. Modell anschliessend eine Simulation durchgeführt, dann kommen noch die Einstellungen hinzu, welche im Modell nicht gespeichert sind und bei jedem Simulationsstart erneut eingegeben werden müssen. Dazu gibt es ebenfalls eine Liste.

### sonar-LAB



### sonar-SIM



### Bemerkungen :

- In der Abbildung sind mögliche Einstellungen aufgezählt, was natürlich nicht heisst, dass der Benutzer auch alle benutzen muss.
- Wie ersichtlich, können gewisse Einstellungen in beiden Programmen vorgenommen werden.
- Diese Blatt ist nur eine Richtschnur und nicht für jeden erdenklichen Fall vollständig.

# Vorgehensweise zur Erzeugung von Primitivkörpern

Anleitung :

- Einfache Primitivkörper können direkt mit dem Toolbar ‚New Primitive‘ grafisch interaktiv erzeugt werden -> 3D Objekte
- Für komplexere Primitivkörper (z.B. Prismen) muss der Weg über Rohdaten genommen werden
- Es gilt:
  - 1. Rohdaten erzeugen (X-Y-Ebene) -> Create Element -> Create Contour -> 2D Rohdaten
  - 2. Create Object (2D -> 3D) -> Extrudieren -> die Objekte befinden sich in ihrer Nulllage (in der Nähe des Koordinatenursprungs)
  - 3. Die Objekte in ihre vorgesehene Raumlage bewegen -> Rotation -> Translation
  - 4. Den Objekten Eigenschaften zuschreiben (Dichte, Farbe, ...)
- Werden Primitivkörper mit Makros erzeugt, dann ist diese Vorgehensweise prinzipiell dieselbe.

# Primitivkörper erzeugen

Zweck:

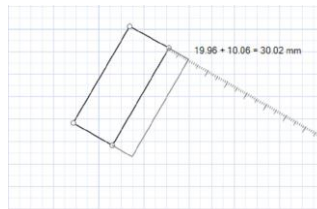
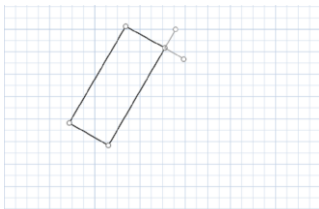
- Welche Möglichkeiten gibt es einen Primitivkörper zu erzeugen?

Anleitung :

- Es gibt mehrere Möglichkeiten ...

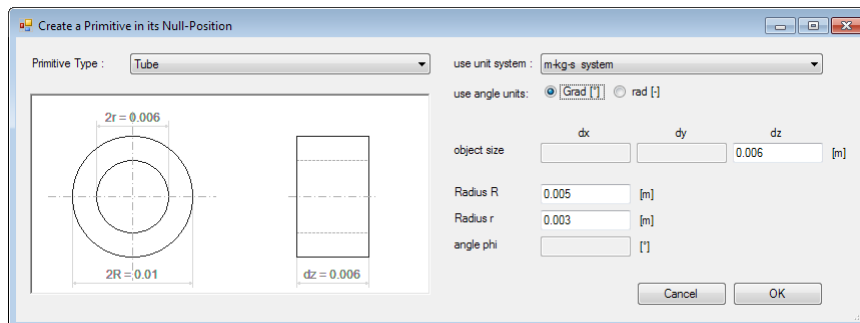
## 1. grafisch interaktiv :

- Siehe: Interaktive Definition von Primitivkörpern (Primitives)
- Nach dem Anklicken eines bestimmten Primitives im Tool ‚New Primitive‘ wird dieses direkt mit der Maus in der Front View gezeichnet, anschliessend angeklickt und manipuliert.



## 2. mit einem Dialog :

- Menu / Construction / New Primitive (Null-Position) -> öffnet den folgenden Dialog
- Sie bestimmen im Dialog den Primitiv-Typ und seine konkreten Dimensionen



## 3. mit Hilfe des Makro Tools :

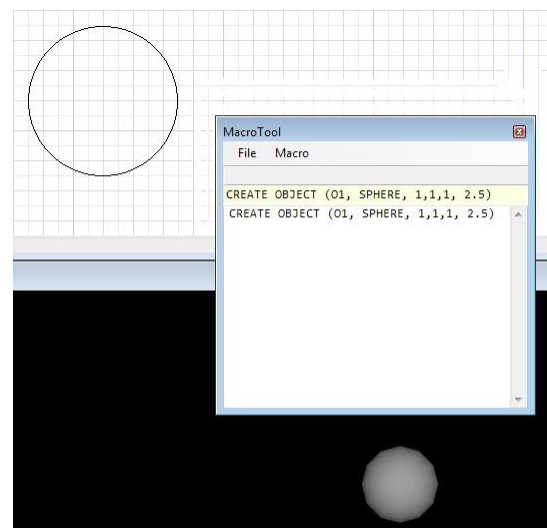
Am rechten Bildschirmrand befindet sich das ‚Makro Tool‘  
Tippen Sie direkt in das Fenster des Tools die folgende Anweisung:

- CREATE OBJECT (O1, SPHERE, 1, 1, 1, 2.5)

Benutzen das Menu im Makro Tool:

- Macro Tool / Makro / Execute

Natürlich verwendet man das ‚Macro Tool‘ normalerweise nicht zur Erzeugung einzelner Objekte, sondern in einem grösseren Kontext zur Erzeugung ganzer Baugruppen. Aber prinzipiell funktioniert auch das.



# Interaktive Definition von Primitivkörpern (Primitives)

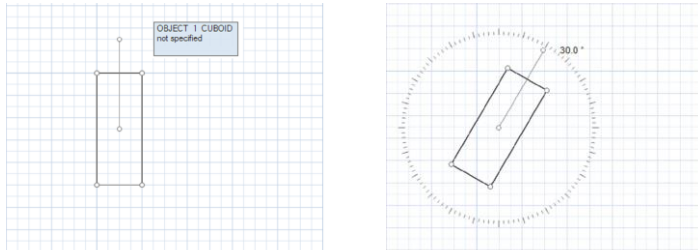
Zweck:

- ,Primitives‘ können auf verschiedenen Wegen erzeugt werden. Diese Anleitung zeigt wie es grafisch interaktiv am Bildschirm geschieht

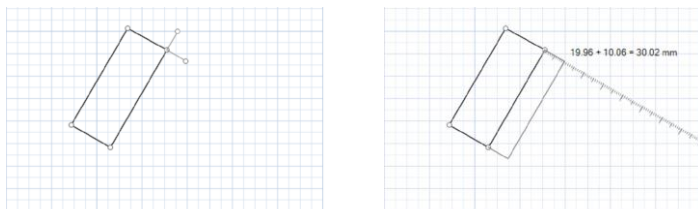
Toolbar ‘New Primitive‘ :



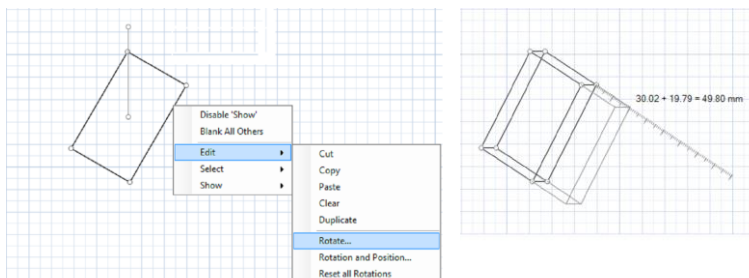
Aktionen am Bildschirm:



Nach der Wahl des Quadersymbols in der Symbolleiste definieren wir einen neuen Quader indem wir in der Frontansicht mit der Maus ein Rechteck auf den Bildschirm zeichnen. Im Quader ist eine Strecke eingezeichnet welche für Drehungen benutzt werden kann. Das oberste Ende dieser Strecke wird mit der Maus gepackt und im Kreis gedreht. Die Drehung erfolgt mit einem Raster deren Schrittweite (in Grad) in den Preferences eingestellt werden kann.



Drückt man die Maus über einem Eckpunkt, dann werden zwei mögliche Streckungs-richtungen eingeblendet. Entscheidet sich der Benutzer in diesem Beispiel für eine Streckung Richtung 4 Uhr, indem er die gedrückte Maus in die betreffende Richtung bewegt, dann werden ihm eine Skala und eine numerische Anzeige eingeblendet.



Mit der rechten Maustaste über der Berandung des Quaders wird das Kontextmenu aufgerufen. Der Befehl ‘Rotate‘ öffnet den entspr. Dialog. Eine Rotation von 30° um die y-Achse ergibt die Quaderstellung in der Abbildung rechts. Natürlich könnte diese Drehung auch wieder grafisch interaktiv, diesmal in der Aufsicht, durchgeführt werden.

Am Ende kann auch hier wieder eine Streckung durchgeführt werden wie abgebildet. Die Verschiebung des Quaders im Raum erfolgt wie üblich durch ‘packen’ am Rand des Quaders.

## Bemerkung:

- Diese Basisoperationen können beliebig oft wiederholt und in allen 2D-Ansichten durchgeführt werden
- Mit diesen grafisch interaktiven Operationen kann folglich die Grösse des Objektes, als auch die Position und Drehung im Raum festgelegt werden.
- Für die Eingabe der Eigenschaften des Objektes bedient man sich wie bei den anderen Definitionsmethoden des Dialogs ,Object Properties‘

# Komplexe Körper

Zweck:

- Wie sollen die einzelnen Primitivkörper untereinander verlinkt werden, wenn diese zu grösseren, komplexeren Körpern zusammengesetzt werden?
- Anordnung und Steifigkeit der Links.

Zu Beachten:

- Letztlich geht es darum, komplexeren Objekten aus mehreren Primitives eine realistische Gesamtsteifigkeit zu geben.
- Diese Steifigkeit wird massgeblich durch folgende Einflussgrössen bestimmt:
  - Anzahl der Links
  - Räumliche Anordnung der Links
  - Art der Links (Link Typ) → wir reden hier vorerst nur von normalen Links
  - Lnk-Konstanten der Links

Anzahl der Links :

- Ein einzelner Link entspricht einem Kugelgelenk, zwei Links entsprechen einer Drehachse, drei Links, welche nicht in einer Linie liegen, entsprechen einer festen Verbindung.
- Folglich sind für eine feste Verbindung zwischen zwei Primitives mindestens drei Links notwendig. Oft werden aber auch vier oder sogar noch mehr Links gesetzt, um die Verbindung zwischen zwei Objekten isotroper zu machen.

Räumliche Anordnung der Links:

- Bei der Wahl der Position der Links gilt es zu beachten, wie diese die Steifigkeit bezüglich der Biegung beeinflussen.
- Eng beieinander liegende Links reagieren empfindlich auf Biegung und Torsion, während weiter auseinander liegende diesbezüglich breiter abgestützt sind. Oft legt man deshalb Links absichtlich weiter auseinander, auch wenn diese dadurch ausserhalb des Materials zu liegen kommen. Man könnte dieselbe Wirkung zwar auch erreichen, indem man einfach die Link-Konstanten entsprechend erhöht (Hebelgesetz). Dies bringt allerdings wieder andere Nachteile mit sich. Denken Sie nur an den Zeitschritt. Links ausserhalb des Materials ist eine ganz normale Sache, die durchaus erlaubt ist.
- Letztlich sollte man die Beanspruchungsarten (Zug, Biegung, Torsion) nicht isoliert betrachten und die entsprechenden Links auch nicht unabhängig anbringen. Wird eine Link-Konstante erhöht, dann werden letztlich alle Beanspruchungsarten bezüglich ihrer Steifigkeit beeinflusst.

Link Konstanten :

- Ein gutes Vorgehen ist deshalb, die folgende Reihenfolge zu wählen
1. Die Link-Konstanten bezüglich ihrer longitudinalen Beanspruchung (Zug-Beanspruchung) berechnen. Bei einem gegebenen Emodul ergibt sich diese Linkkonstante als
  2.  $c_L = E \cdot \dots$
  3. Der gefundene Wert ist der Gesamtwert der Linkkonstanten die wir in einer Querschnittsfläche anbringen.
  4. Würden wir uns dafür entscheiden in diesem Querschnitt 4 Links zu setzen, dann hätte jeder dieser Links eine Link-Konstante von  $c_{Li} = c_L / 4$
  5. Damit ist die Steifigkeit in Zugrichtung abgeglichen. Wir haben nun die Freiheit diese 4 Links in der Querschnittsfläche so anzuordnen, dass auch die Biegebeanspruchung richtig funktioniert. Man kann diese Anordnung der Links berechnen und bekommt als Resultat die folgenden Verhältnisse:

# Wie kann man viele gleiche und geordnete Objekte erzeugen?

Zweck:

- In einem Modell sollen viele gleiche Objekte in einer regelmässigen Anordnung erzeugt werden.
- Das Problem soll allgemein gelöst werden. Die Objekte sollen eine beliebige Form haben

Makros :

- Das Problem kann gelöst werden, indem das gesamte Modell mit einem Makro erzeugt wird.
- Eine zweite Lösung besteht darin das erste Objekt manuell zu erzeugen und anschliessend lediglich das Vervielfältigen und Anordnen mit einem Makro zu automatisieren.
- 1-dimensionale Ketten von Objekten kann man mit einem Makro und einer entspr. Berechnung der Koordinaten als Funktion der Loop Parameter erreichen.
- 2- und 3-dimensionale Anordnungen von Objekten lassen sich mit einem sonar-script-code erzeugen

Pseudocode :

```
LOOP FOR (parameter)
•   Selektiere das Objekt an der Position (x,y,z)
•   Dupliziere dieses Objekt an Ort
•   Rotiere das Objekt ggf. in seine räumliche Drehlage
•   Verschiebe das Objekt an seinen Bestimmungsort (parameter)
END FOR
```

Filter:

- Mit der Loop-Struktur, wie oben angeschrieben, lassen sich vorerst nur reguläre und einfach begrenzte Anordnungen erzeugen
- Mit sog. Filtern lassen sich komplexere Berandungen von Objektstrukturen erzeugen

Pseudocode:

```
LOOP FOR (parameter)
  DO IF (Formel(parameter) < Wert) -- Filter
    •   Selektiere das Objekt an der Position (x,y,z)
    •   Dupliziere dieses Objekt an Ort
    •   Rotiere das Objekt ggf. in seine räumliche Drehlage
    •   Verschiebe das Objekt an seinen Bestimmungsort (parameter)
  END IF
END FOR
```

- Mit einem Filter werden nur diejenigen Objekte erzeugt, welche das Filterkriterium erfüllen
- Mit einem Filterkriterium kann man z.B. quaderförmige, kugelförmige, polyederförmige, usw. Anordnungen erzeugen
- Der einfachste Filter ist eine Ebene. Alle Objekte hinter der Ebene werden nicht erzeugt.
- Kugeln und Zylinder sind weitere geeignete Testberandungen für dieses Ausschlussprinzip
- Durch eine Kombination vieler derartiger Filter können komplexe Anordnungen von Objekten erzeugt werden

Pseudocode:

```
LOOP FOR (parameter)
  DO IF (Filter 1)
    DO IF (Filter 2)
      DO IF (Filter 3)
        . . .
        Objekt erzeugen
        . . .
      END IF
    END IF
  END IF
END FOR
```

## Bemerkungen:

- Eine weitere Möglichkeit ist die Erzeugung eines grossen regulären Haufens regulär angeordneter Objekte mit anschliessender manueller Elimination aller unerwünschter Objekte
- In der Anlage findet man mehrere Makros zu diesem Thema (dichteste Kugelpackungen, Maschinenkette, usw.)

# Wie erzeugt man Antriebe?

Zweck:

- Welche Möglichkeiten gibt es um in einem Modell Antriebe zu realisieren?

Antriebsarten:

- Wir unterscheiden im Folgenden zwei Arten von Antrieben: Rotationsantriebe und lineare, translatorische Antriebe.

Rotationsantriebe:

- Ein Rotationsantrieb wird erzeugt indem auf ein bestimmtes Objekt ein Drehmoment ausgeübt wird.
- Dieses Drehmoment kann direkt auf ein bereits vorhandenes Objekt am Eingang eines Antriebsstrangs ausgeübt werden. Das betreffende Objekt, auf welches das Drehmoment wirkt, benötigt keinen Stator um das Gegenmoment abzufangen (deus ex machina) Dem mechanischen System wird auf diesem Weg einfach ständig Energie zugeführt.
- Ein Rotationsantrieb lässt sich aber auch mit einem zusätzlichen Objekt (Stator) lösen. Zwischen diesem Stator und einem weiteren Objekt (Rotor) werden dann (vom Benutzer) jederzeit entgegengesetzt gleich grosse Drehmomente erzeugt. Der Stator kann dabei ein bewegliches oder ein fixiertes Objekt des Modells sein. Ist der Stator fest verankert, dann ist er im Prinzip obsolet.
- Ist der Stator allerdings ebenfalls ein bewegliches Objekt, dann ist es wichtig, dass der Benutzer dafür sorgt, dass Moment und Gegenmoment immer gleich gesetzt werden, andernfalls wird (actio = reactio) verletzt. Und wenn der Stator beweglich ist, dann kann dies die Bewegung des gesamten Mechanismus in unerwarteter Weise verändern. Es obliegt deshalb der Obhut des Benutzers, sich in diesem Fall um diese Angelegenheiten zu kümmern, weil das Setzen von Antrieben grundsätzlich ein Eingriff in das vermeintliche geschlossene System eines Modells ist. Gibt es zwischen zwei Objekten sog. Links, dann sorgt das System automatisch für Aktion und Reaktion zwischen diesen Objekten. Gibt es aber zwischen einem Stator und einem Rotor keine Verbindungen über welche Kräfte und Momente fließen können, dann muss Aktion und Reaktion zwischen den beiden Objekten künstlich erhalten werden, indem immer auf beide entgegengesetzt gleich eingegriffen wird.
- Man kann hier die Analogie eines einfachen Gleichstrom-Elektromotors anführen. In der Realität wirkt in diesem Elektromotor in jedem Augenblick ein Drehmoment auf die drehbare Spulenachse und ein entgegengesetzt gleich grosses Moment auf die Magnete, welche am Gehäuse befestigt sind. Will der Benutzer diesen Motor physikalisch korrekt nachbilden, dann muss er beide Kräfte bzw. Momente ins Spiel bringen.

Lineare Antriebe:

- Für Linearantriebe steht das sonar-Objekt ‚Actuator‘ zur Verfügung.
- Darüber hinaus können lineare Bewegungen aber auf dem gleichen Weg erzeugt werden wie bei Rotationsantrieben, d.h. mit und ohne Gehäuse und mit den gleichen Möglichkeiten zur Befestigung. Externe Kräfte können direkt auf Objekte angesetzt oder unter Benutzung von Kontrollsystemen zum Einsatz gebracht werden.
- Auch hier gelten die gleichen Voraussetzungen in Bezug auf Aktion und Reaktion wie beim Rotationsantrieb erklärt.

Kräfte und Momente:

- Die eigentlichen Antriebskräfte auf die Objekte können auf verschiedenen Wegen zur Wirkung gebracht werden.
- Konstante Antriebs-Kräfte oder -Momente können als sog. externe Kräfte oder ext. Drehmomente im Dialog ‚Object Preferences‘ gesetzt werden.
- Variable Kräfte oder Momente lassen sich mit einem ‚Control System‘ berechnen und setzen. Hierzu stehen verschiedene Arten von Kontrollsystemen zur Verfügung. Sehen Sie dazu die Kontrollsysteme.

## Bemerkungen :

- Antriebe sind in der Regel Energiequellen und/oder Energiesenken welche dem Modell Energie zuführen oder von diesem abführen.
- Mit der Einführung von Antrieben ist das Modell kein geschlossenes System mehr. Dieser Umstand muss bei der Messung und Auswertung von physikalischen Grössen in Zusammenhang mit der Energie berücksichtigt werden.

# weiche Materialien

Fragen:

- Wie modelliert man weiche Materialien wie Kunststoffe, Gummi, usw.?

Anleitung :

- Wir unterscheiden hier funktionell zwei Arten von weichen Materialien.
  - 1. Objekte die einfach nur als Ganzes an der Oberfläche eingedrückt werden
    - Beispiel: O-Ring
  - 2. Objekte die als Ganzes stark deformiert werden.
    - Beispiel: Lippendichtung
    - Beispiel: Helix förmige Feder

Fall 1 :

- Objekte dieser Klasse können als Starrkörper modelliert werden. Man gibt Ihnen aber eine weiche Oberfläche, so dass sie bei Kollisionen oder bei äusserer Krafteinwirkung relativ stark in das Gegenobjekt eindringen können.
- Die Interaktionskonstante kann ggf. mit einer Hilfssimulation welche nur diese beiden Objekte einschliesst, ermittelt werden

Fall 2 :

- Das Objekt muss aus einer grösseren Anzahl von Primitivkörpern aufgebaut werden die untereinander elastisch verbunden sind.
- Ev. muss die Biegsamkeit kann auch hier mit einer Hilfssimulation verifiziert werden um die Link-Konstanten, welche man näherungsweise berechnen kann, abzugleichen.

# Zahnrad, Kettenrad

Zweck:

- Wie erzeugt man ein Zahn- oder Kettenrad?

Vorgehen:

- Ein Zahnrad wird erzeugt indem man in der Frontansicht (X-Y-Ebene) die Kontur eines einzelnen Zahns zeichnet oder importiert.
- Man achtet darauf, dass der Zahn relativ zum Ursprung (Z-Achse) die richtige radiale Distanz besitzt und so dimensioniert ist, dass dieses Zahnbild beim späteren Duplizieren um die Z-Achse herum nahtlos aneinandergefügt werden kann.
- Mit einer speziellen Funktion wird das bereitgestellt Zahnbild automatisch x-mal dupliziert, um die Z-Achse rotiert und seitlich aneinandergefügt, so dass eine komplette geschlossene Zahnradkontur entsteht.
- Sämtliche Elemente werden neu gruppiert und in eine einzige ‚Contour‘ umgewandelt
- Eventuell wird noch eine zentrale Bohrung in Form eines Kreises gezeichnet
- Die ‚Contour‘ und die Bohrung werden selektiert und extrudiert.

Makro:

```
BEGIN SCRIPT Zahnrad_Z38 -----
IMPORT COLLECTION_LINE_ARC(E1, FILENAME, "C:\sonar Import\Zahnrad_Z38.txt")
CONCATENATE ELEMENTS (E1, RING, 0.0, 0.0, 38) - Zahnbild -> Zahnrad
UNGROUP ELEMENTS (ALL)
GROUP ELEMENTS (ALL) - gruppieren das gesamte Zahnradbild
CREATE ELEMENT(E2, CIRCLE, 0, 0, 0, 0, 0, 1.0, 0.54) - Bohrung im Ursprung
CREATE CONTOUR_LINE_ARC (C1, E1) - aus den Rohdaten eine CONTOUR erzeugen

DESELECT ALL
SELECT CONTOUR(C1) -- Zahnrad
SELECT ELEMENT(E2) -- Bohrung
CREATE OBJECT(O1, PRISM_LINE_ARC, SELECTION, EXTRUSION, 0.72) - Zahnrad extrudieren
SET PROPERTY (O1, DENSITY, 7.8)
-- ggf. weitere Eigenschaften setzen
DESELECT ALL
-- end of script -----
```

Manuelle Erzeugung des Zahnbildes:

- Es wird mit Sicherheit immer so sein, dass das Zahnbild eines einzelnen Zahns in irgendeiner Form bereits vorliegt, sei es weil es analytisch berechnet oder von einem Anbieter solcher Maschinenteile übernommen wurde.
- Es wird also davon ausgegangen, dass eine Linien-Bogen-Kontur eines einzelnen Zahns vorliegt.
- Es gibt demnach folgende Möglichkeiten ein solches Zahnbild zu übernehmen
  - Mühsamer Weg: manuelles Zeichnen von Linien und Bogen mit den sonar-Zeichen-Tools
  - Besserer Weg: Importieren eines Linien-Bogen-Kontur-Files (Textfile) mit dem Dialog ‚Create/Show Contour‘Die folgenden Bilder zeigen von links nach rechts diesen Weg.
- Zum Schluss: Ein Element selektieren und ausführen von : Menu / Functions / Concatenate partial contour to a closed shape

```
ARC
0.0, 0.000018
4.368310e-01, 4.621702e+00
3.298310e-01, 4.630568e+00

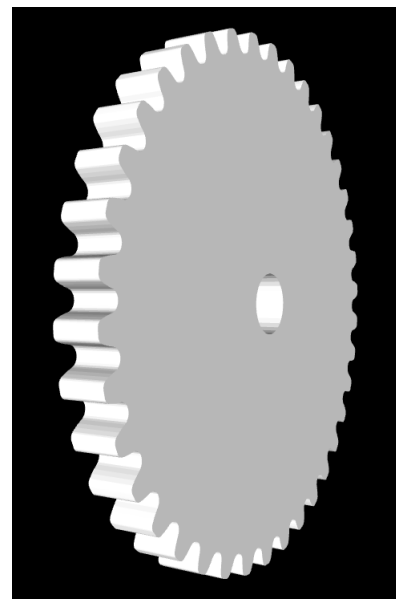
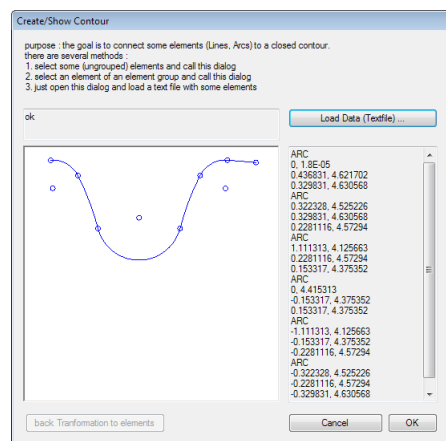
ARC
0.322328, 4.525226
3.298310e-01, 4.630568e+00
2.281116e-01, 4.572940e+00

ARC
1.111313, 4.125663
2.281116e-01, 4.572940e+00
1.533170e-01, 4.375352e+00

ARC
0.0, 4.415313
-1.533170e-01, 4.375352e+00
1.533170e-01, 4.375352e+00

ARC
-1.111313, 4.125663
-1.533170e-01, 4.375352e+00
-2.281116e-01, 4.572940e+00

ARC
-0.322328, 4.525226
-2.281116e-01, 4.572940e+00
-3.298310e-01, 4.630568e+00
```



# Plane, Blache, Segel, Folie

Zweck:

- Wie erzeugt man eine Plane (Zeltplane, gewebeartige, textile Plane, Kunststoffplane)?

Vorgehen :

- In sonar wird eine Plane als Gitterstruktur aus flachen cuboid-Objekten erzeugt welche untereinander elastisch und biegsam verbunden sind.
- Die Verbindung zwischen benachbarten Elementen besteht jeweils aus 2 normalen Links (am Rand) und einem Biege-Link (in der Mitte). Die Verbindung funktioniert folglich wie eine Türe mit 2 Scharnieren und einer Biegefeder in der Mitte.
- Eine Plane kann mit einem Makro gemäss dem folgendem Schema erzeugt werden:
- Der Beispiel- Makro erzeugt eine Plane bestehend aus  $20 \times 30 = 600$  Quader-Objekten mit einer Elementfläche von je  $25 \times 25$  mm.

Makro :

```
BEGIN SCRIPT Plane
-- Planendicke: 2.0 mm
-- Masse: 750 x 500 mm

SET VALUE (NEW_GROUP_NR)

LOOP FOR (J,0,19,1) -- 20 Elemente
  LOOP FOR (I,0,29,1) -- 30 Elemente

    CREATE OBJECT (O1, CUBOID, I*2.5, J*2.5, 0.0, 0, 0, 0, 2.5, 2.5, 0.2)
    SET PROPERTY (O1, GROUP_NR, LAST_GROUP_NR)

    -- Links
    DO IF (I>0) -- X-Richtung
      -- wir setzen zwischen 2 Elementen jeweils 3 Punkte
      CREATE ELEMENT (E1, POINT, I*2.5 - 1.25, J*2.5 - 1.0, 0)
      CREATE ELEMENT (E2, POINT, I*2.5 - 1.25, J*2.5 + 1.0, 0)
      CREATE ELEMENT (E3, POINT, I*2.5 - 1.25, J*2.5, 0)

      -- wir erzeugen einen Link am ersten Punkt E1
      DESELECT ALL
      SELECT ELEMENT (E1)
      SELECT OBJECT (LAST_OBJECT)
      SELECT OBJECT (LAST_OBJECT-1)
      CREATE LINK (K1, NORMAL, SELECTION)

      -- für E2 und E3 analog. Für E3 ersetzen wir ,NORMAL' durch ,BENDING'
      -- mit weiteren Anweisungen ,SET PROPERTY (K1,...) setzen wir die Link-Eigenschaften
    END IF

    DO IF (J>0) -- Y-Richtung
      -- wir setzen zwischen 2 Elementen jeweils 3 Punkte
      CREATE ELEMENT (E1, POINT, I*2.5 - 1.0, J*2.5 - 1.25, 0)
      CREATE ELEMENT (E2, POINT, I*2.5 + 1.0, J*2.5 - 1.25, 0)
      CREATE ELEMENT (E3, POINT, I*2.5, J*2.5 - 1.25, 0)

      -- wir erzeugen einen Link am ersten Punkt E1
      DESELECT ALL
      SELECT ELEMENT (E1)
      SELECT OBJECT (LAST_OBJECT)
      SELECT OBJECT (LAST_OBJECT-30) -- die Zahl 30 korrespondiert mit der Loop-Var.(I)
      CREATE LINK (K1, NORMAL, SELECTION)

      -- für E2 und E3 analog. Für E3 ersetzen wir ,NORMAL' durch ,BENDING'
      -- mit weiteren Anweisungen ,SET PROPERTY (K1, ...) setzen wir die Link-Eigenschaften
    END IF
  END FOR -- I
END FOR -- J

-- mit weiteren Anweisungen ,SET GROUP_PROPERTY (O1, ...) setzen wir die Obj. Eigenschaften
-- rotate and shift Plane to its final position mit Anweisungen der Form
TRANSLATE OBJECTGROUP(O1,...)
ROTATE OBJECTGROUP(O1,...)
-- end of script
```

Bemerkungen :

- Wenn in einem Modell mehrere Planen untereinander interagieren sollen, zum Beispiel weil sie aufeinander zu liegen kommen, dann ist darauf zu achten, dass die interagierenden Planen gleich grosse Elemente besitzen. Ungleich grosse Elemente erzeugen bei der Interaktion inhomogene Interaktions-Kräfte was ein Zerknittern der Planen Oberfläche an der Kontaktfläche mit entsprechenden Fehlfunktionen zur Folge hat.

# Rotationskörper

Zweck:

- Wie erzeugt man einen Rotationskörper?

Vorgehen :

- Ein Rotationskörper wird grundsätzlich hergestellt, indem man einen Profilquerschnitt bereitstellt und diesen um eine Achse rotiert.  
Der Reihe nach:
- 1. Linien-Bogen-Kontur  
In der X-Y-Ebene wird mit den Zeichen-Tools eine geschlossene Linien-Bogen-Kontur erzeugt.  
Die Kontur soll bereits den richtigen Abstand zur X-Achse haben. Man kann sich vorerst vorstellen, dass die Kontur anschliessend um die X-Achse rotiert wird.
- 2. Alle Elemente der Linien-Bogen-Kontur selektieren und gruppieren  
In der Regel benutzt man dazu das Selektionsrechteck um alle Elemente in einem Schritt zu selektieren und wählt anschliessend Menu / Edit / Group / Elements / Selected
- 3. Die Gruppe von Elementen wird in eine ‚Contour‘ umgewandelt. Dazu selektiert man die Gruppe mit der Maus und wählt Menu / Construction / New.Edit Contour... und bestätigt mit ‚OK‘.  
Die Kontur erscheint im ‚Object Tool‘ jetzt im Ordner ‚Contours‘.
- 4. Die Kontur mit der Maus selektieren und in einen Rotationskörper verwandeln mit Menu / Construction / Revolve Contour  
Es erscheint ein Dialog, in dem Sie die Erzeugung des Rotationskörpers nochmals bestätigen oder abbrechen können.  
Falls Sie bestätigen, dann wird der Rotationskörper erzeugt und erscheint in allen Ansichten.
- 5. Das Drehen und Positionieren des Rotationskörpers geschieht anschliessend normal wie mit allen anderen Objekten.

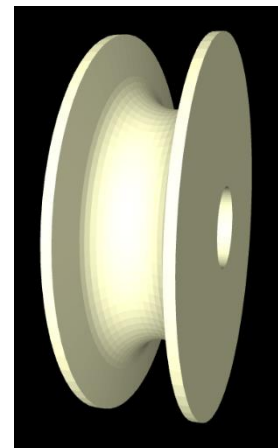
## Beispiel:

Das folgende Makro erzeugt eine Seilrolle.

Beachten Sie, dass das Makro exakt dieselben Aktionen durchführt wie wir sie auch manuell der Reihe nach zu erledigen haben.

```
-----
-- SEILROLLE
CREATE ELEMENT (E1, LINE, 2.0, 1.6, 0, 2.0, 7.25, 0)
CREATE ELEMENT (E2, LINE, 2.0, 7.25, 0, 1.65, 7.25, 0)
CREATE ELEMENT (E3, LINE, 1.65, 7.25, 0, 1.480610, 5.759595, 0)
CREATE ELEMENT (E4, ARC, 0, 6.0, 0, -1.480610, 5.759595, 0, 1.480610, 5.759595, 0, -1)
CREATE ELEMENT (E5, LINE, -1.480610, 5.759595, 0, -1.65, 7.25, 0)
CREATE ELEMENT (E6, LINE, -1.65, 7.25, 0, -2.0, 7.25, 0)
CREATE ELEMENT (E7, LINE, -2.0, 7.25, 0, -2.0, 1.6, 0)
CREATE ELEMENT (E8, LINE, -2.0, 1.6, 0, 2.0, 1.6, 0)

GROUP ELEMENTS(ALL)
CREATE CONTOUR_LINE_ARC(C1, E1)
SELECT CONTOUR(C1)
REVOLVE CONTOUR(O1, C1)
-- end of script
-----
```



# Datenimport / Überblick

Frage:

- Welche Möglichkeiten sind in der ‚sonar‘ Software für den Datenimport aus externen Quellen vorhanden?

Kontur (Rohdaten Import) :

- Konturen sind Linien-Bogen-Kombinationen welche zusammen eine geschlossene Berandung bilden.
- Konturen werden als Querschnittsprofil für den Extrusions-Prozess verwendet.
- Für den Import benutzen wir den Dialog ‚Create/Show Contour‘
- Der Dialog erwartet ein Textfile mit einer 2D-Datenreihe in folgendem Format. Das Textfile wird typischerweise in EXCEL vorbereitet.

LINE	(reserviertes Wort)
x1, y1	(Punkt 1)
x2, y2	(Punkt 2)
ARC	(reserviertes Wort)
x1, y1	(Bogenzentrum)
x2, y2,	(Startpunkt des Bogens)
x3, y3	(Endpunkt des Bogens)

- .....  
Menu / Construction / Create/Show Contour...

Polyline (Rohdaten Import) :

- Eine Polyline ist eine Folge von Punkten welche fortlaufend mit Strecken verbunden werden.
- Die Punkte beschreiben näherungsweise eine 3D-Kurve im Raum.
- Der Import geschieht mit dem Dialog ‚Import Polyline‘
- Der Dialog nimmt die Daten per Copy&Paste aus dem Clipboard entgegen. Alternativ wird mit dem Dialog ein Textfile geöffnet.
- Die Daten werden typischerweise in EXCEL vorbereitet.
- Das Datenformat besteht aus drei Zahlenkolonnen

x1, y1, z1	(dreidimensionale Punkte)
x2, y2, z2	
x3, y3, z3	
.... .... ....	

- Menu / Construction / Import 3D-Polyline

Grid (Rohdaten Import) :

- Benutzen Sie den Dialog ‚Import Grid‘
- Der Dialog akzeptiert zwei unterschiedliche Datenformate um damit zwei unterschiedliche Typen von Netzen zu erzeugen.
- Sehen Sie bezüglich dem Datenformat das entspr. Anleitungsbuch oder das Handbuch.
- Menu / Functions / Import Grid

DXF (Rohdaten Import) :

- Öffnen Sie mit dem Menu-Befehl ‚Import DXF‘ ein DXF-File.
- Das Programm importiert im Anschluss die Daten automatisch, transformiert sie in das interne Format so weit möglich und stellt die Daten in der X-Y-Ebene des Programms sonar-LAB dar.
- Der Benutzer verarbeitet die geladenen Daten in der gleichen Weise weiter, wie er das mit intern definierten Rohdaten tut.
- Menu / File / Import DXF

Merge Model (sonar Daten Import) :

- Die Merge Funktion führt zwei oder mehrere sonar Modelle zusammen.
- In jedem sonar-Modell kann mit der Merge-Funktion ein weiteres sonar-Modell hinzu geladen und damit hinzu addiert werden. Das Resultat einer Merge-Operation ist immer die Summe von zwei Modellen.
- Dieser Vorgang kann beliebig oft wiederholt werden. Damit kann letztlich ein Modell nach dem Baukastenprinzip zusammengebaut werden.
- Menu / File / Merge Model

# Primitivkörper verbinden (Links)

Zweck:

- Wie werden 2 Primitivkörper miteinander verbunden?

Anleitung :

- Als Erstes muss man sich überlegen, von welcher Art die Verbindung sein soll?
  - 1 Link = Kugelgelenk
  - 2 Links = Drehachse
  - 3 Links = feste Verbindung (falls die 3 Links nicht in einer Linie liegen)
- Der klassische Weg einen einzelnen Link zu erzeugen ist mit Hilfe eines Rohdaten-Punktes, welcher den Ort des Links festlegt.
- Beispiel:



- Wir gehen davon aus, dass die relative Position der beiden Objekte zueinander bereits richtig eingestellt ist.
- Wir setzen einen Rohdatenpunkt am Ort wo die beiden Objekte verbunden werden sollen. Dazu gibt es zwei Möglichkeiten:
  1. Menu / Construction / New Point ...
  2. Manu / New Element / Element Type = Point ...
- Mit Hilfe der shift-Taste selektieren wir alle drei Teile, d.h. die beiden Objekte und den Punkt.
- Menu / Construction / Create Link Normal

Sollen für eine feste Verbindung weitere Links zwischen den beiden Objekten gesetzt werden, dann verfährt man völlig analog.

## Bemerkungen :

- Der Parameter ‚NORMAL‘ gibt an, dass ein ganz normaler Link erzeugt werden soll. Darüber hinaus gibt es sog. Biege- und Torsions-Links. Die werden aber an einer anderen Stelle besprochen.
- In einem Makro wird ein normaler Link auf die gleiche Art erzeugt. Die Anweisung heisst:  
`CREATE LINK (K1, NORMAL, O1, O2, E1)`  
Es gilt dabei allerdings zu beachten, dass die beiden Objekte O1 und O2 vorher im gleichen Makro erzeugt worden sein mussten, damit die Referenzen O1 und O2 gültig sind. Das gleiche gilt für den Rohdatenpunkt E1.
- Falls man in einem Makro trotzdem eine Link-Verbindung zu einem Objekt herstellen will, welches zusammen mit einem anderen Makro entstand, dann gibt es in sonar weitere Möglichkeiten:
- Beispiel :  
`CREATE ELEMENT (E1, POINT, x0, y0, z0)`  
`SELECT ELEMENT (E1)`  
`SELECT OBJECT (POINT, x1, y1, z1)`  
`SELECT OBJECT (POINT, x2, y2, z2)`  
`CREATE LINK (K1, NORMAL, SELECTION)`

In diesem Makro erzeugen wir zuerst einen Rohdatenpunkt den wir anschliessend selektieren.

Als nächstes selektieren wir das Objekt welches den virtuellen Punkt (x1, y1, z1) bzw. die betreffende Koordinate enthält.

Dasselbe tun wir mit dem zweiten Objekt

Jetzt sind alle 3 Teile selektiert und wir setzen mit den selektierten Teilen einen Link.

Wir müssen bei dieser Prozedur lediglich darauf achten, dass die Selektion der Objekte mit den virtuellen Punkten eindeutig ist, d.h. wir müssen diese Punkte dort setzen, wo wirklich nur ein einziges Objekt den Punkt enthält.

# Transparente Objekte

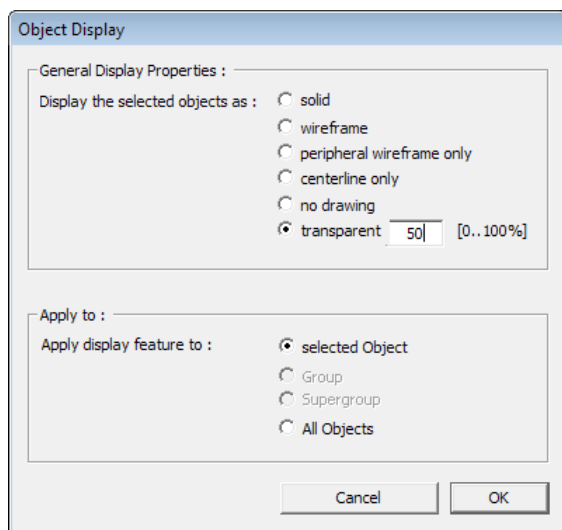
Zweck:

- Wie macht man ein Objekt transparent?

Vorgehen:

Um ein Objekt transparent zu machen gibt es unterschiedliche Methoden. Alle diese Möglichkeiten kann man in sonar-SIM im gleichen Dialog durchführen.

- Wir befinden uns in sonar-SIM
- In einer Ansicht ein Objekt , allenfalls ein Objekt einer Objektgruppe, selektieren
- Ansicht -> Rechtsklick (Objekt) -> Context Menu / Object Display..., oder
- Menu / Display / Object Display...

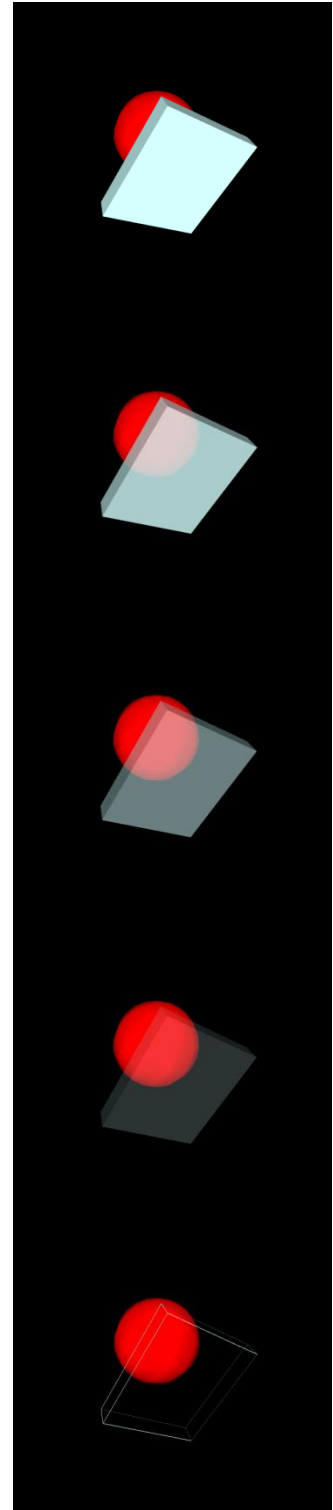


Die Bildreihe zeigt für den dargestellten Quader von oben nach unten die folgenden Einstellungen:

- Solid
- Transparent 20%
- Transparent 50%
- Transparent 80%
- Wireframe

## Bemerkungen:

- Mit einer Transparenz von 100% wäre der Quader nicht mehr zu sehen. Dasselbe Ergebnis bekommt man mit der Option ,no drawing' in den Dialog Einstellungen. Die betreffenden Objekte werden in diesem Fall beim Zeichnen einfach übergangen
- Die Sichtbarkeit eines Objektes hat nichts zu tun mit seinen physikalischen Eigenschaften. Ein unsichtbares Objekt erfüllt weiterhin alle Funktionen die man physikalisch oder geometrisch an das Objekt stellt. Insbesondere nimmt es unverändert an der Simulation teil.
- Die Selektion von ,Wireframe' zeichnet immer ein Drahtmodell des betreffenden Objektes und ist in diesem Sinne 100%-ig transparent. Aber man kann die äussere Form des Objektes trotzdem erkennen, was bei einer Einstellung von ,100% transparent' in der Dialog-Auswahl nicht der Fall ist.
- Man kann die im Dialog gesetzten Einstellungen ggf. auf die ganze Gruppe anwenden, welcher das selektierte Objekt angehört. Dies natürlich nur dann, wenn das Objekt auch Mitglied einer Gruppe ist.
- Die eingestellten Eigenschaften der Transparenz werden mit dem Modell gespeichert.



# Materialeigenschaften?

Zweck:

- Wie ordnet man einer Gruppe oder Supergruppe von Objekten die gewünschten Materialeigenschaften zu?

Vorgehen :

- Wir zeigen zuerst den Ablauf bei Verwendung eines Makro, anschliessend für das manuelle Setzen

Makro :

- Eine Supergruppe selektieren  
Für die weitere Verwendung ordnen wir einer Objektvariablen (z.B. ,O1') ein selektiertes Objekt der betreffenden Supergruppe zu

```
DESELECT ALL
SELECT OBJECT (POINT, 1.0, 2.0, 3.0)
SET VALUE (O1 = SELECTION)
```

- Die Werte für die Link-Konstanten setzen.  
Die Link-Konstanten müssen entweder bereits gesetzt sein oder als Erste gesetzt werden.  
Am einfachsten setzt man diese Werte automatisch durch die Angabe eines Emoduls (z.B. Stahl = 2.1).

```
SET SUPERGROUP_PROPERTY(O1, C_LINK, E_MODUL, 2.1, 0.25)
```

- Das Materialmodell auswählen  
Wir setzen hier beispielhaft das Materialmodell Nr. 005 (isotropic hardening). Dieses Materialmodell ist eines der am meisten benutzten Modelle und hat das Aussehen wie nebenstehend abgebildet.

```
SET SUPERGROUP_PROPERTY(O1, C_LINK, MATERIAL_MODEL, 5)
```

- Als weitere Schritte setzen wir, wie in der Grafik angeschrieben, die Werte für  $\sigma_0$  und  $\sigma_{\max}$

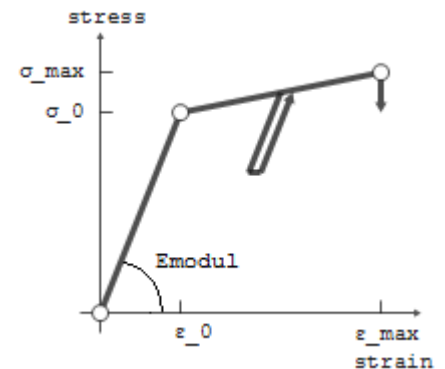
```
SET SUPERGROUP_PROPERTY(O1, C_LINK, STRENGTH_CALC, SPECIFIC, 5E-3, 0.25)
SET SUPERGROUP_PROPERTY(O1, C_LINK, STRENGTH_ULTIMATE, SPECIFIC, 5.5E-3, 0.25)
```

- Bei Überschreiten des Wertes  $\sigma_{\max}$  können verschiedene Aktionen angestossen werden. Es sind folgende Werte möglich:  
NO\_ACTION, BREAKUP, COLORING, SIGNAL

```
SET SUPERGROUP_PROPERTY(O1, C_LINK, OVERLOAD_ACTION, BREAKUP)
```

- Wir setzen den Wert für die maximale Dehnung ,  $\epsilon_{\max}$ ' in Prozenten. Für eine maximale Dehnung von 4% würden wir folgende Anweisung setzen

```
SET SUPERGROUP_PROPERTY(O1, C_LINK, STRAIN_LIMIT, PERCENT, 4.0)
```

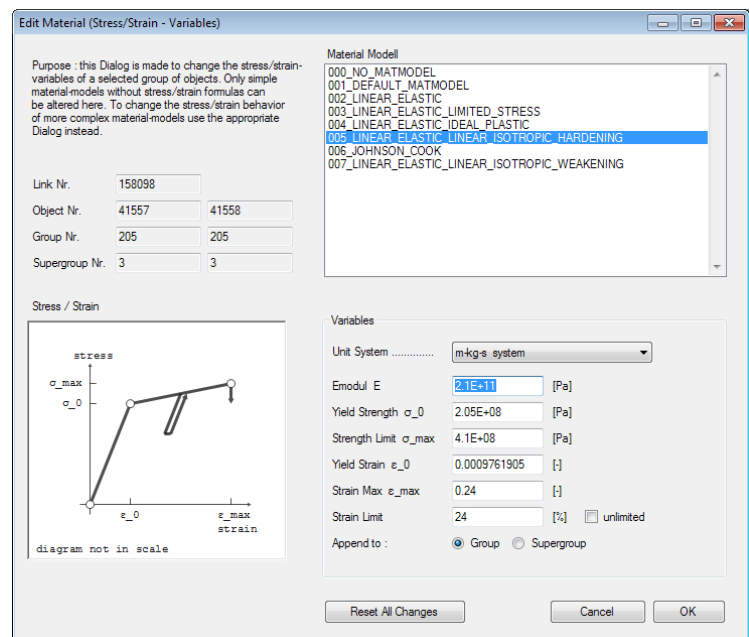


Manuell :

- Der manuelle Ablauf ist analog. Er spielt sich zur Hauptsache in folgendem Dialog ab:  
Menu / Functions / Edit Material / Stress-Strain-Variables...
- Alle Eingaben gemäss dem oben dargestellten Ablauf lassen sich in diesem Dialog in die entsprechenden Felder eintragen.
- Je nach Wahl des Materialmodells stehen eventuell weniger Eigenschaften zur Verfügung, bzw. müssen weniger Eigenschaften gesetzt werden.
- Wählt der Benutzer zusätzlich die Option ,unlimited', dann ändert sich an der Kurvenform grundsätzlich nichts. Es gibt einfach keine Bruchgrenze. Das Material kann über die ausgeschaltete Bruchgrenze hinaus weiter gedehnt werden. Die Kurve wird dazu unbegrenzt linear extrapoliert.

Bemerkungen :

- Der Emodul ist die Steigung der Kurve im Hook'schen Bereich. Dies gemäss der Definition des Emoduls als  $\sigma = E \cdot \epsilon$



# Oberflächenhärte (Objekte)

Zweck: Die Einstellung der Oberflächenhärte eines Objektes und die Auswirkungen bei zu weicher Oberflächenhärte.

Erklärungen:

- Die Oberflächenhärte entscheidet darüber wie hart oder wie weich zwei Objekte miteinander kollidieren.
- Die Kollision kann hart wie Stahl oder weich wie Gummi sein.
- Ein Beispiel für eine hohe Oberflächenhärte ist eine Stahlkugel. Ein Tennisball wäre ein Beispiel für ein weiches Objekt.
- Wird ein Tennisball mit voller Wucht an eine Betonwand geschlagen, dann zeigen Filmaufnahmen in Zeitlupe wie der Ball bei der Kollision flach gedrückt wird. Dasselbe geschieht grundsätzlich bei jeder Kollision. Bei Stahlkugeln ist einfach die Grösse der Deformation viel geringer.



Abb.; Ein Tennisball zum Zeitpunkt der maximalen Kollisionstiefe. Links in der Realität, Rechts in der Simulation mit Starrkörpern

- Die Oberflächenhärte kann man sich als Federkonstante einer virtuellen Druckfeder vorstellen, welche bei der Kollision der Bewegung bzw. dem Eindringvorgang entgegenwirkt.
- In ‚sonar‘ übernimmt die sog. Interaktionskonstante eines Objektes die Funktion dieser Grösse.
- Die Interaktionskonstante kann rein qualitativ (sehr hart, hart, ..., weich, usw.) oder als Zahlenwert eingestellt werden.
- Die Einstellung der Interaktionskonstante erfolgt im Dialog ‚Edit Object Properties‘

Menu / Construction / Edit Object / Properties... / Panel Interaction / Interaction Const.

- Die Interaktionskonstante wird möglichst an die Realität angepasst, um die Eindringtiefe in der Simulation an die wirklichen Verhältnisse anzupassen.
- Allerdings ist das Setzen des korrekten Wertes für die Interaktionskonstante viel weniger wichtig als man denken könnte. Auch wenn der gesetzte Wert um einen Faktor 10 daneben liegt, erzielt man in der Regel gleichwertige Resultate. Objekte die beispielsweise zu weich sind führen dazu, dass die Kollisionen mit ihnen zu grösseren Eindringtiefen führen und etwas länger dauern. Dies wirkt sich allerdings selten auf die nachfolgenden Bewegungen des Systems aus. Die Auswirkungen dieser minimalen Verschiebungen in der Flugbahn oder in der Zeitskala sind meistens geringer als die Fehler die man sich mit einer Simulation durch andere Unzulänglichkeiten einhandelt.
- Erst wenn die Interaktionskonstante um mehrere Grössenordnungen zu weich ist, entstehen wirkliche Probleme in der Simulation. Es kann in diesem Fall passieren, dass die maximale Eindringtiefe so gross wird, dass sie geometrisch nicht mehr eindeutig einem bestimmten Oberflächenabschnitt des betreffenden Objektes zugeordnet werden kann und der ganze Interaktionsvorgang im eigentlichen Sinne die Orientierung verliert. Im Extremfall kann ein Objekt ein anders durchdringen, weil die maximale Eindringtiefe bzw. der Umkehrpunkt der gegenseitigen Bewegung überhaupt nie erreicht wird. In Bezug zur Abbildung oben mit dem Tennisball würde dieser bis in die Mitte der Betonplatte abgebremst und anschliessend auf der anderen Seite wieder hinausbeschleunigt, weil die Relativgeschwindigkeit zwischen Ball und Wand bis in die Plattenmitte noch nicht Null geworden ist.
- Es gibt im Dialog ‚Edit Object Properties‘ wo wir die Interaktionskonstante editieren, eine lineare und eine quadratische Interaktionskonstante. Die eigentliche berechnete abstossende Kraft zwischen den beiden Objekten ist immer die Summe von beiden. Ohne die Dämpfungsterme, welche auch noch hinzukommen, berechnet sich diese Kraft vereinfacht und grundsätzlich wie folgt:

$$F = c_1 * h + c_2 * h^2 \quad (c_1, c_2: \text{Interaktionskonstanten, } h: \text{Interaktionstiefe})$$

- Der quadratische Term ist besonders bei sehr dünnen Objekten wie Folien nützlich, weil dort wenig Interaktionstiefe zur Verfügung steht.
- Bei der Kollision von Objekten mit unterschiedlichen Interaktionskonstanten hat das weichere Objekt den massgeblichen Einfluss. Schiesst man eine Stahlkugel in einen Brotteig, dann spielt die Oberflächenhärte der Kugel keine Rolle mehr. Der Brotteig bestimmt, wie tief die Kugel eindringt. Die Interaktionskonstante von zwei gemischten Objekten ‚a‘ und ‚b‘ wird in ‚sonar‘ berechnet als:

$$C_1(a,b) = 2 * (c_1(a) * c_1(b)) / (c_1(a) + c_1(b))$$

# Simulationszyklus schematisch

Zweck:

- Das abgebildete Schema zeigt schematisch, wie ein einzelner Simulationszyklus in der sonar Software durchgeführt wird.
- Ein Simulationszyklus berechnet die gesamte Physik eines Modells für einen Zeitschritt  $\Delta t$
- Am Ende eines Zyklus befindet sich das Modell in einem neuen Zustand und die Berechnungen beginnen wieder von vorne.

Berechnungsablauf (1 Zyklus) :

- Die meisten Berechnungen werden auf möglichst viele Rechenkerne verteilt (Multiprocessing)

## START ZYKLUS

- EXECUTION CONTROL FUNCTIONS
- DIVERSE VORBEREITUNGEN
- BERECHNUNG DER OBJEKTUMGEBUNGEN
- KRAEFTE
  - INTERACTION
  - LINKS
  - ZENTRALGRAVITATION
  - EXTERNE KRAEFTE
  - VISKOSITAET
- CONTROL SYSTEMS
  - CONTROL SYSTEMS (AUTOMATIC)
  - CONTROL SYSTEMS (LINEAR FUNCTIONS)
  - CONTROL SYSTEMS (BY FORMULAS)
  - CONTROL SYSTEMS (SONAR SCRIPT)
- INTEGRATION
  - BESCHLEUNIGUNG
  - GRAVITATION (LABORATORY)
  - GESCHWINDIGKEIT
  - POSITION
- DATEN AQUISITION
- ZEITSCHRITT
- GRAFIKFUNKTIONEN
- DATENAUFZEICHNUNGEN
- LOGFILE FUNKTIONEN
- PERIODISCHE SICHERUNG (MODELLZUSTAND)

## END ZYKLUS

# Simulation >>> Welche Daten können gewonnen werden?

Zweck:

- Welche Möglichkeiten gibt es, während einer Simulation Daten und Informationen zu gewinnen?

## Datengewinnung:

3D-Ansichten:

- Farbliche Darstellung der Objekte gemäss ihrem Zustand im Hinblick auf eine beliebige physikalische Variable (Zugspannungsverlauf, Geschwindigkeitsverlauf, usw.)

Grafik:

- Darstellung : Var. X / Var. Y

Daten:

- Export: Aus jeder geöffneten Grafik heraus
- Export Format: Var.X / Var.Y als Textdatei mit 2 Zahlenkolonnen im Format (Var.x, TAB, Var.y, RETURN)
- Export Intervall: Die Daten werden in festgelegten Zeit-Intervallen (Simulationszeit) exportiert.
- Import: Die Daten können direkt mit EXCEL geladen und zu Grafiken weiterverarbeitet werden

Filme:

- Format: Filmaufzeichnungen der 3D-Ansichten mit der eingebauten Quicktime Technologie von Apple (Quicktime Movie).
- Grösse: Export vorgegebener Filmformate (640x640, 1280x640, 640x1056)
- Mehrfach-Filmaufzeichnungen aller offenen 3D-windows
- Weiterverarbeitung der Filme nach Simulationsende mit der Software ‚Quicktime Pro‘

Screenshots:

- Tastatur -> Taste Print Screen -> Bild wird in Clipboard gespeichert -> Programm Microsoft Paint -> Paste -> Weiterverarbeiten

Logfiles:

- Diverse Logfiles speichern und dokumentieren weitere Informationen (Events, Dokumentationsunterlagen, Test-Unterlagen)

Moduldaten:

- Die meisten Zusatzmodule zeichnen weitere Daten auf

# Die Genauigkeit von Simulationen

Fragen:

- Wie genau sind Simulationen wie sie mit ‚sonar‘ durchgeführt werden?
- Durch was und auf welchem Weg entstehen Abweichungen von der Realität?

Antworten:

- Abweichungen entstehen nicht, weil die Berechnungsmethode wie sie in sonar durchgeführt wird, grundsätzlich fehleranfällig wäre oder weil die Methode der Integration der Bewegungsgleichungen in Millionen von Rechenzyklen zu Abweichungen führt.
- Abweichungen entstehen dort, wo der Mensch mit Vereinfachungen oder mit ungenauen Zahlen in das System eingreift.
- Abweichungen entstehen hauptsächlich in den beigezogenen physikalischen Gesetzen zur Berechnung von zusätzlichen Kräften wie
  - Reibung,
  - Kollisionseigenschaften, Stossabsorption,
  - Materialeigenschaften (Steifigkeiten)
  - elastisch-plastische Eigenschaften (Spannungs-Dehnungs-Verhalten)
- Besonders in Simulationen mit plastischen Verformungen von Materialien wird die Möglichkeit falsche Werte einzusetzen grösser.
- Deshalb lässt sich die Frage nach der Grösse der möglichen Fehler in Simulationen nicht absolut sondern nur relativ beantworten.
- Die Genauigkeit einer Simulation ist letztlich so gut wie die Wahl der verwendeten Materialparameter.

Bemerkungen:

- Es ist eine gute Idee, Simulationen mit ungenauen oder unbekannten Materialangaben mit mindestens drei Varianten zu berechnen.
  - mit pessimistischen Materialwerten
  - mit durchschnittlichen Materialwerten
  - mit optimistischen Materialwerten

# Schwingungsanalyse

Zweck:

- Die folgenden Erklärungen streifen die Themen Eigenschwingungen, Eigenwerte, Eigenfrequenz, deterministische und chaotische Schwingungen

Erklärungen:

- Für einen rein expliziten Code wie ‚sonar‘ gibt es keinen direkten Weg, die Eigenfrequenz eines deterministischen schwingenden Systems in Form einer Zahl zu ermitteln.
- In einem expliziten Code kann man Eigenschwingungen in einem Modell aber auf die gleiche Weise anregen, wie man das mit einem entsprechenden Experiment auch durchführt: Mit einer Frequenz in der Nähe der Eigenfrequenz.
- Auch in ‚sonar‘ gilt: Je näher die Frequenz der Anregung an die Eigenfrequenz des schwingenden Systems herankommt, umso höher wird die Schwingungsamplitude.
- In ‚sonar‘ ergeben sich Eigenschwingungen folglich auf ähnlich natürliche Art wie in einem realen Experiment
- Darüber hinaus kann man in einem expliziten Code wie ‚sonar‘ aber auch chaotische Schwingungen untersuchen. aber
- In einer sonar-Simulation können Eigenschwingungen, wie auch immer diese entstehen, grafisch aufgezeichnet werden. Auf diesem Weg bekommen wir somit auch Informationen über die ‚Eigenfrequenz‘ usw.

# Datensicherung

Zweck:

- Die folgenden Punkte sind Empfehlungen um eine möglichst hohe Datensicherheit zu gewährleisten und den Verlust von Daten zu minimieren.

Hilfestellungen und Empfehlungen :

sonar-LAB :

- Das Programm lässt eine automatische periodische Modellsicherung zu. Die Funktion wird in den Voreinstellungen des Programms aktiviert.
- Der Benutzer gibt die Häufigkeit der Datensicherung selbst vor.
- Falls diese Funktion deaktiviert ist, empfehlen wir dem Benutzer, seine Arbeit periodisch unter einem neuen Namen zu sichern. Typischerweise geschieht das nach dem Schema: filename\_001, filename\_002, usw., d.h. man speichert das Modell in Arbeit mit einer stetig aufsteigenden Nummer. Dieses Vorgehen hat den Vorteil, jederzeit wieder auf eine ältere Version zurückgreifen zu können.
- Wir empfehlen, wichtige Zwischenschritte in der Modell-Erzeugung (sog. Meilensteine) definitiv unter einem entsprechenden Namen abzuspeichern. Eventuell wird sogar eine zweite Kopie eines Meilensteins angelegt.

sonar-SIM :

- Das Simulationsprogramm bietet die Möglichkeit, den Fortgang einer Simulation periodisch zu speichern. Diese Sicherungsmöglichkeit wird in den Voreinstellungen aktiviert und ist ‚per Default‘ eingeschaltet.
- Diese Sicherungskopien sind dazu gedacht, Programmabstürze und Stromausfälle abzufangen und die Möglichkeit zu bieten, eine Langzeitsimulation ab dem Punkt der letzten Sicherung fortzusetzen.
- Diese automatischen Sicherungen geschehen abwechselungsweise unter zwei verschiedenen Namen. Dies für den Fall, dass ein Computer direkt beim Sichern von einem Stromausfall überrascht wird. Die ältere der beiden Sicherungskopien dürfte dann intakt bleiben.
- sonar-SIM legt bei jedem Programmstart automatisch eine Kopie des Zustandes des Modells an, wie er vor dem Start vorliegt. Der Benutzer ist jederzeit in der Lage, zum Zustand des Modells vor der Simulation zurückzukehren, wenngleich er wahrscheinlich selbst auch Kopien dieses Zustandes angelegt hat.

Bemerkungen :

- Speicherplatz ist heutzutage im Überfluss vorhanden. Es macht keinen Sinn Speicherplatz auf Kosten der Sicherheit zu sparen.
- Kopien werden relativ schnell und unproblematisch erstellt. Eine erledigte Arbeit wegen Datenverlust zu wiederholen ist nervenaufreibend.
- Immer wenn man sich die Frage stellt; ...“soll ich wieder sichern?“,... sollte man sich in Erinnerung rufen, wieviel Arbeit verliert man, wenn gerade jetzt in diesem Moment ein Stromunterbruch geschieht?

# Objekt Zustand (physikalisch)

Zweck: Wie und mit welchen Mitteln kann der aktuelle physikalische Zustand eines Objektes in einem Modell während einer Simulation eingesehen und dargestellt werden?

Möglichkeiten:

- Dialog Object Properties (Menu / Construction / Edit Object / Properties...)
  - Physics (static)
  - Physics (dynamic)
  - Ext. Forces
- Grafische Darstellung
  - physikal. Variable als Funktion der Zeit
  - physikal. Variable als Funktion einer anderen Systemvariablen
- Datenaufzeichnung
  - Grafische Darstellung + Record Data -> File
- Dialog (physikalische Variablen)
  - Darstellung der physikal. Variablen eines ausgewählten Objektes
- Quicktime Movie
  - Dynamische farbliche Darstellung der Objekte mit einer physikalischen Variablen
- Print Report
  - Die wichtigsten physikalischen Variablen einer ausgewählten Objektgruppe zusammengefasst in einem Report
- Edit Material / Stress/Strain Variables...
  - Darstellung von gesetzten und berechneten Materialvariablen
- Dialog Display Group Properties
  - Die aus den Objekten einer Objektgruppe berechneten Gesamt- oder Durchschnittswerte von physikal. Variablen.

# Modell & Simulation dokumentieren

Zweck:

- Welche Mittel stehen zur Verfügung um ein erstelltes Modell zu dokumentieren?
- Aus einer Anzahl Hilfsmitteln werden hier zwei wichtige Dokumente herausgegriffen.

1. Physikalische Eigenschaften:

- Es wird eine Datei erstellt, welche für alle Gruppen und Supergruppen die massgeblichen Eigenschaften auflistet

2. Simulations-Ereignisse:

- Es wird eine Liste erstellt, mit den wichtigsten Ereignissen in Zusammenhang mit Gruppen, Supergruppen und ihren Materialmodellen (Überschreitung der Streckgrenze, Materialbruch, usw.)

Zugriff:

1. sonar-LAB / File / Print Report (physical properties)
2. sonar-SIM/ Edit / Preferences / Simulation / Logfile Events

```
file = C:\sonar Models\72. Debris Fence Sochi\6. final assembly\Szenario-1\Szenario-1_v8\model-17_Sim.135_iActStützen=on_v=100kmph_noBreakup.snir
date = 08.02.2015 15:17:07

-----
supergroup 1
-----
number of objects in supergroup = 301
sample object in supergroup = 34
link Const. = [0.19811320754717...1.115625]
material model = MATMODEL_005_LINEAR_ELASTIC_LINEAR_ISOTROPIC_HARDENING
young modulus (Emodul) = 210 [GPa]
yield strength (sigma_0) = 270 [MPa]
max. strength (sigma_max) = 470 [MPa]
yield strain (epsilon_0) = 0.00128571428571429 [-]
max. strain (epsilon_max) = 0.24 [-]
critical strain = 24 [%]
interaction Const. = [1...1]
interaction mode = PASSIVE
total mass = 31.8353568 [kg]
average velocity = 0 [m/s]

supergroup 2
-----
number of objects in supergroup = 301
sample object in supergroup = 335
link Const. = [0.19811320754717...1.115625]
material model = MATMODEL_005_LINEAR_ELASTIC_LINEAR_ISOTROPIC_HARDENING
```

Abb.: Report (physical properties)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	logfile: C:\sonar\LogFiles\LogFile_Events.txt												
2	model: C:\sonar\Models\62. Mobile Debris Fence\11. Anlage 1-9_ geknickteDiagonale.snir												
3	start: Donnerstag 20.06.2013, 21:45:15												
4	computer: D4400												
5	cycle Nr.	time [s]	microbe 1 Nr.	microbe 1 Name	microbe 2 Nr.	microbe 2 Name	link Nr.	link Type	Pos.x [cm]	Pos.y [cm]	Pos.z [cm]	Event	
6	50295	0.01507353	18866	unit_cylinder	18879	unit_cylinder	75012	Normal	14.24792162	390.9328064	3399.622145	max. link strength reached	
7	151756	0.04551183	47038	LEVEL0	47037	LEVEL0	181120	Normal	91.74810454	372.2721443	3602.556518	link broken	
8	273753	0.08211093	47439	LEVEL0	47438	LEVEL0	182320	Normal	87.31481165	518.113251	3623.73014	link broken	
9	283649	0.08507973	42639	unit_cuboid	42646	unit_cuboid	163566	Normal	29.57830359	296.3122412	3994.219848	link broken	
10	293368	0.08799543	42645	unit_cuboid	42652	unit_cuboid	163572	Normal	55.09224159	292.9948074	3994.091772	link broken	
11	298942	0.08966763	42640	unit_cuboid	42647	unit_cuboid	163567	Normal	36.00939633	295.3823906	4003.241066	link broken	
12	301714	0.09049923	42641	unit_cuboid	42648	unit_cuboid	163568	Normal	40.92051742	293.7112313	4006.728923	link broken	
13	303717	0.09110013	42642	unit_cuboid	42649	unit_cuboid	163569	Normal	48.32259123	292.2870512	4008.156269	link broken	
14	304555	0.09135153	42562	unit_cuboid	42569	unit_cuboid	163489	Normal	19.66617812	203.0548492	3991.613724	link broken	
15	306420	0.09191103	42563	unit_cuboid	42570	unit_cuboid	163490	Normal	19.67186379	202.9705982	3996.667819	link broken	
16	306974	0.09207723	42643	unit_cuboid	42650	unit_cuboid	163570	Normal	55.17385068	290.9616628	4010.973435	link broken	
17	306987	0.09208113	42792	unit_cuboid	42799	unit_cuboid	163719	Normal	76.59820711	521.6884944	3993.984023	link broken	
18	308269	0.09246573	42644	unit_cuboid	42651	unit_cuboid	163571	Normal	59.17693797	291.0031535	4010.292157	link broken	
19	309204	0.09274623	42564	unit_cuboid	42571	unit_cuboid	163491	Normal	23.42937342	202.7439047	3998.327466	link broken	
20	311463	0.09342393	42786	unit_cuboid	42793	unit_cuboid	163713	Normal	59.48151585	508.8253375	4000.144054	link broken	
21	311534	0.09344523	47818	LEVEL0	47817	LEVEL0	183454	Normal	29.13527172	591.8505736	3213.250603	link broken	
22	342668	0.10278543	45223	unit_cuboid	45232	unit_cuboid	174466	Normal	23.4062076	453.2511362	270.3011172	link broken	
23	344465	0.10332453	45222	unit_cuboid	45231	unit_cuboid	174465	Normal	31.94367111	453.8802867	271.4855764	link broken	

Abb.: Logfile (events)

# Zeitschritt

Frage:

- Wie kann der Benutzer den Zeitschritt für die Simulation beeinflussen?

Erklärungen :

- Eine sonar-Simulation wird iterativ in einer Folge von kleinen Zeitschritten durchgeführt. Innerhalb eines Zeitschrittes werden die vollständigen Berechnungen des gesamten Systems einmal durchgeführt. Wir nennen das einen Zyklus.
- Die Berechnung des Zeitschrittes wird in der sonar-Software automatisch vom System durchgeführt.
- Der Benutzer kann entgegen den Empfehlungen den Zeitschritt zwar direkt verändern bzw. setzen, in der Regel führt dies allerdings zu einer 'Explosion' des Modells.
- Die sonar Software basiert rein rechnerisch letztlich auf den elementaren newtonschen Bewegungsgleichungen. Alles was in 'sonar' während einer Simulation geschieht, wird explizit auf diese beiden elementaren physikalischen Gesetze zurückgeführt:
  - $\text{Aktion} = \text{Reaktion}$
  - $\text{Kraft} = \text{Masse} * \text{Beschleunigung}$
- Die Courant-Friedrichs-Stabilitätsbedingung und die Arbeit des Computer-Pioniers John von Neumann machen die folgende Aussage:
  - Ein simuliertes Differentialgleichungs-System der expliziten Art wie es in 'sonar' verwendet wird konvergiert nach beliebig vielen Zyklen gegen die exakte Lösung, vorausgesetzt, der Zeitschritt bleibt unter einem gewissen, maximal zulässigen kritischen Wert der aus den Simulationsparametern berechnet werden kann.
- Der Benutzer ist in der Lage den Zeitschritt indirekt zu beeinflussen. Folgende physikalische Größen beeinflussen diesen:
  - Die Interaktionskonstanten der Objekte (Oberflächenhärte)
  - Die Link-Konstanten (Steifigkeit der Verbindungen)
  - Die Geschwindigkeit der Objekte
  - Die Grösse der Objekte
  - Der Zeitschritt sinkt mit höheren Interaktions- und Link-Konstanten, ebenso mit höheren Geschwindigkeiten und wenn die Objekte kleiner werden. Dasselbe umgekehrt führt zu einem grösseren Zeitschritt.
- Für die Berechnung des Zeitschrittes ist immer dasjenige Objekt oder derjenige Link verantwortlich, welches den Wert am meisten 'herunterzieht'. Will der Benutzer den Zeitschritt auf diesem Weg beeinflussen, dann muss er folglich die richtigen Objekte oder Links modifizieren.
- Die Geschwindigkeit und die Objektgrösse kann er in der Regel nicht beeinflussen. Als Einflussgrösse bleiben die Interaktionskonstante und die Link-Konstante übrig. Je 'weicher' bzw. je niedriger diese gesetzt werden, umso höher kann der Zeitschritt werden. Andererseits entfernt man sich damit aber auch zusehends von den realen Werten.
- Die für den Zeitschritt verantwortlichen Objekte und Links werden im Tool 'Simulations-Parameter' angezeigt.

# Reibung

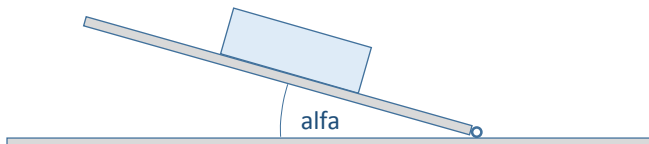
Zweck:

- Erklärungen zum Themenkreis ‚Reibung‘

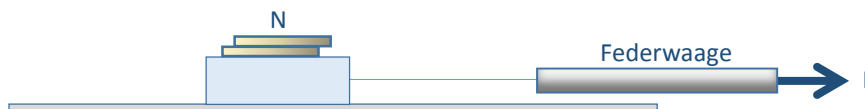
Erklärungen :

- sonar berücksichtigt die Reibung bei allen Kontakten und Stößen, sofern die Reibung eingeschaltet wurde (-> Preferences)
- sonar kennt bilaterale, unilaterale und globale Reibwerte. Bilaterale Werte haben die höchste, globale die niedrigste Priorität, wenn für eine gewisse Objektpaarung der Reibungskoeffizient bestimmt wird.
- sonar berücksichtigt die normale lineare Reibung der Form :  $R = \mu * N$  (R: Reibung;  $\mu$ : Reibungskoeffizient; N: Normalkraft)
- Der Reibungskoeffizient ist eine gemessene Grösse für eine bestimmte Materialpaarung unter bestimmten Zustandsbedingungen.
- Die Reibung ist seit je her eine problematische Einflussgrösse, da der Reibungskoeffizient bei gewissen Materialpaarungen in einem weiten Bereich schwanken kann. So reagiert beispielsweise das Material ‚Papier‘ sehr stark auf Änderungen der Luftfeuchtigkeit.
- Bei vielen Simulationen kommt man nicht umhin Versuche und Messungen der Reibung am realen Objekt anzustellen.
- Im Weiteren macht es oft Sinn, bestimmte Simulationen mit unterschiedlichen Reibwerten durchzuführen und die Auswirkungen dieser möglichen Veränderungen in den Resultaten zu kommentieren und zu dokumentieren.
- In der sonar-Dokumentation und - Software werden z.T. Werte für die Reibung von Materialpaarungen zur Verfügung gestellt. Es handelt sich dabei um Standardwerte wie sie auch im Internet gefunden werden können.
- Die Ergebnisse von Simulationen bei denen die Reibung eine zentrale Rolle spielt, weil gewisse Teile z.B. über längere Zeit über einen Reibschluss in Kontakt stehen, sind letztlich so genau wie das für die Reibungskoeffizienten zutrifft.
- Bei anderen Simulation wo keine ‚Gleitphasen‘ zwischen Objekten stattfinden, spielt der Reibungskoeffizient sicherlich eine kleinere, vielleicht sogar unbedeutende Rolle.
- Keine Theorie dieser Welt kann dem Benutzer die genauen Reibwerte für sein Modell liefern. Die Reibung ist ein Themenkreis wo der Benutzer letztlich seine eigenen Entscheidungen treffen muss. Folgende Ratschläge sollen ihn leiten:
  - Verwenden Sie als Ausgangspunkt Standardwerte aus der Literatur und aus dem Internet
  - Stellen Sie Überlegungen an zur Schmierung oder anderen Einflüssen zwischen den beiden Objekten und korrigieren Sie den Reibwert ggf. entsprechend.
  - Manchmal ist es möglich Reibwerte sehr einfach zu messen
- Einfache Messmethoden zur Bestimmung der Reibwerte

- 1. Schiefe Ebene:  
Ein schräges Brett mit aufgelegtem Zielmaterial hat einen bestimmten Winkel (alfa) zum Tisch wenn die Probe zu rutschen beginnt. Daraus ergibt sich  $\mu = \tan(\alpha)$



- 2. Zugversuch:  
Beschwerte Materialprobe mit einer Federwaage horizontal über Zielmaterial ziehen und die Kraft F ablesen.  
Die Probe kann dabei mit Münzen zusätzlich belastet werden. Daraus ergibt sich  $\mu = F / N$   
Die Punktepaare (N, F) für versch. Gesamtgewichte in eine Grafik auftragen -> Lineare Funktion -> bester Wert für  $F / N$   
(N = Gesamtmasse (Probe + Münzen))

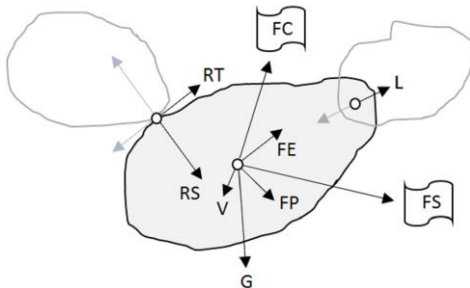


- Federwaagen gibt es für die Gewichts-Bereiche [0.....5g, 10g, 30g, 60g, 100g, 300g, 600g, 1000g, ... ]
- [www.pesola.ch](http://www.pesola.ch)

# Kräfte

Zweck:

- Erklärungen zum Themenkreis ‚Kräfte‘.
- Welche Kräfte können auf ein Objekt wirken?



Interne Kräfte :

- RS: elastisch-plastische Stosskräfte, Kollisionskräfte
- RS: Kontaktkräfte
- RT: Reibungskräfte
- L: Link-Kräfte (Objektverbindungen)
- V: Viskosität

Externe Kräfte :

- FE: konstante externe Kräfte und Momente auf einzelne Objekte
- G: einstellbares gleichgerichtetes konstantes Gravitationsfeld (Labormassstab)
- G: bilaterale Gravitationskräfte (für astronomische Dimensionen)
- FP: Punktkurven (manuell definierbar)
- FE: externe Punktkurven (data files)
- FC: Kontrollsysteme mit math. Formeln
- FS: sonar script (Kontrollsystem)

Erklärungen :

- Als expliziter Code funktioniert in ‚sonar‘ letztlich alles über Kräfte. Ausgenommen sind Zwangsbewegungen.
- Will der Benutzer in einem Modell einen Einfluss auf die Objekte geltend machen, dann kann er das mit einem geeigneten Kontrollsystem formulieren.
- Die internen Kräfte werden, falls sie eingeschaltet sind, während einer Simulation laufend autom. berechnet. Der Benutzer hat die Möglichkeit die Wirkung dieser Kräfte über die zugeordneten Konstanten und Parameter zu beeinflussen.
- Die externen Kräfte sind verschiedene Möglichkeiten zusätzliche Kräfte auf die Objekte auszuüben. Gewisse Krafteinflüsse lassen sich mit mehreren dieser Möglichkeiten realisieren. Der Benutzer entscheidet nach seinen Vorlieben.
- Alle Kräfte werden einmal pro Simulations-Zyklus berechnet, ausgewertet und zugeordnet. Die Berechnungen dieser Kräfte erfolgt also im Takt des Zeitschrittes.
- Es gibt Kräfte welche immer auf den Schwerpunkt der Objekte wirken (z.B. Gravitation) und solche welche auch Momente erzeugen (z.B. Reibung).
- Allgemein gesprochen wirkt jeder Krafteinfluss als eine sog. Dyname, bzw. als Kombination einer translatorischen Kraft und eines Drehmomentes. Es ist die Summe aller vektoriellen Kräfte und Momente auf ein Objekt welche für die Beschleunigung im nächsten Rechenzyklus verantwortlich ist..

# Zwangsbewegungen

Zweck: Was sind Zwangsbewegungen und wie wirken sie?

Erklärungen:

- Zwangsbewegungen sind Einflüsse auf die Bewegung eines Objektes welche nicht den normalen Weg über die Kräfte nehmen.
- Zwangsbewegungen setzen, in Umgehung der Beschleunigung, direkt die Geschwindigkeit oder die Position eines Objektes auf bestimmte Werte.
- Dasselbe gilt für die Rotation. Eine Zwangsbewegung setzt direkt die Winkelgeschwindigkeit oder den Winkel eines Objektes auf bestimmte Werte.
- Eine Zwangsbewegung kann die Werte für die Geschwindigkeit oder Position eines Objektes laufend mit einem Kontrollsystem berechnen oder auf einen konstanten Wert setzen. Kontrollsysteme sind der normale Weg für die Definition von Zwangsbewegungen.
- Wird ein Objekt zwangsbewegt, dann wird im Allgemeinen eine stetige Bewegung angestrebt. Sprünge sind nicht erwünscht. Sprünge können zu pathologischen Situationen führen, wenn Objekte sich durch die sprunghafte Bewegung plötzlich überlappen und das Modell zur Explosion bringen.
- Zwangsbewegungen werden oft in Antrieben verwendet. Besonders Rotationsantriebe sind eine häufige Anwendung dafür.
- Zwangsbewegungen können zu unstatthaften oder widersprüchlichen Situationen führen. Das sonar-System hindert den Benutzer nicht daran, damit bewusst oder ungewollt Unfug zu treiben: Beispiel: Ein Objekt wird zwangsbewegt und erreicht eine feste, unbewegliche und undurchdringliche Berandung. Die stossenden und abstossenden Kräfte schaukeln sich gegenseitig hoch bis etwas bricht oder eines der Teile einen maximalen Grenzwert überschreitet und durch die Berandung hindurchbricht bzw. diese ignoriert. Allerdings sind solche Vorgänge nicht mehr abgesegnet und haben nichts mehr mit der normalen Funktion des Programms zu tun.
- Man muss sich bewusst sein, dass eine Zwangsbewegung stur und ungeachtet des Widerstandes ihre Bewegung ausführt. Der gesamte Rest des Modells wird dieser Zwangsbewegung unterworfen.
- Die Resultate von zwangsbewegten Systemen sind mit Vorsicht zu geniessen. Wird ein mechanisches System z.B. mit einem zwangsbewegten Motor in Gang gesetzt und auf eine bestimmte Drehzahl hochgefahren, dann heisst das natürlich nicht, dass der vorgesehene Motor überhaupt in der Lage wäre das fragliche System auf diese Drehzahl zu bringen und anschliessend mit konstanter Drehzahl zu betreiben.
- Wollte man über diesen Sachverhalt Näheres erfahren, dann müsste man das System besser mit einem Drehmoment-gesteuerten Antrieb versehen, welcher die Drehmomentcharakteristik des betreffenden Motors berücksichtigt.

# Interaktion (Überblick)

Zweck:

- Welche Möglichkeiten gibt es rund um die Interaktion zwischen Objekten und um auf diese Einfluss zu nehmen?

Überblick:

- Interaktion Aktivieren:
  - Die Interaktions-Möglichkeiten und -Funktionen sind in jedem Objekt automatisch vorhanden und müssen nicht erzeugt werden.
  - Interaktions-Hauptschalter (Ein/Aus) -> Preferences / main switch
- Interaktion Einschränken:
  - Es gibt drei Objektklassen (ACTIVE, PASSIVE, NO INTERACTION). Jedes Objekt kann einer Klasse angehören
  - Interaction Rules
    - bilaterale Regeln zwischen zwei bestimmten Objekt Nummern
    - Interaktionsregeln zwischen Objekt Namen
  - Einschränkung der Interaktions-Tests auf bestimmte Richtungen (AXIAL, RADIAL, ALL DIRECTIONS)
- Interaktions-Geometrie:
  - gezielt gesetzte Interaktionspunkte auf Objekten welche bei gewissen Interaktionen zwischen Objekten benutzt werden
- Interaktions-Physik:
  - Interaktionsstärke bzw. Oberflächenhärte der Objekte bei Interaktionen
    - Lineare und quadratische Interaktionskonstanten für lineare und progressive abstossende Kräfte bei Stößen
  - Die sog. Interaktionsmethode bestimmt über das elastisch-plastische Verhalten bei Kollisionen (z.B. E/P = 10/90)
    - Anteil der absorbierten Energie bei einem Stoss, Stoss-Hysterese
- Interaktions-Auswertungen:
  - Kollisionskräfte bei Stößen, grafische Darstellungen

# Interaktions-Kontrolle

## Zweck:

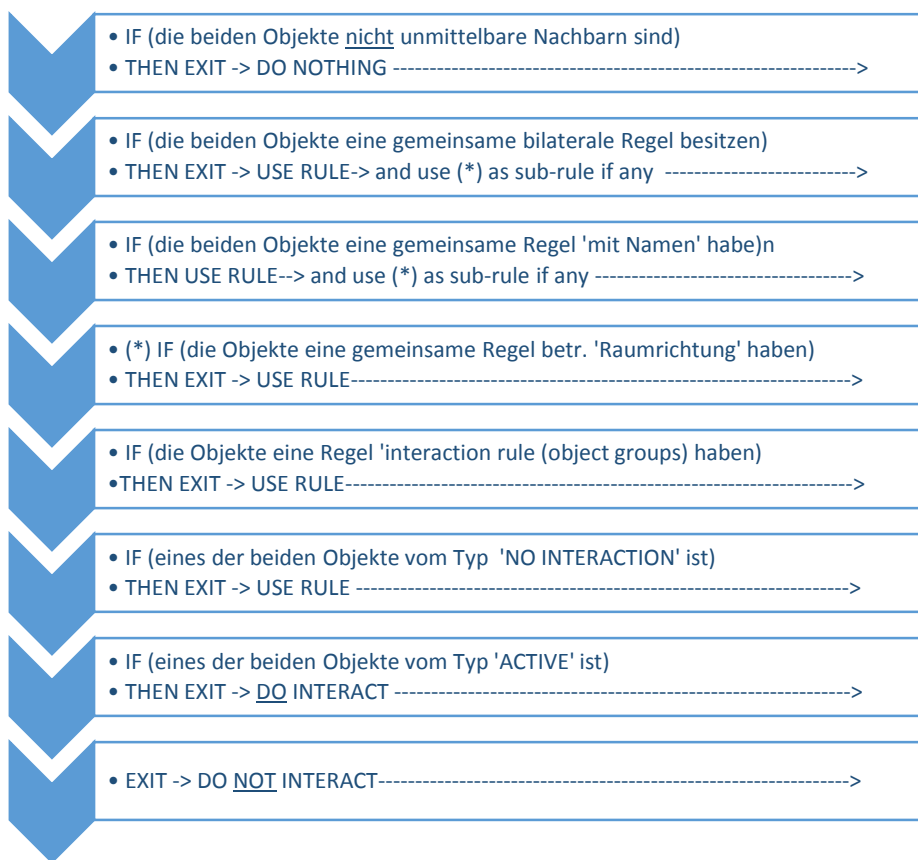
- sonar hat eine automatische Kollisionskontrolle. Für eine maximale Rechenleistung während einer Simulation wird die Interaktion zwischen den einzelnen Objekten, d.h. die Überprüfung ob zwei Objekte miteinander interagieren, gemäss nachfolgendem Schema durchgeführt.

## Zugriff :

- Der Benutzer hat verschiedene Möglichkeiten diesen Ablauf zu steuern und zu beeinflussen, indem er den Objekten entsprechende Eigenschaften (RULE's) zuschreibt.

## Ablauf :

- Der Ablauf funktioniert nach einem Ausschlussprinzip.
- Ziel ist, alle Paarungen von Objekten die letztlich nicht interagieren sollen, möglichst rasch auszuschliessen
- Das folgende Schema zeigt von oben nach unten den Ablauf der Entscheidungen. Auf jeder Linie kann der Ablauf ggf. zugunsten oder zu Ungunsten einer Interaktion abgebrochen werden (THEN EXIT, THEN USE RULE, usw.)
- Diese Test's werden in jedem Rechenzyklus für jede mögliche Objektpaarung einmal ausgeführt.



## Bemerkungen :

- Zwei Objekte sind unmittelbare Nachbarn, wenn eine Kollision zwischen ihnen bereits stattfindet oder unmittelbar bevorstehen könnte
- Bestimmte Regeln (RULE's) zwischen zwei Objekten oder Objektgruppen können eine Kollision entgegen den allgemeinen Regeln ermöglichen oder definitiv ausschliessen.
- Nachdem dieser Ablauf verlassen wurde, sei es auf einer bestimmten Linie oder ganz am Schluss, steht fest, ob die beiden Objekte grundsätzlich interagieren werden, falls sie sich berühren.
- Falls das Ablaufschema zum Schluss geführt hat, ja, die beiden Objekte sollen interagieren, dann werden die beiden Objekte an die betreffenden Interaktionsfunktionen weitergegeben um eine mögliche Interaktion ggf. zu berechnen.

# Interaktionskontrolle (Strategie)

Zweck: Wie sollen die Interaktionseigenschaften (ACTIVE, PASSIVE, NO INTERACTION) auf die Objekte in einem Modell verteilt und angewendet werden?

Strategie :

- Ein gutes Beispiel für die Darlegung, wie man die Interaktionseigenschaften auf die Objekte verteilt, ist ein Flipperkasten.
- In einem Flipperkasten würde man der Kugel oder den Kugeln, falls es mehrere gibt, die Eigenschaft ‚ACTIVE‘ geben.
- Die beiden Flipper und alle Inseln und Berandungen welche mit den Kugeln in Kontakt kommen könnten, würde man auf ‚PASSIVE‘ setzen.
- Aussenkonturen, welche nie die Gelegenheit bekommen mit den Kugeln zu kollidieren, gäbe man die Eigenschaft ‚NO INTERACTION‘

Begründung:

- Die Kugeln sind die einzigen Objekte welche mit allen anderen Objekten in Berührung kommen können -> ACTIVE
- Ein Flipper stösst nie mit dem zweiten Flipper oder einer Insel zusammen -> PASSIVE. Diese Teile bieten sich als Kollisionspartner an, kümmern sich aber nicht aktiv darum.

Korrekturen:

- In einem Flipperkasten gibt es manchmal zusätzliche bewegliche Teile, welche in ihrer Endstellung an anderen Objekten anschlagen, ähnlich wie Türen am Türrahmen. Diese Kollisionen finden immer zwischen zwei bestimmten Teilen statt. Würde man beide Teile wie bisher angenommen auf ‚PASSIVE‘ belassen, dann würden sie untereinander nicht kollidieren. Statt nun eines der Beiden Objekte auf ‚ACTIVE‘ zu setzen könnte man in einer solchen Situation eine sog. ‚BILATERALE REGEL‘ einführen, welche besagt, dass genau diese beiden Objekte eben doch kollidieren sollen, obwohl sie beide ‚PASSIVE‘ sind.
- Bilaterale Regeln sind vom Rechenaufwand her gesehen sehr effizient, weil die Interaktionsfunktionen sich gezielt nur auf die beiden fraglichen Objekte beschränken und keine weiteren und unnötigen Berechnungen in Gang setzten

## AMIGO Playfield Chart

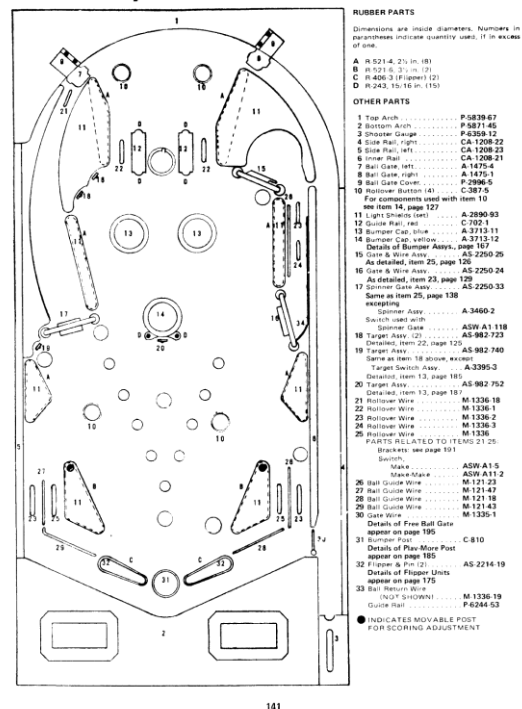


Abb. : The Internet Pinball Database, [www.ipdb.org](http://www.ipdb.org)

# Interaktionskontrolle (ACTIVE, PASSIVE, NO INTERACTION)

Zweck: Was bedeutet ACTIVE und PASSIVE und wie wird diese Eigenschaft gesetzt?

Regeln:

- Jedem Objekt kann eine der drei Eigenschaften ACTIVE, PASSIVE, NO INTERACTION zugeordnet werden.
- Diese Eigenschaften bestimmen, ob Objekte untereinander interagieren oder sich ignorieren.
- Das Setzen dieser Eigenschaft hat letztlich zum Ziel, die Berechnung der Interaktionen zwischen den Objekten zu beschleunigen indem unnötige Berechnungen vermieden werden.

ACTIVE:

- Die Eigenschaft ‚ACTIVE‘ besagt, dass das betreffende Objekt sich selbst aktiv darum kümmert, ob es mit einem anderen Objekt zusammengestossen ist.

PASSIVE:

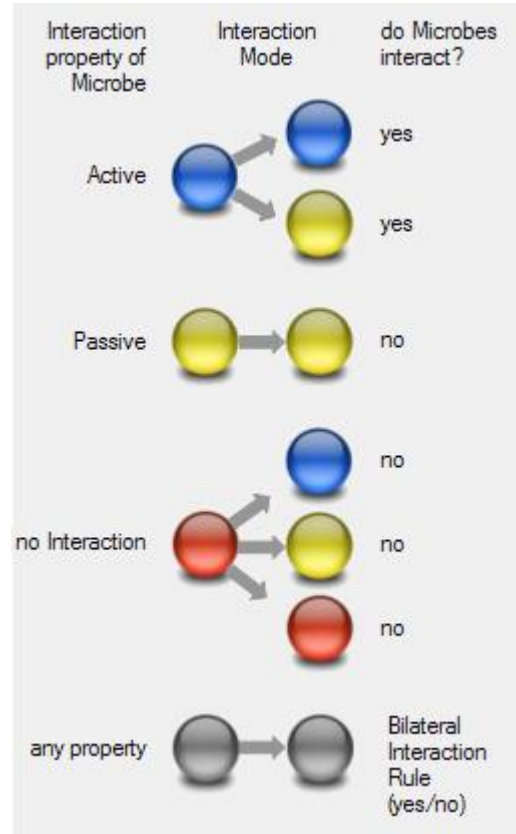
- Ein Objekt mit der Eigenschaft ‚PASSIVE‘ hingegen stellt sich auf den Standpunkt, dass es zwar bereit mit einem anderen Objekt zu kollidieren, aber die anderen Objekte müssen sich schon selbst darum kümmern.

NO INTERACTION:

- Das Objekt kollidiert definitiv nicht mit anderen Objekten.

- Die nebenstehende Tabelle zeigt alle Interaktionskombinationen und welche von diesen zu einer Interaktion führt. Es ist wichtig sich die beiden folgenden Kombinationen zu merken
  1. Passive Objekte interagieren untereinander nicht.
  2. Eine Kombination findet nur statt, wenn eines der beiden Objekte aktiv ist.

- Die drei Eigenschaften ACTIVE, PASSIVE, NO INTERACTION bestimmen das allgemeine Verhalten der Objekte in einem Modell. Mit sog. bilateralen Interaktionsregeln hat man die Möglichkeit in einem Modell alle Ausnahmen von dieser Regel festzulegen. Bilaterale Interaktionsregeln haben eine höhere Priorität als die allgemein gültigen Regeln.



Die Interaktionseigenschaften der einzelnen Objekte werden in den Objekt-Eigenschaften gesetzt:

Menu /...

Die bilateralen Interaktionsregeln lassen sich in folgendem Dialog definieren:

Menu / ...

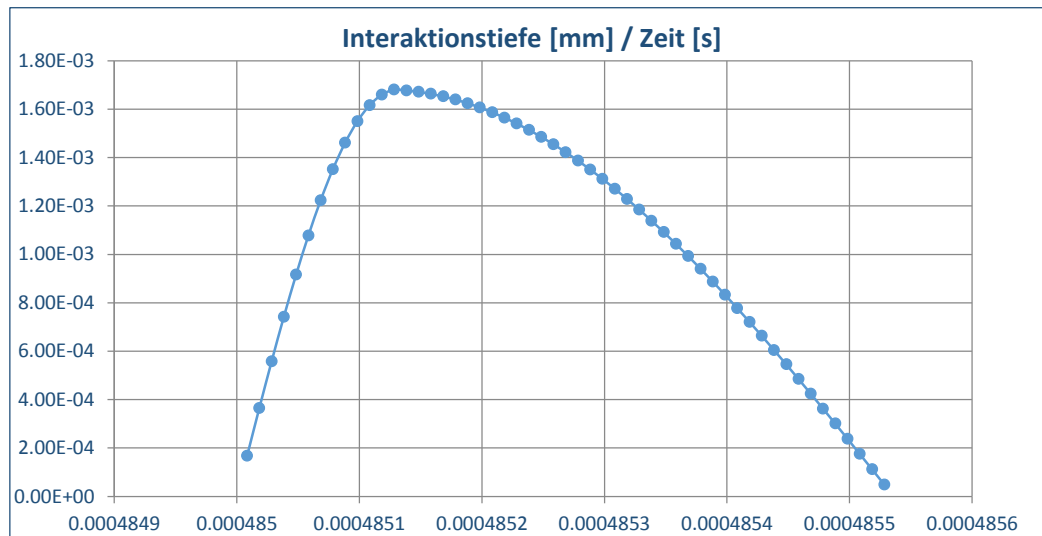
# Interaktion (Ablauf)

Zweck:

- Wie läuft eine Kollision zwischen zwei Objekten im zeitlichen Ablauf genau ab?

Messung :

- Die folgende grafische Darstellung zeigt die Interaktionstiefe zwischen zwei Objekten während ihrer Kollision.
- Es handelt sich um die Kollision eines kleinen Partikels auf eine harte ruhende Unterlage.
- Das Partikel hatte eine Relativgeschwindigkeit von 20 m/s
- Der Zeitschritt während dieser Simulation betrug  $1.0\text{E-}8\text{s} = 0.01\text{ Mikrosekunden} = 10\text{ Nanosekunden}$ .
- Die Stossparameter während diesem Vorgang waren auf ( $E/P = 10/90$ ) eingestellt (10% Elastizität, 90% Plastizität)



- Die Kurve zeigt, dass das Partikel eine maximale Eindringtiefe von ca. 1.7 Mikrometer erreicht und dass der Stossvorgang ca. 0.53 Mikrosekunden dauert.
- Die Punkte auf der Kurve zeigen den Zustand des Partikels als Funktion der Zykluszahl. Demzufolge wird der gesamte Stossvorgang in 53 Rechenzyklen abgehandelt.
- Der Eindringvorgang dauert ca. 13 Zyklen während das Trennen der Objekte anschliessend ca. 40 Zyklen dauert.
- Das Lösen der Objekte dauert deshalb länger als das Eindringen, weil der gesamte Vorgang einer Hysterese unterworfen ist. Beim Eindringen wirken grössere Kräfte als beim Lösen.
- Der Trennungsvorgang dauert folglich 3-mal so lange wie der Kollisionsvorgang. Entsprechend ist die absolute Geschwindigkeit des Partikels nach dem Stoss noch  $1/3$  dessen was es vorher hatte.
- Die kinetische Energie beträgt anschliessend nur noch  $1/3 * 1/3 = 1/9$  des Wertes vor dem Stoss. Dies in Übereinstimmung mit den Einstellungen ( $E/P = 10/90$ ).

# Interaktion (Experiment)

Zweck:

- Wie läuft eine Kollision zwischen zwei Objekten im zeitlichen Ablauf genau ab?
- Die folgende Versuchsanordnung der Firma (xy) zeigt ein Grundlagenexperiment zum elastischen Stoss.
- Ein kleiner Exkurs in die Physik

Funktion :

- Die Schiene (Air Track) enthält auf ihrer gesamten Oberfläche eine grosse Anzahl kleiner Bohrungen aus denen Pressluft ausgeblasen wird. Der sog. 'Glider' gleitet daher auf einem Luftkissen praktisch widerstandslos entlang der Schiene (Air Track).
- Der Glider enthält an beiden Enden gebogene Blattfedern, mit denen er mit anderen 'Glidern' oder mit einem festen Endanschlag (mit Force Sensor) kollidieren kann.
- Diese Blattfedern erzeugen praktisch verlustfreie, rein elastische Stösse.
- Indem ein fahrender auf einen stehenden 'Glider' aufläuft, kann man die Impulsübertragung zwischen den Glidern demonstrieren. Alternativ kann man einen 'Glider' auf den Kraft Sensor auffahren lassen und dabei die Kraft als Funktion der Zeit aufzeichnen.
- Als weitere Möglichkeit kann man die Blattfedern am 'Glider' durch Druckfedern ersetzen, welche mittels einer Raste so funktionieren, dass sie nur eingedrückt werden können, aber die gespeicherte Kraft nicht wieder zurückgeben. Damit lassen sich anschaulich plastische Stösse demonstrieren.



Abb. from University of California Los Angeles (UCLA), Physics&Astronomy Instructional Resource Lab

- Diese Experimente zeigen letztlich alle das grundsätzliche Funktionieren von Stössen in zeitlich gestreckter Auflösung.
- Stösse zwischen realen Körpern funktionieren prinzipiell genauso. Die Deformation der Oberfläche und die Stossdauer ist bei harten Materialien einfach viel kleiner.
- Auch in 'sonar' funktionieren Stösse in ihrem zeitlichen und mathematischen Ablauf analog.
- Die Apparatur kann dazu herangezogen werden zu demonstrieren, dass die elementaren newtonschen Bewegungsgleichungen nicht nur bilanzmässig zutreffen, indem der Zustand der Objekte vor und nach dem Stoss aufgezeichnet werden. Es kann im Detail nachgewiesen werden, wie diese grundlegenden Gesetze zu jedem beliebigen Zeitpunkt oder während jedem noch so kleinen Zeitintervall bzw. Zeitinkrement während des Stosses erfüllt sind. Und das ist das Wesentliche. In jedem noch so kleinen, beliebig herausgegriffenen Zeitintervall während der Stossdauer gilt (ACTIO = REACTIO) absolut auf 20 Stellen nach dem Komma.
- Aus dieser Erkenntnis heraus folgt direkt die Impulserhaltung des Stosses.

$$\int K_1 dt = \int m_1 b_1 dt = m_1 \int b_1 dt = m_1 \int dv_1 = m(v_{12} - v_{11}) = I_{12} - I_{11} = \Delta I_1 = \text{Impulsänderung}$$

wegen (actio = reactio) gilt zu jedem Zeitpunkt  $K_2 = -K_1$

$$\int K_2 dt = \int -K_1 dt = - \int K_1 dt = -\Delta I_1$$

Folgerung: Wäre der Impuls nicht erhalten, dann müsste mindestens während einem sehr kleinen Zeitraum das Gesetz 'actio = reactio' verletzt worden sein um diese Differenz zu erzeugen. Andernfalls ist rein rechnerisch keine Differenz möglich. Aber 'actio = reactio' ist so grundlegend, dass wir nur den Schluss ziehen können: Der Impuls bleibt beim Stoss erhalten. Die Impulserhaltung und 'actio = reactio' hängen also direkt zusammen. Das Eine steht und fällt mit dem Anderen.

## arcball function – moving the model in 3D

Zweck:

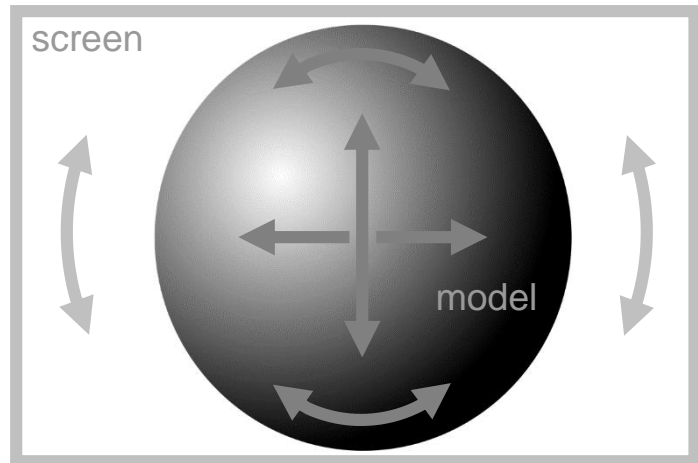
- Die Darstellung des Modells im 3D-Window soll mit der Maus in ihrer räumlichen Lage manipuliert werden

Zugriff:

- Diese Funktion steht immer automatisch zur Verfügung

Funktion:

- Stellen Sie sich gedanklich eine virtuelle Kugel vor die ihren Bildschirm, wie nebenstehend abgebildet, ausfüllt.
- Die Kugel sei in ihrem Zentrum gelagert und frei um alle Achsen drehbar
- Stellen Sie sich weiter vor, dass ihr Modell fest mit dieser Kugel verbunden ist. Ihr Modell sei in die Kugel eingegossen.
- Versuchen Sie nun diese virtuelle Kugel durch Auflegen der Hand bzw. mit der Maus in eine beliebige Richtung zu drehen. So wie sich erwartungsgemäss die Kugel drehen würde, so wird sich Ihr Modell drehen.
- Die Distanz zum Objekt wird mit dem Mausrad eingestellt (Zoom-Funktion)



### Bemerkungen:

- In den Einstellungen zum Programm lässt sich die Empfindlichkeit der Funktion in Bezug auf die Mausbewegung einstellen :  
Einstellungen : Menu / Preferences / ...



Kugelbrunnen an der Seepromenade in Zürich, Schweiz

## snap function – exact object positioning

Zweck:

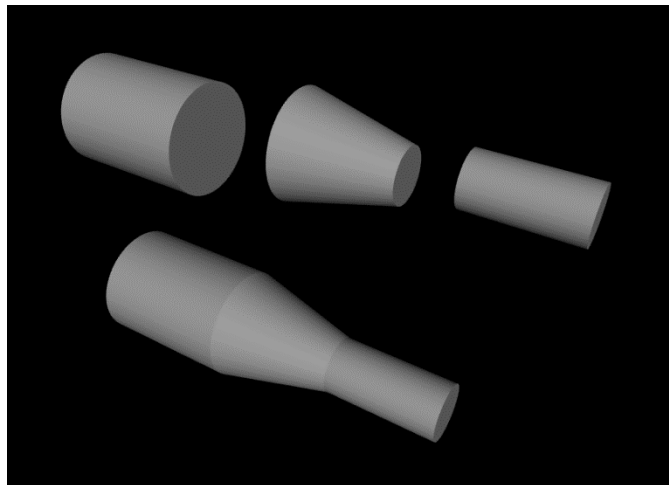
- Mehrere Objekte sollen manuell, aber trotzdem analytisch exakt zu einer grösseren Einheit zusammengeschoben werden.

Zugriff:

- Die Verschiebungen der Objekte werden mit der Maus in den einzelnen 2D-Ansichten durchgeführt (Translation)
- Die hier beschriebene sog. 'snap'-Funktion ist default-mässig eingeschaltet, kann aber ausgeschaltet werden.
- Einstellungen: Menu / Preferences / ...

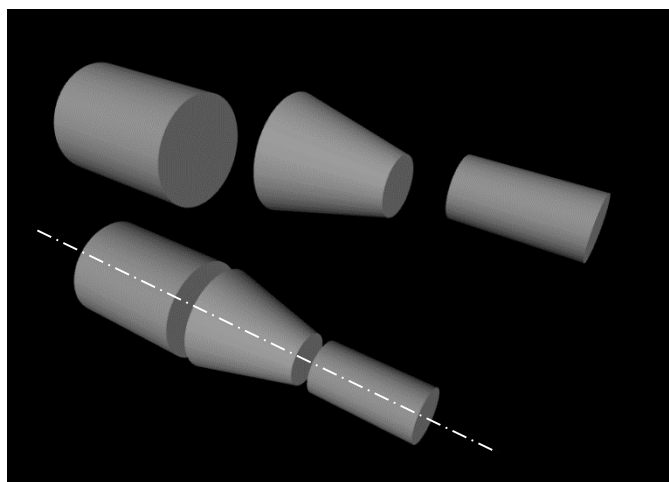
Funktion:

- Wird das mit der Maus bewegte Objekt in der Nähe des Zielobjektes losgelassen, dann findet immer dann ein 'Einschnappen' auf das Zielobjekt statt, wenn die Positions-Abweichungen in allen drei Dimensionen einen bestimmten Maximalwert unterschreiten.
- Dieser kritische Maximalwert kann vom Benutzer in den Einstellungen gesetzt werden (snap distance).
- Die Einheit der 'snap distance' ist 'Pixels'
- Ein guter Durchschnittswert für die 'snap distance' ist 6 Pixels.



### Bemerkungen:

- Um ein Objekt im dreidimensionalen Raum an ein anderes Objekt heranzuführen, sind in der Regel mindestens zwei Translationen in zwei verschiedenen Ansichten notwendig. Der 'snap'-Vorgang, wie oben dargestellt, findet in diesem Fall erst mit der zweiten Translation statt.
- Ist das bewegte Objekt noch nicht nahe genug an ein anderes Objekt herangeschoben worden, aber mit diesem 'in Achse', dann findet auch ein seitlicher 'snap'-Vorgang auf diese gemeinsame Achse statt, so wie in der nachfolgenden Abbildung dargestellt. Dieser 'snap'-Vorgang wird aber nur dann durchgeführt, wenn die Achsen der beiden Objekte in der betreffenden Ansicht senkrecht zur Bildelebene stehen.



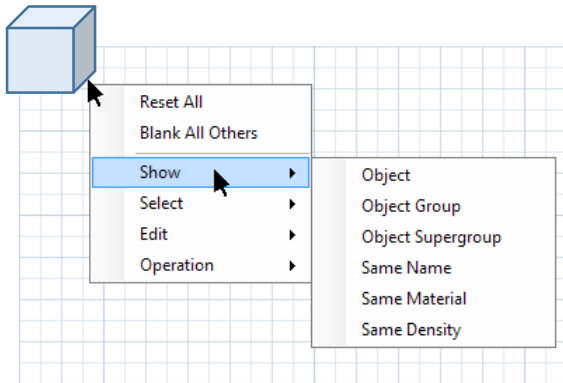
# Kontext Menu

Zweck:

- Gewisse Funktionen an Objekten können zusätzlich, manche ausschliesslich, über das sog. Kontext Menu aufgerufen werden.

Zugriff :

- Fahren Sie in einer Ansicht (View) mit der Maus über die Berandung eines Objektes bis der Cursor seine Farbe wechselt und in einem sog. Kontextfenster der Typ und der Name des gewünschten Objektes angezeigt werden.
- Drücken Sie in dieser Lage die rechte Maustaste.
- In diesem Kontext Menu haben Sie nun verschiedene Möglichkeiten auf Funktionen in Zusammenhang mit dem betreffenden Objekt zuzugreifen.



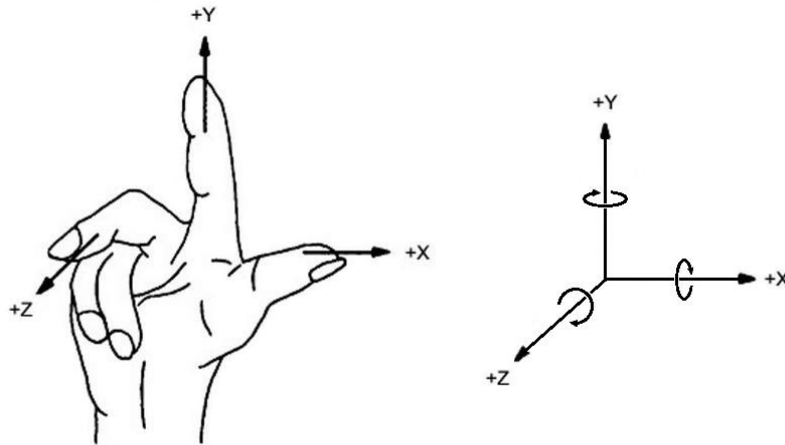
Kontext Menu in sonar-LAB:

- |                            |   |
|----------------------------|---|
| • Reset All                | setze alle im Kontextmenu durchgeführten Veränderungen an der Darstellung zurück  |
| • Blank All Others         | mit ‚Show‘ wird eine Auswahl von Objekten hervorgehoben. ‚Blank All Others‘ macht alle anderen Objekte vorübergehend unsichtbar |
| • Show                     |   |
| ○ Object                   | zeige das selektierte Objekt in einer Signalfarbe   |
| ○ Object Group             | zeige die zugehörige Gruppe des selektierten Objektes in einer Signalfarbe  |
| ○ Object Supergroup        | zeige die zugehörige Supergruppe des selektierten Objektes in einer Signalfarbe   |
| ○ Same Name                | zeige alle Objekte mit dem gleichen Namen (selektiertes Obj.) in einer Signalfarbe  |
| ○ Same Material            | zeige alle Objekte aus dem gleichen Material (selektiertes Obj.) in einer Signalfarbe   |
| ○ Same Density             | zeige alle Objekte mit der gleichen Dichte (selektiertes Obj.) in einer Signalfarbe   |
| • Select                   |   |
| ○ Object                   | selektiere Objekt, ganze Gruppe oder ganze Supergruppe  |
| ○ Object Group             |   |
| ○ Object Supergroup        |   |
| • Edit                     |   |
| ○ Properties...            | Öffne den Dialog ‚Object Properties‘ mit den Daten des selektierten Objektes  |
| ○ Rotation and Position... | Öffne den Dialog ‚Rotation und Position...‘ mit den Daten des selektierten Objektes   |
| ○ Prism Contour...         | Falls das selektierte Objekt ein ‚Linien-Bogen-Prisma‘ ist, dann öffne Dlg.(Contour)  |
| ○ Resize...                | Öffne den Dialog zur Skalierung des selektierten Objektes mit einem Faktor.   |
| ○ Mirror...                | Öffne den Dialog zum Spiegeln des Objektes an einer Hauptebene  |
| ○ Point Curve...           | Öffne die zugeordnete Punktkurve zum selektierten Objekt (falls es eine gibt)   |
| ○ External Point Curve...  | Öffne die zugeordnete externe Punktkurve zum selekt. Objekt (falls vorhanden)   |
| • Operation                |   |
| ○ Cut                      | (wie die Standard Funktionen im Hauptmenu / Edit)   |
| ○ Copy                     |   |
| ○ Paste                    |   |
| ○ Clear                    |   |
| ○ Duplicate                |   |

# Die Dreifingerregel

Zweck:

- Die sog. Dreifingerregel ist eine Gedankenstütze
- Die Dreifingerregel definiert die relative Ausrichtung und positive Drehrichtung der drei Koordinatenachsen (X, Y, Z).



sonar Regeln:

- Benutzen Sie für die Regel die rechte Hand
- In sonar gelten die Achsenbezeichnungen wie abgebildet
- Der Bildschirm liegt in der X-Y-Ebene
- Die positiven Drehrichtungen um die einzelnen Achsen wirken immer im Gegenuhrzeigersinn (in der betr. Achsenrichtung betrachtet)
- Merken Sie sich insbesondere die positive Drehrichtung der Z-Achse, welche aus dem Bildschirm heraus auf den Betrachter zeigt. Diese Information wird in sonar häufig benötigt, wenn Objekte, die in Z-Richtung extrudiert wurden, eine Rotation bzw. Winkelgeschwindigkeit zugeordnet werden muss.

# Objekt Variablen

Zweck:

- Auf welche Variablen eines Objektes hat der Benutzer Zugriff?
- Hinter jedem Objekt steht ein sog. „record“ von Daten, welcher alle Informationen des betreffenden Objektes enthält.
- Viele dieser Daten sind nur für die internen Berechnungen angelegt. Im Folgenden zählen wir nur diejenigen Objekt-Daten auf, auf die der Benutzer direkt über Dialoge zugreifen kann. Über sonar-script sind ev. weitere Zugriffsmöglichkeiten gegeben.
- Die Auflistung eines Teils dieser inneren Objektstruktur soll für den Benutzer eine weitere Anlaufstelle sein, bei der Frage, welche Informationen eines Objektes zugänglich sind.

Variablen :

```
enum primType:           // Primitive-Typ (Kugel, Quader, Zylinder, usw.)
int iActDirection:       // Interaktionsrichtung -> default: MODE_IAct_ALL_DIRECTIONS
int objGroupNr:          // Gruppenzugehörigkeit -> group number (level 0)
int objGroupNr_S1:       // Supergruppenzugehörigkeit -> super group (level 1)
primitiveData primData:  // more data...
ext3D kPt[max *]:        // lokale Coord. der Link-Punkte im Objekt (max *)
ext3D ctrlPt[max *]:     // lokale Coord. der Interaktions-Ctrl.Punkte (max *)
BOOL ctrlPointsActive;   // Aktivierung/Deaktivierung der Interaktions-Ctrl.Punkte
ext3D bPt;               // aktuelle globale Position
ext3D sPt;               // lokale Position des Schwerpunktes (default: bPt = sPt)
struct ext3D f_extern;    // externer Kraftvektor im Schwerpunkt des Objektes
struct ext3D dm_extern;   // externes Drehmoment im Schwerpunkt des Objektes
struct ext3D u;           // aktueller Geschwindigkeitsvektor des Objektes
struct ext3D omg;         // aktueller Winkelgeschwindigkeitsvektor des Objektes
BOOL angX_locked;        // Rotationsachsen (bezüglich lokaler Nullposition) blockieren
BOOL angY_locked;        // ..
BOOL angZ_locked;        // ..
double ro;               // Materialdichte
double Emodul;           // Elastizitätsmodul
double M;                // Masse
double V;                // volumen
struct ext3D tta;         // Trägheitstensor bezüglich Hauptachsensystem bzw. Nullposition(Obj.)
BOOL tta_AutoCalc;        // soll das Trägheitsmoment autom. berechnet / manuell gesetzt werden
char name[32];            // der Objektname
char material[32];        // Bezeichnung des Objektmaterials
int matRefNr;             // Referenznummer bei Verwendung eines Standardmaterials
double c_friction_dynamic; // dynamischer Reibungskoeffizient
double c_friction_static; // Haftreibungskoeffizient
int colorIndex;           // Referenznummer bei Verwendung einer Standardfarbe [0..24]
char colorName[32];       // Name einer beliebigen RGB-Farbe
int cRed;                 // Rot Komponente der RGB Farbe [0..255]
int cGreen;               // Grün Komponente der RGB Farbe [0..255]
int cBlue;                // Blau Komponente der RGB Farbe [0..255]
int cTrans;               // Transparenz des Objektes [0..100%]
enum objectGravity gravity; // Gravitationskraft aktivieren
BOOL inUse;               // ist das Objekt in Gebrauch?
BOOL visible;             // ist das Objekt am Bildschirm sichtbar?
BOOL wireframe;           // soll das Objekt als wireframe gezeichnet werden
BOOL simMember;           // ist das Objekt fest im Raum verankert (simMember=false)?
BOOL hasViscosity;        // hat das Objekt Viskositätseigenschaften
BOOL hasFrictionRule;     // hat das Objekt zusammen mit anderen Objekten Reibungsregeln?
BOOL iActDamping;         // ist die Interaktionsdämpfung für das Objekt aktiviert
int iActMethod;           // Elastisch-Plastisches Verhalten bei Interaktionen
double iActAbsorbFactor;   // wie gross ist der Energie-Absorptionsfaktor bei Stößen?
enum interactionMode iActMode; // Interaktionsart (aktiv, passiv, keine Interaktion)
double c_iActLinear;       // lineare interaktionskonstante
double c_iActQuad;        // quadratische Interaktionskonstante
struct Matrix3D RT;        // die aktuelle Drehmatrix des Objektes relativ zur Nullposition
```

# Weshalb wir das ,cm-g- $\mu$ s' – Einheitensystem benutzen

Fragestellung: Weshalb wird in der Simulationssoftware sonar bei den Berechnungen das cm-g- $\mu$ s – Einheitensystem verwendet?

## Antwort:

- Ein Computer rechnet immer mit endlich langen Zahlen. Man bezeichnet die Länge der Zahlendarstellung, die ein Computer für die Berechnung von reellen Zahlen verwendet als ,signifikante Stellen'
- Bei der Addition reeller Zahlen, die sich erheblich in ihrer Grössenordnung unterscheiden, gehen signifikante Stellen verloren.
- Wenn wir davon ausgehen, dass die Zahlen im Computer ursprünglich in der wissenschaftlichen Notation gespeichert sind und der Computer im weiteren in der Lage wäre 12 signifikante Stellen zu speichern, dann könnten zwei Testzahlen a und b vorab wie folgt definiert sein:

a = 1.23456789012E+0  
b = 1.23456789012E-8

- Summe a + b

```
1.23456789012
0.0000000123456789012
-----
1.2345679024656789012
```

Der Computer kann aber gemäss unserer Annahme nur 12 Stellen speichern, also gehen die überhängenden Stellen von ,b' bei der Addition verloren. Das Resultat lautet:

= 1.23456790246

- Wäre der Unterschied der Grössenordnung der Zahlen in unserem Beispiel statt nur 8 sogar mehr als 12 Stellen, dann könnte dies dazu führen, dass bei Additionen alles verloren geht, was die kleinere Zahl zur Grösseren beitragen möchte. Man könnte dann unendlich oft die Zahl ,b' zu ,a' hinzuaddieren und die Zahl ,a' bliebe konstant.
- Man kann den Verlust signifikanter Stellen zu einem grossen Teil vermeiden, indem man ein Einheitensystem wählt in dem alle relevanten Grössen von der gleichen Grössenordnung sind.
- In unseren sonar Simulationen liegen im sog. Labormassstab im Durchschnitt der Anwendungen die folgenden Verhältnisse vor:
  - Distanzen haben die Grössenordnung von 1 Zentimetern
  - Gewichte haben die Grössenordnung von 1 Gramm
  - Zeitschritte während den Simulationen sind von der Grössenordnung einer Mikrosekunde
- Aus diesem Grund rechnen wir in sonar intern im [cm-g- $\mu$ s]- System.

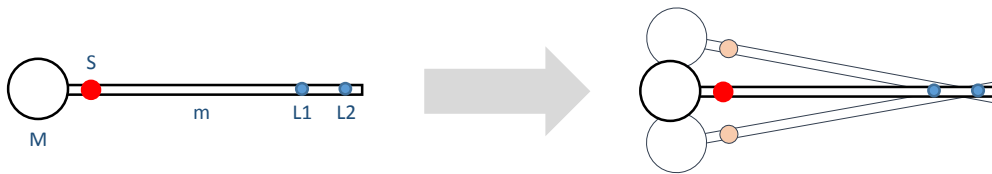
## Der Schwanz wedelt mit dem Hund (the tail wags the dog)

Problem :

- Die Positionierung von Verbindungen (Links) in Bezug auf die Schwerpunkts Lage der Objekte ist wesentlich und bereitet in einer Simulation bei ungünstiger Anordnung ggf. Stabilitätsprobleme.

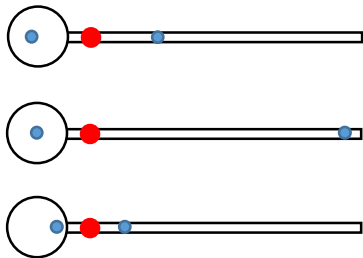
Beispiel:

- Eine einseitige Hantel mit einer grossen Masse  $M$  wird über eine kleine Masse  $m$  weit ausserhalb des Gesamtschwerpunktes  $S$  mit zwei Links  $L1$  und  $L2$  an zwei Fixpunkten oder einem anderen Objekt befestigt.
- 



Eine Anordnung dieser Art ist in höchstem Mass instabil und neigt zu Schwingungen. Befestigungs-Links sollten wenn immer möglich so gesetzt werden, dass sich der Gesamtschwerpunkt der beiden Objekte innerhalb des von den Links aufgespannten Raumes befinden.

Gute Lösungen:



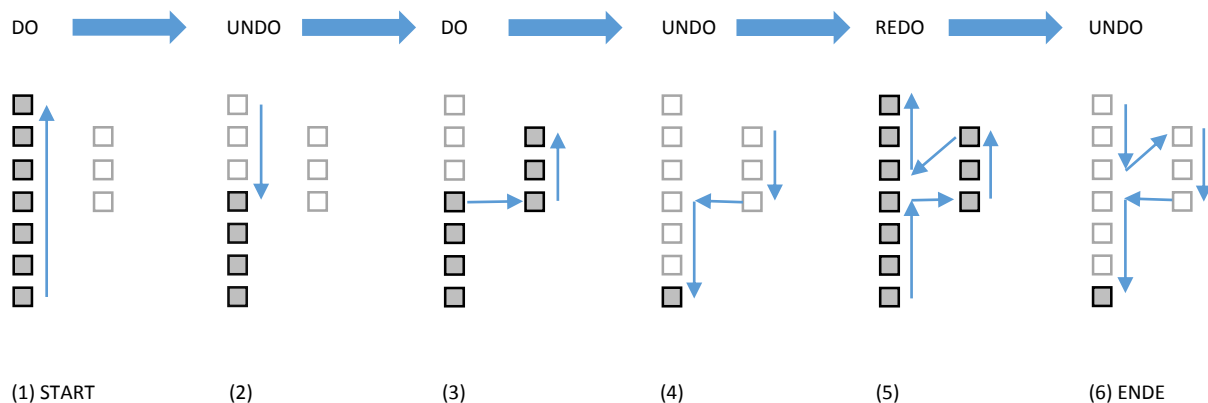
Bemerkung:

- Dieselben Kriterien gelten bereits für die Verbindungen zwischen der grossen Masse  $M$  und der kleinen Masse  $m$  der Hantel.
- In sonar darf ein Link zwischen zwei Objekten auch ausserhalb der Oberfläche bzw. der Volumina der beiden Massen liegen.
- Lässt sich eine Link-Anordnung der beschriebenen Art nicht vermeiden, dann kann den androhenden störenden Schwingungen nur mittels härteren Links und erhöhten Dämpfungswerten der Links entgegengewirkt werden. Diese Eigenschaften müssen für einen gegebenen Fall durch Probieren ermittelt werden.

## Wie funktioniert ‚Undo‘ und ‚Redo‘?

Das Programm sonar-LAB beinhaltet einen mehrstufigen UNDO- und REDO-Mechanismus. Der Benutzer ist damit in der Lage das Modell in mehreren Schritten in einen früheren Zustand zurückzusetzen. Die folgende Abbildung zeigt die zu erwartende Zustandsfolge eines Modells am Beispiel einer fortgesetzten Positionsänderung eines einzelnen Objektes, dargestellt als ein Quadrat. Das Quadrat wird in diesem Beispiel fortgesetzt manuell mit der Maus verschoben. Diese Verschiebungen werden wie dargestellt mit der UNDO- und REDO-Funktion schrittweise aufgehoben oder wieder hergestellt. Die Ausgangslage des Objektes ist am Ort ‚Start‘ links unten. Der Reihe stellt die Abbildung gemäss der Nummerierung Folgendes dar:

- (1) : Ausgehend von der Position ‚START‘ wird das Quadrat mit der Maus in 6 Schritten jeweils um eine Strecke nach oben verschoben.
- (2) : Es wird dreimal der Befehl ‚UNDO‘ ausgelöst
- (3) : Ausgehend von der aktuellen Position wird das Quadrat mit der Maus zuerst um eine Strecke nach rechts und anschliessend in zwei weiteren Schritten nach oben verschoben
- (4) Es wird sechsmal der Befehl ‚UNDO‘ ausgeführt, womit sich das Quadrat wieder an der Startposition befindet
- (5) Jetzt kommt der Befehl ‚REDO‘ ins Spiel. Zuerst wird dieser sechsmal angewendet und das Quadrat befindet sich an seiner obersten Position am Seitenast. Wird der Befehl ‚REDO‘ nun weiter aufgerufen, dann nimmt das Quadrat wieder die alten Positionen auf dem linken Ast ein und erreicht schliesslich wieder die oberste Position auf dem linken Ast ganz oben.
- (6) Jetzt wird fortgesetzt der Befehl ‚UNDO‘ aufgerufen und das Quadrat wird auf dem gleichen Pfad, den es bei der Aktion (5) genommen hat, wieder zurück verschoben. Am Ende befindet sich das Quadrat wieder in der START-Position.



## Mit welchen Dialogen können Links bearbeitet werden?

Wir kennen verschiedene Einzel-Links (NORMAL, BEND, TORSION) und Link-Kombinationen (CTR120, EXCL120, NBT, NNB, usw.) Die folgende Tabelle versucht einen Überblick herzustellen darüber, welcher Dialog für welche Aufgabe und welchen Typ von Link herangezogen werden kann.

Aufgabe	Zugriff	Dialog
Die Link-Eigenschaften eines einzelnen normalen Links ändern	Object Tool / double click Select Link in View / Menu Edit / Edit Link...	Edit Regular Link
Eine bestimmte physikalische Eigenschaft der Links einer Selektion oder Gruppe von Objekten ändern	Select Link in View Menu Functions / Material / Material Property...	Edit Material Property
Die Link-Eigenschaften einer Auswahl von Links ändern	Select Link in View oder Select Object in View Menu Functions / Collective Change / Links	Collective Change of Links
Sich einen Überblick über alle Link-Eigenschaften einer (CTR120)-Link-Kombination verschaffen	Object Tool / double click	Display Link Combination (CTR120)
Die Link-Steifigkeit einer oder mehrerer Link-Kombinationen (CTR120, EXCL120) ändern	Select Link in View Menu Functions / Material / Change Stiffness (CTR120-Links / EXCL120-Links)...	Change Stiffness (CTR120-Links) Change Stiffness (EXCL120-Links)
Einer Objektgruppe bzw. deren Links ein Materialmodell (Spannungs-Dehnungs-Diagramm) und alle diesbezüglichen Variablenwerte zuordnen	Select Link in View Menu Functions / Material / Stress/Strain Variables...	Edit Material (Stress/Strain-Variables)
Einer Objektgruppe oder Auswahl an Objekten ein Fließkriterium bzw. eine Gestaltänderungshypothese zuordnen	Select Link in View Menu Functions / Material / Yield Criteria...	Yield Criteria &/ Failure Theory

# Grenzen der Modellierung (Stand 2017)

## Zweck:

- Wie sieht die sog. ‚envelope‘ des Anwendungsspektrums der sonar Software aus?

## Grenzen:

- Anzahl Objekte
- Modellgrösse, Ausdehnung
- Modell Komplexität
- Simulationszeit (Welt Uhr)
- Technische Disziplinen, Materialien, Media, Physik

## Anzahl Objekte ::

- Vom Standpunkt der Software-Technologie und der Programm-Entwicklung her gesehen bestehen keine Grenzen.
- Die Software unterstützt aber eine fest vorgegebene Anzahl Objekte welche laufend der aktuellen Leistungsfähigkeit von Workstation-Computern angepasst wird.
- sonar-2015: Anzahl Objekte = 64'000; Anzahl Partikel = 250'000; Anzahl Links = 200'000

## Modellgrösse, Ausdehnung

- Es sind Modelle in der Grösse der Mikrotechnologie bis zu astronomischen Dimensionen möglich.

## Modell Komplexität :

- Die Komplexität der Modelle ist grundsätzlich unbeschränkt. Dies gilt sowohl für die Komplexität einzelner Objekte als auch für die Vielfalt wie diese untereinander interagieren.

## Simulationszeit (Welt Uhr)

- Es gibt praktische Grenzen in der Simulationszeit. Das sind Grenzen an die jedes Simulationssystem stösst. Sehr langsame Vorgänge können durch die Grenzen, wie sie von einem expliziten Code wie ‚sonar‘ bezüglich dem Zeitschritt gesetzt werden, so lange dauern, dass eine Simulation praktisch nicht mehr sinnvoll ist.

## Technische Disziplinen :

- sonar ist eine Software zur Simulation mechanischer Systeme (Mehrkörperdynamik Simulation)
- sonar kann demzufolge überall dort eingesetzt werden, wo die newtonsche Mechanik gilt.
- Thermodynamische Vorgänge und Gesetze der Wärmelehre sind nicht implementiert.
- Untere Grenzen: quantenmechanische Vorgänge oder Vorgänge im biochemischen Bereich werden nicht simuliert.
- Obere Grenzen: dunkle Materie und dunkle Energie, die in astronomischen Dimensionen eine Rolle spielen, sind nicht implementiert

## Materialien:

- sonar simuliert zur Hauptsache Festkörper (Starrkörper, elastisch-plastisch verformbare Körper)
- Mit dem Modul ‚Partikel‘ sind zusätzlich viskose Massen und Flüssigkeiten simulierbar. Diese Flüssigkeiten dürfen dank dem Partikelmodell zerspritzende Flüssigkeiten sein.
- Luft oder allg. Gase können indirekt an einer Simulation teilnehmen. In diesem Sinne können in einem Modell bestimmte vorgegebene, ev. sogar zeitlich veränderliche, Luftströmungen als Vektorfelder definiert werden, welche auf alle Objekte einen berechneten Luftwiderstand ausüben. Es findet dabei aber keine Rückkopplung von den Partikeln auf die Luftströmung statt.

