

# sonar

## Tutorial

DEUTSCH

Version 3.0  
Status : for general use  
Author : Fritz Leibundgut, L&G Software  
© L&G Software 2018



# Inhalt

---

|               |   |
|---------------|---|
| Vorwort ..... | 1 |
|---------------|---|

## Einführung 2

|   |    |
|---|----|
| Numerische Simulation .....   | 2  |
| Ein Vergleich zwischen analytischen und numerischen Lösungsmethoden ..... | 3  |
| Simulationsprogramme und Methoden.....                                    | 7  |
| Begriffserklärungen.....  | 7  |
| Die Modellierung in sonar.....  | 9  |
| sonar Programm Module .....   | 11 |
| Aufbau eines sonar Modells.....   | 13 |
| Bestandteile eines sonar-Modells .....                                    | 14 |
| Elastische Verbindungen .....   | 17 |
| Möglichkeiten der Modellerzeugung .....                                   | 18 |
| Bestandteile eines sonar-Modells .....                                    | 19 |
| Funktionsumfang des Basismoduls.....                                      | 20 |

## Die Erzeugung von einfachen Primitivkörpern 21

|  |    |
|--|----|
| Erzeugungsmethoden .....                             | 21 |
| Erzeugung eines Primitivkörpers mit Makros .....     | 21 |
| Erzeugung eines 'Primitives' mit Dialogen .....      | 22 |
| Grafisch-interaktive Erzeugung eines Primitives..... | 24 |
| Die Nulllage eines Objektes.....                     | 25 |
| Weitere Abhängigkeiten.....                          | 26 |

## Prismatische Körper 27

## Extrudierte Linien-Bogen-Konturen 29

|  |    |
|--|----|
| Der Dialog Create>Show Contour.....                | 29 |
| Das Format eines Textfiles.....                    | 30 |
| Die Nachbearbeitung von extrudierten Konturen..... | 31 |
| Komplexe Linien-Bogen Konturen.....                | 32 |

## Unverrückbare Objekte 34

|                        |    |
|------------------------|----|
| Ebenen .....           | 34 |
| Fixpunkte.....         | 34 |
| Grids .....            | 35 |
| fixierte Objekte ..... | 35 |

## Rotationskörper 36

## Netzstrukturen (Grids) 38

|   |    |
|---|----|
| Überblick .....                               | 38 |
| Definition einer Netzstruktur .....           | 39 |
| Berechnungen an der Oberfläche .....          | 40 |
| Die Weiterverarbeitung der Netzstruktur ..... | 40 |

## Objektgruppen 42

|  |    |
|--|----|
| Verwaltung von Objektgruppen .....         | 42 |
| Darstellung von Gruppeneigenschaften ..... | 43 |
| Anwendungen von Objektgruppen .....        | 44 |

## Material Modelle 47

|  |    |
|--|----|
| Überblick .....                                | 47 |
| Dialog Spannungs/Dehnungs-Variablen .....      | 48 |
| Die Funktion des Materialmodells im Code ..... | 48 |
| Dialog Spannungs-Dehnungs-Funktionen .....     | 49 |
| Dialog Vergleichsspannungs-Hypothese .....     | 49 |
| Beispiel .....                                 | 50 |
| Möglichkeiten der Darstellung .....            | 51 |
| Der Dialog Edit Material Property .....        | 51 |
| Der Elastizitätsmodul (young modulus).....     | 51 |

## Federn 53

|                   |    |
|-------------------|----|
| Zugfedern .....   | 54 |
| Druckfedern.....  | 55 |
| Blattfedern ..... | 56 |

## Aktuatoren und Dämpfer 57

## Rohdaten 60

|  |    |
|--|----|
| Ein Element grafisch interaktiv erzeugen ..... | 60 |
| Der Dialog New Element .....                   | 61 |
| Der Dialog New Point .....                     | 62 |
| Ein Element per Makro erzeugen .....           | 62 |
| Format / Elementpunkte .....                   | 63 |
| Elemente gruppieren.....                       | 64 |
| Copy, Past, Duplicate (Elemente).....          | 64 |
| Konturen.....                                  | 65 |
| Mögliche Probleme mit Konturen .....           | 65 |

## Daten-Import 67

|                               |    |
|-------------------------------|----|
| Import Kontur (Textfile)..... | 67 |
| Import 3D-Polyline .....      | 67 |

|  |    |
|--|----|
| Import Grid.....                         | 68 |
| Import DXF File.....                     | 69 |
| Merge (sonar Files zusammenführen) ..... | 70 |

## Objekte Modifizieren 71

|  |    |
|--|----|
| Objekt-Eigenschaften ändern.....       | 71 |
| Objektgrösse ändern.....               | 72 |
| Objekte Spiegeln .....                 | 73 |
| Extruded Line-Arc-Prism editieren..... | 73 |

## Objekte Drehen & Positionieren 75

|                     |    |
|---------------------|----|
| Positionieren ..... | 75 |
| Drehen .....        | 76 |

## Die Bewegung der Objekte in Simulationen einschränken 77

|                                  |    |
|----------------------------------|----|
| Rotationsachsen einfrieren ..... | 77 |
| Alle Bewegungen einfrieren.....  | 77 |
| Bewegung Stoppen .....           | 78 |
| Modell Beruhigen.....            | 78 |

## Objekte verbinden (Links) 80

|                                  |    |
|----------------------------------|----|
| Einführung.....                  | 80 |
| Normaler Link.....               | 80 |
| Biege-Link.....                  | 81 |
| Torsions-Link.....               | 81 |
| Link-Kombinationen .....         | 81 |
| Die Wirkung der Link-Kraft ..... | 82 |
| Link erzeugen .....              | 82 |

## Simulation 84

|   |    |
|---|----|
| Zeitschritt .....                         | 84 |
| kritische Faktoren einer Simulation ..... | 86 |

## Kräfte auf Objekte 87

|   |    |
|---|----|
| Überblick.....                                  | 87 |
| Elastisch-plastische Stosskräfte.....           | 87 |
| Kontaktkräfte .....                             | 89 |
| Reibungskräfte .....                            | 90 |
| Link-Kräfte .....                               | 91 |
| Viskosität .....                                | 92 |
| Konstante Externe Kräfte .....                  | 93 |
| Gleichgerichtetes konst. Gravitationsfeld ..... | 94 |
| Zentrales Gravitationsfeld .....                | 94 |

## Kontrollsysteme 96

|   |     |
|---|-----|
| Punktkurven .....                       | 96  |
| Externe Punktkurven .....               | 97  |
| Kontrollsysteem mit math. Formeln ..... | 98  |
| Kontrollsysteem mit sonar script .....  | 100 |
| Simulationsabbruch .....                | 101 |
| Geschwindigkeitsbeschränkung .....      | 102 |

## Interaktionssteuerung 103

|  |     |
|--|-----|
| Einleitung .....                       | 103 |
| aktive und passive Objekte .....       | 103 |
| Bilaterale Interaktionsregeln .....    | 105 |
| Interaktionsregeln mit Namen .....     | 107 |
| Ablauf der Interaktionssteuerung ..... | 108 |

## Grafische Darstellungen 110

|                               |     |
|-------------------------------|-----|
| Einleitung .....              | 110 |
| Grafik Variablen .....        | 110 |
| Die Funktion der Grafik ..... | 111 |
| Datenaufzeichnung .....       | 111 |
| Grafik Zurücksetzen .....     | 112 |

## Filmaufzeichnungen 113

|  |     |
|--|-----|
| Quicktime Filme aufzeichnen .....          | 113 |
| Quicktime Filme komprimieren .....         | 115 |
| Für Fortgeschrittene: Bewegte Kamera ..... | 117 |

## Preferences 119

|  |     |
|--|-----|
| Programm-spezifische Einstellungen ..... | 119 |
| Hinweise .....                           | 119 |

## Reports & Logfiles 121

|  |     |
|--|-----|
| Report (Physical Properties) .....       | 121 |
| Logfile Geometry (selected object) ..... | 122 |
| Logfile Interactions .....               | 122 |
| Logfile Events .....                     | 123 |
| Logfile Macro / Logfile Syntax .....     | 123 |
| Tipps .....                              | 124 |

## **Vorwort**

---

Das vorliegende Tutorial richtet sich an Ingenieure in der Entwicklung und Konstruktion, im technisch-wissenschaftlichen Bereich und in der Lehre und Forschung. Das Handbuch versucht sowohl dem von Zeitnot getriebenen Anwender der einen raschen Einstieg in die Software sucht, als auch dem interessierten Benutzer der gerne etwas 'darum herum' erfahren möchte, etwas zu bieten. Ersterer kann dieses Buch vorerst beiseitelegen und direkt mit den Beispielen zur Erstellung erster Modelle beginnen und das Buch als Nachschlagewerk benutzen. Die einführenden Modelle findet man in:

1. sonar samples, Kapitel 'Ein erstes einfaches Modell: chaotisches Pendel'
2. sonar samples, Kapitel 'Uhrwerk'
3. sonar samples, Kapitel 'Experimente mit Drähten'

Die Beschreibung der betreffenden Modelle ist so angelegt, dass keine Vorkenntnisse zur Bedienung der sonar Software erforderlich sind.

Das vorliegende Tutorial gibt nicht erschöpfend Auskunft über die Bedienung der sonar Programme. Bei Fragen zur Bedienung eines bestimmten Dialogs empfehlen wir als nächste Anlaufstelle die sog. Arbeitsblätter. Auf einem Arbeitsblatt wird versucht in möglichst kompakter Form auf einer Papierseite die Bedienung eines bestimmten Dialogs, eines Bedienungselements oder einer Funktion der Software zu erklären. Die Arbeitsblätter bilden so gesehen zusammen ein weiteres Manual in einer anderen Struktur.

Die sonar Software ist ein Projekt deren Entwicklung im Jahre 2005 begann und auf absehbare Zeit nicht abgeschlossen sein wird. Die Entwicklung wird permanent weitergeführt, die Software stetig gemäss neuen Bedürfnissen der Benutzer erweitert und mit neuen Technologien angereichert. Das eigentliche Simulationsprogramm sonar-SIM wurde für eine maximale Leistung von Anbeginn an auf Mehrkern-Rechner und parallele Verarbeitung ausgelegt. Es wird erwartet, dass die Simulationstechnologie in den nächsten Jahren und Jahrzehnten stark an Bedeutung gewinnen wird. Die in Zukunft zur Verfügung stehende Rechenleistung wird die Möglichkeiten der Simulationstechnologie stark fördern und für eine zunehmende Verbreitung sorgen. Was die Akzeptanz der Simulationstechnologie im Allgemeinen bis heute behinderte waren eine nicht zufriedenstellende Leistung auf Desktoprechnern und eine anspruchsvolle Bedienung die sich an speziell ausgebildete Fachkräfte richtete. Die sonar Software will in beiden Belangen neue Massstäbe setzen und sich in diese Richtungen konstant weiterentwickeln. In diesem Sinne werden Anregungen aus der Benutzerwelt geschätzt und sehr ernst genommen.

im Herbst 2018  
L&G Software  
Fritz Leibundgut

# Einführung

---

## Numerische Simulation

---

Die numerische Simulation ist die rein rechnerische Simulation auf einem Computer. Ein anderer Ansatz wäre die reale physikalische Modell-Simulation in einem Laboratorium, wo man ein konkretes, meist verkleinertes, Modell der Wirklichkeit durchspielt und an ihm Messungen vornimmt. Die numerische Simulation ist im eigentlichen Sinn erst seit der Existenz von Computern möglich. Somit wurden die ersten numerischen Simulationen am Ende des zweiten Weltkrieges mit den ersten Relais- und Röhrenrechnern durchgeführt. So richtig Fahrt aufgenommen haben die numerischen Simulationen allerdings erst nach der Einführung der ersten Programmiersprachen. Am Anfang war das die eigentliche Maschinensprache, daraus abgeleitet anschliessend die sehr prozessornahen 'Assembly-Language', die auch heute noch für gewisse Zwecke verwendet wird. Die erste höhere Programmiersprache im heutigen Sinn erschien im Jahre 1957 von IBM und hieß FORTRAN (FORmula TRANslation). Eine grosse Zahl wissenschaftlicher Computerprogramme läuft auch heute noch unter dieser Sprache, die im Laufe der Zeit natürlich mehrmals erweitert und modernisiert wurde.

Die numerische Simulation ermöglicht die rechnerische Bewältigung von sehr rechenintensiven iterativen Berechnungen. Vor dem Computerzeitalter musste der Berechnungsingenieur eine technische Problemstellung soweit vereinfachen und in ein Berechnungsmodell verpacken, dass er dieses analytisch lösen konnte. Heute sind wir in der Lage, vergleichsweise komplexe physikalische Zusammenhänge in ihrem veränderlichen zeitlichen Verlauf durchzuspielen, weil wir in der Lage sind, jedes beliebige mathematische Gleichungssystem numerisch zu integrieren. Damit können wir physikalische Systeme nicht nur exakter lösen, sondern werden in die Lage versetzt, im Prinzip jedes mechanische System berechnen zu können, insbesondere auch solche die analytisch nicht lösbar sind. Einfache Beispiele zu diesem Sachverhalt sind das Fadenpendel und das sog. n-Körperproblem. Beim mathematischen Pendel sind zur analytischen Lösung Idealisierungen notwendig, ansonsten es sehr schnell komplizierter wird und in den Bereich elliptischer Integrale führt. Andererseits ist bereits das 3-Körperproblem in seiner allgemeinen Form nicht mehr analytisch lösbar. Auf Supercomputern werden heute Milliarden-Körper-Probleme gelöst

Die ersten Anwendungen der numerischen Simulation fanden im militärisch-wissenschaftlichen Komplex in amerikanischen Forschungseinrichtungen wie Los Alamos, Lawrence Livermore, Sandia Labs usw. statt. Es sind dieselben Standorte die auch heute noch die schnellsten Computer der Welt besitzen und in der Rangliste der sog. Top-500 immer ganz vorne dabei sind oder diese anführen. Das Stichwort zu diesen Berechnungen heisst 'Finite Differenzen Methode'. Wir kommen zu dieser Methode später zu sprechen. Vorläufig nur so viel: Es handelt sich um die rechenintensivste Simulationsmethode. Simuliert wurden damit von Anbeginn an die Reaktion von konventionellen Sprengstoffen in Zusammenhang mit Nuklearwaffen in den Phasenzuständen: Fest, Reaktion und Ausbreitung. Das spaltbare Material einer Kernwaffe wird immer über sehr raffinierte Anordnungen von konventionellen Sprengstoffen in die kritische Phase überführt. Das gleiche gilt für die Initierungsmechanismen des Zündsystems. Die numerische Simulation spielt zur Erforschung und Optimierung dieser Zusammenhänge eine entscheidende Rolle. Während die ersten Code's der Finite-Differenzen-Methode eindimensional waren, kamen in den 70-er Jahren des letzten Jahrhunderts 2-dimensionale und in den 80-er Jahren schliesslich 3-dimensionale Programme auf. Diese Entwicklung ging Hand in Hand mit der entsprechenden Entwicklung der Supercomputer.

Heute, wo die Durchführung von praktischen Nukleartests eingestellt wurde, wird das gesamte nukleare Arsenal der Supermächte unter Zuhilfenahme von numerischen Simulationsmethoden im Hinblick auf Weiterentwicklung, Unterhalt, Versagensanalyse, Alterungs- und Störungsanalyse gewartet. Dies ist letztlich der Grund, weshalb die nationalen Forschungseinrichtungen der USA

die Entwicklung der Supercomputer vorantreiben. Forscher in den Bereichen von Klimawandel, Astrophysik, usw. sind in diesem Sinne Nutzniesser, welche auch einen Teil der Rechenzeit an diesen Anlagen abbekommen. Zum anderen leisten sich heute auch rein wissenschaftliche Forschungszentren Supercomputer und verteilen die Rechenleistung auf Universitäten und Hochschulen.

## Ein Vergleich zwischen analytischen und numerischen Lösungsmethoden

---

Im Folgenden wollen wir an einem konkreten Rechenbeispiel ganz kurz den Unterschied der beiden Lösungswege darstellen. Wir tun dies am Beispiel des Fadenpendels, einer Problemstellung mit einer analytischen Lösung die jeder Ingenieur kennt.

Newton :  $F(t) = m \cdot b(t)$  (EQ 1) **Die analytische Lösung**

mit der Kraft (Fadenpendel) :  $F(t) = -m \cdot g \cdot \sin(\varphi(t))$  (EQ 2)

und der Winkelbeschleunigung (Fadenpendel) :  $b(t) = L \cdot (\partial^2/\partial t^2) \varphi(t)$  (EQ 3)

sowie der Einschränkung auf kleine Auslenkungen :  $\sin(\varphi(t)) \approx \varphi(t)$  (EQ 4)

erhalten wir die Differenzialgleichung des idealen mathematischen Fadenpendels

$(\partial^2/\partial t^2) \varphi(t) + g/L \cdot \varphi(t) = 0$  (EQ 5)

mit den Lösungen der Schwingung, der Periodendauer bzw. der Frequenz wie wir sie alle kennen. Die Lösung ist im analytischen Fall folglich eine Gleichung als Funktion der Zeit die uns für jeden Zeitpunkt angibt, wo sich das Pendel genau befinden wird und welche zugehörige Geschwindigkeit es dort hat.

Die numerische Simulation geht einen anderen Weg. Die numerische Simulation liefert uns keine vergleichbare Formel wo das Pendel sich zum Zeitpunkt  $t = t_1$  oder  $t = t_2$  befinden wird. In einer expliziten numerischen Simulation wird der Vorgang des Pendelns explizit durchgespielt, so als ob wir ein reales Pendel beobachten würden. In der Simulation starten wir z.B. mit einem ausgelenkten Pendel, lassen es los und beobachten was passiert. In der eigentlichen Software geschieht zu diesem Zeitpunkt Folgendes. Auf sämtliche Teile des Pendels (Pendelmasse, Pendelstab) wirkt die Gravitationskraft. Wir gehen davon aus, dass sich die Kräfte im Laufe einer Mikrosekunde nicht merklich ändern werden. Somit sind wir in der Lage mit den elementaren Newtonschen Gleichungen voraus zu berechnen, wo sich das Pendel eine Mikrosekunde später aufhalten wird. Wir führen diese Berechnungen für jedes Massteil in einem Modell einzeln und komponentenweise durch, d.h. für jede der drei Raumrichtungen (x, y, z) separat. Die Pendelmasse würde sich während einem Zeitschritt  $dt$  und in x-Richtung wie folgt von A nach B bewegen

### Die numerische Lösung

Zeit  $t = t_0$ :

- Kraft:  $F_x(t)$
- Geschwindigkeit:  $v_x(t)$
- Position:  $x(t)$

Zeit  $t + dt$ :

- Newton:  $b_x(t) = F_x(t) / m$
- Geschwindigkeit:  $v_x(t+dt) = v_x(t) + b_x(t) \cdot dt$
- Position:  $x(t+dt) = x(t) + (v_x(t)+v_x(t+dt))/2 \cdot dt$

In y- und in z-Richtung werden analoge Berechnungen durchgeführt. Am Ende eines sog. Zyklus, wenn wir die neue Geschwindigkeit und Position sämtlicher Teile neu berechnet haben, liegt ein neuer Zustand des Systems vor, so wie er sich eine Mikrosekunde später einstellt. Jedes Teil hat eine klitzekleine kaum sichtbare Positionsänderung erfahren und sich vielleicht auch um einen ebenso kleinen Winkel gedreht. Dieser neue Zustand führt dazu, dass sich indirekt auch die Kopplungs Kräfte zwischen den Objekten etwas ändern. Es entstehen neue Lagerkräfte. Wenn wir nun unter diesen neuen Voraussetzungen in den nächsten Zyklus, für die nächste Mikrosekunde, starten, dann werden wir die nächsten Berechnungen für den nächsten Zeitschritt 'dt' unter entsprechend veränderten Bedingungen durchführen, denn die veränderten Kräfte ergeben andere Beschleunigungen und damit andere Geschwindigkeitsänderungen, usw.

Durch ständiges Wiederholen dieses Vorgangs (Iteration) arbeiten wir uns auf der Zeitskala vorwärts. Daraus sehen wir den Unterschied zur analytischen Methode. In der Simulation fliegen wir quasi mit und wir wissen nicht zum voraus wohin die Reise geht. Um festzuhalten wo wir waren, müssen wir unsere Wege und Geschwindigkeitsänderungen aufzeichnen, damit wir am Schluss überblicken können was wir eigentlich erlebt haben. Ein weiterer Unterschied den wir erkennen können ist die Tatsache, dass die explizite numerische Berechnung keine spezifischen Kenntnisse über das Modell haben muss. Die iterativen Berechnungen sind weitgehend universell und funktionieren mit jeder Art von Modell. Ein expliziter Code kann deshalb jedes mechanische System berechnen, egal welcher Art und wie komplex es auch sein mag. Und um auf das Pendelsystem zurückzukommen, die typische Pendelbewegung entsteht in unserer numerischen Simulation wie von Geisterhand gesteuert von selbst. Im Gegensatz zur analytischen Lösung, wo wir das Modell im Prinzip bereits in die Berechnung mit eingebaut hatten, "weiss" unsere Simulation nicht um was es geht. Und trotzdem kommt hinten auf wundersame Weise die korrekte Pendelbewegung heraus. Wie kann das System so etwas leisten ohne diese Vorkenntnisse? Nun, die wirkliche Natur weiss ja eigentlich auch nicht was wir mit ihr anstellen. Die real existierende Pendelmasse hat ebenso wenig ein Gehirn welches ihr sagt: "Achtung, du befindest dich im Moment im Aufschwung eines Pendelsystems und musst nun darauf achten, dass Du dich auch so benimmst und im nächsten Moment etwas langsamer wirst um auf der Sinuskurve zu bleiben". Die wirkliche Natur reagiert einfach auf das was wir tun und was in jedem Moment mit ihr geschieht. Sie reagiert darauf mit den elementaren newtonischen Bewegungsgleichungen. Auf alle einwirkenden Kräfte reagiert sie mit Gegenkräften wie Newton sie vorschreibt. Und das ist genau das, was wir mit einem expliziten Code wie sonar auch tun.

Betrachten wir dazu nochmals das letzte Gleichungssystem, so stellen wir fest, dass diese Gleichungen rein gar nichts über unser Modell aussagen. Im allerersten Zeitschritt nach dem Simulationsstart werden folglich einfach alle Teile die nicht raumfest und unbeweglich verankert sind ein ganz kleines Stück Richtung Erde fallen, denn die Gravitationskraft ist vorerst die einzige wirkende Kraft im ganzen Pendelmodell. Bereits im nächsten Zeitschritt aber wird im Lagerpunkt durch diese kleine Verschiebung eine Kollision zwischen Pendelachse und Lager festgestellt werden, was eine zusätzliche Kraft ins Spiel bringt: die Lagerkraft. Diese Lagerkraft ist vorerst sehr klein, wird aber im weiteren Simulationsverlauf von Zyklus zu Zyklus grösser werden und schliesslich, vielleicht nach 50 Zyklen, was einer Zeitdifferenz von ca. 50 Mikrosekunden entspricht, das Gewicht des Pendels erreichen. Dies wird dazu führen, dass das Pendel nicht mehr weiter fallen wird. Es wird sich ein Gleichgewichtszustand zwischen Pendelgewicht und Lagerkraft einstellen. Der beschriebene Vorgang ist keine akademische Gedankenkonstruktion, sondern letztlich genau das was in einem realen Pendelsystem auch geschieht. Wir ahmen also wirklich die Natur auf ihrem elementarsten Niveau und auf universelle Art und Weise nach. Wir simulieren das Kräftespiel zwischen allen verbundenen und interagierenden Teilen eines mechanischen Systems.

Eine gewöhnliche Simulation dieser Art verschlingt schnell einmal eine Million Rechenzyklen. Es können aber auch 10 oder 100 Millionen oder noch mehr werden. Ein berechtigter Einwand, den die meisten Benutzer früher oder später hervorbringen ist der Folgende: Wie kann ich glauben, dass ein solchermassen simulierter Vorgang in Millionen von nacheinander ausgeführten Rechenzyklen überhaupt noch präzise und zuverlässig ist? Weshalb divergiert dieser Vorgang nicht allmählich und führt nach ein paar tausend Zyklen in eine völlig andere Richtung? Zu diesem Thema gibt es wissenschaftliche Abhandlungen. Auch der bekannte Computerpionier John von Neumann hatte sich diese Frage gestellt, als er während des zweiten Weltkriegs am Manhattan Projekt arbeitete. Er führte damit wissenschaftliche Untersuchungen weiter, die bereits früher

veröffentlicht wurden und heute als Courant-Friedrichs-Stabilitätsbedingung bekannt sind. Das Ergebnis dieser Untersuchungen lässt sich wie folgt zusammenfassen.

*Ein simuliertes Gleichungs-System der besprochenen Art 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.*

Die Berechnung des Zeitschrittes wird in der sonar-Software automatisch vom System durchgeführt. Der Benutzer kann den Zeitschritt aber auch verändern. In der Regel führt eine Erhöhung allerdings zu einer 'Explosion' des Modells. Ist der Zeitschritt auch nur geringfügig zu hoch, dann bauen sich sehr schnell numerische Schwingungen auf, welche das gesamte Modell zum Bersten bringen. Man sitzt dann vor einem leeren Bildschirm. Dies ist an sich auch eine beruhigende Information. Kommt die Simulation an ein Ende, dann war der Zeitschritt offenbar durchgehend klein genug und das Resultat ist somit im Rahmen der gegebenen Anfangsbedingungen mit höchster Wahrscheinlichkeit auch korrekt.

Die analytische Lösung des Fadenpendels, wie weiter oben angeschrieben, ist übrigens nur im Spezialfall von kleinen Auslenkungen des Pendels richtig (Equation 4). Die numerische Lösung dagegen ist immer korrekt, d.h. für jede Auslenkung des Pendels.

Ein rein expliziter Code wie 'sonar' löst ein physikalisches Problem und die Bewegungsanalyse all seiner beteiligten Objekte nicht damit, indem er intern auf eine Formelsammlung zugreift und mit allen bekannten physikalischen Formeln die Bewegung der Objekte stückweise berechnet. Es wird auch nicht auf Differentialgleichungen zugegriffen welche all den Teilaufgaben einer Problemstellung zugeordnet werden könnten. Das wäre eine dilettantische Art zu simulieren.

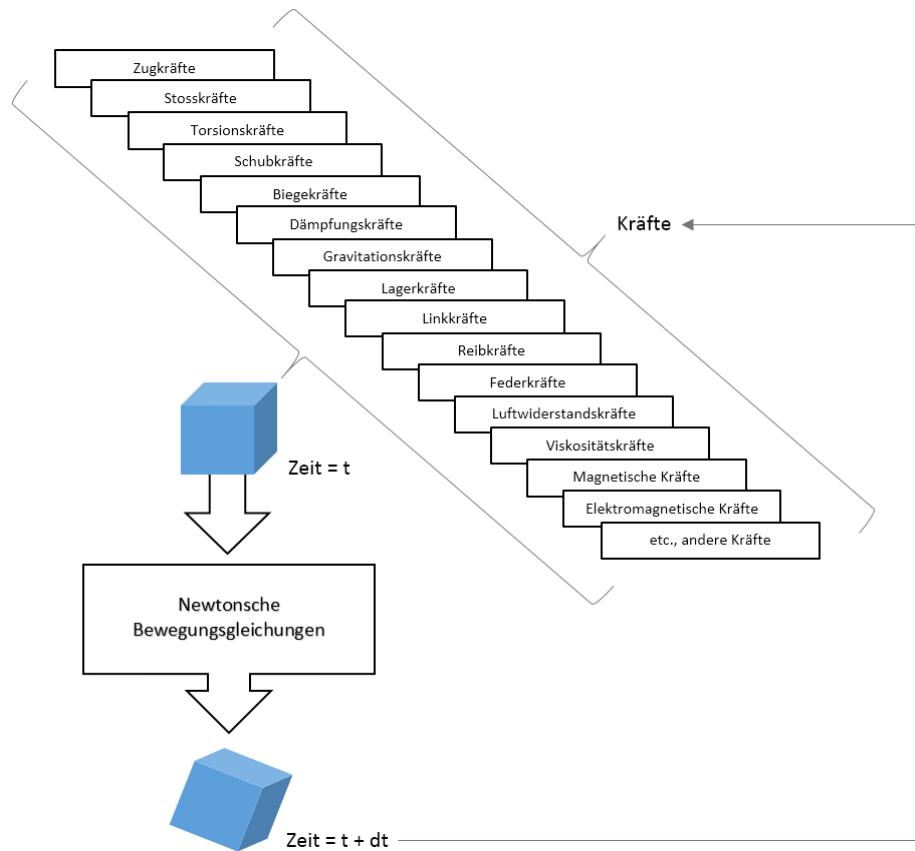
## Zusammenfassung

Jede Differentialgleichung und jede physikalische Formel, wie sie in einer Formelsammlung steht, wurde irgendwann von irgendjemandem aus den elementaren Newtonschen Bewegungsgleichungen hergeleitet. Jede physikalische Formel kann folglich weiter zerlegt werden, bis wir am Ende bei Newton landen.

- Kraft = Masse · Beschleunigung
- Actio = Reactio

Das ist Physik. Alles andere ist nur darüber gestülpte Mathematik. Viele Herleitungen von Formeln werden von Ingenieuren während einer Ausbildungszeit erlernt, aber später vergessen, weil man für das analytische Rechnen mit Stift und Papier nur noch die resultierende Formel braucht wie sie im Formelbuch steht.

Unsere 'sonar' Software rechnet auf der Basis der elementaren Newton'schen Bewegungsgleichungen, also auf der elementarsten Stufe der Physik und auf dem wahrhaftigen Niveau von Ursache und Wirkung. Der Eine oder Andere mag an diesem Punkt fragen: "Warum so kompliziert?" oder "Warum nehmen wir nicht gleich die fertige Formel, statt alles nochmals auf diesem elementaren Niveau durchzuspielen?" Die Antwort lautet: Schauen Sie nochmals zurück an den Anfang diese Kapitels. Bereits ein sehr einfaches Problem wie das Fadenpendel konnten wir analytisch nur sehr eingeschränkt lösen. nämlich für den Fall, wo die Auslenkung des Pendels sehr klein ist. Für grössere Winkel, also für alle anderen Fälle ist die Lösung nicht mehr gültig. Bereits das 3-Körper Problem ist analytisch nicht mehr lösbar. Mit der numerischen Simulation lösen wir das Milliarden-Körper-Problem. Jedes mehr oder weniger komplexe Modell ist letztlich voll von solchen Einschränkungen. Deshalb war und ist es bis heute die Kunst des Ingenieurs für seine Problemstellung analytisch ein Modell zu erzeugen, welches der realen Lösung möglichst nahe kommt.

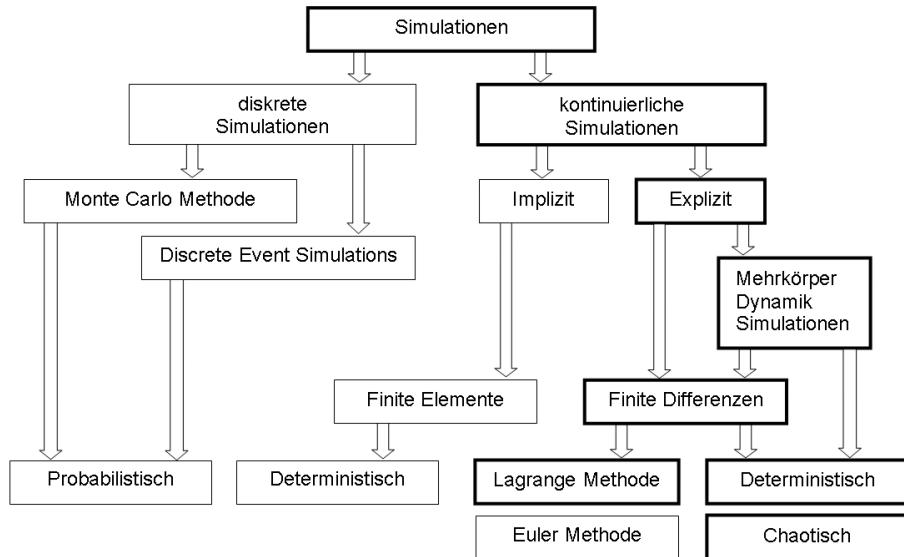


**FIGURE 1.** Der iterative Ablauf einer expliziten Simulation

Mit der numerischen Simulation wie 'sonar' sie verwendet, entfallen all diese Einschränkungen, Vereinfachungen und Simplifizierungen die früher notwendig waren um Probleme analytisch lösen zu können. Mit einem rein expliziten Code wie 'sonar' kann grundsätzlich jedes mechanische Problem beliebiger Komplexität uneingeschränkt und ohne Kompromisse sehr realitätsnah simuliert werden. Mit sonar Simulationen sind sogar Ergebnisse möglich, an die vorher niemand gedacht hatte. Es können Effekte passieren, die man vorher nicht kannte oder die man nicht für möglich hielt. Dies ist möglich, weil 'sonar' das gesamte Kräftespiel in einem Modell auf der unmittelbaren Stufe von Ursache und Wirkung berechnet, d.h. genauso wie die Natur selbst auch funktioniert.

## Simulationsprogramme und Methoden

Man unterscheidet in der Technik verschiedene Klassen und Arten von Simulationsprogrammen, die sich insbesondere auch in Bezug auf die Rechenmethoden unterscheiden.



**FIGURE 2.** Die fett umrandeten Begriffe zeigen die 'sonar'-Welt

In mancherlei Hinsicht sind die Anwendungsgebiete der verschiedenen Methoden teilweise überlappend und lassen sich nicht immer eindeutig zuordnen. Ohne Anspruch auf Vollständigkeit zu erheben, soll die sonar-Umgebung wie folgt in die bestehende Begriffswelt eingeordnet werden.

- sonar ist ein sog. Mehrkörperdynamik-Simulationsprogramm
- führt kontinuierliche Simulationen durch
- ist ein rein expliziter Programm-code
- ist verwandt mit den sog. Finite-Differenzen-Methoden.
- eignet sich besonders für dynamische bis hochdynamische Problemstellungen
- lässt die Berechnung von deterministischen und chaotischen Modellen zu

sonar ist .....

Diese Begriffe werden nachfolgend etwas genauer erklärt. Als erstes unterscheiden wir zwei Klassen von Simulationen mit den Begriffen 'diskrete' oder 'kontinuierliche' Simulationen. Diese unterteilen sich weiter wie folgt:

### Begriffserklärungen

Die Monte Carlo Simulations-Methode stützt sich auf die Wahrscheinlichkeitsrechnung. Ein Monte Carlo Modell basiert auf einer endlichen Menge von möglichen Ereignissen. Das Eintreffen eines jeden Ereignisses wird mit einer bestimmten Wahrscheinlichkeit festgelegt. Betrachtet man nun mögliche Ereignisfolgen, so werden diese Basiswahrscheinlichkeiten entsprechend miteinander multipliziert und geben die Wahrscheinlichkeit für das Eintreffen eines ganzen Ablaufs von Ereignissen an. Das wesentliche am Ganzen ist nun, dass sich das Modell ständig verändert und dass sich damit auch die Wahrscheinlichkeiten für das Eintreffen sämtlicher Einzelereignisse kontinuierlich ändern. Es ist dieser Simulationsmethode deshalb eigen, dass

Monte Carlo Simulation

immer eine sehr grosse Anzahl von Szenarien durchgespielt werden müssen, um letztlich eine hinreichend genaue Gesamtwahrscheinlichkeit eines Vorgangs zu ermitteln.

Beispiel: Simulation der Abschusswahrscheinlichkeit eines Flugzeuges mit einer Fliegerabwehrkanone. Verschiedene Parameter wie Zieldistanz, Flugbahn des Flugzeuges, projizierte Grösse des Ziels, seine Geschwindigkeit, usw. werden dabei in vielen kleinen Schritten im Rahmen der möglichen Bandbreite variiert. Mit jedem einzelnen Satz von Daten wird anschliessend die Wahrscheinlichkeit für den Abschuss des Flugzeuges berechnet. Die Summe der Resultate führt schliesslich zu einer Gesamtwahrscheinlichkeit, dass das Flugzeug abgeschossen wird. Weitere Wahrscheinlichkeitswerte zeigen im Einzelnen auf, welche Teile des Ziels wie oft getroffen werden. Wieder andere Wahrscheinlichkeiten sagen schliesslich aus, wie letal das Eindringen eines Geschosses in den fraglichen Bereich des Ziels ist. Damit liessen sich letztlich Empfehlungen abgeben, welche Komponenten des Ziels besser geschützt werden könnten um die Abschusswahrscheinlichkeit herabzusetzen, usw.

## Diskret

Diskrete Simulationen bewegen Objekte in diskreten Ereignis-Schritten, welche als Folge des Ortes, der Zeit oder mit einer gewissen Wahrscheinlichkeit eintreffen. Diskrete Simulationen haben manchmal auch zu tun mit der Wahrscheinlichkeitsrechnung und der Statistik. Man behandelt mit diesen Methoden Ereignisfolgen, die funktionell, sequentiell oder mit einer Regelmässigkeit eintreffen. Beispiele sind: Simulationen von Verkehrsproblemen, Fluss von Menschenansammlungen, Börsenspekulationen, Warteschlangen (queues), Fluss von Paketen auf Förderanlagen bei der Post, usw.

## Kontinuierlich

Kontinuierliche Simulationen sind exakte Berechnungen auf konkreten physikalischen Grundlagen die in steter Abfolge bzw. kontinuierlich durchgespielt werden. Mit Hilfe von Bewegungsgleichungen wird die Bewegung von physikalischen Objekten durch Raum und Zeit unter Berücksichtigung aller Kräfte exakt berechnet. Die Berechnung der Flugbahn einer Rakete zum Mond wäre ein solches Beispiel.

## Mehrkörperdynamik Simulation

Das ist die Welt in der wir uns mit sonar bewegen. Wir sprechen von einer sog. kontinuierlichen Simulation, wo Objekte in sehr kleinen Zeitschritten von typischerweise Mikro- bis Nanosekunden durch den Raum bewegt werden. Die Basisobjekte in sonar sind sog. 'solids'. Je nach Ausbaustufe und zusätzlichen Tools wird diese Methode weiter verfeinert in Richtung Finite Differenzen.

## Finite Elemente

Als 'Finite Elemente' versteht man umgangssprachlich in der Regel eine implizite Rechenmethode für die Strukturanalyse von Festkörpern. Das Ziel ist die Ermittlung der Spannungsverteilung unter bestimmten Lasten und die Berechnung von Eigenwerten und Eigenschwingungen von Einzelobjekten oder ganzer Baugruppen. Dazu wird eine Konstruktion in diskrete Objekte aufgegliedert oder die Einzelteile mit inneren Netzstrukturen versehen. Die Auflösung bzw. Genauigkeit der Ergebnisse ist von der Grössenordnung der Zelldimensionen des Netzes. Bei der Finiten Elemente-Methode interessiert man sich in der Regel für den Endzustand bzw. den Gleichgewichtszustand unter einer äusseren Last und weniger dafür, wie man dorthin gelangt. Bei der nachfolgenden Finite-Differenzen Methode ist es umgekehrt. Dort interessiert man sich mehr um die eigentliche Bewegung, wie genau und längs welchem Pfad bewegen sich die Teile und wie verformen sie sich dynamisch als Funktion der Zeit.

## Finite Differenzen

Auch die Finite Differenzen Methoden verfeinern die solids mit inneren Netzstrukturen. Allerdings liegen den Finiten Differenzen rein explizite Rechenmethoden zugrunde. Dabei unterscheidet man zwei Methoden (Lagrange Formalismus, Euler Formalismus). Die Lagrange Methode bewegt während der Simulation die Knotenpunkte des inneren Netzes auf Grund von Druck- und Spannungsdifferenzen zwischen benachbarten Zellen. Der Lagrange Methode ist es folglich eigen, dass sich das innere Netz mit seinem Objekt mitbewegt. Anders gesagt, das Netz ist das Objekt. Diese Methode erlaubt nebst der Bewegungssimulation der Objekte als Ganzes, die explizite dynamische Simulation der inneren Spannungen und der

### Deformation von Körpern.

Bei der Euler Methode liegt dem Modell ein raumfestes Gitter zugrunde, welches den gesamten Simulationsbereich abdeckt. Anschliessend werden die Objekte in diese Gitterstruktur eingefüllt und bewegen sich während der Simulation unter Zuhilfenahme von Transportgleichungen durch das Gitter hindurch. Diese Methode wird eingeschränkt auch für Festkörper verwendet, zur Hauptsache aber in der Simulation von Flüssigkeits- und Gasströmungen eingesetzt. Die Luftströmung um einen Flugzeugflügel ist ein typischer Anwendungsfall für diese Methode (virtueller Windkanal).

Implizite und explizite Rechenmethoden sind zwei mathematische Klassen welchen völlig unterschiedliche Rechenmethoden zugrunde liegen. Das klassische Finite-Elemente Programm ist ein impliziter Programmcode, welcher den Gleichgewichtszustand und Eigenwerte eines mechanischen Systems berechnet. Ein System mit 'n' Freiheitsgraden führt zu einem Gleichungssystem mit 'n' Gleichungen und 'n' Unbekannten und weiter zu einer  $[n \times n]$ -Matrix, also in die Matrizenrechnung. Dies ist ein Erkennungsmerkmal für ein implizites System. Das Lösen dieses Matrix-Systems ergibt einen Lösungsvektor der Dimension 'n' welcher z.B. die Gleichgewichtskräfte aller 'n' Elemente angibt. In einem impliziten System interessiert man sich folglich nicht 'explizit' für die Bewegungen in einem mechanischen System, sondern mehr für die End- bzw. Gleichgewichtszustände beim Anlegen von bestimmten Kräften.

**Implizit**

Ein expliziter Programmcode funktioniert viel mehr wie die physikalische Wirklichkeit. Ein Modell besteht aus physikalischen Objekten, an welchen unterschiedliche Kräfte angreifen und welche, einmal in Bewegung versetzt, untereinander interagieren. Man interessiert sich vorerst nicht für Gleichgewichtszustände, sondern konzentriert sich darauf, die Bewegung der einzelnen Objekte als Folge der Krafteinflüsse mit Hilfe der elementaren newtonischen Bewegungsgleichungen zu lösen. Man geht davon aus, dass sich die Kräfte des mechanischen Systems für die nächste Mikro- oder Nanosekunde nicht merklich ändern und kann unter dieser Voraussetzung die voraussichtliche Geschwindigkeits- und Weg-Änderung der Objekte für diese Zeit vorausberechnen. Dies führt zu einem neuen Zustand des Systems mit neuen Kräften, wie er eine Mikro- oder Nanosekunde später vorliegt. Durch ständiges Wiederholen dieses Prozesses (Iteration) wird letztlich die Bewegung des gesamten Systems in Millionen oder Milliarden von Zyklen durchgespielt. Explizite Simulationsprogramme gehören zu den rechenintensivsten Computerprogrammen die es gibt und erfordern Hochleistungsrechner.

**Explizit**

Deterministische Modelle sind vorhersagbare, theoretisch vollständig berechenbare Modelle. Mit ihnen lässt sich der räumliche Verlauf der Bewegung eines Objektes und damit der Aufenthaltsort in der Zukunft vorausberechnen. Das einfache Fadenpendel ist ein deterministisches System.

**Deterministisch**

Für chaotische Modelle ist das nicht der Fall. Chaotische Modelle haben die Eigenschaft, dass eine unendlich kleine Änderung an den Anfangsbedingungen des Modells nach endlicher Zeit zu beliebig grossen Abweichungen des Verhaltens des Modells führt. Das sog. chaotische Pendel ist ein Beispiel für dieses Verhalten. Das Wetter und seine Vorhersage ist ein weiteres Beispiel.

**Chaotisch**

## Die Modellierung in sonar

Die Begriffe 'Modell' und 'Simulieren' sind eng miteinander verbunden. Ein Modell ist eine Abbildung eines Teilbereichs der Realität in den Bereich der Mathematik, Physik und Ingenieur-Wissenschaften. Ein Modell kann ein konkreter Nachbau der Natur in einem kleineren Massstab sein. So werden z.B. sog. Geröll- und Schlammlawinen (Muren) in einem kleineren Massstab in Laboratorien nachgestellt und erforscht. Auch Schiffs- und Flugzeugmodelle in Wasser- bzw. Windkanälen sind nichts Anderes als verkleinerte Varianten von realen Objekten.

**Das Modell**

Besonders seit dem Aufkommen der Computertechnik und der numerischen Simulation auf Computern haben Simulationen generell stark an Bedeutung gewonnen. So werden z.B. Windkanalversuche an neuen Flugzeugentwicklungen zu einem grossen Teil durch rein rechnerische Simulationsverfahren ersetzt. Das ursprüngliche Problem, ein möglichst gutes Abbild der Natur zu schaffen, hat sich dadurch aber eher noch verstärkt. Viele Teile des Windkanalmodells, welche vorher noch konkret als physikalische Media in ihrer realen Form verwendet wurden, so z.B. die Luft, müssen jetzt im Modell auch noch in ihrer gesamten physikalischen Komplexität nachgebildet werden.

Die Kunst des Modellierens besteht darin, was wesentlich ist in das Modell zu übernehmen und Details, welche wenig oder keinen Einfluss auf die Ergebnisse haben, wegzulassen. Beim Modellieren kommt auch hier der Satz von Einstein zum Tragen der heisst: Man soll alles so einfach wie möglich betrachten, aber nicht einfacher. Manchmal ist auch ein Vorgehen mit einer schrittweisen Verfeinerung sinnvoll. Nach dem Motto "vom Groben zum Feinen" versucht man in einem ersten Anlauf ein einfacheres Modell zu simulieren und verändert anschliessend gewisse Objekte oder fügt weitere Details hinzu.

## Wie modellieren?

Der sonar-code simuliert die wahren äusseren Formen von realen Maschinenteilen und Mechanismen wie sie in der Technik vorkommen oder existieren könnten. Der Benutzer von sonar sollte immer bestrebt sein, die reale Welt möglichst realistisch in die Simulationswelt von sonar zu übernehmen. Es ist z.B. nicht möglich zwei Teile mit einem 'Strich-Hebel' zu verbinden, welcher nur aus einer Linie besteht. Es ist auch nicht möglich, für einen beweglichen Schieber einen 'Strich' bzw. eine geometrische Linie als Linearführung einzusetzen. Das gibt es in der realen Welt nicht und in sonar ebenso wenig. Alle funktionellen Teile in sonar müssen Volumenkörper sein und müssen eine träge Masse haben. Alle Körper in sonar haben folglich eine definierte geschlossene Oberfläche und interagieren untereinander automatisch über diese. Wann immer zwei Körper zusammenstossen oder sich berühren, dann registriert dies sonar automatisch und veranlasst alles Notwendige um diese Interaktion physikalisch korrekt zu berechnen und zu simulieren. Das Programm bedient sich dazu der Eigenschaften der Körper, welche letztlich wieder der Benutzer bestimmt. Der Benutzer sagt, welche Masse, welches Trägheitsmoment, welche Oberflächenhärte und welchen Reibungskoeffizient, usw. ein Objekt hat, wenngleich das Programm den Benutzer auch in der Bestimmung dieser Grössen unterstützt, wann immer das möglich ist. So berechnet sonar im Allgemeinen alle physikalischen Eigenschaften auf Grund einer Materialangabe und der äusseren Form eines Körpers automatisch.

## Objekt-Verbindungen

Etwas abstrakter wird die Modellierung erst dann, wenn sich die Frage nach Verbindungen zwischen Objekten stellt. In der Technik gibt es vielerlei Möglichkeiten, zwei Körper miteinander zu verbinden. Um nur einige aufzuzählen: Schrauben, Nieten, Schweißen, Punktschweißen, Kleben, usw. Während man sich noch gut vorstellen kann, eine Niet- oder Schraubverbindung auch mit entsprechenden formbehafteten Körpern nachzubilden, so ist das bei einer Schweiß- oder Klebe-Verbindung nicht mehr ohne weiteres möglich. Bei Verbindungen müssen also abstraktere Funktionen eingeführt werden, welche diese Verbindungen physikalisch korrekt nachbilden. In sonar ist das der sog. 'Link'. Ein Link ist im Prinzip eine gedämpfte Feder mit der Ruhelänge gleich Null, welche zwei Objekte punktförmig verbindet. Jede Verbindung in sonar ist folglich eine elastische Verbindung. Dies deckt sich mit der physikalischen Wirklichkeit. Selbstverständlich kann der Benutzer die Elastizität und die Dämpfungseigenschaften dieser Links beliebig numerisch beeinflussen, wenn er will. Er kann auch einfach qualitative Vergleichswerte setzen (z.B. weich wie Gummi oder hart wie Stahl). Das gleiche gilt für die Oberflächen der Objekte. Auch diese können untereinander weich oder hart und elastisch oder eher plastisch kollidieren. Der Benutzer kann diese Oberflächenhärte der Teile beliebig festlegen.

## Wenn Sie aus der CAD-Welt kommen..

Wenn Sie aus der CAD-Welt kommen und mit der Konstruktion von Maschinen oder Apparaten vertraut sind, dann sind Ihnen vielleicht bereits sog. Bewegungs- und Animationsprogramme begegnet, welche mit den meisten CAD-Programmen mitgeliefert werden. Programme dieser Art erlauben es dem Konstrukteur und Ingenieur, seine Konstruktion in Bewegung zu sehen, gewisse

Kollisionsprüfungen durchzuführen und sogar physikalische Eigenschaften wie Massen und Geschwindigkeiten einzubringen um damit Filme des bewegten Systems zu erzeugen. Für einfache Problemstellungen sind diese Hilfsmittel durchaus nützlich, man muss sich aber immer auch die Grenzen dieser rudimentären Berechnungen vor Augen halten. Diesen Berechnungsprogrammen ist es eigen, dass sie die Eingabe von allen physikalischen Beziehungen zwischen den Objekten verlangen und bereits an ihre Grenzen stoßen, wenn komplizierte Oberflächen in Kontakt treten sollen. Oft wird die räumliche Bewegung von Objekten mit mathematischen Funktionen und entsprechenden Kurven festgelegt. Solange man sich nur für die Beobachtung der relativen Bewegung mehrerer gekoppelter Objekte interessiert geht das völlig in Ordnung. Wenn aber die Analyse von physikalischen Größen ins Spiel kommt, dann sind diese Tools definitiv nicht mehr zuverlässig.

Falls Ihnen diese Art zu arbeiten bekannt vorkommt, dann müssen Sie sich umstellen. In sonar werden Objekte nicht künstlich mit Formeln und Raum- bzw. Punktkurven in Bewegung versetzt. Letztlich wäre zwar auch dies möglich, aber viel einfacher ist es in sonar beispielsweise eine Kurvenscheibe und ein Hebelsystem welches eine komplexe Bewegung erzeugt, mit in ein Modell aufzunehmen. Man bekommt anschliessend die komplizierte räumliche Bewegung des betreffenden Objektes und des gesamten verbundenen Antriebsstrangs geschenkt. Und das alles ohne dass der Benutzer sich um die Details der Kontakte zwischen den Objekten Gedanken machen muss. Weil sich sonar um alle Interaktionen zwischen den Objekten automatisch kümmert, entstehen die kompliziertesten Bewegungen von Getrieben, Kurvenscheiben und Hebelsystemen völlig selbstständig ohne Zutun des Benutzers. Und diese Bewegungen werden zudem mit allen Mitteln der Physik simulationstechnisch berechnet. Der Benutzer legt nur die Randbedingungen fest, d.h. welches Drehmoment oder welche Drehmomentfunktion die Antriebswelle in Bewegung versetzt, aus welchem Material die einzelnen Teile sind, welche Reibwerte benutzt werden sollen, usw.

Der Benutzer von sonar kann Daten vom CAD-System importieren, falls er mit einem solchen System arbeitet. Dazu steht die sog. DXF-Schnittstelle zur Verfügung, welche für sonar bis auf weiteres die Standardschnittstelle zur CAD-Welt darstellt. DXF ist ein Austauschformat des Herstellers Autodesk. Die Anbindung an dieses Format geschah deshalb, weil ein DXF-file sehr einfach aufgebaut ist, im Textformat vorliegt und von einem Benutzer sogar mit jedem Texteditor selbst verändert oder manipuliert werden kann. Allerdings wird in der Regel nur ein geringer Teil der Daten, wie sie normalerweise in einem DXF-file vorhanden sind, tatsächlich in die sonar-Software importiert. Dies liegt letztlich daran, dass der grundlegende Datenaufbau in sonar sich unterscheidet mit der Art und Weise, wie der Benutzer in einem CAD-Programm seine Objekte erzeugt. Immerhin, wo möglich werden die DXF-Daten in das eigene sonar-Format transformiert.

## Datenübernahme

Im Weiteren können in sonar auch andere Textformate von Geometrie-Informationen verwendet werden, die der Benutzer selbst mit EXCEL, WORD, EDITOR oder anderen Programmen vorbereitet hat. Wir kommen darauf später zu sprechen.

Die Philosophie hinter 'sonar' ist: Sie definieren ein mechanisches System und spezifizieren die Rand- und Umweltbedingungen. Sie haben dem Programm nicht zu sagen, was geschehen soll und wie sich die einzelnen Teile der Maschine und das System als Ganzes zu bewegen haben. Das erledigt das Programm für Sie. Wenn Sie das alles schon wüssten, dann müssten Sie letztlich nicht mehr simulieren. Sie stellen ein Modell in die virtuelle physikalische Welt von 'sonar', drücken auf den Knopf, lassen das Modell laufen und sehen zu, was passiert.

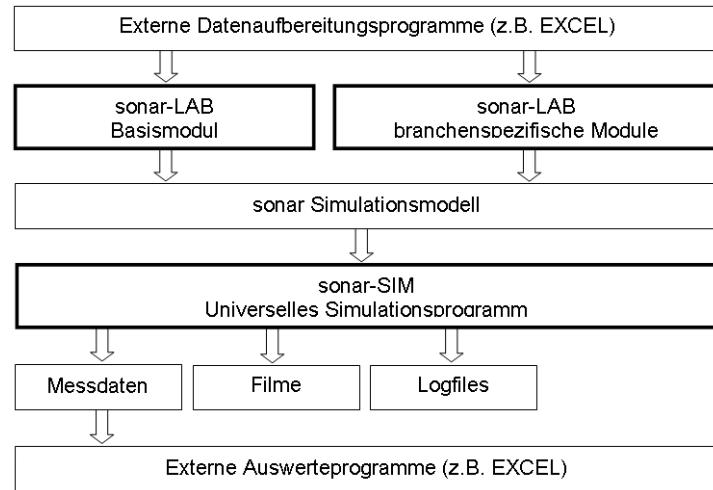
## Philosophie

### **sonar Programm Module**

---

Die sonar Software ist ein modulares Programm Paket. Wir unterscheiden zum ersten zwischen mehreren Modell-Erzeugungsprogrammen und einem eigentlichen universellen Simulationsprogramm mit dem wir bestehende Modelle simulieren. Die Modell-Erzeugungsprogramme unterteilen sich weiter in ein zentrales Basisprogramm und mehrere branchenspezifische Module. Diese Unterteilung wurde nicht deshalb eingeführt, weil Modelle

unterschiedlicher Branchen letztlich anders simuliert werden, sondern weil jede Branche bevorzugte und spezielle Objekte besitzt, welche die Software mit entsprechend angepassten branchenspezifischen Erzeugungsmethoden unterstützt. Die Modell-Erzeugungsprogramme bezeichnen wir als sonar-LAB Umgebung, das eigentliche zentrale Simulationsprogramm als sonar-SIM. Das vorliegende Manual befasst sich mit dem Basismodul von sonar-LAB.



**FIGURE 3.** sonar Programm Umgebung

## Aufbau eines sonar Modells

Nach dem Prinzip 'vom Groben zum Feinen' erkunden wir die Modellierungsmöglichkeiten von sonar vorerst ganz grob gemäss der folgenden Unterteilung. Anschliessend gehen wir auf einzelne dieser Punkte etwas genauer ein. Ein sonar Modell setzt sich in der Regel aus unterschiedlichen konkreten und abstrakten Bausteinen und Objekten zusammen, mit denen der Benutzer ein möglichst realitätsnahes Modell der Wirklichkeit aufbaut. Wie immer gilt, dass der gute Ingenieur ein Gespür dafür entwickelt, wo Abstraktionen der Realität zulässig sind, und wo eine speziell detaillierte Modellierung notwendig ist um am Ende ein ausgewogenes Modell zu erhalten.

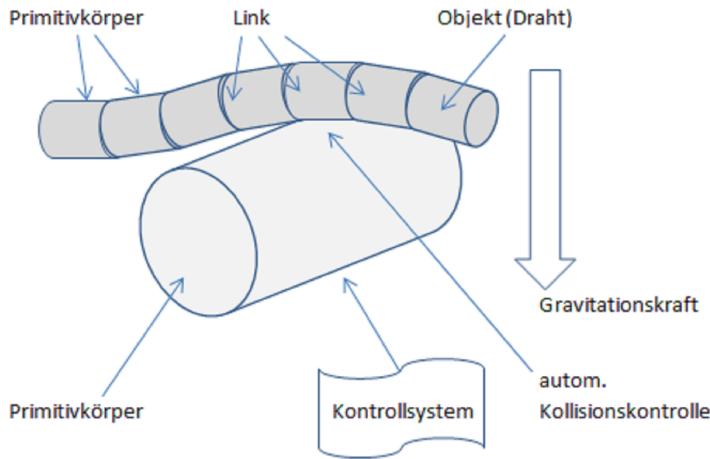


FIGURE 4. ein sonar Modell (schematisch)

Das abgebildete, schematische Teilmodell, welches Teil eines grösseren Modells sein könnte enthält z.B. folgende sonar-Bestandteile:

- |  |                           |
|--|---------------------------|
| • Objekte (Maschinenteile), bestehend aus                              | <b>Modellbestandteile</b> |
| • Primitivkörpern, welche verbunden sind mit                           |                           |
| • Links (elastisch-plastische Verbindungen)                            |                           |
| • externe Kräfte (Gravitation, Kraftfelder, Reibung, Viskosität, usw.) | <b>Randbedingungen</b>    |
| • automatische Kollisionskontrolle                                     |                           |
| • Kontrollsysteme (Steuerungen)  |                           |
| • Datenaufzeichnung  | <b>Auswertungen</b>       |
| • grafische Darstellungen  |                           |
| • Filme  |                           |

Wir wollen im Folgenden die wichtigsten sonar-Bestandteile kurz zusammenfassen und dabei bei jedem Baustein erwähnen, wie genau dieser mit der Realität übereinstimmt und welche Kompromisse ggf. gemacht wurden um diesen in die Simulationsumgebung zu überführen.

## Bestandteile eines sonar-Modells

| Objekt              | Beschreibung  | Abstraktionen   | Bemerkungen   |
|---------------------|---|---|---|
| Rohdaten            | Rohdaten sind reine Zeichendaten, sei es weil sie als solche gezeichnet oder aus externen Quellen importiert wurden. Rohdaten bilden so gesehen eine Vorstufe bei der Erzeugung von Objekten, speziell von Prismen aller Arten. Rohdaten nehmen an den Simulationen nicht teil, sind aber in den Ansichten auch während den Simulationen sichtbar. Man kann sie in diesem Sinne auch für dekorative oder für Mess-Zwecke verwenden (Referenzlinien, Referenzpositionen).  | reine Zeichendaten  | Punkt, Strecke, Kreisbogen, Kreis, Quadstrip, Polygon, Konturen   |
| Primitivkörper      | Primitivkörper sind die einfachsten physikalischen Objekte. Es sind die Grundbausteine (Atome), aus denen komplexere Objekte aufgebaut werden. Primitivkörper können nicht weiter zerlegt oder unterteilt werden. Sie können aber beliebig klein gemacht werden. Primitivkörper sind Starrkörper mit einer elastischen Oberfläche. Primitivkörper nehmen auch allein und ohne zusätzliche Verbindungen vollumfänglich an den Simulationen teil.   | Primitivkörper sind nicht deformierbar  | Kugel, Zylinder, Quader, Kegel, Kegelstumpf, Rohr, Rohrsegment, Torus, Torussegment, Prismen  |
| Objekt              | Ein Objekt setzt sich aus mehreren Primitivkörpern zusammen die zusammen eine komplexere Form bilden. Alle Primitivkörper eines Objektes werden entweder manuell oder automatisch durch Funktionen untereinander verlinkt und sind damit elastisch-plastisch verbunden.   | Ein Objekt mit einer hohen Auflösung an Primitivkörpern kommt der Wirklichkeit sehr nahe. | Die Auflösung der inneren Elastizität von Objekten wird von der Grösse der einzelnen Primitivkörper bestimmt  |
| Fixpunkt            | Ein Fixpunkt ist ein fest verankerter raumfester Befestigungs-punkt. Mit Links können an ihm Primitivkörper und alle Arten von Objekten befestigt werden. Fixpunkte sind feste Stützpunkte oder Ankerpunkte welche eine Verbindung zur Aussenwelt herstellen. Ohne Abstützung wäre ein Modell ständig in der Schweben. Eine Befestigung an einem Fixpunkt ist dank dem Link eine elastische Verbindung, soweit die Elastizität des betreffenden Links es zulässt.   | Ein Fixpunkt hat keine Ausdehnung (geometrischer Punkt)                                   | Alternativ zu Fixpunkten können auch raumfest verankerte Objekte (no simulation member) diese Funktion übernehmen   |
| Ebene               | Eine Modell-Ebene ist auch geometrisch gesehen eine Ebene. Eine Ebene hat keine Dicke, ist raumfest verankert und hat unendliche Ausdehnungen. Eine Ebene wird mit einem dreidimensionalen Vektor definiert, welcher mit seinem Fusspunkt senkrecht auf der Ebene steht. Eine Ebene wird als unbewegliche Wand benutzt und kann ein Modell in eine bestimmte Raumrichtung begrenzen. Man benutzt Ebenen auch um sog. Halbmodelle herzustellen.  | idealisierte geometrische Ebene mit unendlichen Ausdehnungen                              | Die Interaktionen finden an Ebenen genauso statt wie an der Oberfläche eines raumfest verankerten Quaders   |
| physikalische Feder | Eine physikalische Feder kann man sich vorstellen als eine reale Feder welche man in mehrere Hundert Stückchen zerschneidet und diese anschliessend wieder mit Links verbindet. Zur automatischen Erzeugung von unterschiedlichen Federn gibt es im Basispaket verschiedene Dialoge. Physikalische Federn können auch sehr realistisch an ihren Objekten befestigt oder angehängt werden.   | Eine physikalische Feder kommt der Realität in jeder Beziehung sehr nahe                  | Zugfeder, Druckfeder, Blattfeder, Torsionsfeder<br>physikal. Federn simulieren auch das innere dynamische Schwingungsverhalten des Federkörpers                           |
| mathematische Feder | Mathematische Federn haben keinen physikalischen Körper und sind rein mathematische Gebilde. Sie existieren nur als Kraftfunktion zwischen den Objekten an denen sie befestigt werden. Die Darstellung der math. Federn im Modell geschieht über sog. Drahtmodelle, um sie von den physikalischen Federn zu unterscheiden. Rein zeichnerisch wird aber auch die math. Feder ständig den sich verändernden Längenverhältnissen angepasst. Das virtuelle Federbild (Drahtmodell) bewegt sich kontinuierlich im Raum mit den angedockten Objekten. | rein mathematisches Gebilde   | Eine mathematische Feder hat keine inneren Schwingungen und keine seitlichen Auslenkungen. Die Kraft wird allein aus der Distanz der beiden Befestigungspunkte berechnet. |

TABLE 1.

| Objekt                         | Beschreibung  | Abstraktionen   | Bemerkungen   |
|--------------------------------|---|---|---|
| mathematischer Aktuator        | Ein math. Aktuator ist ein elektrisches Stellglied, ein Pneumatik Zylinder, ein Hydraulikzylinder, usw. was immer sie wollen. Das Aktuator Modell legt nicht fest mit welchem Medium der Zylinder betrieben wird. Der Aktuator hat keinen inneren Aufbau. sondern besteht im Prinzip nur aus zwei Teilen, einem Kolben und einem Zylinder. Beide an ihrem Ende mit einem schematischen Befestigungspunkt. Dem Aktuator wird mit einem Dialog eine innere math. Funktion zugeordnet. Diese Funktion steuert die Länge des Aktuators.                 | math. Gebilde. Der Aktuator hat keine Masse und keine Trägheit.                             | Ein physikalischer Aktuator könnte man herstellen, indem man diesen aus Primitiv-körpern aufbaut und ihm einen math. Aktuator als innere Funktion einverleibt.  |
| mathematischer Dämpfer         | Der math. Dämpfer kann man bezüglich der Abstraktion mit dem math. Aktuator vergleichen. Wie dieser besteht er nur aus zwei Teilen und einer math. Funktion welche seine innere Funktion festlegt.  | math. Gebilde. Der Dämpfer hat keine Masse und keine Trägheit                               | analog wie der math. Aktuator   |
| Draht                          | In sonar ist ein Draht ein biegbares Objekt, zusammenge-setzt aus einer Kette von Primitivs welche untereinander mit geeigneten Link-Kombinationen verbunden sind. Drähte dieser Art bewegen sich unter Krafteinfluss in allen Belastungsrichtungen (Zug, Biegung, Torsion) entsprechend ihren Vorbildern.  | Drähte kommen der Realität sehr nahe  | Unter schlagartigen Krafteinflüssen werden auch die induzierten Schwingungen nachgebildet   |
| Seil (Blockmodell)             | Wir bezeichnen als Blockmodell eine vereinfachte Seilkonstruktion, welche aus einer Kette einheitlicher starrer Zylinder aufgebaut ist, ganz ähnlich wie biegsame Drähte. Diese Seile bewegen sich bezüglich allen Belastungsarten wie ihre Vorbilder, allerdings werden dabei keine Erkenntnisse über die unterschiedlichen inneren Belastungen der einzelnen Drähte gewonnen.   | Der Litzen- oder Seilquerschnitt wird durch einen vollen Querschnitt (Zylinder) ange-nähert | geeignet für Grossanlagen wie Seilbahnen oder für den Bereich Hebe- und Fördertechnik   |
| physikalische Litzen und Seile | physikalische Seile werden wie ihre Vorbilder aus einzelnen Litzen und die wiederum aus einzelnen Drähten aufgebaut. In einem eigenen Modul stehen dem Benutzer entsprechende Funktionen zur Verfügung, um beliebige Litzen und Seile nach den entsprechenden Herstellungsvorschriften zu modellieren.  | es gibt keine Abstraktionen. Diese Litzen und Seile kommen der Wirklichkeit sehr nahe       | Litzen und Seile interagieren und deformieren sich unter Last in den Nuten von Seilrädern absolut realistisch   |
| Link                           | Links sind punktförmige Verbindungen zwischen Primitivkörpern, Objekten, Fixpunkten und weiteren speziellen Objekten (Aktuatoren, Dämpfer, usw.). Ein Link ist ein Feder-Dämpfungselement mit einer Auslenkung von Null im unbelasteten Zustand. Ein einzelner Link ist funktionell ein elastisches Kugelgelenk. Zwei Links bilden eine elastische Drehachse und drei oder mehr Links eine feste, aber elastische Verbindung zwischen zwei Objekten. Die Lagerkräfte und die Reibung im Lager sind unter Kontrolle des Programms und des Benutzers. | Eine Drehachse bestehend aus Achse und Lager wird auf eine Linie reduziert.                 | Ein Drehlager kann grundsätzlich auch aus einer zylindrischen Achse und einer Lagerbüchse modelliert werden. Allerdings bringt dieses Vorgehen meistens keinen Gewinn, denn alle Lagerparameter können auch mit Links ins Spiel gebracht werden |

TABLE 1.

Ein sonar-Modell setzt sich in der Regel aus mehreren Objekten zusammen. Objekte kann man sich letztlich als Maschinenteile vorstellen. So gibt es in einem sonar-Modell im Allgemeinen Maschinenteile welche der Benutzer von Grund auf nach seinen Vorstellungen modelliert. Es gibt aber auch sog. Standard-Maschinenteile, die der Benutzer z.B. mit Hilfe eines Dialogs direkt abrufen, dimensionieren bzw. parametrisieren und positionieren kann (z.B. eine Zugfeder).

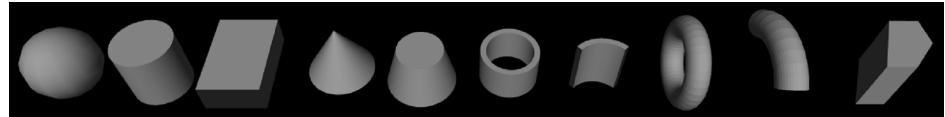
#### Aufbau eines Modells

Ein Objekt wiederum ist im Allgemeinen eine Kombination aus mehreren Grundkörpern, sog. Primitivkörpern. Eine beliebige Anzahl von Primitivköpfen werden in sonar zu komplexeren Strukturen vereinigt und bilden so ein Objekt bzw. ein Maschinenteil.

#### Aufbau eines Objektes

## Primitivkörper

Das Basismodul von sonar-Lab besitzt vorerst unter anderen die folgenden Primitivkörper:



**FIGURE 5.** eine Auswahl von Primitivkörpern (nicht vollständig)

## Modellierung

Die Erzeugung eines Modells geschieht zur Hauptsache in den Programmen der sog sonar-LAB Umgebung selbst. Nicht selten werden aber zusätzliche Methoden verwendet. So kann z.B. für die Erzeugung einer Freiformfläche bzw. Gitterstruktur eine EXCEL-Tabelle eingelesen und die Punkte übernommen werden. Es ist auch möglich 2-dimensionale Konturen einzulesen um diese anschliessend in sonar-LAB zu rotieren oder extrudieren und auf diesem Weg einen Rotationskörper oder einen prismatischen Körper zu erzeugen. sonar-LAB versucht dort wo es sinnvoll erscheint, solche Schnittstellen zu externen Programmen herzustellen.

## Modellierungsphilosophie

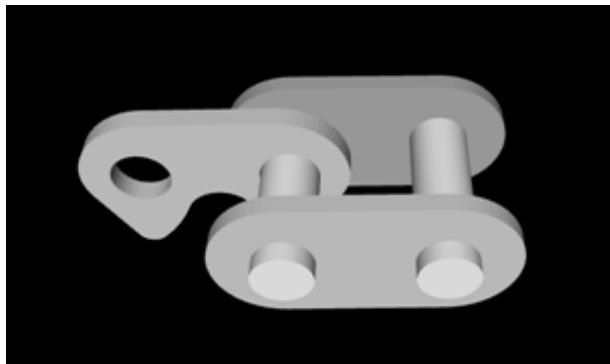
sonar-LAB ist kein CAD-Programm. sonar-LAB ist auch kein ausgewachsenes 'solid-modeling'-Programm. sonar-LAB versucht grundsätzlich immer, aus den vorhandenen elementaren Bausteinen (Primitivkörper) ein komplexeres Objekt zusammenzustellen oder besser gesagt 'zusammen zu addieren'. Wie in einem CAD-Programm kann man sich diesen Prozess als Boolesche Operation (Vereinigung) vorstellen. Es gibt in sonar allerdings nur die Operation 'vereinigen' bzw. 'addieren'. Es ist also bis auf weiteres nicht generell möglich aus einem Primitivkörper oder einem Objekt nachträglich ein Loch herauszuschneiden indem ein Zylinder vom betr. Objekt subtrahiert wird. Trotzdem lässt sich die hinter sonar steckende Modellier-Philosophie am ehesten mit der sog. CSG-Methode im CAD Bereich vergleichen (CSG: constructive solid geometry). Dieser Methode liegt die Idee zugrunde, alle komplexen Objekte durch boolesche Operationen von Primitivkörpern zu erzeugen.

## Bohrungen

Bohrungen in Maschinenteilen dienen in der Praxis oft dem Zweck der Lagerung (Achse in Bohrung) oder als Durchgangsloch für eine Schraube oder einen Bolzen. In unserer Simulationsssoftware ist für diesen Zweck nicht zwingend eine Bohrung notwendig, weil diese Aufgabe in der Regel anders gelöst wird. Eine drehende Achse kann in einer Bohrung auch durch das Setzen von zwei Verbindungspunkten zwischen den beiden Körpern befestigt werden. Das Material muss für diesen Zweck auch nicht wirklich aus der Bohrung entfernt werden, damit die Achse durchgeschoben werden kann. Achse und Bohrung bzw. die zwei betreffenden Körper durchdringen sich einfach und sind an zwei Referenzpunkten längs der Drehachse miteinander verbunden. Im Übrigen wird beim Herstellen dieser Punktverbindungen autom. eine Regel gesetzt, welche sagt, dass die beiden Objekte nicht miteinander interagieren sollen, obwohl ihre Oberflächen miteinander kollidieren.

Wo eine Bohrung für die Funktion eines Mechanismus wirklich notwendig ist, weil man sich für die Details zwischen Achse und Bohrungsoberfläche interessiert, muss man sich anders behelfen. Ist wirklich ein Durchgangsloch gefragt, so kann man am betreffenden Ort z.B. einen Primitivkörper des Typs 'Rohr' einbauen. Im weiteren sind einzelne Primitivkörper, wie z.B. das Prisma, in der Lage, wirkliche Löcher zu haben welche beim Erzeugungsprozess als negative Konturen übernommen und interpretiert werden. Auf diese Weise können z.B. zwei Laschen mit je einer Bohrung mit einem zylindrischen Bolzen durch diese Bohrungen gekoppelt werden. Die beiden Laschen würden in diesem Fall tatsächlich durch die Oberflächen von Bolzen und Bohrung zusammengehalten indem die Kollision bzw. Interaktion zwischen diesen Oberflächen explizit simuliert würde. Auch ein allenfalls vorhandenes Spiel zwischen Achse und Bohrung würde dann

dynamisch realistisch nachgebildet. Mehr dazu später im Kapitel 'Extrudierte Linien-Bogen Konturen'.



**FIGURE 6.** Bolzen verbinden die Laschen einer Maschinenkette

## Elastische Verbindungen

Wenn sich ein Modell wie beschrieben aus mehreren Objekten und diese sich wiederum aus mehreren Primitivkörpern aufbauen, dann stellt sich als nächstes die Frage, was diese Objekte und Primitivkörper letztlich zusammenhält. Es liegt in der Absicht des Benutzers, dass sich Primitivkörper, die zusammen ein Objekt bilden, sich auch zusammen und gleichförmig durch den Raum bewegen. Die Primitivkörper eines Objektes sollen, was ihre äussere Form betrifft, zusammen eine Einheit bilden. Das Objekt wurde schliesslich nur deshalb in Primitivkörper aufgegliedert, um auf diesem Weg leichter eine komplexe Struktur zu konstruieren. Aber die Primitivkörper bilden zusammen ein einziges Objekt.

Primitivkörper,  
Objekte

In der sonar Software sind Primitivkörper starre Objekte welche untereinander elastisch zu grösseren Strukturen (Objekten) verbunden werden. Objekte, welche sich aus mehreren Primitivkörpern zusammensetzen, haben demzufolge eine gewisse innere Elastizität. Die elastischen Verbindungen welche zwei Primitivkörper verbinden nennen wir Links. Links kann man sich als Federn vorstellen welche in ihrer Ruhelage die Länge Null haben. Werden zwei Primitivkörper die durch einen Link verbunden sind auseinander gezogen, dann beginnt eine Federkraft zu wirken, welche diese Körper wieder zusammenziehen will. Im Allgemeinen werden Primitivkörper die fest miteinander verbunden sein sollen, mit mehreren Links an unterschiedlichen Positionen verbunden. Stellen Sie sich Links als punktförmige elastische 'Schweisspunkte' vor, welche die Summe aller Primitivkörper eines Objektes zusammenhalten. Vielleicht ist es aber auch besser sich diese Verbindungspunkte als Kugelgelenke vorzustellen, denn werden zwischen zwei Primitivkörpern nur zwei Links gesetzt, dann bilden diese beiden Links eine Drehachse zwischen den beiden Primitivkörpern. Dieser Sachverhalt wird in der Tat oft benutzt, um zwei ganze Objekte drehbar miteinander zu verbinden.

Links

Mit der Wahl der Anzahl Primitivkörper eines Objektes und der Grösse der Federkonstanten der Links kann die Elastizität bzw. die Steifigkeit und die Beweglichkeit bzw. Verformbarkeit eines Objektes gesteuert werden. Manchmal kümmert sich die sonar-Software automatisch um diese Details, manchmal greift der Benutzer selbst ein. Wird über ein Makro oder eine Menu-Funktion eine Zugfeder erzeugt, dann berechnet und setzt das Programm die Links und ihre Konstanten automatisch. Auch eine Zugfeder ist ein Objekt, welches sich aus einer grösseren Anzahl von Primitivkörpern zusammensetzt welche untereinander elastisch verbunden sind. Die Links werden automatisch so dimensioniert, dass die Feder als Ganzes die physikalischen Eigenschaften (Federkonstante) besitzt, die der Benutzer als Ziel vorgibt oder die auf Grund eines Federdatenblattes erwartet werden.

Steifigkeit der Objekte

## Möglichkeiten der Modellerzeugung

---

### Vorteile von Makros

Es ist ein Ergebnis aus der praktischen Anwendung der Software, dass komplexere Modelle vorteilhaft immer in Verbindung mit Makros erzeugt werden. Sei es, indem ein Modell ausschliesslich und vollständig durch das Schreiben eines Makro oder durch die Übernahme und Anpassung eines bestehenden Makro aus einer Makro-Bibliothek erzeugt wird. Besonders dann, wenn ein Modell viele sich wiederholende Teile besitzt, ist die Verwendung von Makros unverzichtbar. Makros sind auch eine hervorragende Dokumentation eines Modells, weil in ihnen nicht nur die geometrischen, sondern auch die physikalischen Eigenschaften festgehalten sind.

### Gemischte Verwendung von Makros mit klassischen Zeichenmethoden (not yet implemented)

Makros können auch zusammen mit klassischen Zeichenmethoden verwendet werden, indem z.B. ein paar Teile zuerst grafisch interaktiv am Bildschirm gezeichnet und anschliessend mit einem entsprechenden Menu-Befehl in einen Makrotext umgewandelt werden, um diesen Text schliesslich in einem zweiten Makro weiter zu verwenden. Oder umgekehrt können ein paar Objekte zuerst durch die Ausführung eines Makro auf den Bildschirm gebracht und anschliessend manuell, d.h. grafisch-interaktiv verändert werden. Zum Schluss könnte man den neuen Makrotext per Copy & Paste wieder weiterverwenden.

### Makrosprache lernen

Die Rückführung von gezeichneten Objekten oder von ganzen Modellen in den Makrotext ist eine hervorragende Möglichkeit zum Erlernen der Makrosprache. Es beantwortet die Frage: "wie würde man das abgebildete Modell mit Hilfe eines Makro erzeugen?".

### Rückführung von Objektgruppen in ihre ursprünglichen Erzeugungsfunktionen (not yet implemented)

Bei der Rückführung eines Modells in einen Makrotext ist das Programm im Übrigen so intelligent, dass es weiss, wie eine Gruppe von Objekten ursprünglich erzeugt wurde. Hat der Benutzer z.B. in einem Modell mit einer entsprechenden Menu-Funktion oder einem Makro ein U-Profil erzeugt, dann sind alle Teile des U-Profil entsprechenend gekennzeichnet. Ein U-Profil besteht so gesehen aus einer grossen Anzahl von Quadern (Cuboids). Überführt der Benutzer dieses Modell später wieder in einen Makrotext, dann wird das U-Profil im zu erzeugenden Makrotext nicht in eine grosse Anzahl von einzelnen Quadern zerlegt. Die einzelnen Quader werden im Makro wieder in die entsprechende Funktion zur Erzeugung eines U-Profil zusammengefasst.

### Halbfabrikate, Normteile

In der Umgebung des Basismoduls von sonar-Lab stehen ein paar Dialoge zur Erzeugung von Standardobjekten zur Verfügung. In den branchenspezifischen Modulen gibt es eine Vielzahl mehr. So kann z.B. eine Zugfeder mit Hilfe eines Dialogs oder der Makro-Entsprechung dieser Funktion erzeugt werden. Auf diesem Weg können einem Modell direkt Teile hinzugefügt werden. Makros zur Erzeugung von Normteilen sind dabei oft parametrisierbar.

### Objekt Library

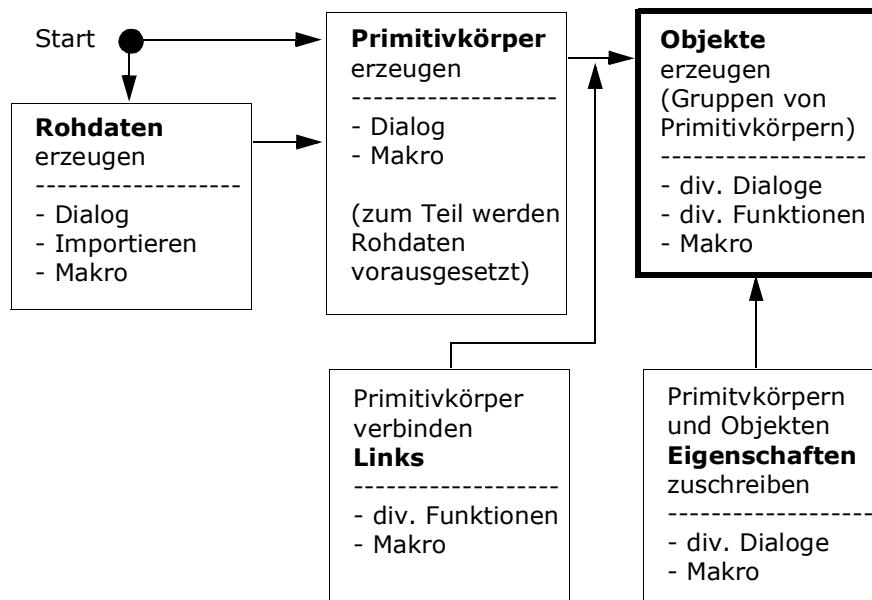
Jeder Benutzer kann sich seine eigene Teile-Bibliothek aufbauen, in denen er Objekte oder Objektgruppen, die er in Zukunft wiederverwenden will, sammelt. Mit Hilfe der sog. Merge-Funktion können solche Teile direkt in ein Modell hineinkopiert werden. Statt die eigentlichen Objekte kann ein Benutzer natürlich auch ihre Makro-Entsprechungen sammeln. In der Regel ist diese Variante noch flexibler.

### Die Verwendung von externen Dateien (TEXT EXCEL)

In der Regel können externe Dateien sowohl in einem Makro als auch im entspr. Menu-Befehl für die grafisch-interaktive Bearbeitung gleichermaßen verwendet werden. Will der Benutzer z.B. zur Erzeugung einer Freiformfläche in Form einer zweidimensionalen Gitterstruktur eine externe Datei verwenden, dann stehen ihm die folgenden Möglichkeiten offen:

- der Benutzer öffnet einen Menu-Befehl zur Erzeugung eines gewissen Objektes und kann im betreffenden Dialog über eine Taste das gewünschte File auswählen und in die Erzeugung einbinden.
- der Benutzer verwendet in einem Makro einen Befehl zur Erzeugung eines gewissen Objektes und kann im Makrobefehl auch gleich die file-Referenz für das zu verwendende File eingeben.

Die folgende Abbildung soll einen ersten Überblick über die Möglichkeiten zur Erzeugung eines sonar-Modells geben. Die Abbildung erhebt noch keinen Anspruch auf Vollständigkeit.



**FIGURE 7.** Modellerzeugung in sonar

## Bestandteile eines sonar-Modells

Nebst den besprochenen Grundkörpern (Primitives), den daraus aufgebauten Objekten und den verbindenden Links gibt es im Grundbaukasten von sonar noch weitere Bausteine für ein Simulationsmodell. Dazu gehören

Im Basispaket von sonar sind eine Anzahl von Standard Objekten mit zugehörigen Erzeugungsmöglichkeiten (Dialoge) eingebaut. Dazu gehören Zug- und Druckfedern, Aktuatoren und Dämpfer. Wichtig werden dann aber die verschiedenen Erweiterungs-Tools welche dem Benutzer branchenspezifisch eine grosse Auswahl von weiteren Objekten in die Hand geben. Jeder Benutzer kann sich auch direkt mit sonar oder mit sonar-Macros zusätzlich seine eigene Bibliothek von Standardobjekten schaffen.

### Standard Objekte

sonar besitzt, wie bereits erwähnt, eine Makrosprache (sonar script). Mit dieser Sprache lassen sich komplette Modelle definieren und in dieser Form auch speichern. Darüber hinaus wird 'sonar script' aber auch als Steuerungs- bzw. Kontrollsystem-Sprache während der eigentlichen Simulation eingesetzt. Der geometrische und physikalische Zustand eines jeden Objektes kann während einer Simulation kontinuierlich von einem Kontrollsystem abgefragt werden. Mit diesen Informationen kann das Kontrollsystem mit 'sonar script' weitergehende Berechnungen durchführen und anschliessend gewisse Resultatwerte in das Simulationsmodell zurückspeisen um damit den Zustand oder eine physikalische Größe eines Objektes zu ändern. So kann z.B. die Hublänge eines Aktuators in einem Modell auf Grund beliebiger Ereignisse und Zustände im Modell gesteuert werden. Mit diesen Möglichkeiten ist es letztlich möglich eine virtuelle Maschine mit einer virtuellen Steuerung zusammen zu simulieren. 'sonar script' kommt in dieser Form allerdings erst im eigentlichen Simulationsprogramm sonar-SIM zum Zug.

### Steuerungen

Das Basismodul enthält Basisfelder wie Gravitationsfelder und lässt die Definition von weiteren benutzerspezifischen Feldern zu. Kraftfelder sind letztlich einfach orts- oder zeitabhängige,

### externe Kräfte und Kraftfelder

externe Kräfte welchen alle Objekte ausgesetzt sein können, welche sich im fraglichen Raum bewegen. Mittels eines bestimmten Dialogs kann aber auch eine externe Kraft mit einem gegebenen Verlauf (Punktkurve, math. Funktion) auf ein einzelnes Objekt definiert werden.

## **Funktionsumfang des Basismoduls**

---

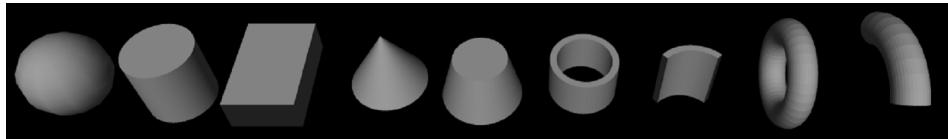
Für die Erzeugung und Verwaltung eines Modells und seinen Bestandteilen stehen, um einen ersten unvollständigen Überblick zu geben, folgende Funktionen zur Verfügung:

- Dialoge zur Erzeugung sämtlicher Primitivkörper. Entsprechende Makrobefehle für den gleichen Zweck.
- Funktionen für die Erzeugung von Standard-Objekten wie Aktuator, Dämpfer, Antriebe, math. oder physikal. Federn,
- Funktionen für die Verwaltung, Änderung von Primitivkörpern und Objekten (Kopieren, Duplizieren, Einsetzen, Löschen, usw.)
- Funktionen für das interaktive und nummerische Bewegen von Objekten (Translation, Rotation, Skalieren, Spiegeln)
- Funktionen für das Einstellen der Kollisionseigenschaften zwischen den Objekten
- Funktionen für die Verwaltung von Objektgruppen
- Funktionen für das Einstellen / Ändern von Materialeigenschaften (Masse, Emodul, Streckgrenze, Bruchdehnung, Reibung, Materialmodell, usw.)
- Funktionen für das Erzeugen und Justieren von Links
- verschiedenen Arten von Kraft-Feldern und Kraftfunktionen auf Objekte
- Funktionen für den Import von 2D-Daten, Kurven und Konturen (DXF, EXCEL, sonar-2D, Text, usw.) -> Extrude-Funktionen, Freiformflächen
- Kontrollsysten-Sprache (sonar script)
- Funktionen für die Erzeugung von Dokumentationsmaterial zum Modell.

# Die Erzeugung von einfachen Primitivkörpern

---

Als einfache Primitivkörper bezeichnen wir die folgenden Formen:



Sphere, Cylinder, Cuboid, Cone, Frustum, Tube, Tube Segment, Torus, Torus Segment

**FIGURE 8.** Primitivkörper

Diese Objekte werden deshalb als 'einfach' bezeichnet, weil sie mit einem einzigen Dialog und ohne weitere Input Daten erzeugt werden können. Andere Primitivkörper wie Prismen oder extrudierte Linien-Bogen-Konturen basieren hingegen auf vorab hergestellten Elementen wie Polygone oder Gruppen von Linien und Bogen.

## Erzeugungsmethoden

---

Primitivkörper lassen sich generell auf verschiedenen Wegen erzeugen. Dazu gehören:

- durch Anwendung eines Makrobefehls
- mit Hilfe eines Dialogs
- grafisch interaktiv direkt am Bildschirm

indirekte Methoden sind:

- durch Kopieren und Verändern eines existierenden Körpers
- durch Funktionen welche ihrerseits Primitivkörper erzeugen

## Erzeugung eines Primitivkörpers mit Makros

---

Ein Makro ist eine Liste von Anweisungen in der sog. 'sonar script' -Sprache. Ein Makro wird als Textfile geschrieben und gespeichert oder als Anweisung direkt im sog. Makro-Tool am Bildschirm eingegeben und ausgeführt. Im Allgemeinen wird ein Primitivkörper mit all seinen Eigenschaften nicht in einem Schritt mit einem einzigen Makro erzeugt. Der Primitivkörper wird zuerst bezüglich seinen äusseren Dimensionen und ev. auch noch in Bezug auf seine Position im Raum definiert. Damit ist der Primitivkörper vorerst in seiner äusseren Gestalt vorhanden. Anschliessend spezifiziert man alle weiteren Eigenschaften des Primitivkörpers mit Hilfe seiner Referenznummer und weiteren Makrobefehlen.

```
CREATE OBJECT (07, CUBOID, 30, -20, 12, 32, -16, 15)
ROTATE OBJECT (07, 31, -18, 13.5, 0, 0, 0.7854)
SET PROPERTY (07, COLOR_RGB, 200, 200, 200)
SET PROPERTY (07, DENSITY, 7.8)
```

### Beispiel

Die erste Linie erzeugt die Geometrie eines Quaders im Raum. Die Koordinaten bestimmen zwei räumlich gegenüberliegende Ecken des Quaders. Der erste Eckpunkt befindet sich in (30, -20, 12), der zweite in (32, -16, 15). Der Schwerpunkt des Quaders befindet sich folglich in (31, -18, 13.5). Der Quader hat die Kantenlängen  $dx = 2$ ,  $dy = 4$  und  $dz = 3$ . Die Kanten verlaufen soweit parallel zu den Koordinatenachsen (x, y, z) denn es wurde noch keine Drehung des Körpers definiert. Diese wird schliesslich mit der zweiten Anweisung umgesetzt. Der Körper wird um  $45^\circ$

um seinen Schwerpunkt (31, -18, 13.5) um eine Achse parallel zur z-Achse gedreht.

Die dritte Anweisung weist dem Quader eine graue Farbe zu. Die RGB-Farbkomponenten (Rot, Grün, Blau) müssen im Intervall [0..255] liegen. Farben mit gleichen Farbkomponenten R=G=B sind alle grau. Dabei bedeutet [0, 0, 0] = schwarz und [255, 255, 255] = weiss. Unser Quader bekommt mit der obigen Anweisung folglich die Farbe hellgrau.

Die letzte der vier Linien gibt dem Quader die Dichte von Stahl [7.8 g/cm<sup>3</sup>]. Da das Volumen des Quaders durch die Definition der Geometrie bereits bestimmt ist, setzt das Programm mit der letzten Anweisung automatisch auch die Masse des Quaders auf den Wert: Masse = Dichte \* Volumen. Im Weiteren berechnet das Programm ebenso automatisch die Hauptträgheitsmomente des Quaders neu. Beachten Sie, dass die in Makros verwendeten Zahlenwerte (bis auf weiteres) immer im [cm g µs]-System angegeben werden müssen.

Die erste Makro-Linie, welche den Quader erzeugt, speichert den Index des neuen Primitivkörpers intern als 'o7' ab. Anstelle von '7' kann jede andere ganze Zahl verwendet werden. Wenn wir den Quader erzeugen, dann geben wir ihm folglich gleichsam auch einen 'Namen' bzw. eine Referenzbezeichnung. Deshalb können nachfolgende Makro-Linien auf diesen Namen zurückgreifen und zum Beispiel sagen: Der Primitivkörper, welcher 'o7' heisst, soll hellgrau werden.

## Fazit

Sie brauchen vor dieser Makrosprache keine Berührungsängste zu haben. Es handelt sich um eine einfache, für jedermann verständliche Befehlssprache die weitgehend selbsterklärend ist.

## Erzeugung eines 'Primitives' mit Dialogen

Auch die Definition eines Primitives unter Zuhilfenahme von Dialogen wird in der Regel in mehreren Schritten durchgeführt. Würde man alle Einstellungsmöglichkeiten eines Objektes in einen einzigen Dialog verpacken, dann würde dieser sehr komplex und unübersichtlich. Aus diesem Grund wurden diese Aufgaben auf mehrere Dialoge verteilt, genauso, wie auch die Erzeugung per Makro in mehreren Anweisungsschritten erfolgt, wie auf der vorangehenden Seite als Beispiel dargestellt ist. Auf diesem Weg wird die Definition eines Objektes in einem gewissen Sinne 'modular'. Die beiden letzten der drei nachfolgenden Dialoge lassen sich so gesehen auch nachträglich zum Editieren bzw. Ändern eines Objektes einsetzen.

1. Mit einem ersten Dialog erzeugt der Benutzer die Geometrie d.h. die äussere Gestalt des Objektes. Nach diesem Schritt ist das Objekt bereits am Bildschirm bzw. im Modell sichtbar.

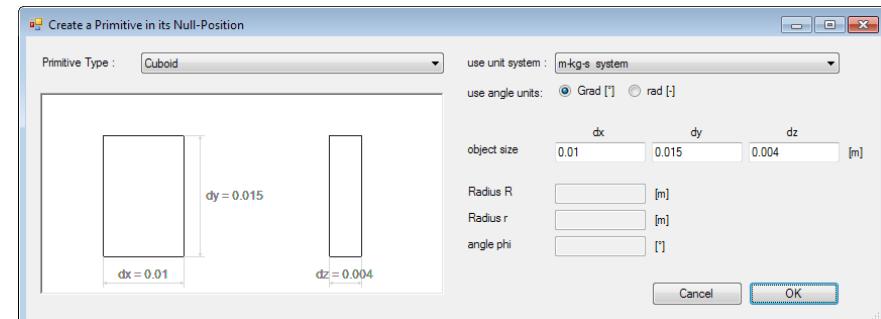
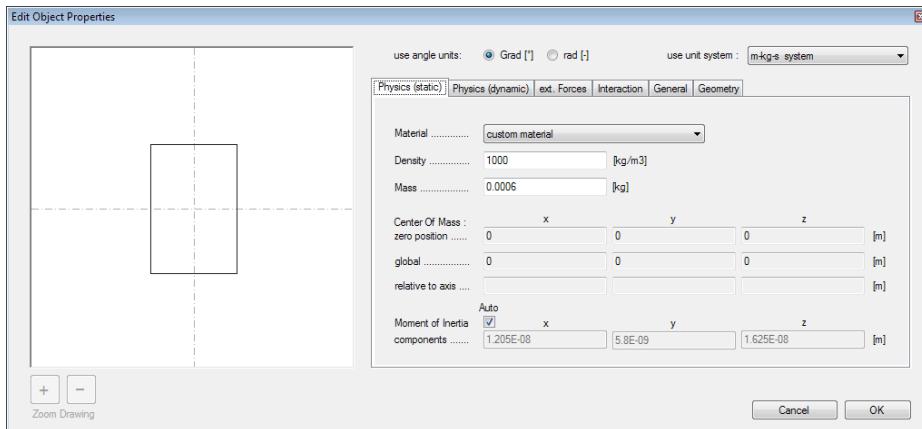


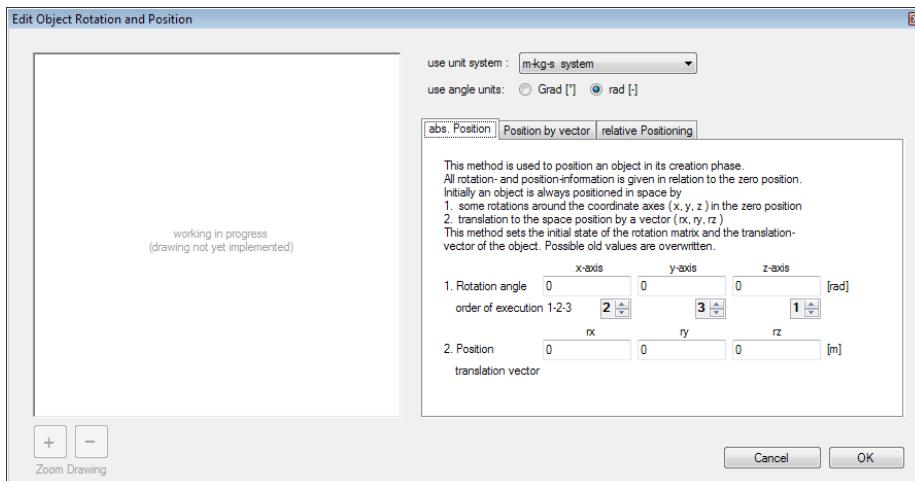
FIGURE 9. Erzeugung eines Objektes: Menu / Create / New Primitive

- ein zweiter Dialog spezifiziert anschliessend die individuellen physikalischen Eigenschaften, das Kollisionsverhalten und allgemeine Merkmale des Objektes, wie z.B. die Farbe, usw.



**FIGURE 10.** Objekt Eigenschaften: Menu / Edit / Edit Object/Properties...

- Oft folgen weitere Anweisungen zur Drehung und Positionierung des neuen Objektes im Raum. Dazu steht ein weiterer Dialog zur Verfügung.



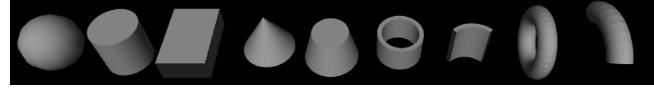
**FIGURE 11.** Rotation und Position des Objektes: Menu / Edit / EditObject / Rotation and Position...

Wir erzeugen ein Objekt auf diese Weise also in drei Schritten

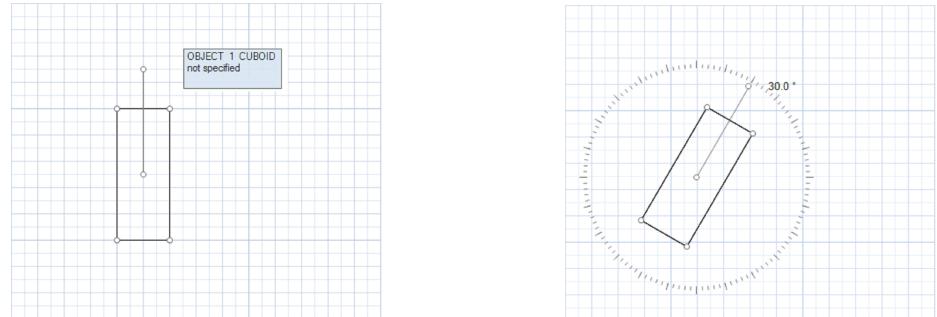
- Geometrie des Objektes erzeugen
- Die Eigenschaften des Objektes festlegen
- Das Objekt räumlich drehen und positionieren

(der zweite und der dritte Schritt könnten auch in umgekehrter Reihenfolge durchgeführt werden)

## Grafisch-interaktive Erzeugung eines Primitives



**FIGURE 12.** Toolbar 'New Primitive'



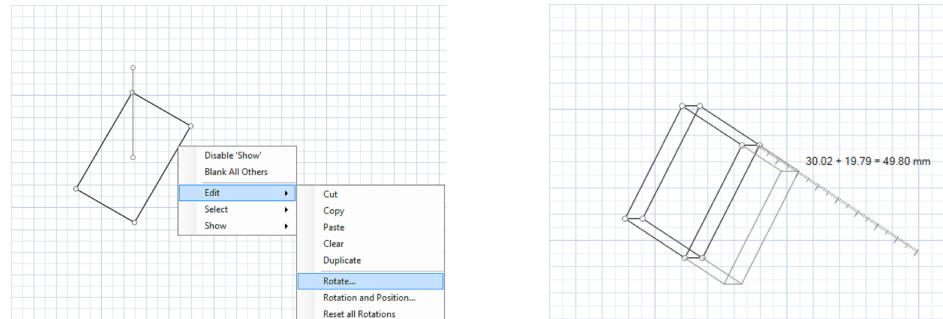
**FIGURE 13.** Nach der Wahl des Quadersymbols in der Symbolleiste definieren wir einen neuen Quader. Im Quader ist eine Strecke eingezeichnet welche für Drehungen benutzt werden kann.

**FIGURE 14.** 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.



**FIGURE 15.** Drückt man die Maus über einem Eckpunkt, dann werden zwei mögliche Streckungsrichtungen eingeblendet.

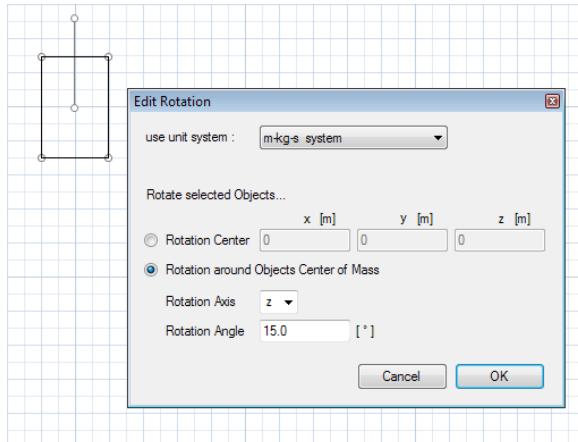
**FIGURE 16.** Entscheidet sich der Benutzer in diesem Beispiel für eine Streckung Richtung 4 Uhr, indem er die gedrückte Maus in die betreffende Richtung bewegt, dann werden ihm eine Skala und eine numerische Anzeige eingeblendet.



**FIGURE 17.** Mit der rechten Maustaste über der Berandung des Quaders wird das Kontextmenü aufgerufen. Der Befehl 'Rotate' öffnet den entspr. Dialog. Eine Rotation von 30° um die y-Achse ergibt die Quaderstellung in der Abbildung rechts.

**FIGURE 18.** Auch hier kann wieder eine Streckung durchgeführt werden wie abgebildet. Die Verschiebung des Quaders im Raum erfolgt wie üblich durch 'packen' am Rand des Quaders.

Falls Sie sich die Frage stellen, weshalb die Rotation um  $30^\circ$  am Ende der Bildserie mit einem Dialog und nicht auch grafisch interaktiv durchgeführt worden ist, dann dürfen wir Ihnen antworten: Sie haben recht, diese Rotation um die y-Achse hätte man tatsächlich auch in der Aufsicht grafisch interaktiv durchführen können. Wir haben hier in dieser Bildserie die beiden Methoden, ein Objekt zu drehen, aus folgendem Grund vermischt: Will der Benutzer ein Objekt, sagen wir um  $30.000^\circ$  drehen, dann kann er das sehr einfach grafisch interaktiv. Will er das Objekt aber um einen Winkel von  $31.137^\circ$  drehen, warum auch immer, dann funktioniert das grafisch interaktiv nicht mehr. In diesem Fall muss er auf den Rotations-Dialog, am einfachsten über das Kontext-Menu, zugreifen.



**FIGURE 19.** Eine Objekt-Rotation mit einem Dialog durchführen

Die letzte Bildreihe zeigt das grafisch interaktive Arbeiten am Beispiel eines Quaders. Verglichen mit den beiden anderen Erzeugungsmethoden wird hier die äussere Form und die Lage im Raum bestimmt. Der letzte Schritt, also die Festlegung der Eigenschaften des Quaders, geschieht hier genauso wie wenn das Objekt mit Dialogen erzeugt worden wäre. Man ruft dazu den folgenden Dialog auf:

Menu / Edit / Edit Object / Properties...

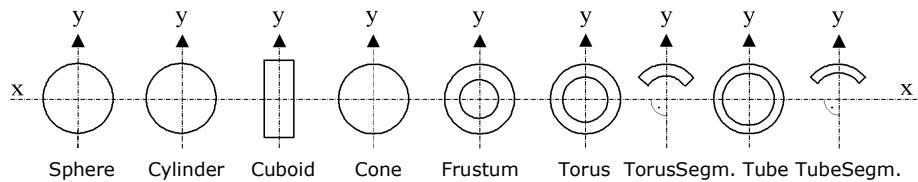
Die anderen Primitivkörper werden bezüglich der grafisch-interaktiven Erzeugungsmethode sinngemäss gleich behandelt. Sie werden mit einer intuitiven Herangehensweise sicherlich das richtige tun. Vielleicht ist es eine gute Idee, in einem leeren Dokument jeden Primitive-Typ einmal grafisch interaktiv zu erzeugen und seine Form und Lage auf alle möglichen Arten spielerisch zu verändern.

## Die Nulllage eines Objektes

Die Nulllage eines Objektes ist per Definition die Lage, in der ein Objekt programm-intern gespeichert ist. Jede Raumlage eines Objektes ist so gesehen eine Bewegungsmatrix des betreffenden Objektes relativ zu seiner Nulllage. Aus diesem Grund wird ein Objekt oft zuerst in seiner Nulllage erzeugt und anschliessend durch eine dreidimensionale Bewegung (Drehung + Translation) an seinen Bestimmungsort verschoben. Dieser Weg wird nicht zuletzt auch deshalb eingeschlagen, weil die Definition eines Objektes in seiner Nulllage einfacher ist, als wenn dieses schief im Raum steht. Einige Primitivkörper bieten auch Erzeugungsfunktionen an, welche bereits bei der Definition die Drehwinkel und Raumkoordinaten entgegennehmen. Wichtig ist in diesem Zusammenhang die Reihenfolge, mit der die Drehungen um die Koordinatenachsen (x,y,z) durchgeführt werden. Drehungen sind nicht kommutativ. Oft ist man vor die Situation gestellt, dass man ein Objekt mit ganz bestimmten Drehungen um gewisse Raumachsen in die endgültige Lage bringen möchte und dass die Reihenfolge dieser Drehungen nicht der vordefinierten Reihenfolge entspricht, wie sie in den entsprechenden Funktionen mit solchen Parametern vorgesehen ist. In diesem Fall erzeugt man das Objekt vorerst im Ursprung bzw. in seiner Nulllage

und dreht es anschliessend mit individuellen Drehungen um einzelne Koordinatenachsen in die Endlage. Es ist wichtig die Nulllage der Objekte zu kennen, denn jede relative Bewegung eines Objektes bezieht sich auf seine vorangehende Lage. Die erste Bewegung bezieht sich somit immer auf seine Nulllage.

Meistens erzeugt man ein neues Objekt in seiner Nulllage oder in einer Position nahe seiner Nulllage. Dies allein deshalb, weil die Koordinaten in dieser Form mehr Aussagekraft haben und leicht überprüft und geändert werden können. Die Nulllage eines Objektes ist in einem gewissen Sinn auch die räumliche Lage eines Objektes wie sie in einer technischen Werkzeichnung zur Darstellung kommt. Meistens liegen die Objekt-Abmessungen in der Nulllage auch in ihrer einfachsten natürlichen Form vor und müssen nicht umständlich aus Raumkoordinaten umgerechnet werden. Nachträgliche Änderungen werden auf diese Weise einfach und verständlich. Die anschliessende Drehung und Verschiebung in die endgültige Raumlage ist einfach.



**FIGURE 20.** Nulllage der Primitivkörper

## Weitere Abhängigkeiten

---

### Gruppierungen

Primitivkörper, welche mit Links zu einem Objekt vereinigt werden, können nicht mehr einzeln und unabhängig bewegt werden. Solche Primitivkörper haben ihre Eigenständigkeit zum grossen Teil aufgegeben und werden automatisch als Gruppe bewegt. Auch andere Primitivkörper-Eigenschaften gehen wahlweise oder notwendigerweise in die Gruppeneigenschaften über.

### Nachträgliche Änderungen

Wie ein Primitivkörper auch immer erzeugt wurde, seine Eigenschaften lassen sich alle nachträglich wieder verändern, indem man einen der Dialoge 'Edit Object...' aufruft. Darüber hinaus gibt es Funktionen welche ganze Gruppen von Primitivkörpern oder Objekten zusammen verändern.

# Prismatische Körper

Extrudierte Polygone nennt man auch Prismen. Wir unterscheiden dabei zwei Arten von Prismen:

- konvexe Prismen
- konkave Prismen

Die Bezeichnungen 'konvex' und 'konkav' beziehen sich auf die Querschnittsfläche welche extrudiert wird. Diese Unterscheidung wird deshalb eingeführt, weil konvexe Prismen im späteren Gebrauch in einer Simulation in mehrererlei Hinsicht einfacher zu berechnen sind. Der erste Schritt beim Erzeugen eines Prismas ist die Erzeugung der Querschnittskontur des Prismas. Zu diesem Zweck öffnet man den Dialog 'Menu / Create / New Element' und selektiert in der Combobox links oben den gewünschten Element-Typ:

- konvexes Prisma -> Polygon
- konkaves Prisma -> Quadstrip

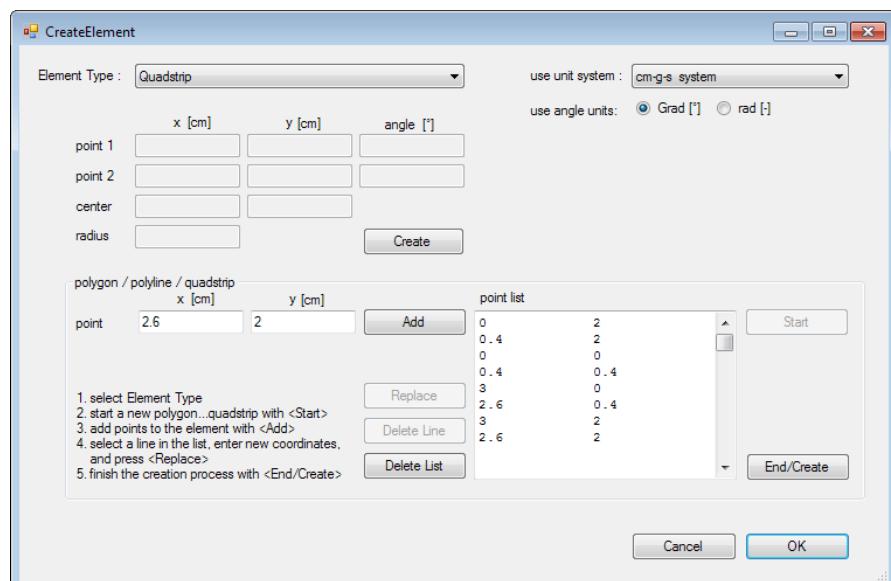
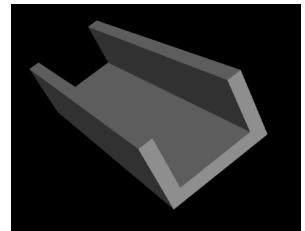


FIGURE 21. Der Dialog 'Create Element'

Für die Eingabe der Punkte des neuen Elementes ist die untere Hälfte des Dialogs zuständig. Wesentlich ist an diesem Punkt die etwas unterschiedliche Eingabereihenfolge der Punkte für die beiden Element-Typen 'Polygon' oder 'Quadstrip':

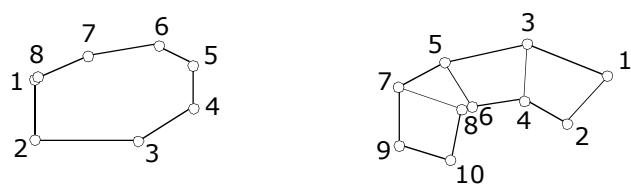


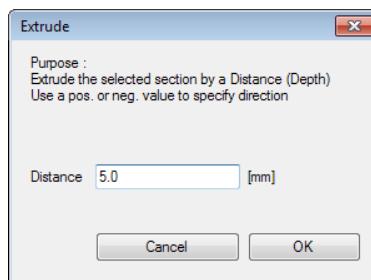
FIGURE 22. Die Reihenfolge und Orientierung der Punkte (Polygon links, Quadstrip rechts)

Die Punktfolge eines Polygons muss im Gegenuhrzeigersinn erfolgen. Dies ist immer die sog. positive Orientierung eines flächenartigen Gebildes. Der letzte Punkt des Polygons muss mit dem ersten zusammenfallen.

Die Punktfolge eines Quadstrips muss so erfolgen, dass die ungeraden Punkte (1, 3, 5, ...) im Gegenuhrzeigersinn der Berandung definiert werden. Ein Quadstrip hat im Übrigen immer eine gerade Anzahl Punkte. Falls dies für eine gegebene Fläche impraktikabel ist, dann können ggf. zwei Punkte zusammengelegt oder ein Zwischenpunkt auf einer Geraden gesetzt werden, wie z.B. der Punkt Nr. 6 im letzten Beispiel. Die Bezeichnung 'quadstrip' ist im Übrigen der Bezeichnung dieser Gebilde in der OpenGL-Grafiksoftware entnommen (Streifen aus Vierecken).

Bevor Sie mit der Eingabe der Punkte beginnen, sollten Sie das entsprechende Einheitensystem anwählen und bevor Sie den Dialog verlassen drücken Sie die Taste 'End/Create' und erst anschliessend 'OK'.

Was nun weiter folgt ist die Extrusion des neuen Querschnitts, der auf dem Bildschirm als blaue Kontur dargestellt ist. Zu diesem Zweck selektieren Sie das soeben erzeugte Element und rufen den Dialog 'Extrude...' auf : (Menu / Create / Extrude Contour...). Setzen Sie die vorgesehene Länge des Prismas.



**FIGURE 23.** Der Dialog 'Extrude'

Der Vorgang der Erzeugung ist damit abgeschlossen und das neue Objekt sollte sowohl in den 2D- als auch in den 3D-Ansichten zu sehen sein. Das Prisma befindet sich im Moment noch in seiner Nulllage und kann wie jeder Primitivkörper mit den beiden Dialogen

Menu / Edit / Edit Object / Properties...  
 Menu / Edit / Edit Object / Rotation und Position...

weiter verarbeitet werden.

## Bemerkungen

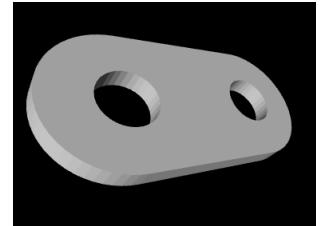
Es ist grundsätzlich nicht falsch, wenn die Querschnittsfläche eines konvexen Prismas mittels einem Quadstrip erzeugt wird. Die Realisierung mit einem konvexen Polygon wäre lediglich geringfügig leistungsfähiger, was sich aber nur beim Einsatz einer grossen Anzahl von Objekten wirklich bemerkbar macht.

Wenn allerdings ein konkav polygon zu einem Prisma extrudiert wird, dann wird bereits die Darstellung dieses Objektes im 3D-Fenster mit an Sicherheit grenzender Wahrscheinlichkeit falsch erfolgen. Die Ursache für diesen Fehler liegt in der OpenGL Grafiksoftware, wie sie in sonar verwendet wird. OpenGL kann grundsätzlich nur konvexe Polygone darstellen. Alle komplizierteren Formen müssen intern in konvexe Polygone zerstückelt werden. Aus diesem Grund erzielt man mit Polygonen, die vorneweg als konvex definiert werden, eine höhere Grafikleistung.

# Extrudierte Linien-Bogen-Konturen

Die Erstellung dieser Gebilde geschieht grundsätzlich in zwei bis drei Schritten:

1. Eine Linien-Bogen-Kontur bereitstellen.
2. Falls das Objekt zusätzlich noch Löcher haben soll, dann müssen auch entsprechende Kreise gezeichnet werden.
3. Extrudieren



## Der Dialog Create/Show Contour

Eine Linien Bogen Kontur kann entweder explizit am Bildschirm mit dem Dialog 'Create Element' Stück für Stück gezeichnet werden oder die Kontur kann als Datei importiert werden. In letztem Fall geschieht die Definition der Linien-Bogen Kontur in einem externen Programm und wird als Textfile bereitgestellt und mit der Taste 'Load Data (Textfile)' importiert. Die letzte Variante ist besonders dann die einzige praktikable Methode, wenn es sich um eine komplexe Berandung wie z.B. ein Zahnrad handelt. Dazu mehr später. Wir sind also in der Lage auf zwei Wegen an den Dialog, wie unten abgebildet, heranzugehen.

1. In der Frontansicht (x-y-Ebene) die Linien und Bogen zeichnen, selektieren und den Dialog Menu / Create / New Contour... aufrufen
2. ohne Selektion den Dialog aufrufen  
>>> die Taste 'Load Data (Textfile)' im Dialog wird aktiviert

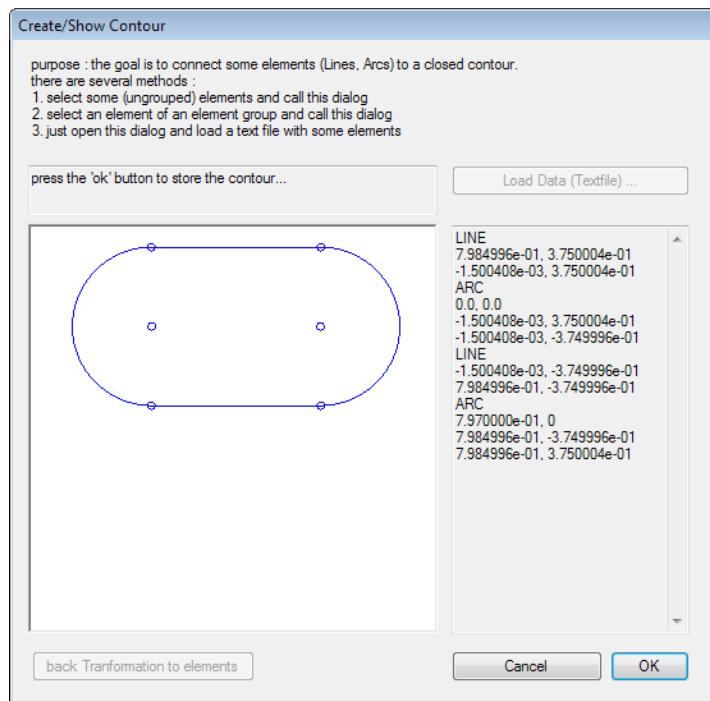
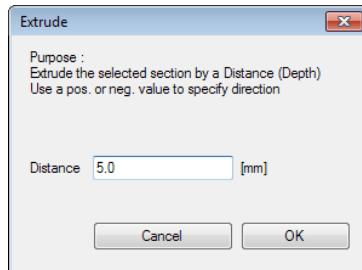


FIGURE 24. Der Dialog Create Contour / Show Contour

Für einfache Konturen, wie im Dialog oben abgebildet, kann die Erzeugung der Linien und Bogen mit dem Dialog 'Create Element' durchaus Sinn machen. In diesem Fall muss nur noch die Taste

‘OK’ gedrückt werden und die Kontur ist für die Extrusion bereit. Diese erfolgt genau gleich wie bereits bei den Prismen, durch Aufrufen des Dialogs ‘Menu / Create / Extrude Contour...’



**FIGURE 25.** Der Dialog Extrude, angewendet auf Linien-Bogen-Konturen

Auch hier erfolgt die Weiterverarbeitung anschliessend mittels:

Menu / Create / Edit Object / Properties...  
 Menu / Create / Edit Object / Rotation und Position...

## Das Format eines Textfiles

Das Textfile in welchem eine geschlossene Kontur aus Linien- und Bogen-Elementen definiert wird, muss ein bestimmtes Format aufweisen. Dieses Format kann leicht in jedem Text- oder Tabellen-Verarbeitungsprogramm geschrieben werden. Obwohl die Zahlenreihe unten aus Platzgründen zweispaltig angeschrieben ist, muss sie im betreffenden Text-Programm in einer Spalte vorbereitet werden. Das abgedruckte Beispiel definiert die Aussenkontur einer sog. Zahnlasche wie nebenstehend abgebildet.



```
**
LINE
7.398821e-01, 3.750002e-01
5.211704e-02, 3.750002e-01
ARC
0.052117, 0
5.211704e-02, 3.750002e-01
-2.945094e-01, 1.430916e-01
ARC
-0.013267, 0.013613
-2.945094e-01, 1.430916e-01
-2.821187e-01, -1.399479e-01
LINE
-2.821187e-01, -1.399479e-01
-9.624104e-02, -4.653798e-01
ARC
-0.017178, -0.420221
-9.624104e-02, -4.653798e-01
4.358308e-02, -4.880332e-01
ARC
0.396000, -0.315425
5.722373e-01, -2.410849e-01
2.197618e-01, -2.410849e-01
ARC
0.805266, 0.013613
1.074118e+00, -1.399479e-01
1.114814e+00, 7.132228e-03
ARC
0.739882, 0.000000
1.114814e+00, 7.132228e-03
7.398821e-01, 3.750002e-01
-0.419172, 0.028427
4.358308e-02, -4.880332e-01
2.197618e-01, -2.410849e-01
ARC
0.396000, -0.315425
5.722373e-01, -2.410849e-01
2.197618e-01, -2.410849e-01
```

**FIGURE 26.** Das Textfile einer sog. Zahnlasche wie nebenstehend abgebildet

Die Datei ist eine Liste von 2-dimensionalen Punkten (x-y-Ebene). Das Format der Linien und Bogen ist wie folgt festgelegt:

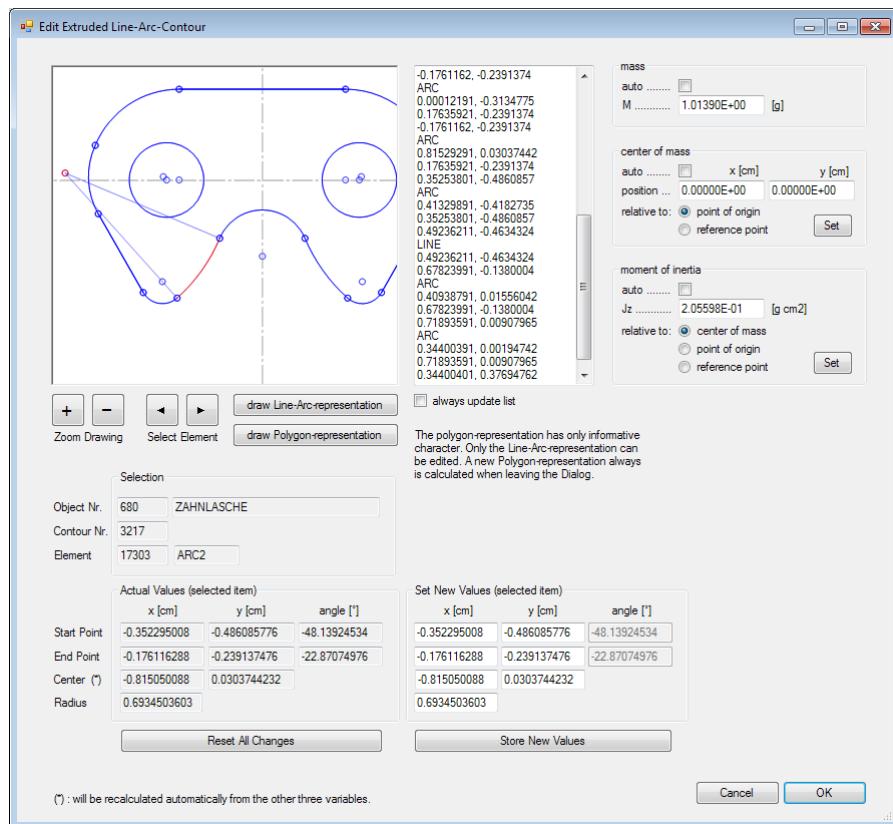
```
LINE
p1.x, p1.y      Startpunkt
p2.x, p2.y      Endpunkt
```

**Textfile Definition für eine Kontur (Linien, Bogen)**

```
ARC
ctr.x, ctr.y      Bogenzentrum
p1.x, p1.y      Startpunkt (Bogen im Gegenuhrzeigersinn)
Radius = (ctr -> p1)
p2.x, p2.y      Endpunkt (Bogen im Gegenuhrzeigersinn)
Endwinkel = (ctr -> p2)
```

## Die Nachbearbeitung von extrudierten Konturen

Unter Nachbearbeitung verstehen wir eine geringfügige Korrektur der bestehenden Kontur und nicht die Umwandlung in etwas völlig Neues. Dafür wäre dieser Dialog, um den es hier geht, nicht geschaffen. Der Benutzer ist mit diesem Dialog in der Lage sog. Fertigungstoleranzen ins Spiel zu bringen und ihre Auswirkungen in Simulationen zu untersuchen. Der Dialog wird aufgerufen mit 'Menu / Edit / Edit Object / Extruded Contour...'.



**FIGURE 27.** Edit Extruded Line-Arc-Contour

Eine typische Handlung mit diesem Dialog könnte der Reihe nach etwa wie folgt aussehen:

- Die Maus wird über das Bild der Kontur bewegt und das zu ändernde Bogenelement kann mit einem Mausklick selektiert werden. Alternativ können dazu auch die Pfeil-Tasten 'Select Element' benutzt werden.
- Die Koordinatenwerte des selektierten Elements werden im unteren Teil des Dialogs zweimal angezeigt. Links unveränderlich als Gedankenstütze wie sie im Moment sind, rechts zum Editieren durch den Benutzer.

- Wollte man den Start- und End-Punkt des selektierten Bogens, wie abgebildet, etwas nach rechts verschieben, dann müsste man die Koordinaten (x,y) dieser beiden Punkte entsprechend ändern. Sobald die Taste 'Store New Values' betätigt wird, werden die eingegebenen Daten umgesetzt und auch in der Grafik angezeigt.
- Wesentlich ist an diesem Punkt festzuhalten, dass mit jeder Änderung auch die beiden Nachbarelemente automatisch verändert werden. Dies immer in der Absicht, eine nahtlose, zusammenhängende, geschlossene Kontur aufrecht zu erhalten.
- Rechts oben im Dialog können ggf. die physikalischen Eigenschaften, die mit einer geometrischen Veränderung einhergehen, neu berechnet und gesetzt werden.

## Bemerkungen

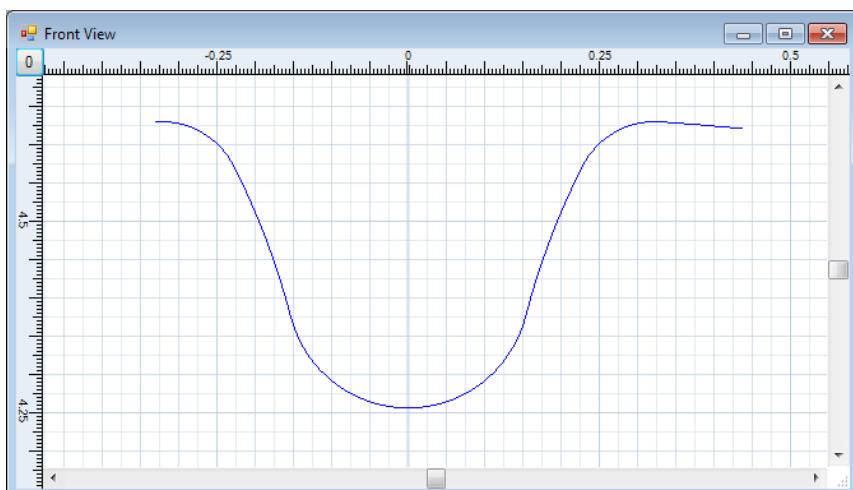
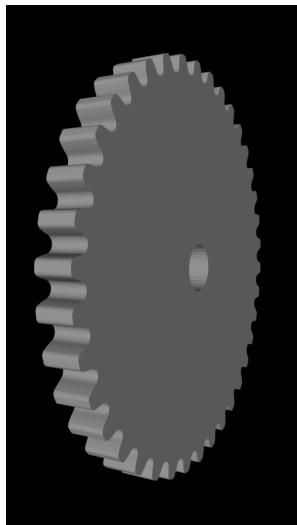
Das Verändern der Geometrie eines Objektes im Modell kann weitergehende Folgen haben. Wenn das Objekt vorher mit anderen Objekten mit Links verbunden war, dann haben sich diese Links in keiner Weise verschoben bzw. mitverschoben. Es liegt in der Verantwortung und Obhut des Benutzers, solche Links ggf. zu löschen und neu zu setzen.

Eine Änderung der geometrischen Lage der Bohrungen kann im Weiteren dazu führen, dass z.B. ein Bolzen der vorher exakt auf der Lochachse lag nun daneben liegt und die Lochberandung schneidet. Falls dies zutrifft müssten entsprechende Gegenmassnahmen ergriffen werden. Eine Möglichkeit wäre, das Modell mit einer Hilfssimulation zu beruhigen. Sehen Sie dazu das Kapitel '*ein Modell beruhigen*'.

## Komplexe Linien-Bogen Konturen

---

Die Erzeugung einer komplexeren Kontur, wie z.B. diejenige eines Zahn- oder Kettenrades läuft fast zwingend über externe Dateien ab. Konkret würde man für die Definition des nebenstehenden Zahnrades in einem externen Programm ein Textfile mit der Kontur eines einzigen Zahns bereitstellen. Der Zahn hätte in diesem Textfile bereits den richtigen Abstand zum Radzentrum (im Ursprung), und entspräche einem Rapport, wie er sich längs dem Umfang Z-mal wiederholt.



**FIGURE 28.** Eine importierte Zahnkontur (1Rapport)

In der Regel liegen diese Daten in der einen oder anderen Form im CAD-System vor und können dort extrahiert werden. Das Textfile des oben dargestellten Zahns hätte das folgende Format:

### Die Daten eines Zahns

```

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
```

Das folgende sonar-script erledigt die ganze Arbeit um aus diesen Zahndaten eine geschlossene Zahnkontur längs dem Umfang zu erzeugen:

```
BEGIN SCRIPT Zahnrad_Z38
-- die Rohdaten eines Zahnkettenrades erzeugen
-- das file 'Z38.txt' entspricht den oben angeschriebenen Daten
IMPORT COLLECTION_LINE_ARC(E1, FILENAME, "C:\Import\Z38.txt")
CONCATENATE ELEMENTS (E1, RING, 0, 0, 38)
UNGROUP ELEMENTS (ALL) -- ungroup loaded pattern
GROUP ELEMENTS (ALL) -- group the ring as a whole
CREATE ELEMENT(E2, CIRCLE, 0, 0, 0, 0, 0, 1.0, 0.54)
-- end of script
```

#### sonar script für Zahnrad-Kontur

- Die erste Linie des script's lädt aus dem angegebenen Ordner die Datei mit den Daten eines einzelnen Zahns.
- Die zweite Anweisung trägt dieses Zahnmuster 38-mal längs dem Umfang ab. Der Ursprung (0, 0) wird dabei als Drehzentrum verwendet
- Die letzte Linie setzt im Zentrum einen Kreis für die vorgesehene Bohrung

Jetzt haben wir eine geschlossene Kontur und ein Kreis. Der Rest der Zahnrad-Definition erfolgt völlig analog wie zu Beginn dieses Kapitels erklärt. Die Zahnkontur und der Kreis werden selektiert und mit dem Dialog 'Extrude' extrudiert.

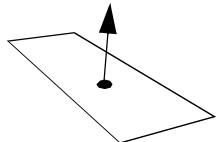
# Unverrückbare Objekte

---

## Ebenen

---

Eine Ebene hat in 'sonar' eine unendliche Ausdehnung. Eine Ebene ist in einem gewissen Sinne eine einfache Methode, in einem Modell eine unverrückbare, raumfeste 'Wand' einzubauen. Der Simulationsbereich kann mit einer Ebene in eine bestimmte Richtung beschränkt werden. Ebenen wirken bezüglich Kollisionen mit Objekten wie die Oberfläche eines anderen Objektes welches räumlich fixiert ist. Eine Ebene kann parallel zu einer Hauptebene (x-y, x-z, y-z) gelegt werden oder im Raum eine beliebige Lage haben. Sechs Ebenen könnten den Simulationsraum z.B. schachtförmig einrahmen. Eine Ebene wird definiert mit einem Vektor, der senkrecht auf der Ebene steht und seinen Fusspunkt in der Ebene hat.



Menu / Edit / New Plane...

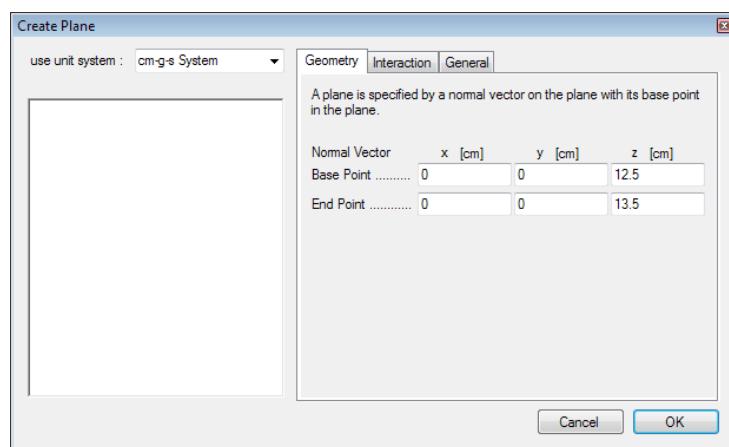


FIGURE 29. Der Dialog Create Plane

## Fixpunkte

---

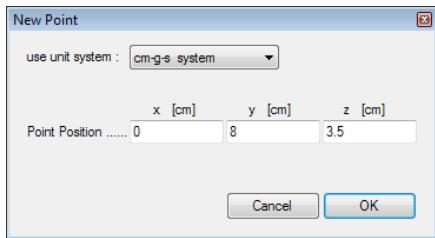
Ein Fixpunkt ist wie der Name sagt ein fester Raumpunkt, welcher als Ankerpunkt für Links benutzt werden kann. Ein Fixpunkt hat für sich allein keinen Einfluss und keine weitere Funktion während einer Simulation. Er spielt im Kollisionsverhalten der Objekte keine Rolle und wird in diesem Sinne nicht wahrgenommen. Er bekommt seine Bedeutung erst dann, wenn an ihm etwas befestigt wird. Dies ist im Prinzip nur mittels Links möglich. Die Freiheitsgrade der Bewegung die ein Objekt haben kann, können mit der Anzahl von Fixpunkten, an denen das betr. Objekt befestigt wird, eingeschränkt werden.

- Objekt mit einem Link an einem Fixpunkt befestigt:  
-> raumfestes Kugelgelenk
- Objekt mit zwei Links an zwei Fixpunkten befestigt:  
-> raumfeste Drehachse
- Objekt mit drei Links an drei Fixpunkten befestigt:  
-> Objekt ist bezüglich Translation und Rotation blockiert

Erzeugt wird ein Fixpunkt mit:

Menu / Create / New Fixpoint...

Menu / Create / Transform Point to Fixpoint



**FIGURE 30.** Der Dialog 'New Point' erzeugt einen Rohdatenpunkt

Ein Objekt welches mit Links an Fixpunkten montiert ist, egal wie viele das sind, ist immer elastisch befestigt. Ist ein Objekt z.B. an drei Fixpunkten aufgehängt, dann kann sich das Objekt immer noch im Rahmen der zulässigen Link-Elastizität bewegen und zwar sowohl translatorisch als auch rotativ. Das Objekt kann folglich je nach Zustand der äusseren Anregungen um seine Befestigungen herum vibrieren. Das Gleiche gilt analog für den Fall von einem oder zwei Links.

## Grids

Grids sind frei formbare Oberflächen. Grids haben eine definierte endliche Ausdehnung. Diese Oberflächen werden mit einer Netzstruktur festgelegt. Wesentlich an diesen Strukturen ist die Eigenschaft, dass die Oberflächen im Allgemeinen keine geschlossenen Volumina, sondern offene Berandungen bilden. Als solche haben Grids keine physikalischen Eigenschaften wie Dichte und Masse. Folglich können sie im Laufe einer Simulation auch nicht beschleunigt oder bewegt werden. Es sind wie die Ebenen raumfeste Gebilde welche eine sinnvolle Berandung für bewegte Objekte anbieten und in dieser Eigenschaft normal an den Interaktionen teilnehmen.

## fixierte Objekte

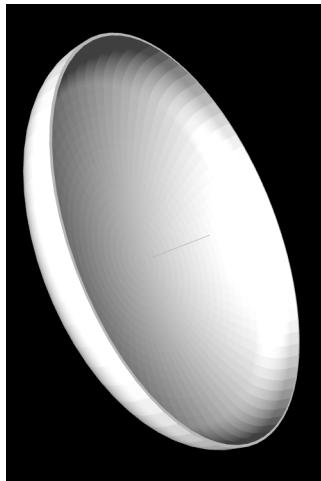
Wird gewünscht, dass ein Objekt unverrückbar fixiert bleibt und auf diese Weise nur noch die Funktion einer Berandung und eines Stosspartners wahrnimmt, dann kann das betr. Objekt als raumfest deklariert werden wie folgt:

Menu / Edit / Edit Object / Properties / General / Fixed in Space (nonrelocatable)

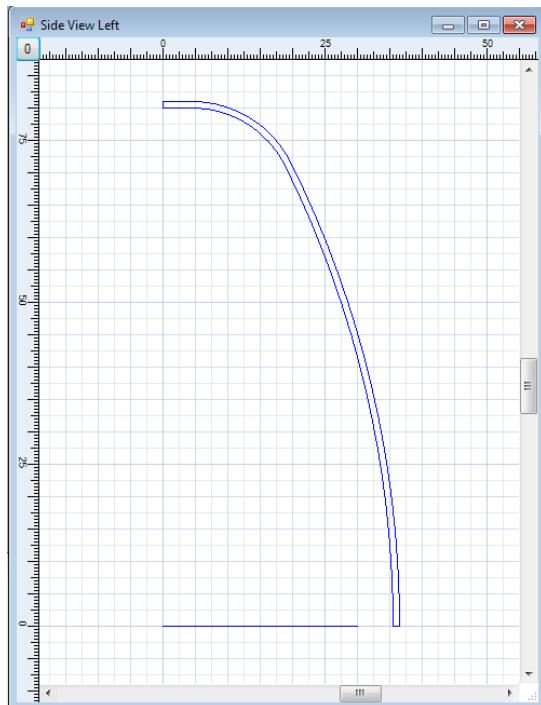
Der Unterschied zur Befestigung an drei unterschiedlichen Fixpunkten ist die Elastizität. Ein Objekt welches als raumfest deklariert wurde ist absolut in Ruhe und hat keine Vibratoren mehr, was immer auch auf das Objekt einwirkt.

# Rotationskörper

---



In einem der letzten Kapitel wurde die Extrusion einer Linien-Bogen Kontur behandelt. Mit Rotationskörpern tun wir im Prinzip dasselbe, mit dem Unterschied, dass die Extrusion einer Kontur nicht linear längs der Z-Achse erfolgt, sondern rotativ um die Z-Achse herum. Aber auch hier besteht die Aufgabe vorerst darin, eine Querschnittsfläche bereit zu stellen. Diese 2D-



**FIGURE 31.** Die Rohdaten-Kontur in der X-Y-Ebene vor der Verschiebung in die Z-Y-Ebene

Kontur wird völlig analog wie bei der Extrusion in der X-Y-Ebene definiert, wozu, wie bereits erwähnt, mehrere Möglichkeiten offen stehen:

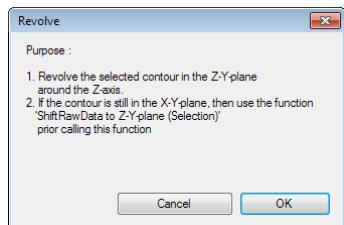
- Dialog 'New Element' -> Linien und Bogen
- Externe Dateien
- Dialog 'New/Show Contour'
- Makro

Ist die Kontur in der X-Y-Ebene definiert, dann wird diese vor dem Rotationsprozess automatisch in die Z-Y-Ebene transformiert. Der Grund für dieses Vorgehen ist letztlich der, dass Rohdaten und damit auch Konturen immer in der X-Y-Ebene definiert werden, die eigentliche Rotation aber immer um die Z-Achse erfolgt. Dies wiederum bedingt, dass die Querschnitts-Kontur für die Rotation in der Z-Y-Ebene bereitgestellt werden muss. Um die Sache für den Benutzer so einfach wie möglich zu halten, erledigt das System diese Transformation im Zuge des Rotationsprozesses in eigener Regie.

Der Benutzer zeichnet also wie gewohnt in der X-Y-Ebene mit mehreren Elementen eine geschlossene Berandung, wandelt diese in eine Kontur um und wählt die eigentliche Funktion für die Erzeugung des Rotationskörpers

Menu / Create / Revolve Contour...

Auf diese Weise sind alle Arten von Rotationskörpern realisierbar: Teller-artige Gebilde, Wellen oder auch rotierte Querschnittsprofile jeder Art (ringförmige Objekte), usw. Jede Linien-Bogen Kontur kann zu einem entsprechenden Rotationskörper weiterverarbeitet werden. Wichtig ist, dass bei Rotationskörpern ohne zentrale Öffnung, so wie im letzten Beispiel, das Querschnittsprofil der Rohdaten bis zur X-Achse reicht ( $Y=0$ ) und ein Stück weit dieser folgt. Es sollte längs der Kontur aber nie Punkte mit negativen Y-Werten geben.



**FIGURE 32.** Dialog -> Menu / Create / Revolve Contour...

Die anschliessende Weiterverarbeitung des Rotationskörpers geschieht mittels:

Menu / Create / Edit Object / Properties...  
Menu / Create / Edit Object / Rotation und Position

Damit werden die physikalischen Eigenschaften und die räumliche Lage und Position eingestellt.

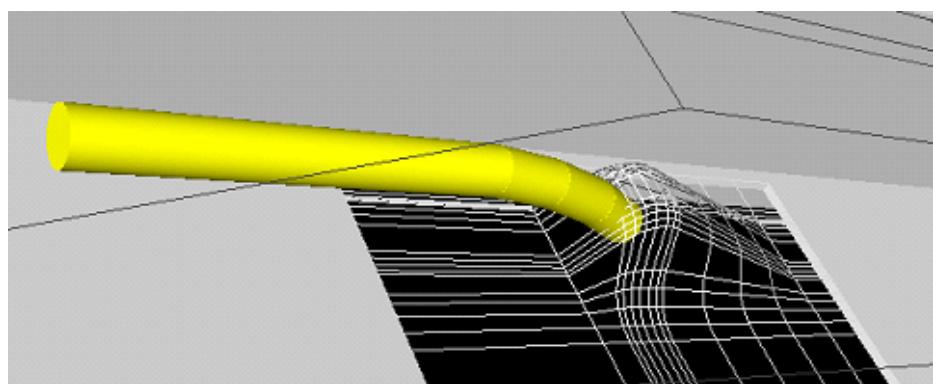
# Netzstrukturen (Grids)

---

## Überblick

---

Netzstrukturen sind Näherungen an Freiformflächen. Frei geformte dreidimensionale Oberflächen werden mit einer angemessenen Anzahl von Netzlinien in Vierecke zerlegt. Die einzelnen Vierecke sind im Allgemeinen keine Ebenen sondern räumlich verwundene bzw. leicht verdrehte Einheiten, welche zusammen die gewünschte Oberfläche annähern. Diese Netze sind räumlich fixierte Objekte welche als Ganzes beliebig gedreht und positioniert werden können, die aber, zumindest in der Basisversion von sonar, während einer Simulation ihren Standort und ihre Rotationslage beibehalten. So gesehen sind die Netze raumfeste Berandungen die den Simulationsbereich mit ihrer Oberflächenform begrenzen. Sie eignen sich zum Bau von Kanälen und von Ausbuchtungen in Kanälen welche z.B. von Partikel durchströmt werden und an diesen Abprallen oder um sie herum strömen. Zusatz-Tools von sonar bieten ev. auch räumlich bewegliche Freiformkörper an.



**FIGURE 33.** Ein aerodynamisch geformter Höcker als Beispiel einer Grid-Struktur

Die Verformbarkeit eines Netzes ist in dieser Form zudem begrenzt. Dies wird deutlich wenn wir zeigen, wie diese Netzstrukturen entstehen. Ein Netz dieser Art wird in der Regel in einer EXCEL-Tabelle definiert. In Anlehnung an das letzte Bild würde man vorerst mit einer absolut planen Netzstruktur beginnen welche wir als X-Y-Struktur bezeichnen wollen. Dieses zweidimensionale Netz, von dem wir ausgehen wollen, würde wie ein Blatt Papier oder wie ein Stück Stoff flach auf dem Tisch liegen, enthielte aber bereits die eingezeichneten Netzlinien. Die Knoten bzw. die Kreuzungspunkte dieser Netzlinien würden nun in der dritten Dimension, senkrecht zur Tischebene, hochgezogen. Dabei würde jeder Knoten genau so weit hochgehoben, bis sich am Ende das Netzbild ergäbe wie es oben abgebildet ist.

Daraus wird ersichtlich, dass damit keine Netze erzeugt werden können, die einen Körper mehr als bis zur Hälfte seiner Oberfläche umschließen. Vielmehr handelt es sich um gerichtete Ausbuchtungen senkrecht zur ursprünglichen Ebene. Technisch gesprochen erinnert uns dieses Vorgehen an eine Stanzmaschine die mit einem Stempel eine Form in ein Blech hineindrückt. Falls der Benutzer komplexere Formen modellieren will, wie z.B. einen geschlossenen Kanal, dann muss er das mit mindestens zwei getrennten Formen erreichen, die von zwei Seiten her den Kanal umschließen.

## Definition einer Netzstruktur

Die Definition von Netzstrukturen erfolgt über externe Dateien, welche in geeigneten Programmen wie EXCEL oder anderen vorbereitet werden. Wir unterscheiden bislang zwei Arten von Datenstrukturen (Typ 1, Typ 2)

Beim ersten Typ handelt es sich um ein allgemeines Netz mit beliebigen Vierrecken. Bei diesem Grid-Typ werden alle Koordinaten-Knotenpunkte ( $x, y, z$ ) explizit vorgegeben. In einem Tabellenkalkulationsprogramm wie EXCEL führt dies zu folgendem Format. Jeder zu definierende Punkt des Netzes benötigt dazu drei Zellen, d.h. eine Kolonne und drei Zeilen für die Koordinaten ( $x, y, z$ ). Die Anordnung der Punkte in der Tabelle entspricht derjenigen in der Realität (siehe Abbildung links).

|   | A   | B   | C   | D   |
|---|-----|-----|-----|-----|
| 1 | x   | x   | ... | x   |
| 2 | y   | y   | ... | y   |
| 3 | z   | z   | ... | z   |
| 4 | x   | x   | ... | x   |
| 5 | y   | y   | ... | y   |
| 6 | z   | z   | ... | z   |
| 7 | ... | ... | ... | ... |
| 8 |     |     |     |     |

FIGURE 34. Systematischer Aufbau einer Netzstruktur (Typ 1) in EXCEL

Der zweite Netztyp hat spezielle Eigenschaften. In dieser Variante werden die Höhen senkrecht zur Bildschirmebene in die Tabelle eingetragen. Der Benutzer blickt so betrachtet von oben auf den Tisch wo ein Geländemodell aufgestellt ist. In der obersten Zeile und der Kolonne am linken Rand werden dazu die x- und y-Koordinaten eines Netzes eingegeben, welches vorerst flach auf dem Tisch liegt. Anschliessend muss der Benutzer aber nicht für jede Zelle eine Höhe angegeben, es genügt das Geländemodell mit einem Profilquerschnitt (p) und einem Längsschnitt (s) zu charakterisieren.

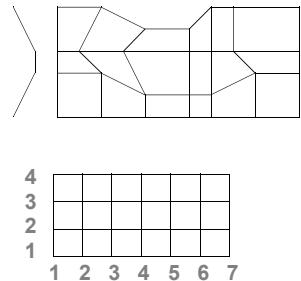
|   | A   | B | C | D   | E | F | G | H | I | J |     |
|---|-----|---|---|-----|---|---|---|---|---|---|-----|
| 1 | 0   | x | x | x   | x | x | x | x | x | x | ... |
| 2 | y   |   |   |     | p |   |   |   |   |   |     |
| 3 | y   |   |   |     | p |   |   |   |   |   |     |
| 4 | y   |   |   |     | p |   |   |   |   |   |     |
| 5 | y   | s | s | s/p | s | s | s | s | s | s | ... |
| 6 | y   |   |   |     | p |   |   |   |   |   |     |
| 7 | y   |   |   |     | p |   |   |   |   |   |     |
| 8 | y   |   |   |     | p |   |   |   |   |   |     |
| 9 | ... |   |   |     |   |   |   |   |   |   |     |

FIGURE 35. Systematischer Aufbau einer Netzstruktur (Typ 2) in EXCEL

Zwischenprofile entlang der sog. sweepline (s) werden aus dem vorgegebenen Profil (p) durch lineare Interpolation berechnet. Das im Überblick zu diesem Kapitel gezeigte Bild hat einen Profilkörper dieses Typs.

Sie können sich vorstellen, dass wir mit dieser Methode die Form eines Autos eingeben wollen. Wir setzen folglich in der Mitte des Autos einen Querschnitt durch das Auto. Anschliessend setzen

Datenstruktur Typ 1  
Ein 'Grid' in seiner geometrischen (oben) und topologischen Ansicht (unten)



|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 4 |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 |
|   |   |   |   |   |   | 7 |

Datenstruktur Typ 2

Legende:

0: unused value

x: x-Koordinaten

y: y-Koordinaten

p: z-Koord. profile

s: z-Koord. sweepline

Beispiel

wir einen Längsschnitt genau in der Mitte des Wagens welcher von ganz vorne bis ganz hinten reicht. Letztlich wird dann die Form der Aussenhülle des Autos automatisch so berechnet, dass der definierte Querschnitt von der Mitte des Autos von ganz vorne bis nach ganz hinten verschoben und dabei seine Höhe immer so skaliert wird, dass der Mittelpunkt des Profils die Linie bzw. die vorgegebene Höhe des Längsschnitts berührt. Es ist wie bei einem Kuchen in Cake-Form welcher von vorne bis hinten in Scheiben zerschnitten wird. Bei uns hat der Cake die Form eines Autos.

Daraus wird ersichtlich, dass die Freiheit der Formgebung bei dieser Netzstruktur beschränkt ist auf ein einziges Profil welches nur noch in der Höhe skaliert werden kann. Auch dieser Netztyp kann keine geschlossene Form haben. Die gesamte Oberfläche der Netzstruktur unterliegt denselben Einschränkungen wie bei einer Cakeform. Man muss den Cake noch herausnehmen können. Die Oberfläche ist maximal eine Halbschale, kann aber auch flacher sein.

## Berechnungen an der Oberfläche

---

Beiden besprochenen Netzstrukturen (Typ 1, Typ 2) ist es eigen, dass die einzelnen Vierecke der Netzstrukturen nicht planar sind. Die meisten Vierecke einer Netzstruktur haben geometrisch gesehen eine verwundene Form. Mit anderen Worten, wollte man die einzelnen Vierecke eines Netzes auf einen Tisch legen, dann würden nicht alle vier Ecken den Tisch berühren bzw. in der Tischebene liegen. Das Ganze erinnert uns an einen vierbeinigen Stuhl mit ungleich langen Stuhlbeinen. Dies bringt Probleme mit sich, wenn man mit solchen Oberflächenelementen Kollisionspunkte berechnen will. Allerdings ist das kein Problem für den Benutzer. Die sonar Software wendet dafür ein automatisches Interpolationsverfahren an, welches für jeden Oberflächenpunkt innerhalb eines solchen Vierecks alle drei Raum-Koordinaten berechnet. Die Berechnungsmethode läuft auf eine Aufteilung des Vierecks in vier Dreiecke hinaus, welche in der Mitte des Vierecks einen gemeinsamen Eckpunkt haben.

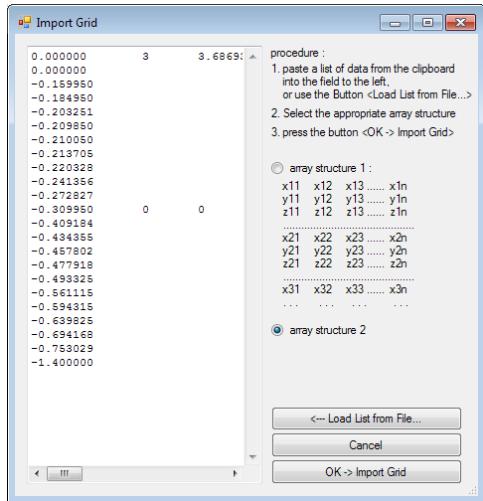
## Die Weiterverarbeitung der Netzstruktur

---

Die in EXCEL definierte Netzstruktur, sei es Typ-1 oder Typ-2, wird als Textfile bzw. mit der Endung 'TXT' auf einem Datenträger gespeichert und bildet so die geometrische Definition des Netzes. Natürlich können wir das Netz in diesem Textformat noch nicht in einer sonar-Simulation direkt verwenden. Dazu fehlen dem Netz auch noch viele Eigenschaften welche für eine Verwendung als physikalisches Objekt notwendig sind. sonar stellt aber einen Dialog zur Verfügung, mit dem dieses Netz sehr schnell in ein sonar-Objekt umgewandelt werden kann. Wenn es also so weit ist, dass Sie das fragliche Netz in ein Modell integrieren möchten, dann wählen Sie den folgenden Dialog:

Menu / Import/Export / Import Grid...

Im Dialog selbst betätigt der Benutzer die Taste 'Load List from File...' und selektiert das Textfile welches er in EXCEL wie beschrieben vorbereitet hat. Je nach Typ des Netzes (Typ 1 oder Typ 2) drückt er den zugehörigen Radio-Button und schliesslich 'OK -> Import Grid'.



**FIGURE 36.** der Dialog Import Grid für das Laden externer Grid-Strukturen. Auf der linken Seite des Dialogs ist ein vorbereitetes File vom Typ 2 zu erkennen, welches mit der Taste 'Load List from File...' geladen wurde. Das File ist in der Horizontalen nicht vollständig sichtbar. Benutzen Sie im Dialog ggf. die Rollbalken.

Ab diesem Moment ist das Grid ein vollwertiges sonar-Objekt welches in allen Ansichten dargestellt und im 'Object Tool' aufgeführt wird und welchem wie gewohnt weitere Eigenschaften zugeordnet werden können. Ein Doppelklick im Object Tool führt Sie weiter.

# Objektgruppen

---

Mehrere Objekte können gruppiert werden. Wir sprechen im Folgenden von zwei Stufen von Gruppen:

- Gruppen
- Supergruppen

Eine Supergruppe kann mehrere Gruppen und auch einzelne Objekte die keiner Gruppe angehören enthalten. Die Zugehörigkeit eines Objektes zu einer Gruppe wird dadurch erkannt, dass das Objekt als Attribut eine automatisch festgelegte Gruppennummer bzw. Supergruppen-Nummer bekommt. Eine Gruppennummer ist eine ganze Zahl grösser als Null. Die Gruppennummer von ungruppierten Objekten ist dagegen Null. Gruppen werden so wie sie erzeugt werden fortlaufend nummeriert. Ein bestimmtes Objekt kann zu einem bestimmten Zeitpunkt nur einer Gruppe und einer Supergruppe angehören. Mehrfachzugehörigkeiten sind nicht erlaubt. Wird ein bereits gruppiertes Objekt einer anderen Gruppe zugeteilt, dann wird die alte Gruppennummer durch die neue überschrieben.

Eine Gruppennummer ist jedoch nicht notwendigerweise eine unveränderliche Zahl für eine Gruppe von Objekten. Sie ist so gesehen keine Identifikationsnummer. Dies wird deutlich, wenn zwei sonar-Dateien zusammengeführt bzw. vereinigt werden, indem die Objekte eines Modells in ein anderes Modell hineinkopiert werden. Oft entstehen da Konflikte mit gleichen Gruppennummern in verschiedenen Dateien. Deshalb gibt das Programm bei diesem Vorgang den einzelnen Gruppen ggf. automatisch unterschiedliche neue Gruppennummern.

Die Gruppen- und Supergruppen-Nummer eines Objektes wird in verschiedenen Dialogen, welche eine Eigenschaft eines bestimmten Objektes editiert, angezeigt. Viele Eigenschaften von Objekten können zur Erleichterung auf ganze Gruppen oder Supergruppen kollektiv angewendet werden. Die Tatsache, dass ein gewisses Objekt einer Gruppe angehört hat vorerst überhaupt keine Konsequenzen. Keine einzige Eigenschaft in seinem gesamten Verhalten wird dadurch beeinflusst. Erst wenn anschliessend an eine Gruppe neue Eigenschaften vergeben werden, wirkt sich das indirekt auf das betreffende Objekt in dieser Gruppe aus. So könnte man einer Gruppe z.B. als Ganzes eine neue Farbe geben. Dies wirkt sich folglich auch auf jedes Objekt in dieser Gruppe aus.

Der Sinn dafür zwischen Gruppen und Supergruppen zu unterscheiden, kann an einem Beispiel am besten erklärt werden. Eine Litze oder ein Seil besteht aus mehreren Drähten. Jeder Draht wiederum besteht aus einer langen Kette von Zylindern die untereinander elastisch verbunden sind. Indem man die einzelnen Drähte zu Gruppen erklärt und das Seil als Ganzes zu einer Supergruppe, kann man auf verschiedenen Stufen auf die Objekte eines Seils zugreifen um sie ggf. zu verändern.

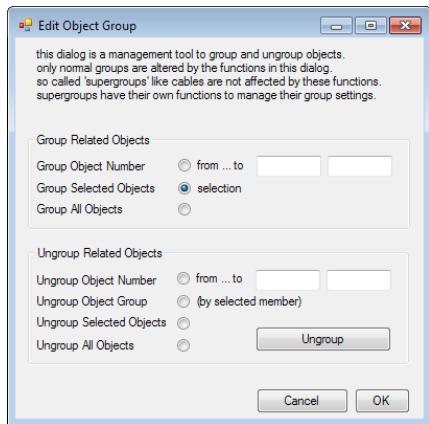
## Verwaltung von Objektgruppen

---

Eine Anlaufstelle um die Gruppeneigenschaften an sich zu verwalten, ist der Dialog 'Edit Object Group'. In diesem Dialog kann einer Selektion von Objekten eine gemeinsame Gruppen- oder Supergruppen-Eigenschaft zugeordnet werden. Mit diesem Dialog ist es aber auch möglich eine bestehende Gruppennummer wieder aufzuheben.

Menu / Edit / Group / to Object Group...  
Menu / Edit / Group / to Object Supergroup...

Menu / Edit / Ungroup / Object  
Menu / Edit / Ungroup / Object Group...  
Menu / Edit / Ungroup / Object Supergroup...  
Menu / Edit / Edit Group Name...



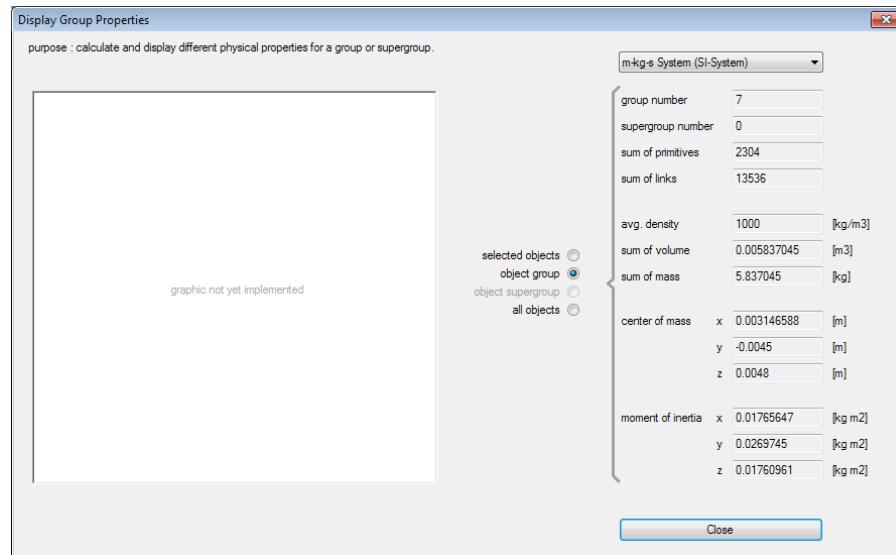
**FIGURE 37.** Der Dialog Edit Object Group

Mit der Vergabe einer Gruppennummer passiert mit den betroffenen Objekten vorerst weiter nichts. Die Einführung einer Gruppennummer ist lediglich eine Vorbereitung für eine andere Funktion, deren Eigenschaften an eine ganze Gruppe zu vergeben oder eine Aktion für alle Objekte in einer Gruppe durchzuführen.

## Darstellung von Gruppeneigenschaften

Manchmal möchte man eine gewisse Eigenschaft für eine ganze Gruppe kennen. Das kann ein Durchschnittswert in der Gruppe oder ein Summenwert über die ganze Gruppe sein. Der Dialog 'group properties' berechnet und zeigt wie dargestellt ein paar wichtige Gruppeneigenschaften.

Menu / Functions / Calculate Group Properties...



**FIGURE 38.** der Dialog 'Display Group Properties'

## Anwendungen von Objektgruppen

### Object Tool

Im sog. 'Object Tool' werden Objekte die einer Gruppe oder Supergruppe angehören auf effiziente und übersichtliche Art und Weise dargestellt. Statt jedes Objekt in der Baumstruktur einzeln aufzuführen, kann der Benutzer durch spezielle Navigationstasten auf die einzelnen Mitglieder der Gruppe zugreifen.

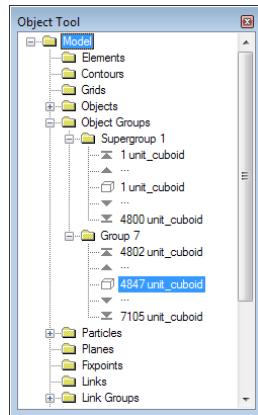


FIGURE 39. Object Tool

### Objekte kollektiv ändern

Wann immer die Eigenschaften für alle Mitglieder einer Gruppe (kollektiv) geändert werden sollen, ist der Dialog 'Collective Change of Objects' eine gute Wahl

Menu / Edit / Collective Change... / Objects

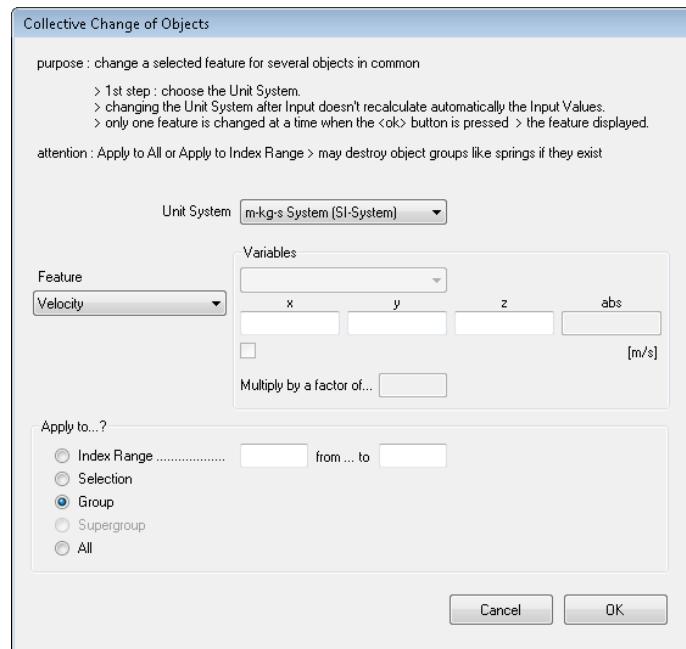


FIGURE 40. Der Dialog 'Collective Change of Objects'

Mit diesem Dialog lassen sich sogar Kombinationen von Eigenschaften zur Selektion einer 'Gruppe' von Objekten heranziehen. So könnte man z.B. die Selektion der Objekte weiter

eingrenzen, indem man sagt, alle Objekte einer Gruppe welche die gleiche Farbe haben, sollen einen neuen Wert für eine bestimmte Variable bekommen.

Das was wir mit Objekten taten, kollektives ändern, das kann man mit einem anderen Dialog auch mit Links tun. Der Dialog erlaubt die Veränderung verschiedenster Variablen in Zusammenhang mit Links. Diese Variablen können im popup-Menu 'Variable' des Dialogs eingestellt werden.

Menu / Edit / Collective Change... / Links

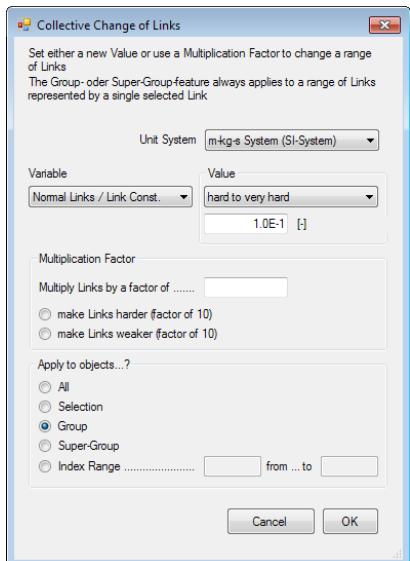


FIGURE 41. Der Dialog 'Collective Change of Links'

Die Materialeigenschaften einer Gruppe werden im nächsten Kapitel gesondert behandelt. Eine einzelne Materialeigenschaft einer Gruppe kann zusätzlich mit dem folgenden Dialog verändert werden

Menu / Material / Material Property....:

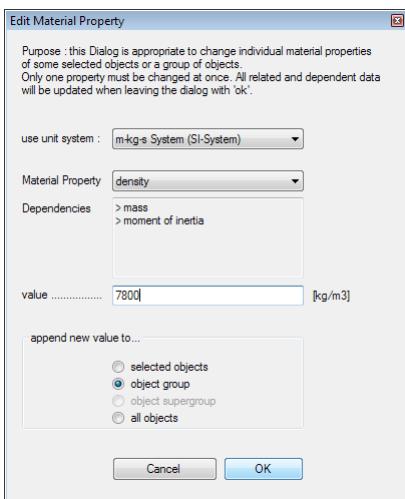


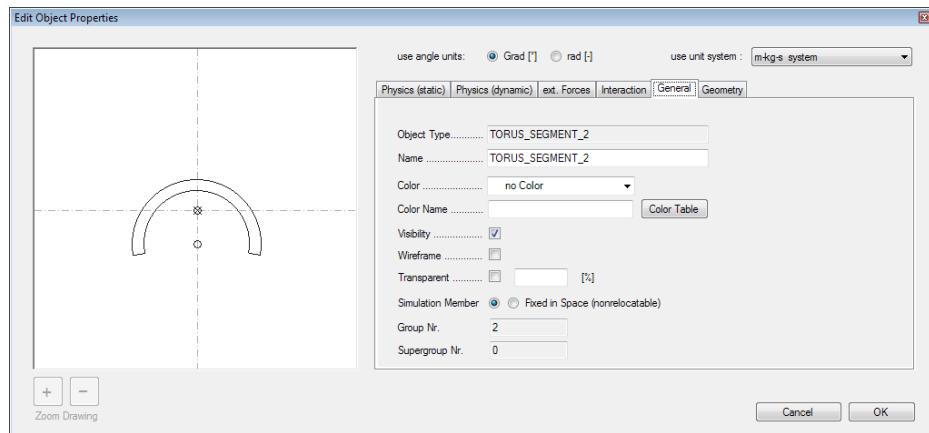
FIGURE 42. Der Dialog 'Edit Material Properties'

Schliesslich erlaubt der oft zitierte Dialog 'Edit Object Properties' das Kontrollieren und Einsehen der group- oder supergroup property, indem in diesem Dialog das Panel 'General' eingeschaltet

## Links kollektiv ändern

## Material von Objektgruppen ändern

wird. Auf dieser Seite des Dialogs wird die Gruppen- und Supergruppen-Nummer des betreffenden Dialogs angeschrieben.



**FIGURE 43.** Der Dialog 'Edit Object Properties' und die Anzeige der Gruppen-Nummern.

## Makro Befehle für Objektgruppen

Eine ganze Reihe von Makro-Befehlen steht in Zusammenhang mit Gruppeneigenschaften. Sie alle sind vom Makro Befehlstyp

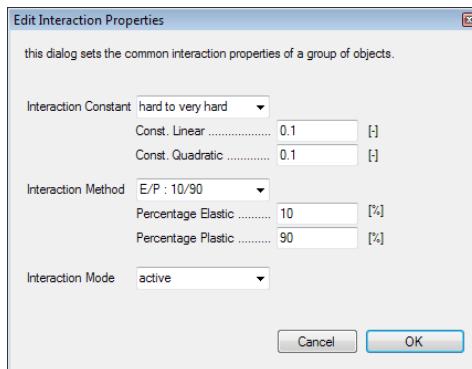
```
SET GROUP_PROPERTY (o1, parameters)
```

Der erste Parameter (o1) trägt dabei die Objektnummer eines Repräsentanten der Objektgruppe. Konsultieren Sie dazu bitte das Handbuch 'sonar script Language Guide'.

## Interaktion von Objektgruppen anpassen

Die Interaktionseigenschaften können für eine ganze Gruppe von Objekten wie folgt verändert werden:

Select an object of a group (View / Object Tool)  
Menu / Interactions / Edit Properties (Groups)...



**FIGURE 44.** Der Dialog 'Edit Interaction Properties'

# Material Modelle

---

## Überblick

---

Materialmodelle wie sie hier beschrieben werden sind Gruppen von Objekten zugeschrieben, welche sich aus stückweise gleichen oder ähnlichen 'Primitives' aufbauen. Einem einzelnen 'Primitive' ein Materialmodell dieser Art zuzuordnen macht keinen Sinn, weil die Eigenschaften des Materialmodells in den Links zwischen den Objekten eingebaut sind. Materialmodelle beschreiben das elastisch-plastische Verhalten von Körpern. Tatsächlich steckt in sofern der Mechanismus dazu aber in den Verbindungen zwischen den Körpern, während die Körper selbst, wie bereits bekannt, Starrkörper sind. Eine weitere Eigenschaft die in dieses Kapitel gehört ist das Bruchverhalten. Diese zwei Eigenschaften von Materialien können in sofern unabhängig voneinander behandelt werden. Während das Materialmodell die Spannungs- Dehnungs-Kurve festlegt, bestimmt das Bruchmodell, wo, längs dieser Kurve, der Bruch eintreten soll. Der Bruch kann von der Dehnung selbst abhängig sein, es gibt aber auch andere Kriterien für einen Bruch. Im Folgenden werden zuerst die Dialoge vorgestellt, welche für das Editieren des Spannungs- Dehnungs-Diagramms vorgesehen sind. Wir unterscheiden in diesem Zusammenhang zwei Dialoge

- Editieren von Spannung/Dehnungs-Variablen
- Editieren von Spannung/Dehnungs-Funktionen

Spannungs-Dehnungs-Kurven beschreiben das Dehnverhalten von Materialien unter Last. Die meisten dieser Kurven haben eine sog. Elastizitätsgrenze, welche unmittelbar oder kurz nach dem Hook'schen Bereich den Punkt der maximalen reversiblen Dehnung markiert. Wird das Material über diesen Punkt hinaus gedehnt, dann kehrt dieses nach der Entlastung nicht wieder in seinen ursprünglichen Zustand zurück. Diese Elastizitätsgrenze wird in der Technik für jedes Material mit einachsigen Zugversuchen an zylindrischen Körpern gemessen und dokumentiert. Damit das Simulationssystem auch bei komplizierteren dreidimensionalen Belastungen eines Körpers an jedem Ort und zu jedem Zeitpunkt automatisch eine entsprechende Vergleichsspannung berechnen und mit der bekannten und gemessenen Spannungs-Dehnungs-Kurve vergleichen kann, gibt es verschiedene Vergleichsspannungs- bzw. Festigkeitshypothesen. Der Benutzer von sofar hat die Möglichkeit in folgendem Dialog unter mehreren bekannten Hypothesen auszuwählen und die betreffende Hypothese gewissen Objektgruppen zuzuweisen

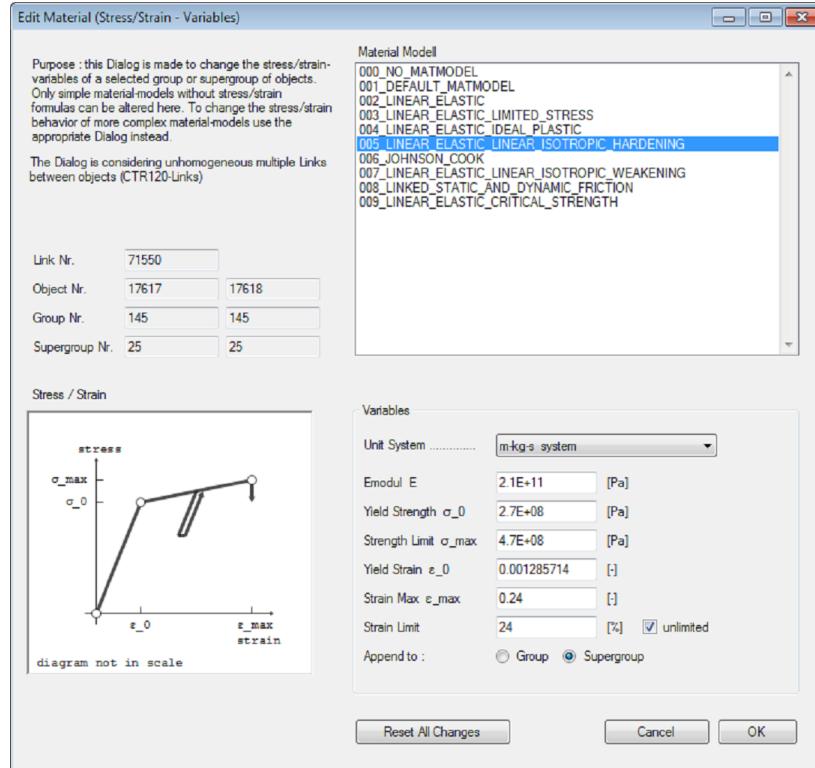
- Dialog Vergleichsspannungs-Hypothese

Schliesslich stehen dem Benutzer weitere ergänzende Dialoge zur Verfügung um gezielte Änderungen am Materialverhalten einzubringen. Im Dialog

- Edit Material Property

hat er z.B. die Möglichkeit eine bestimmte Material-Eigenschaft eines Objektes oder einer Objektgruppe gezielt zu verändern.

## Dialog Spannungs/Dehnungs-Variablen



**FIGURE 45.** Der Dialog mit einer Darstellung in Zusammenhang mit dem selektierten Materialmodell Nr.005 Linear Elastic Linear Isotropic Hardening

Der Benutzer selektiert vor dem Aufruf dieses Dialogs ein einzelnes Objekt einer Objektgruppe, welcher er ein Materialmodell zuordnen will. Der Zugriff auf den Dialog geschieht mit

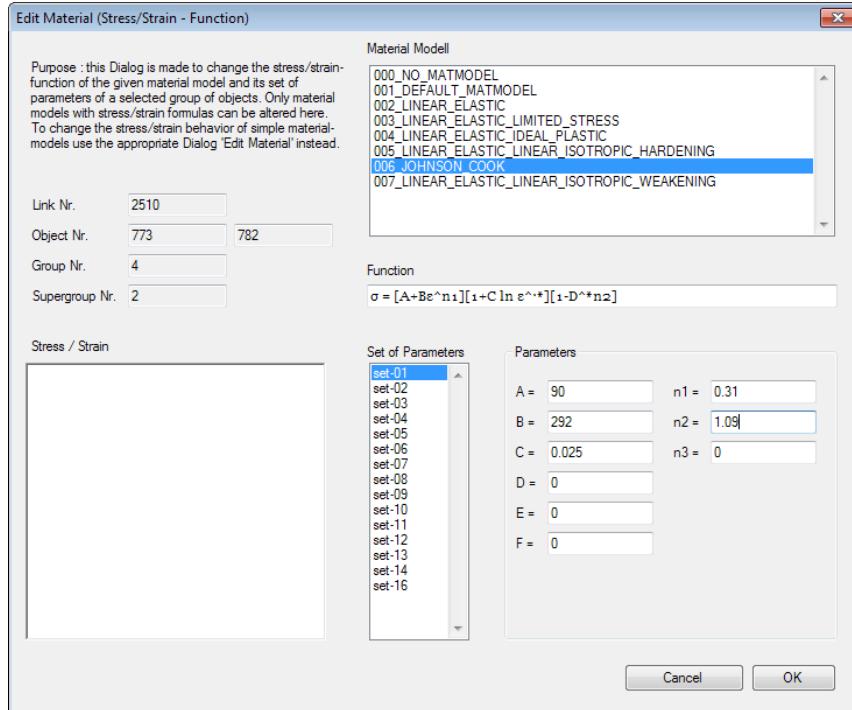
Menu / Material / Stress/Strain Variables...

Die Bedienung des Dialogs ist intuitiv verständlich. Der Benutzer selektiert in der Liste der Modelle ein bestimmtes Modell und bekommt das entsprechende charakteristische Spannungs-Dehnungs-Bild des Modells schematisch angezeigt. Zum gewählten Modell gibt der Benutzer die zugehörigen Materialwerte seiner Wahl in die betreffenden Felder ein.

## Die Funktion des Materialmodells im Code

Die Links zwischen allen Objekten der betreffenden Objektgruppe werden nach dem Verlassen des Dialogs alle so eingestellt, dass sämtliche Eigenschaften des Modells wie erwartet zusammenspielen. Insbesondere werden die Werte aller Link-Konstanten so gesetzt, dass die Elastizität der gesamten Gruppe dem E-Modul entspricht. Für jedes Materialmodell sind im Programm-code Funktionen vorhanden, welche das elastisch-plastische Verhalten der Links steuern. Überschreitet ein Link schliesslich die Bruchdehnung, dann wird die Festigkeit des Links auf null und der Link selbst auf 'unused' gesetzt, d.h. der betreffende Link existiert 'de facto' nicht mehr.

## Dialog Spannungs-Dehnungs-Funktionen



**FIGURE 46.** Dialog Spannungs-Dehnungs-Funktionen

Komplexere Materialmodelle werden mit einem erweiterten Dialog definiert. Statt idealisierte Materialmodelle wie im vorangehenden Abschnitt lassen sich damit im eigentlichen Sinn Spannungs-Dehnungs-Funktionen aufsetzen. Ein Beispiel einer solchen Funktion ist das 'Johnson-Cook Materialmodell', wie in der Abbildung als Beispiel dargestellt. Materialmodelle dieser Art werden allerdings nur für spezielle hochdynamische Anwendungen benötigt. In den meisten Fällen wird der durchschnittliche 'user' die Materialmodelle des vorangehenden Abschnittes 'Spannung/Dehnungs-Variablen' einsetzen.

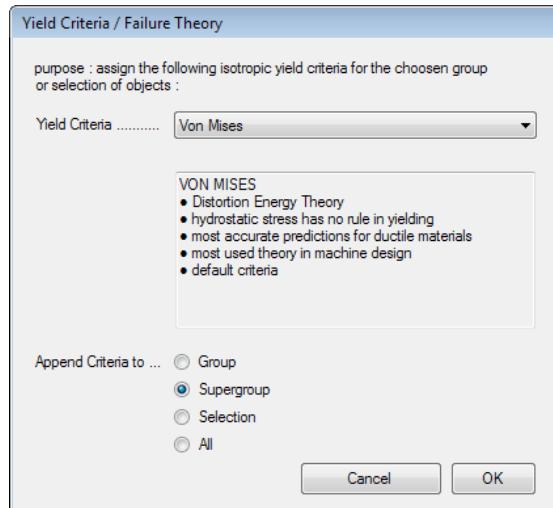
## Dialog Vergleichsspannungs-Hypothese

Die für ein Material angegebenen Materialkennwerte in der Spannungs-Dehnungs-Kurve basieren auf einachsigen Zugversuchen an meist zylindrischen Probekörpern. Ein realer dreidimensionaler Spannungszustand an einer bestimmten Position eines Maschinenteils besteht im Allgemeinen in jeder Raumrichtung aus einer Kombination von Normal- und Schubspannungen. Um einen Vergleichswert dieser Spannungen mit einem einzigen gemessenen und dokumentierten Spannungs-Dehnungs-Punkt in der entsprechenden Kurve herzustellen, müssen die aktuellen Normal- und Schubkräfte mit einer Theorie in einen entsprechenden Vergleichswert umgerechnet werden, welcher schliesslich direkt mit der gemessenen Grösse verglichen werden kann. Für isotrope Materialien stehen uns in sonar folgende Festigkeitshypothesen zur Verfügung

- W.J.M Rankine Maximum Principal Stress Theory (Default)
- St. Venant Maximum Principal Strain Theory
- Beltrami-Haigh Total Strain Energy Theory
- Tresca Maximum Shear Stress Theory
- von Mises Distortion Energy Theory
- Mohr-Coulomb Yield Criterion
- Drucker-Prager Yield Criterion

In sonar werden zur Hauptsache die beiden Theorien 'Rankine' und 'von Mises' eingesetzt. Erstere weil sie sehr einfach ist und wenig Rechenaufwand benötigt, letztere weil sie in der Maschinenindustrie die am häufigsten eingesetzte Theorie ist für Metalle. Den Dialog zum Setzen der Vergleichsspannungshypothese erreicht man mit

Menu / Material / Yield Criteria...

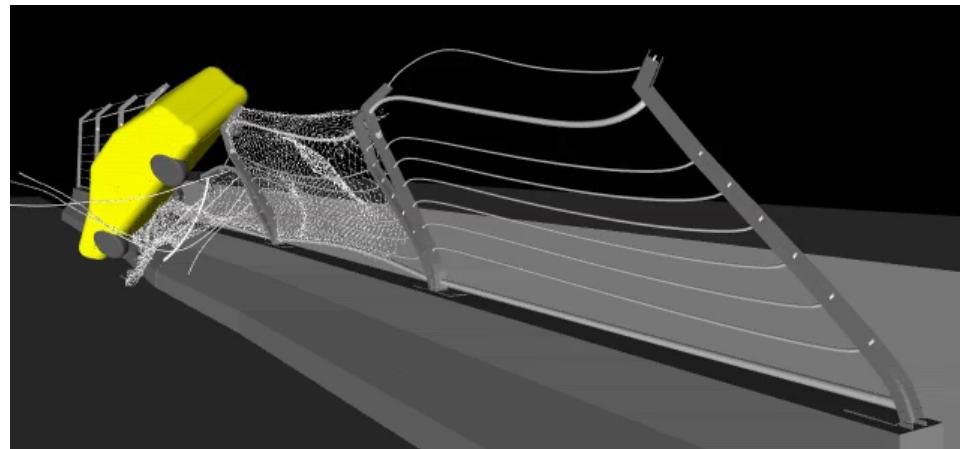


**FIGURE 47.** Dialog Vergleichsspannungshypothese

## Beispiel

---

Das folgende Bild zeigt ein Beispiel eines Modells mit deformierbaren Objektgruppen. Sowohl die senkrechten Stützen (U-Profile) als auch die horizontalen Stangen und Rohre sind elastisch-plastisch deformierbare Körper. Alle diese Teile wurden aus einer grösseren Anzahl von Primitivkörpern aufgebaut, auf ganz ähnliche Weise, wie man einen Körper bei einer Finite-Elemente-Analyse in innere Netzstrukturen unterteilt. Das Modul 'Profile' unterstützt den Benutzer weitgehend bei der Überführung eines starren Profils in ein solches das elastisch-plastisch deformierbar ist.



**FIGURE 48.** Crash Analyse

Im dargestellten Modell wäre für die Seile, Drähte und Stäbe die Rankine-Vergleichsspannungshypothese eine gute Wahl. Man würde folglich die vom System 'default-mässig' gesetzten Einstellungen übernehmen. Bei diesen Teilen handelt es sich um Objekte die hauptsächlich in Achsenrichtung belastet werden. Der Rechenaufwand für eine komplexere

Theorie würde sich in diesem Fall nicht lohnen. Bei den Stützen, die aus einem U-Profil hergestellt werden und die sich z.T. kompliziert verformen und verdrehen, ist der Einsatz der 'Von-Mises'-Vergleichsspannungs-Theorie angebracht. Grundsätzlich gilt, dass der Einsatz der 'Von-Mises'-Theorie nur dort einen Nutzen abwirkt, wo tatsächlich Objekt-Gruppen in einer angemessen hohen räumlichen Auflösung mit inneren Netzstrukturen vorliegen. Die 'Von-Mises'-Theorie berücksichtigt bei der Berechnung der Vergleichsspannung nicht nur die lokalen Zug- und Schärfkräfte, sondern berücksichtigt auch den Anteil des hydrostatischen Druckes im Material welcher auf die kritische Fließgrenze keinen Einfluss hat.

## Möglichkeiten der Darstellung

Das System bietet dem Benutzer farbliche Darstellungen der Spannungsverteilung in einer Objektgruppe. Auf die letzte Abbildung bezogen könnte der Benutzer die 'Rankine'- oder 'Von Mises'-Spannungsverteilung in einem U-Profil dynamisch als Film aufzeichnen. Werden maximal zulässige Spannungen überschritten, dann haben diese Ereignisse Einträge im Event-Logbuch zur Folge. Der Benutzer kann diese Aufzeichnungen jederzeit konsultieren. Kommt es während einer Simulation zu Brüchen, dann läuft die Simulation unter den neuen Verhältnissen weiter.

## Der Dialog Edit Material Property

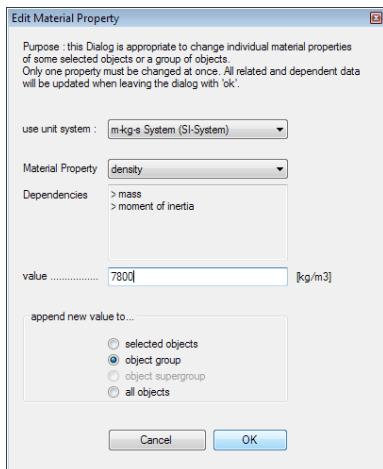


FIGURE 49. Der Dialog 'Edit Material Property'

Dieser Dialog gibt dem Benutzer die Möglichkeit, einzelne wählbare Variablen eines Materials zu ändern. Beispiel: Der Benutzer möchte von einer Objektgruppe die maximal zulässige prozentuale Dehnung von beispielsweise 20% auf 24% erhöhen. Nach der Selektion eines Objektes der betreffenden Objektgruppe im Modell ruft er im Menu den Dialog 'Edit Material Property' auf. Mittels der Combobox in diesem Dialog selektiert er anschliessend das Kriterium 'max. percental strain' und kann im betreffenden Feld einen neuen Wert für diese Grösse eingeben. Mit den anderen physikalischen Eigenschaften in der Combobox verfährt er ggf. völlig analog.

## Der Elastizitätsmodul (young modulus)

Auch den Elastizitätsmodul kann der Benutzer mit dem Dialog 'Edit Material Property' auf die beschriebene Art und Weise ändern. Der Elastizitätsmodul oder abgekürzt E-Modul steht in sonar in einem besonderen Zusammenhang mit den sog. Link-Konstanten. Erzeugt man z.B. eine verlinkte Kette von Zylindern welche zusammen ein verformbares Stangenprofil oder einen Draht darstellen, dann stehen die beiden Grössen E-Modul und Link-Konstante in Bezug auf die Zugbelastung in folgendem Zusammenhang:

$$K = E * \bar{A} / eL$$

(EQ 6)

K: Link Konstante

E: E-Modul

$\bar{A}$ : Querschnittsfläche des Profils

eL: Element Länge der einzelnen Objekte

Im Falle wo nicht ein einzelner Link mit der Link-Konstanten 'K' sondern eine Link-Kombination aus (n) Links die Verbindung zwischen jeweils zwei Objekten herstellt, gilt für die Zugbelastung der folgende Zusammenhang:

$$K = \sum k_i = E * \bar{A} / eL \quad i = [1..n]$$

(EQ 7)

Wenn wir die Querschnittsfläche ' $\bar{A}$ ' und die Länge 'el' der einzelnen Zylinder vorerst als unveränderlich betrachten, dann stehen die beiden Größen 'K' und 'E' in einem linearen Zusammenhang. Ändern wir eine der beiden Größen, dann wird die andere Größe vom Programm entspr. automatisch angepasst. In der Regel gibt der Benutzer den E-Modul des Materials vor und erwartet, dass die Summe der betroffenen Link-Konstanten entsprechend gesetzt werden. Die folgenden Dialoge zeigen den Wert des E-Moduls entweder an oder bieten Editiermöglichkeiten für denselben.

#### Dialog Object Properties

Object Tool / Object -> double click / Dialog Object Properties  
der E-Modul wird im Reiter 'physics static' nur angezeigt

#### Dialog Edit Material Property

Menu / Material / Material Property / young modulus  
der E-Modul steht zum Editieren bereit. Die notwendigen und autom. durchgeföhrten Folgeberechnungen werden im Dialog angezeigt -> Dependencies

#### Dialog Edit Stress/Strain-Var.

Menu / Material / Stress-Strain Variables...  
Der Emodul kann in diesem Dialog geändert werden. Auch für Link-Kombinationen werden die Anpassungen an den Link Konstanten anschliessend automatisch durchgeführt.

#### Dialog Edit Stress/Strain Funct.

Im Dialog 'Spannungs-Dehnungs-Funktionen' schliesslich kommt der E-Modul in dieser einfachen Form wie er für den Hook'schen Bereich gilt nicht mehr vor. Die Normalspannung  $\sigma$  ist dort nicht mehr eine lineare Funktion der Dehnung mit der Konstanten E, sondern wird durch eine komplexere Funktion ersetzt welche diese Aufgabe übernimmt.

# Federn

---

Federn sind sog. Halbfabrikate die in sonar aus einer Anzahl von Primitivkörpern aufgebaut sind. Im Prinzip könnten diese Federn, zumindest die physikalischen, auch vom Benutzer selbst aus elementaren Primitivkörpern selbst manuell konstruiert werden. Ein anderer Weg wäre, diese Federn mit einem Makro zu erzeugen. Ein Erschwernis, das sich bei diesem Vorgehen ergeben würde, wäre die Berechnung der Link-Konstanten in einer Weise, dass die Feder als Ganzes die korrekten Spannungs-Dehnungs-Eigenschaften bzw. die richtige Federkonstante hätte. Der gangbarste Weg wäre wahrscheinlich ein empirisches Vorgehen, indem die Federkonstante einer bestimmten definierten Feder solange verändert wird, bis ihr Verhalten als Ganzes den Erwartungswerten entspricht.

Bereits in der Standard Version von sonar-LAB finden sich deshalb ein paar Funktionen um komplettete Federn mit Hilfe eines Dialogs automatisch erstellen zu lassen. Folgende Federn können mit der Basisversion auf diesem Weg erzeugt werden

- physikalische und mathematische Zugfedern
- physikalische und mathematische Druckfedern
- physikalische Blattfeder

Im Zusatzmodul 'Federn' finden sich eine grosse Anzahl weiterer Federn. Die Helix-förmigen physikalischen Federn sind in sonar aus Torus Segment- und Zylinder-förmigen Primitivkörpern aufgebaut welche untereinander mehrfach mit Links verbunden sind. Die einzelnen Link-Konstanten werden bei der Erzeugung bereits richtig eingestellt, so dass die Feder als Ganzes die definierte Federkonstante bekommt. Eine physikalische Feder verhält sich in einer Simulation sehr detailgetreu entsprechend dem realen Vorbild. Das bedeutet im Einzelnen, dass nicht nur das eigentliche Spannungs- Dehnungs-Verhalten exakt nachgebildet wird, sondern dass auch das gesamte dynamische Verformungsverhalten der folgenden Eigenschaften der Drahthelix nachgebildet wird:

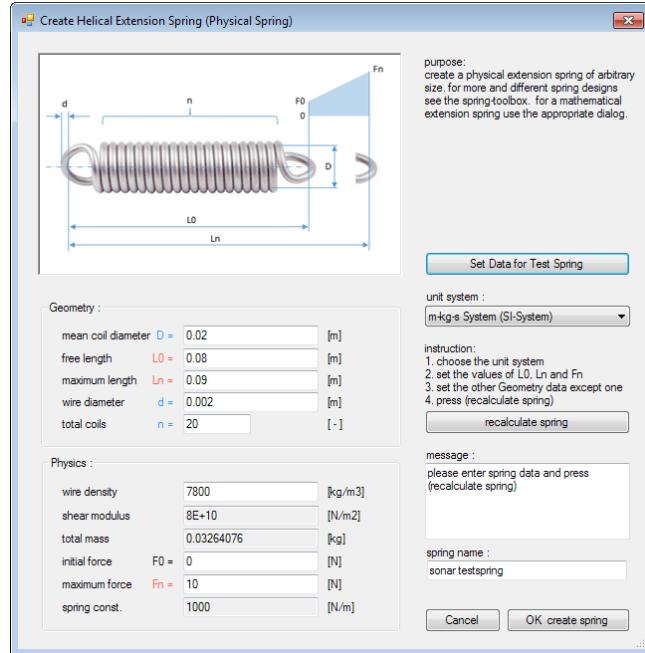
- die Signalfortpflanzung von stossartigen Belastungen durch die Federhelix wird autom. simuliert
- das longitudinale und transversale Schwingungsverhalten der Feder wird explizit mitsimuliert.
- die Eigenschwingungen und das Resonanzverhalten der Federhelix wird autom. simuliert.
- bei entsprechender Einstellung der Eigenschaften kollidieren die Federwindungen wenn sie sich berühren

Bei mathematischen Federn wird lediglich in jedem Rechenzyklus eine Kraft als Funktion der aktuellen Länge berechnet welche kontinuierlich auf die beiden beteiligten Objekte übertragen wird. Die Feder hat aber keine Masse und keine Trägheit. Die Feder hat eigentlich auch kein Abbild. Das Federbild wird lediglich aus den Federdaten in jedem Zyklus neu berechnet und gezeichnet. Es handelt sich aber um ein rein virtuelles Bild der Feder.

Der Zugriff auf alle Feder-Dialoge erfolgt durch

Menu / Create / New Spring / ...

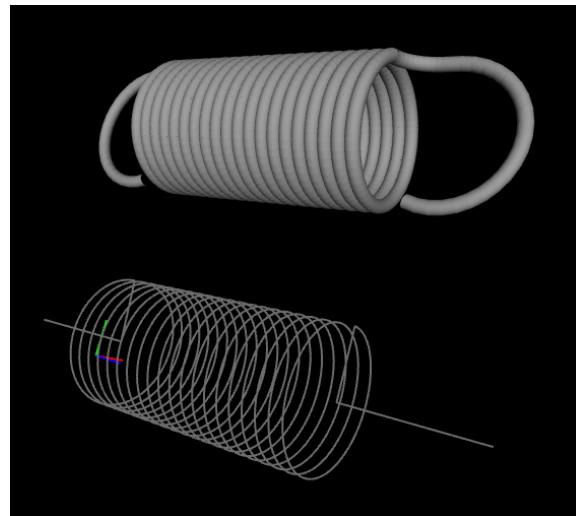
## Zugfedern



**FIGURE 50.** Der Dialog Create Helical Extension Spring (Physical Spring)

Abgebildet ist der Dialog zur Erzeugung der physikalischen Zugfeder. Der entsprechende Dialog für die math. Zugfeder hat das gleiche Aussehen, ausser dass die Felder für die Eingabe von Masse und Dichte fehlen, weil math. Federn keine solche haben.

Im Dialog gibt es eine Taste mit der Bezeichnung 'Set Data for Test Spring' welche sämtliche Einstellungen für eine beispielhafte Feder durchführt und nach dem Drücken der ok-Taste die obere der beiden Federn, wie nachstehend abgebildet, erzeugt. Die untere der beiden Federn ist die mathematische Variante mit den gleichen Einstellungen wie sie mit dem Dialog für die mathematische Zugfeder erzeugt wird.

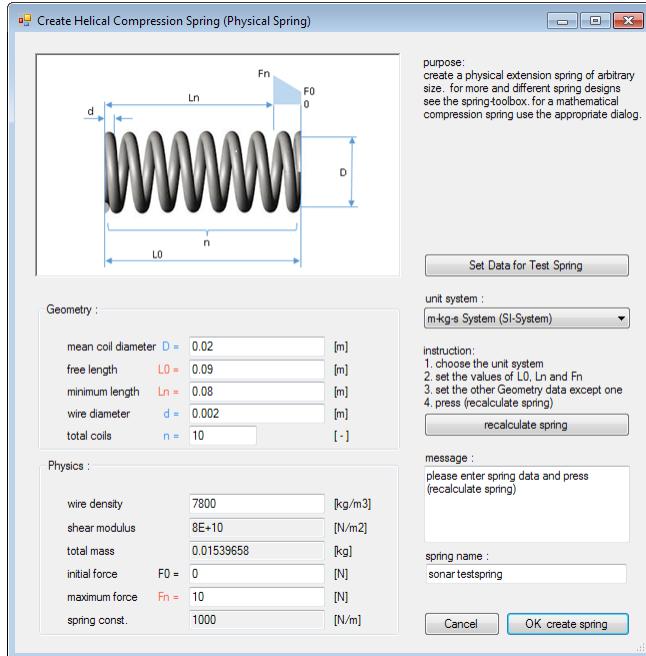


**FIGURE 51.** Eine physikalische und eine mathematische Zugfeder

Physikalische Zugfedern können wahlweise mit Links an anderen Objekten befestigt werden oder einfach wie reale Zugfedern z.B. mit den Ösen über Bolzen gelegt werden worauf sie allein durch

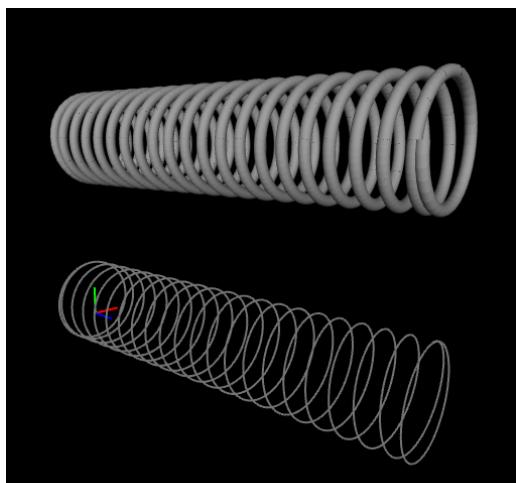
die Kontakt- und Reibkräfte auf ihrer Position gehalten werden. Mathematische Zugfedern hingegen müssen an beiden Enden mit Links an Objekten befestigt werden. Hat ein Modell mathematische Federn, die noch nicht vollständig befestigt sind, dann meldet dies das Programm beim Simulationsstart.

## Druckfedern



**FIGURE 52.** Der Dialog Create Helical Extension Spring (Physical Spring)

Der Dialog für die physikalische und mathematische Druckfeder ist gleich aufgebaut und funktioniert sinngemäß analog. Auch dieser Dialog erlaubt das Abrufen einer beispielhaften Feder mit der Taste 'Set Data Test Spring' welche in der Folge die nachfolgend gezeigte Druckfeder erzeugt.



**FIGURE 53.** Eine physikalische und eine mathematische Druckfeder

Auch in dieser Abbildung wird zum Vergleich die math. Druckfeder daneben gestellt. Was die Befestigung anbelangt gilt dasselbe wie bei der Zugfeder. Eine physikalische Druckfeder kann sich

über Kontakte an seinem Gegenüber abstützen oder mit Links befestigt werden. Math. Druckfedern allerdings müssen mit Links befestigt werden.

## Blattfedern

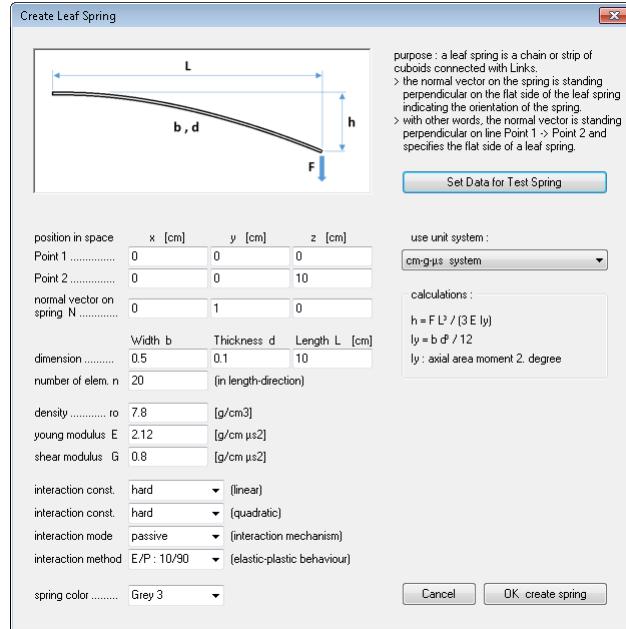


FIGURE 54. Dialog Create Leaf Spring

Blattfedern sind in sonar aus einer Kette von quaderförmigen Blattelementen aufgebaut, also Primitives vom Typ Quader welche untereinander elastisch biegsam verbunden sind. In der Regel werden auf diese Weise ein Duzend bis 40 Elemente verwendet. Im Dialog zur Erzeugung einer Blattfeder kann diese direkt in ihrer räumlichen Lage und Ausrichtung definiert werden. Dies geschieht durch die Eingabe der beiden Endpunkte (Querschnittsmittelpunkte) der Blattfeder und einem Vektor, welcher die Verdrehung um diese Achse vorgibt. Zur Definition dieser Verdrehung stellt der Benutzer einen Vektor senkrecht auf die Blattoberfläche. Natürlich kann der Benutzer die Blattfeder auch unverdreht im kartesischen Koordinatensystem, z.B. längs der Z-Achse, definieren und die endgültige räumliche Lage und Position später festlegen. Der Dialog legt die Blattfeder so aus, dass die Biegesteifigkeit als Funktion des Emoduls und des Querschnittes dem analytisch berechneten Erwartungswert entspricht.

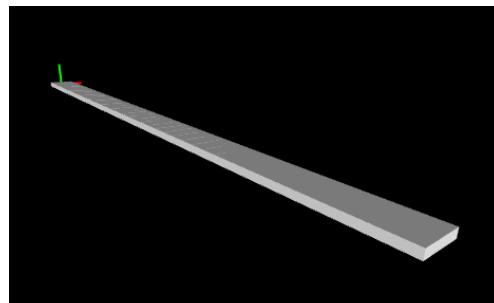
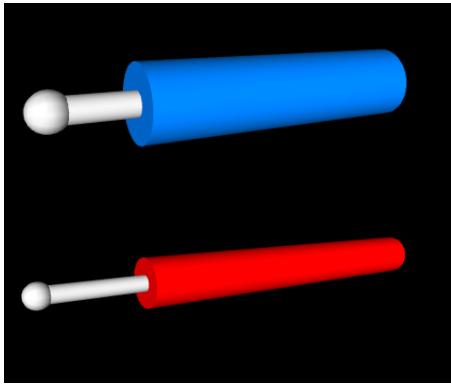


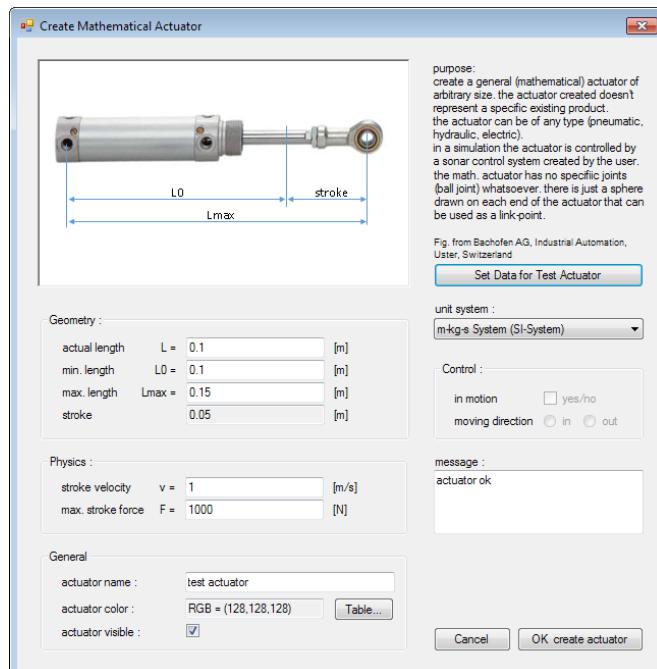
FIGURE 55. Eine typische Blattfeder in ihrer unverbogenen Form wie sie vom Dialog erzeugt wird.

# Aktuatoren und Dämpfer

Diese beiden Halbfabrikate sind im Prinzip mathematische Objekte. Sie haben zwar eine dreidimensional sichtbare Hülle und auch einen beweglichen Zylinder, aber es sind keine Massen und Trägheiten mit ihnen verbunden. Die inneren Funktionen zum Antrieb oder zur Dämpfung der Bewegung des Zylinders sind auch nicht wirklich mit massenbehafteten Teilen vorhanden, sondern durch eine mathematische Funktionalität ersetzt worden. Wie bei der mathematischen Feder üben der Aktuator und der Dämpfer Kräfte zwischen zwei Befestigungspunkten aus die mathematisch definiert werden.



**FIGURE 56.** Ein Aktuator und ein Dämpfer wie sie von den beiden Dialogen erzeugt werden.



**FIGURE 57.** der Dialog Create Mathematical Actuator

Der Zugriff auf die Dialoge geschieht mit

Menu / Create / New Actuator...

Menu / Create / New Damper...

Menu / Edit / Edit Actuator...

Menu / Edit / Edit Damper...

Der von diesem Dialog erzeugte Aktuator hat noch keine Kontrollfunktionen und Steuerungen. Als physikalische Eigenschaften besitzt der Aktuator vorläufig nur eine max. Kraft und eine Hubgeschwindigkeit. Dies sind äussere begrenzende Werte welche die physikalischen Fähigkeiten des Aktuators vorerst einschränken, unabhängig davon wie er nachher gesteuert wird. Darüber hinaus definiert der Benutzer mit diesem Dialog die geometrischen Randbedingungen welche auch den Hubbereich festlegen. Die Befestigung des Aktuators geschieht an den beiden Kugeln die sich an seinen Enden befinden, eine am Gehäuseboden, die andere am Ende des Zylinders. Die Steuerung des Aktuators geschieht über ein sonar script Kontrollsysteem welches den Aktuator über seine Nummer, wie im Object Tool aufgelistet, anspricht. Weitere Informationen zur Steuerung von Aktuatoren finden sich im Manual 'sonar script'.

Der Dämpfer ist vom äusseren Erscheinungsbild her gesehen und wie er an zwei Objekten befestigt wird dem Aktuator ähnlich. Der Unterschied ist seine innere Funktion. Er übt keine aktiven Kräfte aus sondern macht eher das Gegenteil, er dämpft Kräfte durch passive Gegenkräfte, welche der Bewegung immer entgegengesetzt sind.

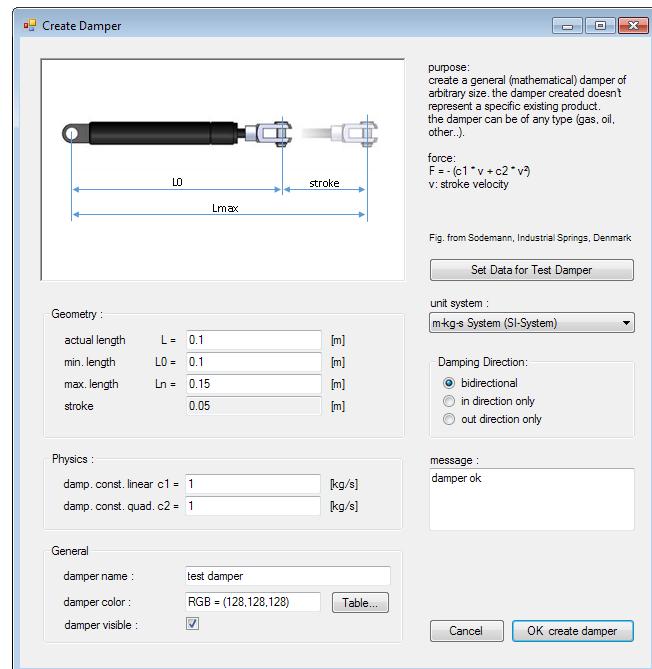


FIGURE 58. der Dialog Create Damper

Der Dämpfer, wie er hier definiert wird, benötigt keine weiteren Kontrollmechanismen. Er ist mit diesem Dialog abschliessend definiert und kann seine Funktion erfüllen. Seine innere Funktion ist ebenfalls mathematisch definiert. Die dynamische Dämpfungskraft, welche der Dämpfer entgegen den beiden Objekten ausübt, ist im Dialog selbst als Hilfestellung nochmals angeschrieben.

$$F = - (c_1 \cdot v + c_2 \cdot v^2) \quad (\text{EQ 8})$$

Die Relativgeschwindigkeit  $v$  wird dabei immer positiv genommen als  $|v|$  bzw. als Betrag der Geschwindigkeit. Die beiden Konstanten sind ebenfalls positiv. Das Minuszeichen in der Formel soll andeuten, dass die Kraft in diesem Sinne immer entgegen der Relativgeschwindigkeit wirkt.

Eine der beiden Konstanten darf natürlich auch auf null gesetzt werden. Die Wirkungsrichtung eines Dämpfers lässt sich im Dialog auf eine Bewegungsrichtung einschränken (ein- oder ausfahren). Die andere Bewegungsrichtung übt dann in einem solchen Fall keine Kraft aus.

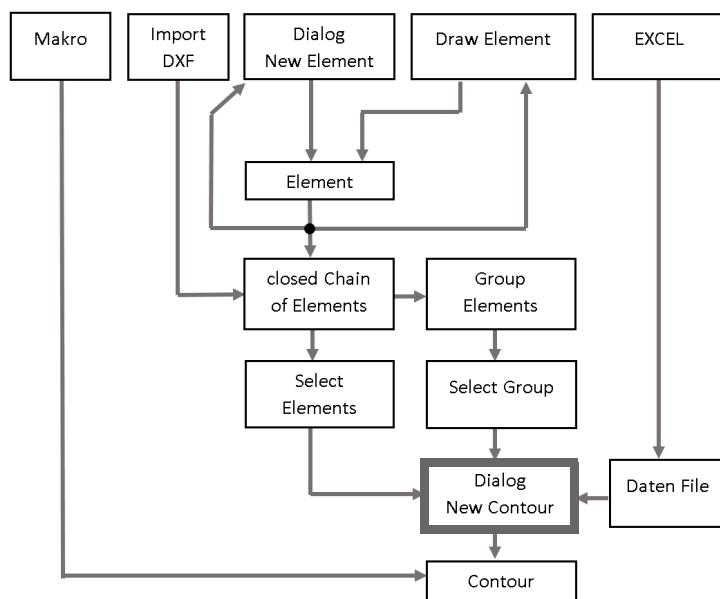
Sowohl für Aktuatoren als auch für Dämpfer gilt: Funktionelle Einheiten mit komplexeren inneren Mechanismen können vom Benutzer mit 'sonar' elementar aufgebaut werden. Komplexere Dämpfungsfunktionen für Dämpfer können mit sonar script implementiert werden.

# Rohdaten

---

Wir sind im bisherigen Verlauf schon mehrmals sog. Rohdaten begegnet. In diesem Kapitel wollen wir diese nochmals vollständig erfassen und die verschiedenen Möglichkeiten, diese zu erzeugen, behandeln. Rohdaten sind meistens 2-dimensional. Wenn nichts anderes angegeben wird zeichnen oder definieren wir Rohdaten immer in der X-Y-Ebene (Front View). Alle Extrude-Funktionen benutzen die X-Y-Ebene als Querschnittsebene und Extrudieren entlang der Z-Achse. Die 'Front View' bzw. die X-Y-Ebene ist folglich unser 2D-Zeichenfenster. In wenigen Fällen benötigen wir auch 3-dimensionale Rohdaten. Diese werden mit dem Dialog 'Import 3D-Polyline' geladen.

Als Rohdaten bezeichnen wir 2- und 3-dimensionale Zeichendaten, so wie sie auch mit einem Zeichenprogramm erzeugt werden. Rohdaten haben noch keine physikalische Bedeutung. Es sind geometrische Elemente wie Linien, Bogen, Kreise, usw. welche die Berandung eines zwei- oder drei-dimensionalen Gebildes beschreiben. Rohdaten bilden oft den Ausgangspunkt zur Erzeugung von 3-dimensionalen Körpern. Wenn wir direkt ein Objekt vom Typ Primitiv erzeugen, z.B. einen Quader, dann überspringen wir in einem gewissen Sinn die Rohdaten, wir erzeugen direkt den physikalischen Körper. Einzelne Objekte sind aber nur auf dem Weg über Rohdaten definierbar, weil ihre Berandung zu kompliziert ist um sie direkt zeichnerisch zu definieren. Zu dieser Sorte von Objekten gehört z.B. eine extrudierte Kontur, deren Berandung vorher aus einer Folge von Linien und Bogen definiert wird. Eine Kontur kann auf verschiedenen Wegen erzeugt werden.



**FIGURE 59.** Mögliche Wege eine Kontur zu erzeugen

## Ein Element grafisch interaktiv erzeugen

---

Erzeugen Sie einfache Elemente wie Punkte, Linie, Bogen und Kreise direkt mit den Werkzeugen in der Tool-Liste im Kopfteil des Bildschirms. Normalen-Vektoren, welche senkrecht auf den Elementen stehen, definiert werden.

## Der Dialog New Element

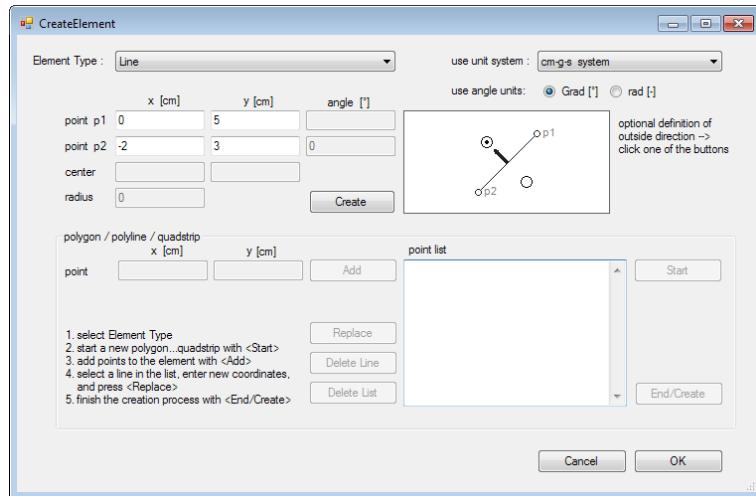


FIGURE 60. Der Dialog Create Element im Zustand (Line, Arc, ...)

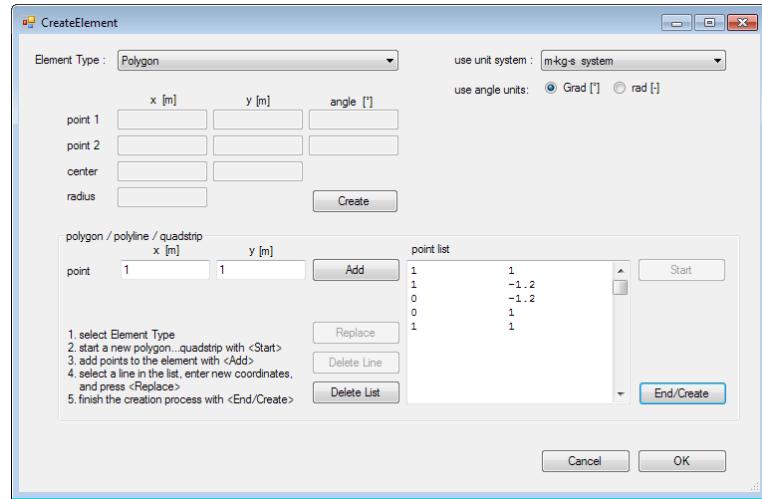
Je nachdem, welche Art von Element der Benutzer im popup-menu links oben selektiert, bekommt der Dialog ein etwas anderes Aussehen mit unterschiedlich selektierten Tasten. Aus diesem Grund ist die erste Handlung beim Öffnen des Dialogs immer die Wahl des Elementes und des Einheitensystems. In der letzten Abbildung ist das Element 'Line' gewählt worden. Wählt der Benutzer stattdessen 'Arc', dann werden für die Eingabe der Daten alle Felder links oben freigegeben. Die zur Verfügung stehende Auswahl an Feldern ist in diesem Fall redundant. Der Benutzer hat dann mehrere Möglichkeiten einen Bogen zu definieren.

Sowohl für den Bogen als auch für den Elementtyp 'Line' ist optional die Angabe der sog. 'Aussenseite' möglich. Diese Eigenschaft ist z.B. bei nicht geschlossenen Oberflächen notwendig, um dem System mitzuteilen, welche Seite als Aussenseite für die Interaktion zuständig sein soll. Der Benutzer legt diese Eigenschaft für alle Elemente die einer offenen Linien-Bogen-Kontur angehören fest. Wird diese Kontur anschliessend extrudiert oder rotiert, dann überträgt sich diese Eigenschaft automatisch auf die gesamte entstehende Fläche. Eine offene Oberfläche dieser Art kann also nur auf einer ihrer beiden Seiten Interaktionen von anderen Objekten wahrnehmen. Die interaktive Grafik rechts oben im Dialog beinhaltet zwei kleine Tasten. Durch das Betätigen einer dieser Tasten legt der Benutzer einen Richtungsvektor fest, welcher seinen Fusspunkt am Element hat und nach aussen zeigt. Für das Element 'Arc' funktioniert das Ganze völlig analog. Sind die erzeugten Elemente nicht für offene und interagierende Oberflächen vorgesehen, dann lässt man die Definition der Aussenseite einfach weg. Diese Information wird dann nicht benötigt und ist folglich obsolet.

optionale Defintion von Normalen-Vektoren

Im Handbuch 'sonar Language Guide' wird im Kapitel 'Partieller Rotationskörper (Grid Segment)' als Beispiel mit einem Macro eine offene Oberfläche erzeugt und die dazu verwendeten Linien und Bogen mit entsprechenden Informationen versehen die angeben, welche Seite die Aussenseite sein soll. Während hier bei der manuellen Erzeugung der Elemente mit einem Dialog ein Mausklick genügt um die Aussenseite festzulegen, müssen in einem Macro die eigentlichen Normalen-Vektoren, welche senkrecht auf den Elementen stehen, definiert werden.

Das nächste Bild zeigt denselben Dialog mit der Element-Wahl 'Polygon':



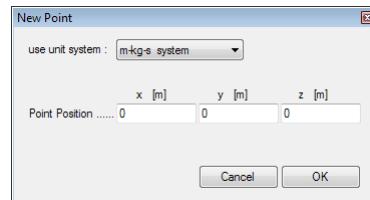
**FIGURE 61.** Der Dialog 'Create Element im Zustand (Polygon, Polyline, Quadstrip)

Links unten im Dialog ist eine kurze Anleitung eingeblendet, wie ein Polygon oder eine Polyline erzeugt werden soll.

1. Taste [Start] drücken
2. Datenpunkt eingeben und Taste [Add] betätigen
3. weitere Datenpunkte eingeben
4. Taste [End/Create] drücken

In diesem Dialog können beliebig viele Elemente erzeugt werden ohne den Dialog zu verlassen. Wenn immer Sie die Taste [Create] bzw. [End/Create] betätigen wird das betreffende Element unmittelbar erzeugt und der Dialog ist bereit für das nächste Element.

### Der Dialog New Point



**FIGURE 62.** Der Dialog 'New Point'

Anstelle eines Punkts mit dem allgemeinen Element-Dialog, wie oben abgebildet, zu erzeugen, kann dieser alternativ auch mit dem sehr einfachen Dialog 'New Point' definiert werden.

### Ein Element per Makro erzeugen

Ein Element per Makro zu erzeugen ist denkbar einfach. Der Makrotext wird mit einem einfachen Textprogramm wie dem 'Editor' geschrieben, als Textfile gespeichert und in sonar-LAB mit dem Befehl

```
macroTool / File / Open Macro
```

geladen. Im Makro Window selbst, welches sich als Tool auf der sonar-Oberfläche befindet, gibt es ein eigenes Menu mit dem Befehl

Die Befehle zur Erzeugung der Elemente lauten:

```

CREATE ELEMENT (E1, POINT, x0, y0, z0)
CREATE ELEMENT (E1, LINE, x1, y1, z1, x2, y2, z2)
CREATE ELEMENT (E1, ARC, x0,y0,z0, x1,y1,z1, x2,y2,z2, orient)
CREATE ELEMENT (E1, CIRCLE, cx, cy, cz, nx, ny, nz, R)

CREATE ELEMENT (E1, POLYGON, n)
DATA(x1,y1,z1, ..., xn,yn,zn)

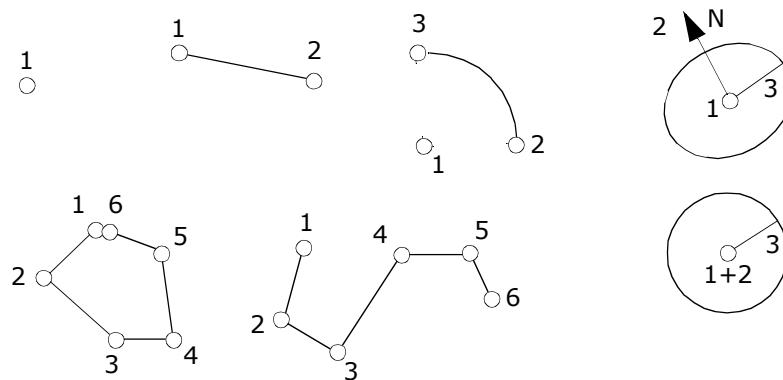
CREATE ELEMENT (E1, POLYLINE, n)
DATA(x1,y1,z1, ..., xn,yn,zn)

CREATE ELEMENT (E1, QUAD_STRIP, n)
DATA(x1,y1,z1, ..., xn,yn,zn)

```

Die drei letzten Elemente Polygon, Polyline und Quadstrip machen Gebrauch von einer nachfolgenden Datenlinie. In der eigentlichen Formel zur Erzeugung des Elementes wird mit dem letzten Parameter (n) immer die Anzahl Punkte angegeben. In der nachfolgenden Datenlinie müssen anschliessend genauso viele Zahlentripel (x,y,z) definiert werden wie Punkte deklariert wurden. Allerdings müssen diese Zahlen nicht alle in eine einzige DATA-Anweisung verpackt werden. Es dürfen der Übersicht halber mehrere DATA-Linien folgen. Typischerweise schreibt man pro Linie drei Zahlentripel in eine DATA-Anweisung.

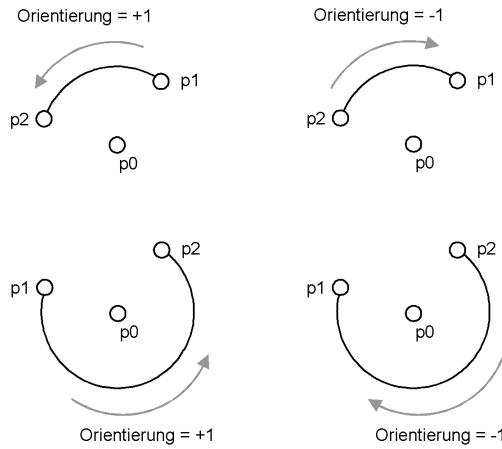
## Format / Elementpunkte



**FIGURE 63.** die Punktfolgen bei der Definition von Elementen

Mit der Eingabe von drei Punkten ( $p_0, p_1, p_2$ ) sind vorerst vier Interpretationen möglich, wie der Bogen gezeichnet und durchlaufen werden soll. Der Benutzer muss sich aber nur merken, einen Bogen konsequent im Gegenuhzeigersinn zu definieren, so wie in der ersten der vier Varianten in der folgenden Abbildung angezeigt. Die Orientierung wird in der Regel immer vom Programm selbst bestimmt, wenn eine Kette von Linien und Bogen zu einer Kontur gruppiert wird. Innerhalb einer Kontur kann es Bogen mit beiden Orientierungsrichtungen geben.

Der Bogen besitzt einen Parameter 'orient', welcher die Orientierung des Bogens nach dem folgenden Schema festlegt:



**FIGURE 64.** Vier Interpretationen eines Bogens mit denselben Punkten p1 und p2

Erwähnenswert ist schliesslich die Art und Weise wie ein Kreis definiert wird. Das Zahlentripel (cx, cy, cz) bezeichnet das Kreiszentrum während (nx, ny, nz) einen homogenen Vektor festlegt, welcher senkrecht auf der Kreisebene steht und mit seinem Fusspunkt am Kreiszentrum befestigt ist.

### **Elemente gruppieren**

Die Gruppierung von Elementen hat eine andere Bedeutung als die Gruppierung von Objekten. Die Gruppierung von Objekten hat den Zweck mehreren Objekten, welche einer gemeinsamen Gruppe angehören, gemeinsame Eigenschaften zuzuordnen. Die Gruppierung erfolgt unabhängig davon, ob die Objekte miteinander verbunden sind oder nicht.

Im Fall der Rohdaten hat die Gruppierung die Aufgabe mehrere Elemente nahtlos aneinanderzufügen. Eine Elementgruppe kann durch eine beliebige Kombination von Linien und Bogen erzeugt werden. Nur Elemente die sich mit ihren Enden nahtlos miteinander verbinden lassen, können zusammen eine Gruppe bilden. sonar lässt offene und geschlossene Gruppen zu. Geschlossene Gruppen lassen sich zu Konturen weiterverarbeiten. Folgende Funktionen stehen dem Benutzer für den eigentlichen Gruppierungsprozess von Elementen zur Verfügung:

Menu / Edit / Group / Elements / Selected  
 Menu / Edit / Group / Elements / All

Die Selektion der Elemente kann ggf. mit einer Mehrfachselektion unter Zuhilfenahme der 'shift'-Taste oder mit dem sog. Umfassungsrechteck erfolgen. Elementgruppen können wieder aufgehoben werden mit:

Menu / Edit / Ungroup / Elements / Selected  
 Menu / Edit / Ungroup / Elements / All

### **Copy, Past, Duplicate (Elemente)**

Es gibt keine Funktionen für diesen Zweck. Elemente werden immer neu erzeugt mit einer der folgenden Möglichkeiten:

- manuell interaktiv am Bildschirm mit der Toolleiste 'New Element'
- mit dem Dialog 'Menu/Create/New Element...'
- innerhalb eines Makros mit 'CREATE ELEMENT (...)'
- indirekt über einen Datenimport

## Konturen

---

Konturen bestehen aus nahtlos aneinander gefügten Linien und Bogen welche zusammen eine geschlossene Kontur bilden. Eine geschlossene Linien-Bogen-Kette ist aber noch nicht automatisch eine sog. 'Contour' im Kontext von sonar, sie muss in eine solche transformiert werden. Der Unterschied ist rein zeichnerisch nicht erkennbar, aber die dahinter liegende Datenstruktur ist eine andere. Eine 'Contour' enthält nebst den Informationen zu den einzelnen Elementen noch weitergehende Daten welche die Beziehung, die Orientierung, die Reihenfolge, usw. der einzelnen Elemente in der Kontur beschreiben. Im sog. Object Tool erscheint eine 'Contour' folglich auch als eigentliches neues 'Element' in einem eigenen 'Contour-Folder'. Konturen bilden in dieser Form den Ausgangspunkt für extrudierte Linien-Bogen-Konturen welche bereits in einem eigenen Kapitel beschrieben wurden. Im Dialog 'Create/Show Contour' können Konturen, falls sich das als notwendig erweisen sollte, wieder in ihre elementaren und ursprünglichen Linien und Bogen zurückverwandelt werden. Ein Doppelklick auf eine 'Contour' im 'Object Tool' öffnet in diesem Fall die Kontur mit diesem Dialog.

## Mögliche Probleme mit Konturen

---

Wann immer eine Kette von Linien und Bogen zu einer geschlossenen Kontur weiterverarbeitet werden soll, kann es vorkommen, dass dieser Prozess nicht einwandfrei funktioniert, weil irgendwo in der Kette der Elemente der Anschluss an das nächste Element nicht einwandfrei gewährleistet ist. sonar benutzt für die Erkennung ob zwei Elemente nahtlos aneinander passen einen Toleranzwert. Beträgt die Distanz zwischen zwei Endpunkten von zwei Elementen weniger als die Toleranz, dann sprechen wir von einer regulären Fortsetzung innerhalb der Linien-Bogen Kontur. Andernfalls wird eine Verbindung der beiden Elemente abgelehnt und das Resultat ist keine geschlossene Kontur.

sonar hat Funktionen, gewisse Mängel in Konturen automatisch zu beheben:

```
Menu / Sketch / Remove / All identical Elements  
Menu / Sketch / Remove / All wrongly defined Elements  
Menu / Edit / Preferences... / Constants, Tolerances / group tol.
```

Besonders beim Daten Import über die DXF Schnittstelle kommt es häufig vor, dass eine grössere Anzahl von Linien oder Bogen mit identischen Endpunkten mehrfach vorhanden sind. Dies kommt nicht zuletzt von daher, dass DXF Files aus 2D-Ansichten von 3D-Modellen abgeleitet werden wo in der Projektion viele Kanten zusammen. Oft ist sich der Benutzer dessen nicht bewusst, weil diese Zeichenelemente meistens nicht sichtbar sind und auch nicht auffallen. Wenn die sonar-Funktion, welche sich bemüht eine geschlossene Kontur zu bilden, versucht, Elemente nahtlos aneinander zu reihen und zu verbinden, dann stören doppelte Elemente enorm. An Endpunkten mit doppelten Elementen hat die Funktion immer zwei Möglichkeiten eine Fortsetzung der Linien-Bogen-Kontur zu finden. Nimmt sie den falschen Weg, dann landet sie meistens in einer Sackgasse.

### Identical elements

Aus dem gleichen Grund, weshalb beim Erzeugen von DXF Files doppelte Elemente entstehen, werden auch solche mit einer Null-Länge gebildet. Linien bzw. Kanten, welche in einer 2D-Ansicht eines 3D-Modells genau in der Projektion, also senkrecht zum Bildschirmebene liegen, werden auf einen Punkt reduziert und haben zweidimensional identische Endpunkte.

### wrongly defined Elements

Die Toleranz für die Gruppierung von Elementen einfach möglichst gross zu wählen, ist keine gute Idee. Ist der Toleranzwert grösser als das kleinste Element in einer Elementkette, dann werden beim Aneinanderfügen der Elemente eventuell Linien oder Bogen übersprungen. Der Toleranzwert sollte also auf die minimale Länge der Elemente abgestimmt werden und immer kleiner sein als die Elementlänge.

### group tolerance

## Vorgehen

Wann immer beim Gruppieren oder beim Bilden einer Kontur das System ein Versagen der Gruppierung von Elementen meldet, wird folgendes Vorgehen empfohlen:

```
Menu / Edit / Ungroup / Elements / All ... Selected  
Menu / Sketch / Remove / All identical Elements  
Menu / Sketch / Remove / All wrongly defined Elements  
Menu / Edit / Group / Elements / All ... Selected  
Menu / Create / New Contour...
```

Wenn der Gruppierungsprozess weiterhin versagt, dann sollte versucht werden den Toleranzwert zu ändern

```
Menu / Edit / Preferences.../ Constants, Tolerances / group tol.
```

## Selektion von Elementgruppen und Konturen

Beim Abfahren einer Gruppe von Elementen, einer Elementgruppe oder einer Kontur mit der Maus wird der Cursor ständig angepasst, je nachdem, was sich gerade unter der Cursorspitze befindet (Linie, Bogen, Endpunkt, Mittelpunkt, usw.) Besteht eine Folge von Elementen aus sehr vielen kleinen und nahe beieinander liegenden Strecken- oder Bogenelementen, dann wird praktisch nur noch der Endpoint-Cursor angezeigt und es wird fast unmöglich ein einzelnes Element oder die Gruppe als Ganzes zu selektieren. In diesem Fall hilft nur ein Hinein-Zoomen, um mehr Platz zu schaffen und auf diesem Weg Zugang auf die Bereiche zwischen den Endpunkten zu bekommen.

# Daten-Import

---

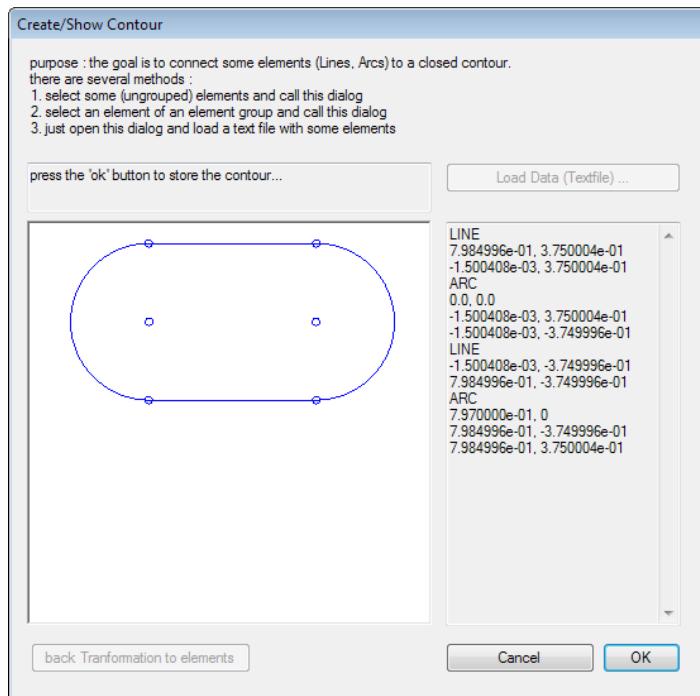
Der Datenimport stellt eine minimale Anbindung zum Datenaustausch mit externen Programmen sicher. In sonar existieren die folgenden Möglichkeiten externe Daten in das System einzubringen:

- Rohdaten Import einer Kontur (Textfile)
- Rohdaten Import einer 3D-Polyline
- Rohdaten Import Grid
- Rohdaten Import im DXF Format (Austauschformat mit CAD System)
- sonar Modelle importieren mit Merge (sonar-Einzelteile, -Baugruppen und -Modelle zusammenführen)

## Import Kontur (Textfile)

---

Das Vorgehen wie Konturen geladen und weiterverarbeitet werden, wurde bereits im Kapitel 'Extrudierte Linien Bogen Konturen' besprochen. Die nächste Abbildung zeigt auf der rechten Hälfte nochmals die Struktur einer solchen Datei, wie sie im betreffenden Dialog eingeblendet wird. Ein Textfile dieser Art kann mit EXCEL und Speicherung im TEXT Format oder mit jedem anderen Textprogramm erzeugt werden.

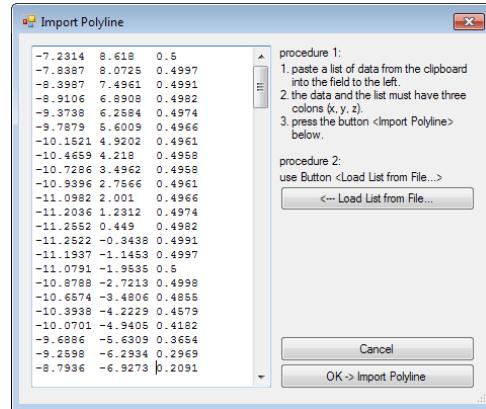


**FIGURE 65.** Dialog Create/Show Contour

## Import 3D-Polyline

---

3D-Polylines werden in gewissen Fällen benötigt um den 3-dimensionalen Pfad einer Extrusion bzw. 'Sweep'-Funktion, wie man sie in der CAD-Welt nennt, festzulegen.

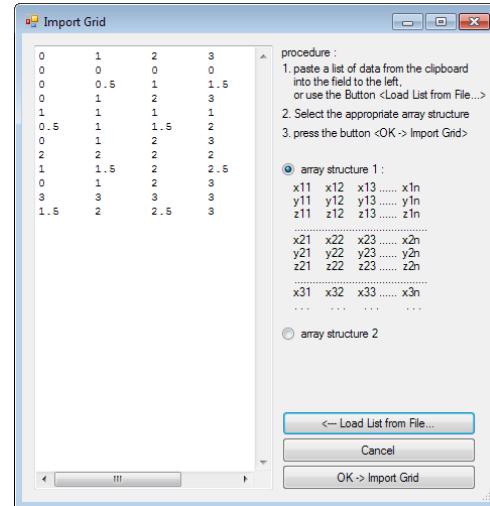


**FIGURE 66.** Der Dialog Import Polyline

Der Dialog wie abgebildet lässt zwei Vorgehensarten zu. Die beiden Methoden sind im Dialog rechts oben angeschrieben:

1. Per 'copy & paste' aus dem EXCEL- oder einem anderen Textprogramm. Man platziert den Cursor in der Tabelle und setzt den Inhalt der Zwischenablage ein. Man hat lediglich darauf zu achten, dass die Liste aus drei mit TAB's getrennten Kolonnen besteht welche auf jeder Linie einen 3D-Punkt (x,y,z) darstellt.
2. Man lädt mit der Taste [Load List from File] ein ebensolches Textfile mit dieser Struktur bestehend aus drei Datenreihen.

## Import Grid



**FIGURE 67.** Der Dialog Import Grid

Im Kapitel Netzstrukturen (Grids) wurde die Verwendung einer Netzstruktur bereits erklärt. Netzstrukturen kann man sich als dreidimensional deformierte Oberflächen oder als Freiformflächen vorstellen. In sonar tragen diese Gebilde den Namen 'Grid' und haben einen eigenen Ordner im 'Object Tool'. Die Definition von Grids kann in der aktuellen Implementierung

nur über extern definierte Netzstrukturen erfolgen. EXCEL ist eine gute Wahl zur Erzeugung dieser Netze. Sehen Sie dazu das Kapitel 'Netzstrukturen (Grids)'

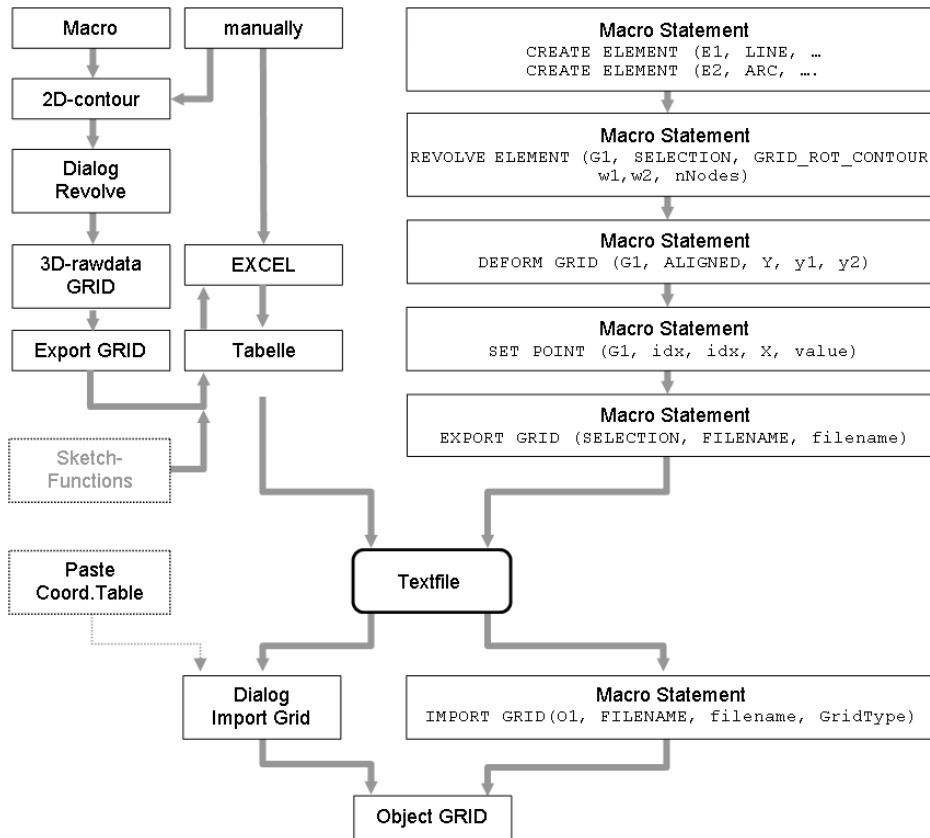


FIGURE 68. Wege zum 'Grid'

## Import DXF File

DXF steht für 'Data eXchange Format' und ist eines der am meisten verbreiteten Austauschformate im CAD Bereich. DXF kann 2D- und 3D-Daten beschreiben, ist einfach im Aufbau und kann mit jedem Textprogramm editiert oder manipuliert werden.

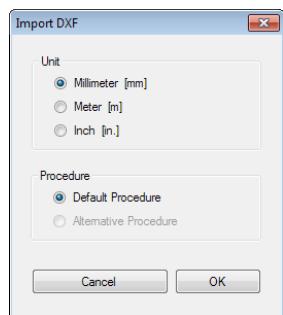


FIGURE 69. Der Dialog 'Import DXF'

Der abgebildete Dialog öffnet sich mittels:

Menu / Import/Export / Import DXF...

Wenn die Taste 'OK' gedrückt wird, öffnet sich der Standard File Dialog zum Öffnen eines Files mit möglichen Dateien im Fileformat '.dxf'. Der in sonar implementierte DXF-Interpreter ist allerdings sehr wählerisch und importiert nur Elemente des Typs die in sonar direkt weiterverarbeitet werden können. In der Regel ist das allerdings auch das was der Benutzer erwartet, denn alles andere ist letztlich nur Ballast der ohnehin wieder entfernt werden müsste. In allen zweidimensionalen Daten werden die Z-Koordinaten auf null gesetzt. Die importierten Elemente befinden sich also in der X-Y-Ebene (Front View).

Wenn der Import abgeschlossen ist, kann der Benutzer all das was er nicht braucht selektieren und löschen und die restlichen Elemente in der üblichen Form weiterverarbeiten.

Menu / Sketch / Remove / All identical Elements  
Menu / Sketch / Remove / All wrongly defined Elements

## **Merge (sonar Files zusammenführen)**

---

Wir befinden uns im Programm sonar\_LAB und haben ein geöffnetes Modell vor uns auf dem Bildschirm. Im File-Menu finden wir die drei Funktionen

Menu / File / New Model  
Menu / File / Open Model  
Menu / File / Merge Model

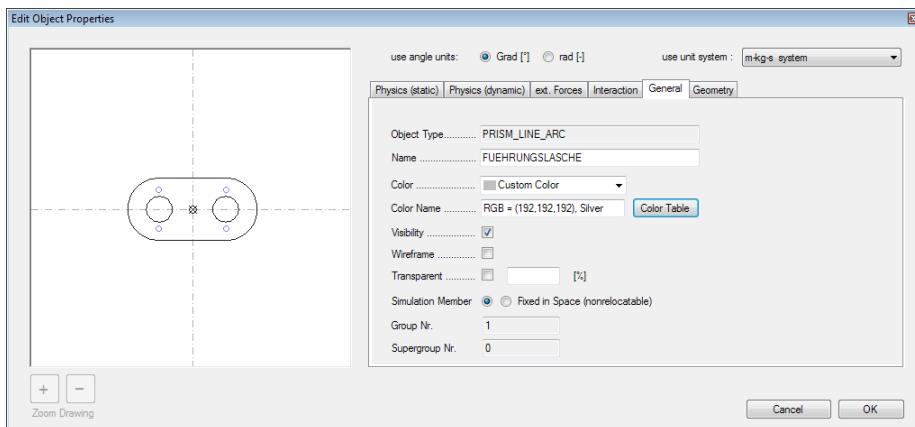
Mit 'Merge Model' öffnen wir ein Model in der gleichen Art wie mit 'Open Model', mit dem Unterschied, dass das offene Modell auf dem Bildschirm nicht gelöscht oder überschrieben wird. Das zu öffnende Modell wird stattdessen an das bereits offene Modell angefügt. Wir haben damit die Summe der beiden Files in unserem Modell. Es ist offensichtlich, dass dies ein starkes Instrument ist um Modellbausteine zusammenzufügen und um Bibliotheks-Teile in ein Modell einzusetzen. Diese Funktion fügt im eigentlichen Sinn Module zusammen. Sehr oft wird diese Möglichkeit auch dazu benutzt um in komplexen Modellen einzelne Baugruppen einzeln aufzusetzen, zu modellieren und zu editieren, nicht zuletzt auch deshalb, weil die Software mit einfacheren Modellen eine höhere Performance hat. Das Zusammenführen der einzelnen Modellteile geschieht dann erst vor der eigentlichen Simulation indem dann mit der Merge Funktion ein Gesamtmodell erzeugt wird. Mit diesem Vorgehen lassen sich auch sehr einfach Varianten und Abwandlungen von Gesamtmodellen erzeugen.

# Objekte Modifizieren

## Objekt-Eigenschaften ändern

Die erste Anlaufstelle zur Veränderung der Eigenschaften eines einzelnen Objektes ist der Dialog 'Edit Object Properties'

```
select Object / Menu / Edit / Edit Object / Properties...  
...oder  
select object / right click on object / context menu / Edit /  
Properties..
```



**FIGURE 70.** Der Dialog Edit Object Properties

Die Daten sind in diesem Dialog auf mehrere Panels verteilt, welche durch die Wahl eines Reiters selektiert werden können.

- statische physikalische Daten
- dynamische physikalische Daten
- externe Kräfte
- Interaktions-Eigenschaften
- Allgemeine Daten (in der Abbildung oben dargestellt)

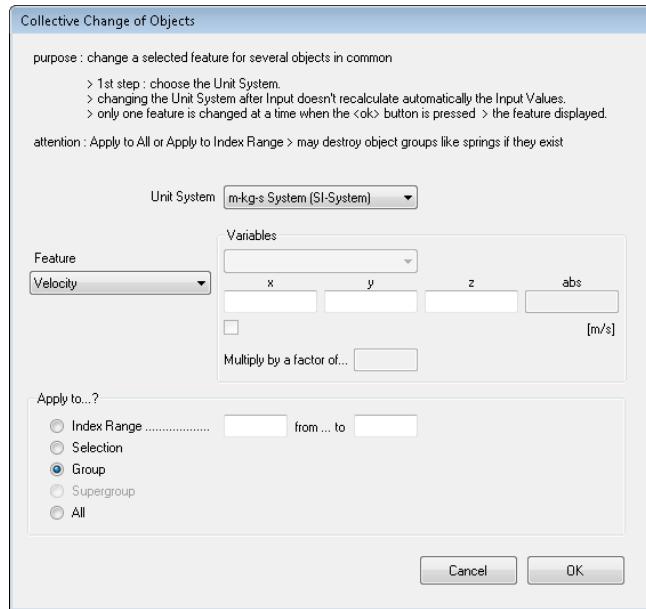
Sollen gewisse Eigenschaften nicht nur an einem einzelnen sondern an einer Selektion oder einer Gruppe von Objekten verändert werden, dann öffnet der Benutzer stattdessen den Dialog 'Collective Change'

```
select Object / Menu / Edit / Collective Change / Objects...
```

In diesem Dialog kann der Benutzer nicht nur neue (kollektive) Eigenschaften eingeben, sondern hat auch viele Möglichkeiten festzulegen welche Objekte diese neuen Eigenschaften erhalten sollen. Das selektierte Objekt dient ihm dabei als Repräsentant einer Gruppe von Objekten, an dem sich der Dialog ggf. weiter orientiert.

## 1 Objekt ändern

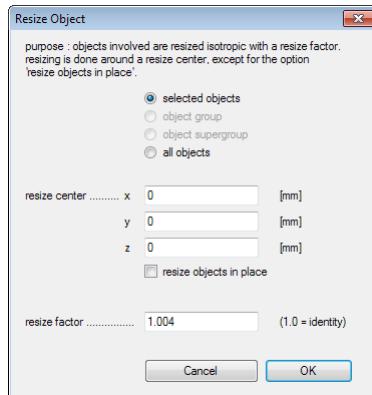
## Eine Gruppe von Objekten ändern



**FIGURE 71.** Der Dialog 'Collective Change of Objects'

Es sind sogar Kombinationen möglich von der Art: "Ändere die Dichte aller Objekte welche zur gleichen Gruppe gehören wie das selektierte Objekt oder den gleichen Namen oder die gleiche Farbe haben", usw.

## Objektgrösse ändern



**FIGURE 72.** Der Dialog 'Resize Object'

Diese Funktion skaliert die Objektgrösse eines einzelnen Objektes, einer Gruppe von Objekten oder eines ganzen Modells. Der Zugriff erfolgt durch eine der beiden Methoden:

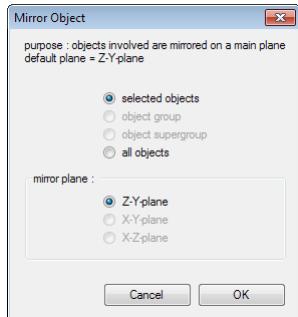
```
select object / Menu / Edit / Edit Object... / Resize...
select object / right click on object / context menu / Edit / Resize...
```

die Option 'resize objects in place' ist vor allem für einzelne und ungebundene Objekte gedacht. Wenn eine ganze Gruppe von Objekten skaliert wird, dann ist diese Option nur statthaft, wenn zwischen den Objekten keine Links existieren. In der Regel erfolgt eine Skalierung einer Gruppe von Objekten allerdings ohne diese Einschränkung. Eine Baugruppe oder ein ganzes Modell wird

skaliert, indem alle Koordinaten sämtlicher Objekte mit einem Skalierungsfaktor multipliziert werden, was einer massstäblichen Vergrößerung oder Verkleinerung des betreffenden Modellteils gleichkommt. Dabei trägt das Programm dafür Sorge, dass auch sämtliche betroffenen Link-Positionen neu berechnet werden.

## Objekte Spiegeln

---



**FIGURE 73.** Der Dialog 'Mirror Object'

```
select object / Menu / Edit / Edit Object... / Mirror...
select object / right click on object / context menu / Edit /
Mirror...
```

Eine Spiegelung ist in sonar immer eine Spiegelung an einer Hauptebene. Es gibt keine Punktspiegelungen und keine Spiegelungen an schiefen Ebenen. Wie der Dialog 'Resize' kann auch dieser Dialog auf einzelne Objekte, Objektgruppen und ganze Modelle angewendet werden. Links zwischen den Objekten werden mitgespiegelt.

## Extruded Line-Arc-Prism editieren

---

Extrudierte Linien-Bogen Konturen besitzen einen speziellen Dialog für den Editierprozess. In einem früheren Kapitel haben wir bereits ausführlich über diesen Dialog und seine Anwendung gesprochen.

```
select object / right click on object / context menu / Edit /
Prism Contour...
```

Dieser Dialog ist nicht dazu gedacht, die Kontur eines Objektes im grösseren Stil oder komplett zu verändern. Vielmehr richtet sich dieser Dialog an den Benutzer, welcher sog. Toleranzfehler und Korrekturen dieser Größenordnung einbringen will. Der Dialog erlaubt es, alle Linien und Bogen einer bestehenden Kontur gezielt zu verschieben um damit die Berandung geringfügig zu verändern. Wird im Dialog ein Element verschoben, dann kümmert sich die Funktion auch darum, dass die Anschlüsse an die benachbarten Elemente jeweils automatisch wieder hergestellt werden.

Verbindungen zu anderen Objekten werden mit diesem Dialog nicht tangiert. Falls es der Benutzer als notwendig erachtet infolge dieser Formänderungen gewisse Links zu verschieben, dann muss er diese manuell löschen und neu setzen.

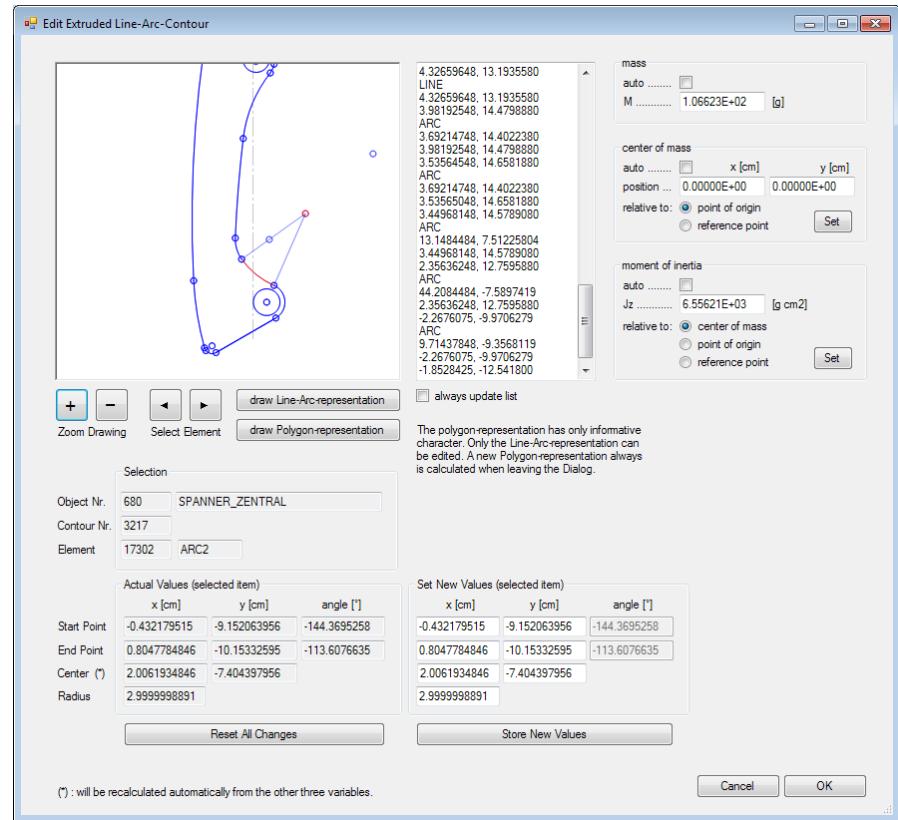


FIGURE 74. Der Dialog 'Edit Extruded Line-Arc-Contour'

# Objekte Drehen & Positionieren

---

Wir reden in diesem Kapitel über das Editieren von Objekten in Zusammenhang mit ihrer Position und Drehlage in einem Modell. Wir befinden uns also im Programm sonar\_LAB. Die Editiermöglichkeiten in sonar\_SIM sind diesbezüglich eingeschränkt.

## Positionieren

---

Interaktive translatorische Bewegungen mit der Maus sind in sonar\_LAB in jeder Ansicht möglich. Der Benutzer greift mit der Maus eine Berandung eines Objektes und verschiebt es an eine andere Position. Ist der angepackte Primitivkörper mit anderen Objekten verbunden, dann wird die gesamte verbundene Kette von Objekten simultan verschoben. Ist eines der Objekte in der verbundenen Kette von Objekten an einem Fixpunkt befestigt, dann verweigert das Programm die Verschiebung. Objekte, in welchen die Eigenschaft "fixed in space" gesetzt ist, dürfen allerdings verschoben werden, sie sind im weiteren Verlauf einfach am Zielort weiterhin unbeweglich verankert.

Interaktiv

Im Weiteren gibt es eine Funktion für die translatorische Bewegung einer Selektion von Objekten wie folgt:

Menu / Dialog

Menu / Edit / Shift (Selection)...

Drückt der Benutzer über einer Objekt-Berandung die rechte Maustaste, dann öffnet sich das Kontextmenu. Mit dem folgenden Kontext-Menu-Befehl ist der Benutzer in der Lage, die Verschiebung eines einzelnen Objektes (Primitive) numerisch einzugeben.

Kontext Menu / Dialog

Kontext Menu / Edit / Rotation and Position...

Im folgenden ein paar Makro-Befehle, wie sie zum Zwecke der Verschiebung von Objekten eingesetzt werden.

Makro

```
TRANSLATE OBJECT (O1 || SELECTION, ABSOLUTE || RELATIVE, dx,dy,dz)  
  
TRANSLATE OBJECTGROUP (O1, ABSOLUTE || RELATIVE, dx,dy,dz)  
TRANSLATE OBJECTGROUP (LAST_GROUP_NR, ABSOLUTE || RELATIVE,  
                      dx,dy,dz)  
TRANSLATE OBJECTSUPERGROUP (O1, ABSOLUTE || RELATIVE, dx,dy,dz)  
TRANSLATE OBJECTSUPERGROUP (LAST_SUPERGROUP_NR,  
                           ABSOLUTE || RELATIVE, dx,dy,dz)
```

O1 : die Objektreferenz eines Objektes einer Objektgruppe oder Objektsupergruppe

LAST\_GROUP\_NR : die letzte erzeugte Objektgruppe

LAST\_SUPERGROUP\_NR : die letzte erzeugte Objektsupergruppe

dx,dy,dz : Verschiebungsvektor im [cm-g-μs]-System

|| : bedeutet 'oder'

## Drehen

---

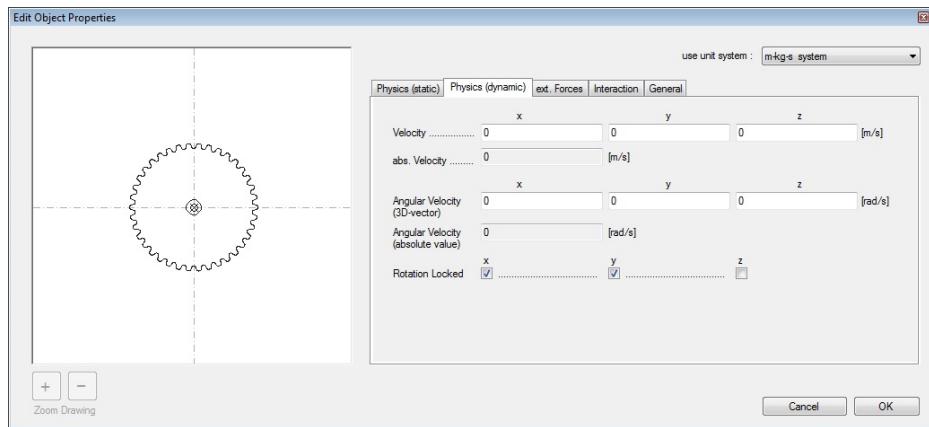
|                        |  |
|------------------------|--|
| Interaktiv             | Einfache Primitivkörper lassen sich direkt grafisch interaktiv in einer Ansicht drehen. Wird ein Primitivkörper am Bildschirm durch Anklicken selektiert, dann wird er in der Farbe 'magenta' dargestellt und mit verschiedenen Ankerpunkten ausgestattet, welche unter Anderen auch eine Drehung mit Hilfe der Maus zulassen.   |
| Menu / Dialog          | Auch für die Rotation einer Selektion von Objekten steht ein Dialog zur Verfügung. Der Benutzer kann in diesem Dialog den Drehpunkt frei wählen<br><br>Menu / Edit / Rotate (Selection)...   |
| Kontext Menu / Dialoge | Mit Hilfe des sog. Kontext Menus bekommen wir Zugang zu den gleichen, bereits erwähnten, Dialogen<br><br>Context Menu / Edit / Rotate...<br>Context menu / Edit / Rotation and Position...   |
| Makro                  | und wie beim Positionieren auch hier ein paar Makros für Drehungen<br><br>ROTATE OBJECT (O1, cx, cy, cz, wx, wy, wz)<br>ROTATE OBJECT (SELECTION, cx, cy, cz, wx, wy, wz)<br><br>ROTATE OBJECTGROUP(O1, cx, cy, cz, wx, wy, wz)<br>ROTATE OBJECTGROUP(LAST_GROUP_NR, cx, cy, cz, wx, wy, wz)<br><br>ROTATE OBJECTSUPERGROUP(O1, cx, cy, cz, wx, wy, wz)<br>ROTATE OBJECTSUPERGROUP(LAST_SUPERGROUP_NR, cx, cy, cz, wx, wy, wz)<br><br>O1 : die Objektreferenz eines Objektes einer Objektgruppe oder Objektsupergruppe<br>LAST_GROUP_NR : die letzte erzeugte Objektgruppe<br>LAST_SUPERGROUP_NR : die letzte erzeugte Objektsupergruppe<br>cx,cy,cz : Drehzentrum im [cm-g-μs]-System<br>wx,wy,wz : Drehwinkel um die einzelnen Achsen in [rad] |

Die gleichzeitige Drehung um mehrere Achsen (wx, wy, wz) in ein- und derselben Anweisung gehorcht bezüglich der Reihenfolge einem gewissen Schema bzw. einer bestimmten Abfolge (Y->X->Z). Damit eine zusammengesetzte Drehung um mehrere Achsen in der Reihenfolge den Erwartungen des Benutzers entspricht, ist es eine gute Idee, Drehungen um mehrere Achsen immer in mehrere Anweisungen aufzulösen, so dass jede Anweisung immer nur eine Drehung um eine bestimmte Achse ausführt. Das folgende Beispiel dreht das Objekt mit der Nummer 17 zuerst um -90° um die Z-Achse und anschliessend weiter um 30° um die X-Achse (Drehzentrum = Ursprung)

|          |   |
|----------|---|
| Beispiel | ROTATE OBJECTGROUP(O17, 0, 0, 0, 0, 0, -1.570796)<br>ROTATE OBJECTGROUP(O17, 0, 0, 0, 0.523598, 0, 0) |
|----------|---|

# Die Bewegung der Objekte in Simulationen einschränken

## Rotationsachsen einfrieren



**FIGURE 75.** Dialog 'Edit Object Properties'

Im Dialog 'Edit Object Properties' auf den wir nach einer Selektion eines Objektes zugreifen mit

Menu / Edit / Edit Object / Properties...

sehen wir im Panel 'Physics (dynamic)' drei Checkboxes welche mit 'Rotation Locked' angeschrieben sind. An dieser Stelle kann der Benutzer jede einzelne Rotationsachse eines Objektes einfrieren bzw. blockieren. Dies hat zur Folge, dass die betreffende Achse keine Drehungen mehr durchführt. Als Beispiel betrachten wir das Zahnrad wie im Dialog abgebildet. Durch das Setzen der beiden Checkboxes 'X' und 'Y' kann das Zahnrad in der Simulation nur noch um seine eigentliche Drehachse ('Z'-Achse) drehen und ist immun auf sogenannte Kippmomente welche versuchen das Rad in einer der anderen beiden Achsen 'X' und 'Y' schief zu stellen. Blockierte Achsen werden bei der Neuberechnung der Winkelgeschwindigkeiten und der Winkel einfach übergangen.

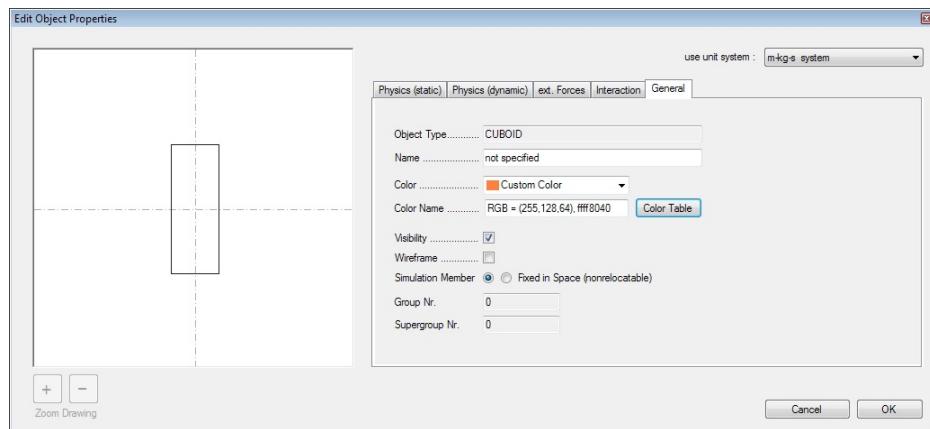
Es ist wichtig sich zu merken, dass die Blockierung dieser Drehungen sich nicht auf die aktuelle Drehlage der Objekte im Raum bezieht, sondern auf die sog. Nulllage in der das Objekt definiert wurde. Bei Rotationsobjekten ist die Drehachse in diesem Sinn immer die z-Achse, denn der Rotationskörper wurde ursprünglich in Richtung der Z-Achse extrudiert, auch wenn er unmittelbar danach im Modell in eine andere Stellung und Position überführt wurde. Die drei Achsen X, Y, Z beziehen sich im Dialog 'Edit Object Properties' also immer auf das lokale Koordinatensystem des Objektes.

Der Benutzer kann wahlweise eine, zwei oder alle drei Achsen einfrieren. Die translatorische Bewegungsfreiheit wird durch diese Einschränkungen nicht tangiert.

## Alle Bewegungen einfrieren

Ein Objekt kann bezüglich seinen Bewegungen absolut fixiert werden indem im Dialog 'Edit Object Preferences' die Einstellung 'Fixed in Space (non relocatable)' aktiviert wird. Dies hat zur Folge, dass sowohl die translatorischen wie auch die Rotations-Bewegungen unterbunden werden. Das

Objekt nimmt aber weiterhin an den Interaktionen teil und seine Interaktionseigenschaften bezüglich Oberflächenhärte und Stossabsorption bleiben weiterhin bestehen.



**FIGURE 76.** Der Dialog 'Edit Object Properties'

Benötigt wird diese Eigenschaft oft dort wo Modelle einen festen Untergrund oder ein Fundament haben an dem der Rest des Modells befestigt wird. An den Befestigungspunkten zwischen Modell und Fundament kann z.B. auch der Kraftverlauf gemessen werden, welche eine laufende Maschine auf das Fundament ausübt. Raumfest verankerte Objekte sind in der Regel Impuls- und Energiesenken, denn die Stossanteile die bei Interaktionen auf diese Objekte ausgeübt werden, haben (im fixierten Objekt) keine Impuls- und Energie-Änderungen zu Folge.

Alternativ könnte man ein Objekt stattdessen auch an Fixpunkten befestigen. Dies hätte zur Folge, dass die Kräfte bei Interaktionen vorerst auf das befestigte Objekt übertragen würden und dieses im Rahmen der elastischen Verbindungen zu den Fixpunkten auch noch kleine Auslenkungen durchführen könnte. Am Ende würden die besprochenen Impuls- und Energieanteile aber genauso abgeleitet, diesmal einfach an den Fixpunkten selbst. Wenn man aber an der Messung der Kräfte auf das fixierte Objekt interessiert ist, dann ist dieses Vorgehen manchmal doch geeigneter als das Objekt einfach als raumfest zu deklarieren.

## Bewegung Stoppen

Manchmal möchte man die Geschwindigkeit und die Winkelgeschwindigkeit eines Objektes oder sämtlicher Objekte des Modells anhalten bzw. auf null zurücksetzen. Für beide Fälle gibt es in sonar-SIM eine Funktion

Menu / Edit / Stop Movement of / all Objects  
 Menu / Edit / Stop Movement of / selected Objects

Diese Rückstellung der Bewegungen ist natürlich jeweils einmalig und nicht anhaltend. Im weiteren Verlauf einer Simulation können sich die betroffenen Objekte erneut in Bewegung setzen. Sie haben einfach einen Zwischenstopp gemacht und werden dann ggf. erneut beschleunigt.

## Modell Beruhigen

Eine nützliche Funktion, welche das Zurücksetzen der Bewegung sämtlicher Objekte in periodischen Zeitintervallen wiederholt, ist die Funktion 'Calm'. Damit lässt sich ein Modell im eigentlichen Sinn 'beruhigen', falls unerwünschte Interaktionen, aus welchem Grund auch immer, für eine gewisse Unruhe gesorgt haben. Durch das periodische Zurücksetzen sämtlicher Bewegungen werden diese gedämpft und kommen allmählich zur Ruhe, vorausgesetzt, der Benutzer setzt das Zeitintervall dieses Vorgangs in einem angemessenen Rahmen. Ein guter Wert

für dieses Zeitintervall ist eine Viertelperiode der Schwingung welche man dämpfen will. Das Öffnen einer Grafik, bevor dieser Vorgang eingeleitet wird, kann helfen den richtigen Wert zu finden und den gesamten Dämpfungsvorgang zu überwachen.

# Objekte verbinden (Links)

---

## Einführung

---

Objekte werden mit sog. Links verbunden. Links sind im Prinzip und etwas vereinfacht ausgedrückt Feder-Dämpfungs-Elemente. Links können aber noch mehr, sie werden eingesetzt um die Funktionen der Materialmodelle umzusetzen. Links haben folglich ein elastisch-plastisches Verhalten, falls man diese Eigenschaften aktiviert. Links' können bei Überlast brechen oder Alarm schlagen um kundzutun, dass sie sich in einem kritischen Zustand befinden.

Links sind also elastische Verbindungen zwischen den Objekten. Folglich können sich Objekte, die auf diese Weise untereinander verbunden sind, relativ zueinander etwas bewegen. Zumindes so viel, wie die Feder- bzw. Link-Konstante dies zulässt. Im Idealfall stellt man diese Link-Konstanten also so ein, dass die relative Bewegung unter den einzelnen Objekten dem entspricht, was sich die reale Struktur unter den gleichen Voraussetzungen elastisch verbiegen würde.

Links sind aber nicht nur elastische Verbindungen, sondern auch Gelenke. Ein einzelner Link zwischen zwei Objekten ist für sich allein genommen äquivalent einem Kugelgelenk. Mit zwei Links erzeugen wir eine Drehachse zwischen den Objekten. Erst mit drei Links die nicht auf einer Linie liegen, erreichen wir eine feste Verbindung. Da alle diese Link-Verbindungen letztlich elastisch sind, kann an ihnen kontinuierlich die Auslenkung bzw. die Kraft abgelesen werden. Wir wissen immer darüber Bescheid, wie gross die Kräfte in den einzelnen Links oder in Kombinationen von Links sind. Es sind ja letztlich wirklich die Links, welche die Objekte zusammenhalten, indem diese bei Auslenkungen jeder Art diametral entgegengesetzte Kräfte auf die entsprechenden Objektpaarungen ausüben und diese wieder zusammenziehen. Ein sonar-Modell ist so gesehen nichts anderes als ein grosses komplexes Feder-Masse-System.

Wir unterscheiden in sonar unterschiedliche Link-Typen und verschiedene Link-Kombinationen. Auf die einzelnen Links bezogen gibt es drei Grund-Typen

- (N) Normaler Link
- (B) Biege-Link
- (T) Torsions-Link

## Normaler Link

---

Einen normalen Link kann man sich als Zug- oder Druckfeder vorstellen. Im unbelasteten Zustand hat die Feder bzw. der Link allerdings die Länge Null. Die beiden Federenden fallen in einem Punkt zusammen. Unter Last dehnt sich der Link aus und bekommt eine endliche Länge. Die Kraft im Link berechnet sich als Funktion der Ausdehnung wie bei einer Schraubenfeder. Im einfachsten Fall ist die Link-Kraft eine homogene lineare Funktion, im Allgemeinen die Summe eines linearen und eines quadratischen Terms. Da die Ruhelänge eines normalen Links gleich Null ist, kann der Link unter einer äusseren Last in alle Raumrichtungen wirken. In der Tat werden zwei Objekte in einem grösseren Verbund auch chaotisch um das gemeinsame Link-Zentrum herumpendeln. Die Link-Kraft  $F'$  für Link Nr. 'K' berechnet sich im Normalfall und im Hook'schen Bereich eines Materialmodells für jede Raumrichtung als:

$$F_k = C_{1,k} \cdot r_{k,i} + C_{2,k} \cdot r_{k,i}^2 + D_{k,i}(r_{k,i}, v_{k,i}) \quad (\text{EQ 9})$$



F: Kraft

K: Link Index

$C_1, C_2$ : lineare und quadratische Link-Konstante

r: Link Auslenkung

D: Dämpfungs-Term

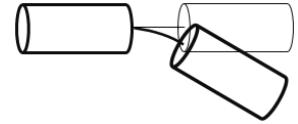
v: relative radiale Geschwindigkeit zwischen den beiden Link-Enden

i: Raumrichtung {x, y, z}

Sobald ein Materialmodell den Hook'schen Bereich verlässt kommen je nach Modell weitere Terme hinzu, welche die Spannung kontinuierlich auf die betreffende Spannungs-Dehnungskurve zurücksetzen und damit plastische Arbeit leisten.

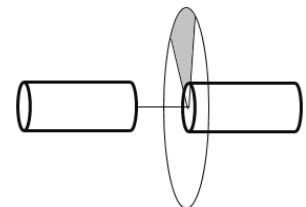
## Biege-Link

Verbindet man zwei Zylinder fluchtend, so dass sie zusammen einen doppelt so langen Zylinder bilden, dann kann der eine Zylinder festgehalten und der andere heruntergebogen werden, so dass die beiden Zylinderachsen nachher schief zueinander stehen. Ein Biege-Link an der gemeinsamen Kontaktfläche der beiden Zylinder wirkt dieser Verbiegung entgegen. Ein Biege-Link wirkt so gesehen wie eine Dreh- bzw. Schenkelfeder deren Drahtenden in die Richtung der beiden Zylinderachsen schauen. In der Einbaulage kann ein Biege-Link bereits eine Auslenkung haben. Man definiert diesen Winkel im entsprechenden Link-Dialog.



## Torsions-Link

Wenn wir uns wieder zwei Zylinder vorstellen die fluchtend aneinander stoßen, dann wirkt ein Torsions-Link wie eine Torsionsfeder deren Windungen auf den einen Zylinder geschoben werden und die mit einem Schenkel auf den zweiten Zylinder wirkt. Einem Torsions-Link kann in der definierten Lage eine Auslenkung zugewiesen werden, so dass in der Einbaulage bereits ein bestimmtes Moment wirkt.



## Link-Kombinationen

Ein Biege- oder Torsions-Link kann für sich allein nicht existieren. Beide Links produzieren in ihrer Eigenschaft nur Drehmomente auf die beiden verbundenen Objekte. Es ist aber niemand da, der dafür sorgt, dass die beiden Objekte am Ort des Links auch wirklich zusammengehalten werden. Nur ein normaler Link kann das. Deshalb werden der Biege- und Torsions-Link immer als zusätzliche Links an bestimmten Drehachsen eingebaut. Die normalen Links sind die Scharniere und der Biege-Link ist die Drehfeder im Scharnier. Link Kombinationen sind deshalb üblich und der Normalfall in der sonar-Software. Folgende Link-Kombinationen liegen als 'Link-Paket' vor:

- NBT Link
- NNNN Link bzw. CTR120 Link
- NNN Link bzw. EXCL120 Link
- NNB Link

Die sonar-Software kennt spezielle Link-Kombinationen welche im Verbund arbeiten. Die Kurzbezeichnungen geben bereits an, wie diese Kombinationen zusammengesetzt sind. So bedeutet die Bezeichnung CTR120-Link dass der Link aus einem zentralen und drei peripheren Links besteht welche im Winkel von 3 mal 120° und einem bestimmten radialen Abstand um den Zentrallink herum angeordnet sind. Im Prinzip werden diese Link-Kombinationen nicht deshalb eingeführt, weil damit neue Berechnungen möglich werden. Alle Link-Kombinationen bestehen letztlich aus den elementaren drei Link-Typen (N), (B) und (T). Der Vorteil dieser Kombinationen liegt mehr im einfacheren Erzeugen und Editieren von Modelleigenschaften. Der Benutzer ist oft vor die Aufgabe gestellt die Eigenschaften einer Objektverbindung bezüglich Streckung, Biegung und Torsion sinnvoll zu kombinieren. Dazu gibt es nicht nur die drei Basis-Links welche genau diese drei Aufgaben erfüllen. Man kann oft dasselbe Ziel mit einer anderen Kombination von Links erreichen, welche für eine bestimmte Aufgabe manchmal besser geeignet ist. So sind z.B. Drähte, welche während einer Simulation stark verbogen werden, besser mit NNN- oder CTR120-Links zu verbinden, welche kombiniert alle Arten von Streckungen, Verbiegungen und Verdrehungen auch rechnen können. Um dem Benutzer die Arbeit zu ersparen, welche Link-Konstanten ein CTR120-Link nun bekommen muss und wie die einzelnen Links relativ zueinander angeordnet werden müssen, damit sie in ihrer Kombination auch die gewünschten Anforderungen bezüglich Biegung

erfüllen, übernimmt die Software diese Arbeit und stellt dem Benutzer ganze Link-Kombinationen zur Verfügung. Allerdings werden dabei nur ganze Baugruppen aus mehreren Objekten unterstützt, welche gemeinsam bestimmte physikalische Eigenschaften bekommen sollen. Link-Kombinationen versuchen letztlich das Zusammenspiel zwischen den Beanspruchungsarten Streckung, Biegung und Torsion durch eine geeignete Anordnung der einzelnen Links der Gruppe zu realisieren. Dabei wird hauptsächlich auch die Geometrie der Objekte berücksichtigt. Ein einfaches Beispiel findet der Leser im Manual 'Samples' wo ein Draht mit verschiedenen möglichen Verbindungsarten erzeugt und editiert wird. Der Benutzer hat darüber hinaus natürlich immer die Freiheit eine einzelne Verbindung zwischen zwei Objekten durch eine beliebige eigene Kombination von Links selbst herzustellen.

Für das Editieren von einzelnen Links und Link-Kombinationen stehen spezielle Dialoge zur Verfügung welche sich automatisch öffnen, wenn der Benutzer einen entsprechenden Link im Object-Tool doppelklickt. In diesen Dialogen ändert der Benutzer eine Link-Kombination immer als Ganzes. Die Integrität der Link-Kombination bleibt so gesehen erhalten.

## Die Wirkung der Link-Kraft

---

### Actio = Reactio

Die berechnete Link-Kraft wirkt, unabhängig davon wie sie berechnet wird, immer und ausnahmslos diametral entgegengesetzt bzw. mit entgegengesetzten Vorzeichen gleichermassen auf die beiden Objekte welche der Link verbindet. Damit wird dem newtonischen Gesetz 'actio = reactio' Genüge getan und die Impulserhaltung im System sichergestellt. Dasselbe gilt natürlich auch für die Rotations-Links wie den Biege- oder Torsions-Link.

### Deaktivierung

Links können direkt oder indirekt ausgeschaltet werden. Materialmodelle mit aktivierten Bruchegenschaften sind in der Lage, Links bei Überschreitung einer Maximallast auszuschalten bzw. zu deaktivieren. Die Links werden nicht gelöscht, aber ausser Kraft gesetzt. Deaktivierte Links werden bei der Berechnung der Kräfte fortan übergegangen und ihre Wirkung wird ignoriert. So gesehen sind sie 'de facto', zumindest für die Berechnungen nicht mehr existent. Die Simulation kann unter diesen veränderten Voraussetzungen normal weiter laufen. Oft will man sogar wissen, was genau geschieht beim Bruch bestimmter Verbindungen.

### Links manuell löschen

Links können prinzipiell auch manuell ausgeschaltet werden, indem die Simulation angehalten, ein Link gelöscht und die Berechnungen anschliessend fortgesetzt werden. Gelöschte Links können nicht wieder aktiviert werden, es sei denn unmittelbar danach mit 'Undo'. Deaktivierte Links werden beim nächsten Sicherungsvorgang definitiv aus dem Memory und dem Modell-File entfernt. Deshalb benutzt man nach solchen Manipulationen am besten einen neuen Filenamen um den alten Zustand des Modells zu erhalten.

## Link erzeugen

---

Um einen neuen Link zu erzeugen muss man dem System mitteilen, welche beiden Objekte verbunden werden sollen und an welchem Punkt diese Verbindung positioniert sein soll. Die Funktion welche den Link setzt geht dabei immer davon aus, dass die beiden Objekte bereits die richtige relative Position zueinander haben. Falls dies noch nicht der Fall ist, dann müssen die beiden Objekte vorab in ihre richtige relative Position verschoben werden. Die beiden Objekte dürfen sich dabei auch überlappen, ohne dass dies zu explosionsartigen Abstosskräften führen wird. Der Grund dafür ist der, dass mit jedem Link gleichzeitig und automatisch auch eine bilaterale Interaktionsregel gesetzt wird, welche besagt, dass die beiden fraglichen Objekte nicht interagieren sollen. Falls der Benutzer also wünscht, aus welchem Grund auch immer, dass keine solche Regel gesetzt werden soll, dann muss er diese anschliessend wieder löschen.

Menu / Create / New Point...  
Objekt 1 selektieren  
Objekt 2 mit gedrückter 'shift'-Taste selektieren  
Punkt mit gedrückter 'shift'-Taste selektieren  
Menu / Create / New Link / Normal

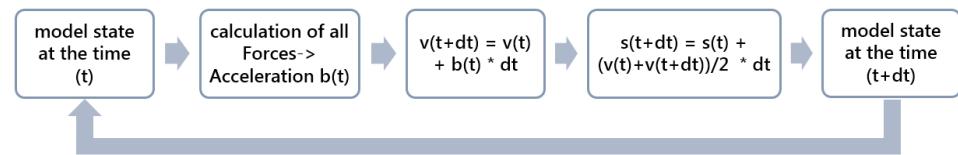
#### Ablauf zur Erzeugung eines Links

Biege-Links (Bending Links) und Torsions-Links (Torsion Links) werden völlig analog erzeugt, mit dem einzigen Unterschied, dass im letzten Schritt der betreffende Menu-Befehl zur Erzeugung des Links entsprechend gewechselt wird.

# Simulation

---

Der Ablauf einer Simulation für einen Zyklus und ein einzelnes Objekt kann vereinfacht wie folgt dargestellt werden:



$t$  : Zeit  
 $b$  : Beschleunigung  
 $v$  : Geschwindigkeit  
 $s$  : Weg

**FIGURE 77.** ein Simulationszyklus für ein einzelnes Objekt: Zeit (t) -> Zeit (t+dt)

## Zeitschritt

---

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. Der Zeitschritt, bzw. der Zyklus ist der Herzschlag eines expliziten Systems wie 'sonar'. Wir werden von Benutzern der sonar-Software oft gefragt, weshalb der Zeitschritt nicht beeinflusst werden kann. Eine typische Frage dieser Art lautet wie folgt:

- Weshalb kann ich als Benutzer den Zeitschritt nicht einfach erhöhen, wenn ich eine entsprechend ungenauere Berechnung in Kauf nehmen möchte? Für eine erste Abschätzung würde mir ein ungenauereres Resultat genügen, wenn ich es stattdessen schneller bekäme.

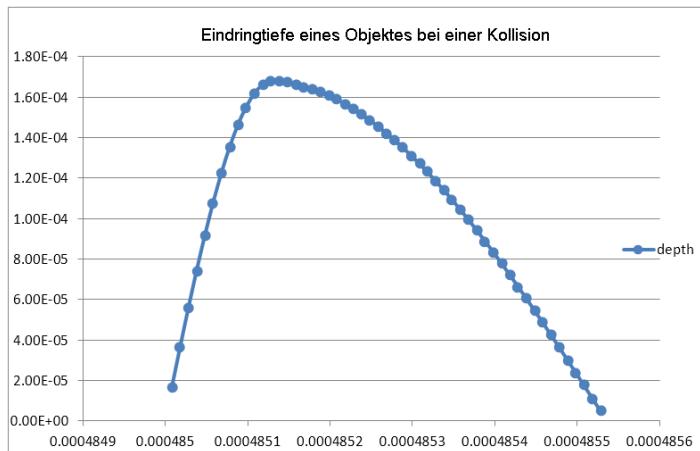
Die Antwort lautet: Das funktioniert nicht mit einem expliziten Code wie sonar. Der Grund dafür liegt in der numerischen Stabilität. Die sonar Software basiert rein rechnerisch letztlich auf den elementaren newtonschen Bewegungsgleichungen:

$$\begin{aligned} \text{Aktion} &= \text{Reaktion} \\ \text{Kraft} &= \text{Masse} \cdot \text{Beschleunigung} \end{aligned} \quad (\text{EQ 10})$$

Alles was in sonar während einer Simulation geschieht, wird letztlich explizit auf diese beiden elementaren physikalischen Gesetze zurückgeführt. Aus diesem Grund bezeichnet man sonar als 'expliziten code'. Wenn sich in sonar zwei Körper berühren, dann wird nicht irgendeine Formel verwendet, welche aus den Stossvektoren die resultierenden Geschwindigkeitsvektoren der Körper nach dem Stoss berechnet. Man könnte das zwar auch so rechnen, man würde damit aber die Gesetzmäßigkeiten eines expliziten Codes verlassen. Man begäbe sich damit physikalisch auf eine andere Ebene. Auf dieser neuen Ebene gelten andere Gesetzmäßigkeiten. So müsste man, um beim Zusammenstoss von zwei Körpern zu bleiben, ständig vorausberechnen, welche Körper während diesem 'längeren' Zeitschritt eventuell bereits kollidierten und müsste dann hinterher nochmals den inzwischen stattgefundenen Stoss berechnen und die neuen Geschwindigkeitsvektoren entsprechend korrigieren und auch die Positionen neu berechnen. Dies könnte aber zur Folge haben, dass die neue Flugbahn eines Objektes bereits zu einer weiteren Kollision geführt hat, usw. Das kann mit einer grossen Anzahl von Objekten am Ende recht kompliziert werden. Mit einer kleinen Anzahl von Objekten kann man mit einem solchen Code wahrscheinlich schneller rechnen. Je grösser die Anzahl der Objekte und je dynamischer eine Simulation wird, umso mehr verschieben sich allerdings die Vorteile in Richtung expliziter Code.

Bei einem expliziten Code wird jeder Stoss zwischen zwei Objekten in eine grössere Anzahl von Zeitschritten aufgelöst. Der gesamte Stoss wird folglich in einer Folge von elementaren physikalischen Ereignissen berechnet. Der Reihe nach passiert etwa folgendes:

Irgendwann während einer laufenden Simulation stellt das Programm fest, dass unsere beiden Objekte 'A' und 'B' während dem letzten kleinen Zeitschritt kollidierten. Die Kollisionstiefe ist aber dank dem kleinen Zeitschritt noch sehr gering. 'sonar' macht nun genau das, was auch in Wirklichkeit geschieht wenn zwei Körper zusammenstossen. In den beiden Objekten bauen sich durch die Deformation der Oberfläche abstossende Kräfte auf. Diese abstossenden Kräfte vergrössern sich mit zunehmender Kollisionstiefe und umgekehrt, nehmen wieder ab, wenn sich die Objekte wieder voneinander lösen. In sonar wird zwar die Oberfläche der Starrkörper nicht wirklich deformiert, aber die abstossenden Kräfte verhalten sich physikalisch genauso wie die Wirklichkeit, d.h. sie sind eine Funktion der Eindringtiefe. Das kann eine lineare Funktion sein oder eine Funktion von höherer Ordnung. Nachdem also festgestellt wurde, dass die beiden Objekte 'A' und 'B' im Begriff sind ineinander einzudringen, werden entgegengesetzte abstossende Kräfte aufgebaut, die sich nach jedem Zeitschritt weiter erhöhen bis die maximale Eindringtiefe erreicht ist. Dies kann z.B. nach vierzig Zeitschritten der Fall sein. Anschliessend bewegen sich die beiden Objekte allmählich wieder voneinander weg, bis sie sich in ca. hundert weiteren Zeitschritten völlig voneinander gelöst haben.



**FIGURE 78.** Die Eindringtiefe als Funktion der Zeit eines kleinen Objektes bei einem Aufprall während einer sonar-Simulation. Die Kraftfunktion bzw. der Kraftstoss verläuft ähnlich.

Diese rechnerische Nachbildung einer Kollision funktioniert natürlich nur dann wie erwartet, wenn während der Stossdauer eine genügend hohe Anzahl von Zwischenschritten stattfindet, so dass der gesamte Stoss quasi kontinuierlich abgearbeitet wird. Aus diesen Überlegungen heraus wird deutlich, dass der Zeitschritt so klein gewählt werden muss, dass auch der härteste mögliche Stoss in einem Modell noch genügend gut in Zwischenschritte aufgelöst wird. Dies bestimmt den maximal zulässigen Zeitschritt für ein Modell. Natürlich werden zur Berechnung des Zeitschrittes noch weitere Einflussgrössen berücksichtigt, welche sich indirekt auf die Stosse auswirken. Dazu gehört die Masse der Objekte, ihre maximale relative Geschwindigkeit, usw. Im Übrigen wird während der gesamten Stossdauer auch die Reibung zwischen den Objekten kontinuierlich berechnet. Hinzu kommt, dass die Stossmodelle im Allgemeinen eine Hysterese haben, d.h. beim Kollidieren wirkt eine höhere Kraft als beim Loslösen. In diesem Sinne wird ein Teil der Kollisionsenergie absorbiert.

Um abschliessend auf unser ursprüngliches Anliegen zurückzukommen, würde also ein zu grosser Zeitschritt letztlich deshalb nicht funktionieren, weil ein Modell mit einem, auch nur geringfügig, zu hohen Zeitschritt explodiert. Sie haben richtig gelesen. Das Modell fliegt auseinander wie eine CAD-Explosionszeichnung, weil sich numerische Schwingungen aufbauen, die zu einem Bersten des Modells führen. Sie sitzen vor einem leeren Bildschirm.

## kritische Faktoren einer Simulation

---

### Zeitschritt

Der Zeitschritt während einer Simulation wird von folgenden Größen beeinflusst:

- Oberflächenhärte der Objekte
- Link Konstanten der Objekt-Verbindungen
- max. Geschwindigkeit der Objekte
- Objektabmessungen

Bei der Festlegung bzw. Berechnung des maximal zulässigen Zeitschrittes ist immer das Objekt verantwortlich, welches den Zeitschritt am meisten 'herunterzieht' bzw. vermindert. Wenn also z.B. das Objekt 27 für den aktuellen Zeitschritt verantwortlich ist, dann hilft es nicht, wenn die Oberflächenhärte irgendwelcher anderer Objekte 'weicher' gemacht werden. Die Eigenschaften des verantwortlichen Objektes müssten ggf. verändert werden. Die verantwortlichen Objekte für den aktuellen Zeitschritt werden während einer Simulation im Fenster 'Simulation Parameter' angezeigt.

Grundsätzlich gilt aber, dass eine Zeitschritterhöhung möglich ist...

- indem man die Interaktionskonstante der Objekte weicher macht. Dies hat zur Folge, dass die Objekte während den Kollisionen tiefer ineinander eindringen. Dadurch erhöht sich auch die Stossdauer.
- indem man die Link-Konstanten der Objektverbindungen weicher macht. Dies hat zur Folge, dass die Amplitude der relativen inneren Schwingungen zwischen den verbundenen Objekten zunimmt und die entsprechende Schwingfrequenz abnimmt.

Wie das im Einzelnen geschieht, darauf kommen wir später zu sprechen.

### Rechenaufwand

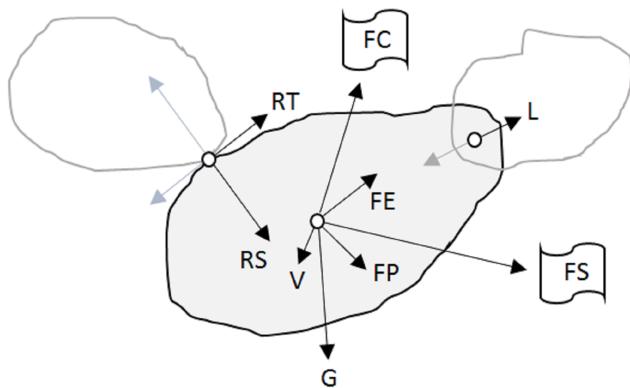
Der Rechenaufwand bzw. die absolute Rechenzeit zur Erreichung eines bestimmten Simulationsziels wird von folgenden Größen beeinflusst

- Anzahl Objekte
- Anzahl der zu berechnenden Interaktionen pro Zyklus.
- Dies wiederum hat etwas zu tun mit den Interaktionsregeln und Interaktionseinstellungen wie der Benutzer sie festlegt. Der Benutzer kann die Anzahl der Berechnungen beeinflussen, indem er die Interaktionseinstellungen der Objekte möglichst so setzt, dass nur diejenigen Begegnungen berechnet werden, die letztlich auch wirklich in die Lage kommen zu interagieren. Wir kommen später darauf zurück.
- Die Komplexität der Objekte.
- Die Interaktion einer komplexen Linien-Bogen Kontur, wie sie z.B. bei einem Zahnrad vorkommt, dauert länger als die Kollision von zwei Kugeln.
- Die Zeichenrate am Bildschirm (Updates pro Sekunde).
- Bei einer Simulation die Stunden dauert ist es nicht notwendig sämtliche Modell-Fenster mit einer Zeichenrate von 10 oder 30 Bilder pro Sekunde zu erneuern. Eine Reduktion dieser Updates kann die Rechenleistung der Software erheblich verbessern.
- Zeitschritt

# Kräfte auf Objekte

## Überblick

Als expliziter Code funktioniert in sonar letztlich alles über Kräfte. Wir zählen als Erstes alle möglichen Kräfte auf und gehen anschliessend auf jede Art von Kraft im Detail ein. Wir reden insbesondere auch darüber, wie man diese Kräfte erzeugt bzw. einstellt und aktiviert. Viele dieser Kräfte sind allgegenwärtig und werden automatisch berechnet ohne dass der Benutzer viel davon mitbekommt. Mit den folgenden Erklärungen erhält der Benutzer ein paar zusätzliche Informationen über die 'Natur' dieser Kräfte und welche Rolle sie in der Software spielen.



**FIGURE 79.** Eine Auswahl von Kräften auf ein Objekt. Externe Momente sind nicht eingezeichnet.

- |  |                       |
|--|-----------------------|
| <ul style="list-style-type: none"><li>• RS: elastisch-plastische Stosskräfte, Kollisionskräfte</li><li>• RS: Kontaktkräfte</li><li>• RT: Reibungskräfte</li><li>• L: Link-Kräfte (Objektverbindungen)</li><li>• V: Viskosität</li></ul>  | <b>Interne Kräfte</b> |
| <ul style="list-style-type: none"><li>• FE: konstante externe Kräfte und Momente auf einzelne Objekte</li><li>• G: einstellbares gleichgerichtetes konstantes Gravitationsfeld (Labormassstab)</li><li>• G: bilaterale Gravitationskräfte (für astronomische Dimensionen)</li><li>• FP: Punktkurven (manuell definierbar)</li><li>• FE: externe Punktkurven (data files)</li><li>• FC: Kontrollsysteme mit math. Formeln</li><li>• FS: sonar script (Kontrollsystem)</li></ul> | <b>Externe Kräfte</b> |

### Elastisch-plastische Stosskräfte

Stosskräfte werden hervorgerufen wenn Objekte kollidieren. Diese Kräfte stehen im Simulationsprogramm sonar absolut im Zentrum, weil alle Stöße explizit durchgespielt werden. Wie an anderer Stelle bereits erwähnt, werden die Wege die zwei kollidierende Kugeln nach einem Stoss nehmen nicht auf der Basis der Flugbahn der Kugeln vor dem Stoss und den Winkelverhältnissen der Begegnung berechnet. Jeder Stoss wird in einer grösseren Zahl von

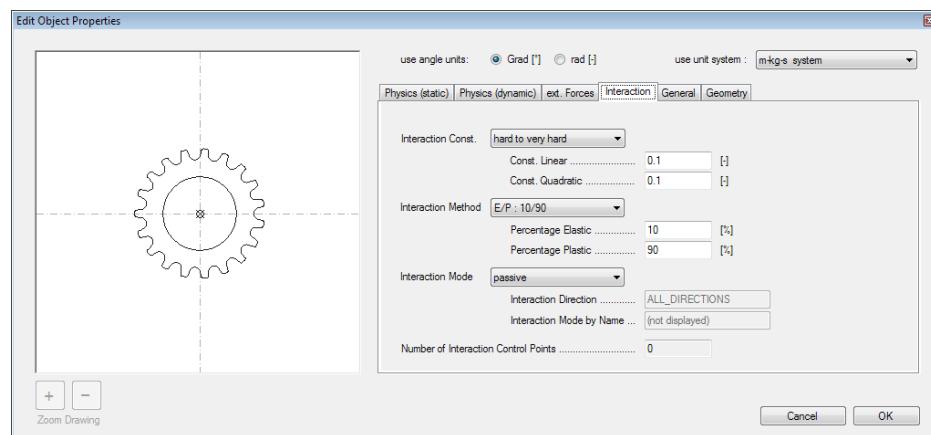
Rechenzyklen explizit mit den elementaren newtonischen Bewegungs-gleichungen durchgespielt (sehen Sie dazu das Kap. 'Simulation').

Der Benutzer ist allerdings in der Lage festzulegen, wie die physikalischen Einflussgrößen eines Stosses sein sollen. In diesem Sinne bestimmt er objektspezifische Eigenschaften wie

- Oberflächenhärte der Objekte  
Damit beeinflusst er die Eindringtiefe der beiden Objekte
- Das Verhältnis zwischen absorbiert und reflektierter Energie beim Stoss  
Diese Eigenschaft führt zu einer Hysterese der Stosskräfte
- den Reibungskoeffizient während dem Stoss (siehe nachfolgendes Kapitel).

Die beiden ersten Eigenschaften, die Oberflächenhärte und der absorbierte Energieanteil beim Stoss, kann der Benutzer individuell für jedes Objekt im Dialog "Properties" festlegen. Im "Preference" Dialog kann er die geltenden Standardwerte dieser Eigenschaften verändern. Der Dialog "Properties" ist vom Hauptmenu als auch vom Context-Menu aus aufrufbar.

Select Object / Menu / Edit / Edit Object / Properties...  
Select Object / click right mouse button on Object / Edit / Properties...



**FIGURE 80.** Der Dialog Object Properties mit dem selektierten Reiter "Interaction"

Der Dialog besitzt mehrere Seiten bzw. Panels die durch Anklicken des entsprechenden Reiters gewählt werden. Im dargestellten Panel "Interaction" werden von oben nach unten die folgenden Eigenschaften gesetzt:

- Interaction Const.  
Die Interaktionskonstante setzt die Oberflächenhärte des Objektes fest. Der Benutzer kann diesen Wert grob qualitativ setzen mittels Begriffen wie weich und hart usw. oder einen beliebigen Wert eingeben. Die Interaktionskonstante hat die Dimension einer Federkonstante. Je höher der Zahlenwert, umso härter wird die Interaktion. Ein guter Anfangswert ist immer 'hard' oder 'hard to very hard', also Werte im Bereich (0.01...0.1). Für mikromechanische Teile wie sie in einer mechanischen Armbanduhr vorkommen sind diese Werte allerdings zu hoch. Dort würde man Werte im Bereich von 1.0E-4 wählen. Die sonar Software setzt bei der Erzeugung eines Objektes einen entspr. berechneten Standardwert auf der Basis der Objektmasse. Meistens fährt man damit gut.  
Darüber hinaus hilft es nicht darüber zu philosophieren, was die Interaktionskonstante physikalisch wirklich ist und wie man entsprechende Werte in einem gegebenen Modell berechnen und setzen könnte. Der Benutzer muss selbst ein Gefühl dafür entwickeln welche Werte für seine Modelle letztlich geeignet sind. Es ist allerdings eine gute Idee, wenn der Benutzer ein einfaches Kollisionsmodell nimmt und mit diesem herumspielt, indem er einmal verschiedene Werte ausprobiert und ihre Wirkung beobachtet. In der Objektsammlung zur sonar Software findet sich dazu ein geeignetes einfaches Modell mit Anleitung.

- Interaction Method

Im betreffenden pop-up-menu schlägt das System ein paar Standardeinstellungen vor wie 'E/P : 10/90' usw. Dieser Ausdruck bedeutet: Elastischer Anteil am Stoss = 10%, plastischer Anteil = 90%.

Diese Einstellung ist übrigens ein guter Durchschnittswert den Sie immer verwenden sollten, wenn Sie sich im allgemeinen Maschinenbau bewegen. Der Benutzer kann einen entsprechenden Wert sehr einfach selbst ermitteln, indem er z.B. eine Stahlkugel aus einer Höhe von einem Meter auf eine Stahlunterlage fallen lässt und grob misst, wie hoch die Kugel zurückspringt. Anschliessend wiederholt er dieses Experiment in sonar und verändert die Einstellungen des elastischen bzw. plastischen Anteils so lange bis Übereinstimmung mit dem Experiment vorliegt. Der Benutzer muss dabei beachten, dass er die Werte auch für die Unterlage entsprechend einstellt und nicht nur für die Kugel allein. Bei Stößen ist es so, dass das weichere Objekt den Ton angibt. Ein fallendes Stück Brotteig wird nie zurückreflektiert, wie hart die Unterlage auch sein mag. Das gleiche gilt für eine Stahlkugel die in den Brotteig fällt.

## Kontaktkräfte

---

Wir sprechen von Kontaktkräften wenn zwei Objekte durch äussere Kräfte aneinander oder aufeinander gedrückt werden ohne dass sich der Kontakt löst. Ein Objekt welches durch die Gravitationskraft bewegungslos auf einem anderen Körper liegt, wäre ein solches Beispiel. In einem ruhenden Mechanismus gibt es in der Regel viele Kontaktkräfte.

Die Kontaktkräfte werden genau gleich behandelt und berechnet wie Stosskräfte. Da Stösse durch die Eigenschaft der Hysterese meistens gedämpft werden, können diese nach dem Abklingen automatisch in Kontaktkräfte übergehen. Ein Mechanismus in Ruhe hat konstante Kontaktkräfte. In sonar stellt sich ein solcher Gleichgewichtszustand, wenn es denn einen gibt, von selbst ein. Damit verbunden stellen sich auch die resultierenden Kontaktkräfte automatisch ein. Dies ist wiederum eine direkt Folge der Tatsache, dass sonar ein expliziter Code ist und alles über die Kräfte abläuft. Wenn wir ein Objekt auf ein anderes Objekt legen, welches seinerseits auf einem festen Tisch liegt, dann wird das oberste Objekt dank der Gravitationskraft solange in das darunter liegende Objekt eindringen, bis die abstossende Kraft das Gewicht des oberen Objektes aufhebt. Ist das geschehen, dann kommt diese Wechselwirkung mit einer kaum erkennbaren abklingenden Schwingung allmählich zur Ruhe und es herrscht Gleichgewicht. Es ist der gleiche Vorgang, wie er in Wirklichkeit auch geschieht. Sie mögen nun einwenden, dass die beiden Objekte in der Realität nicht ineinander eindringen. Das mag sein, aber die Oberflächen der beiden Objekte werden in Wirklichkeit ein klein wenig deformiert oder eingedrückt oder flach gedrückt oder wie immer Sie das nennen mögen. Auch dies geschieht solange bis sich ein Gleichgewicht der Kräfte eingestellt hat. Und der Betrag dieser Deformation ist etwa gleich gross wie die Eindringtiefe in unserer Simulation. Am Ende läuft das Ganze auf dasselbe hinaus.



**FIGURE 81.** Eine Kugel liegt in Ruhe auf einer Unterlage. In der Realität (links) gibt es an der Kontaktstelle einen deformierten Bereich der sich auf die beiden Objekte verteilt. Rechts ist die Simulation mit Starrkörpern dargestellt die ineinander eindringen. Die Überlappung der beiden Objekte ist aus Gründen der Klarheit übertrieben gross gezeichnet.

Kontaktkräfte unterliegen allein der Kontrolle des Programms und können vom Benutzer nicht direkt beeinflusst werden.

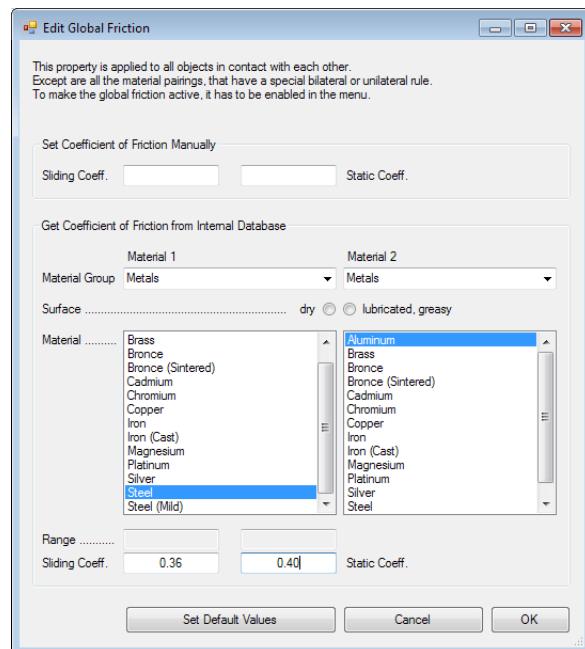
## Reibungskräfte

Reibung entsteht, wenn zwei Körper die miteinander in Kontakt treten längs ihrer Berührungsfläche unterschiedliche tangentiale Geschwindigkeitswerte haben. Dies ist die Art und Weise, wie in sonar die Reibung berechnet wird. Da ein Stoss wie bereits erklärt immer in einer grösseren Anzahl von Rechenzyklen aufgelöst wird, wird in diesem Sinne auch die tangentiale Geschwindigkeitsdifferenz und die Normalkraft während dem Stoss ständig neu berechnet und mittels entsprechenden Momenten auf die beiden beteiligten Objekte übertragen. Das bedeutet, dass in diesem Sinne auch die Reibung zwischen Objekten sehr genau berücksichtigt wird.

Die vektorielle Geschwindigkeitsdifferenz an der Kontaktstelle gibt vor, in welche Richtung die Reibung wirkt. Der Betrag der Reibung kann aus der Normalkraft an der gleichen Stelle auf unterschiedliche Art berechnet werden. Am einfachsten und standardmässig implementiert ist eine Reibung proportional zur Normalkraft. Es sind aber auch nichtlineare Reibwerte möglich.

Der Benutzer hat wie folgt Zugang zu den Reibungskoeffizienten. Als Erstes dürfte das Setzen der sog. globalen Reibung interessieren. Es ist die allgemeine, für alle Objekte geltende Reibung für die nichts Anderes festgelegt wurde

Menu / Ext. Forces / Edit Global Friction...



**FIGURE 82.** Der Dialog Global Friction

Der Benutzer setzt die Reibwerte entweder direkt manuell durch Eingabe entsprechender Werte in die beiden obersten Eingabefelder oder lässt sich mit den Auswahlmöglichkeiten darunter Vorschläge für diese Werte machen.

Sind diese allgemein gültigen Werte gesetzt, dann besteht die Möglichkeit für bestimmte Objekte oder Objektpaarungen andere Werte zu setzen. Diese haben dann für die betreffenden Objekte höhere Priorität als die allgemeinen Werte. Wir unterscheiden dabei zwischen unilateralen und bilateralen Reibwerten. Die bilateralen Werte gelten, wie der Name sagt, für eine ganz bestimmte Paarung von Objekten und nur für diese beiden zusammen. Wenn eines der beiden Objekte mit einem dritten Objekt kollidiert, dann gelten dort wieder die allgemeinen globalen Reibwerte. Die Unilateralen Reibwerte schliesslich gelten für ein ganz bestimmtes Objekt und für alle Kontakte welche dieses Objekt eingehen wird. Als Beispiel könnte man das Kettenrad einer Maschinenkette

ins Auge fassen, welches eine unilaterale Reibung bekommen hat. In der Folge werden alle Kettenglieder und Kettenbolzen welche mit diesem Kettenrad in Berührung kommen von dieser Einstellung profitieren.

Menu / Ext. Forces / Edit Uni-/Bilateral Friction...

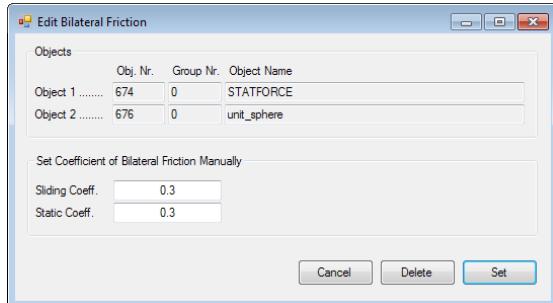


FIGURE 83. Der Dialog Edit Bilateral Friction

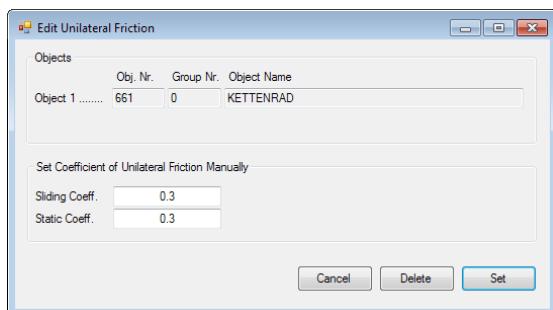


FIGURE 84. Der Dialog Unilateral Friction

## Link-Kräfte

Objekte die miteinander verbunden wurden stehen zusammen über sog. Links in Verbindung. Ein Link ist dabei wie eine Feder mit den beiden Enden an zwei unterschiedlichen Objekten befestigt. Links sind Feder-Dämpfungselemente mit einer Ruhelänge von Null. Die virtuelle Feder, die man sich am Ort des Links vorstellen kann, hat im unbelasteten Zustand folglich keine Länge und besteht aus einem Punkt. Erst unter Last bekommt der Link eine Ausdehnung. Diese Ausdehnung repräsentiert gleichsam die elastische Verlängerung der Verbindung unter Last. Die Kraft in einem Link berechnet sich als

$$F_{\text{Link}} = c_1 \cdot dL + c_2 \cdot dL^2 + D(dL, \partial L / \partial t) \quad (\text{EQ 11})$$

$c_1$  und  $c_2$  sind die sog. Link-Konstanten, welche die Steifigkeit des Links bestimmen.  $dL$  ist die Auslenkung des Links. Der Term 'D' schliesslich ist eine Dämpfungsfunktion.

Link-Kräfte können während einer Simulation aufgezeichnet werden und geben damit Auskunft über die 'inneren' Spannungen in einem grösseren Objekt, welches aus mehreren Primitivkörpern zusammengesetzt wurde. Ein einfaches Beispiel dafür ist ein Draht aus einer Kette von Zylindern. Der Benutzer kann während einer Simulation die dynamische Spannungsverteilung im Draht farblich darstellen und zwecks späterer Analyse im EXCEL Format aufzeichnen.

Wenn man sich einen grösseren Verbund von Objekten vorstellt die untereinander alle mit Links verbunden sind, dann sind es diese Links welche die Kräfte von einem Objekt zum nächsten

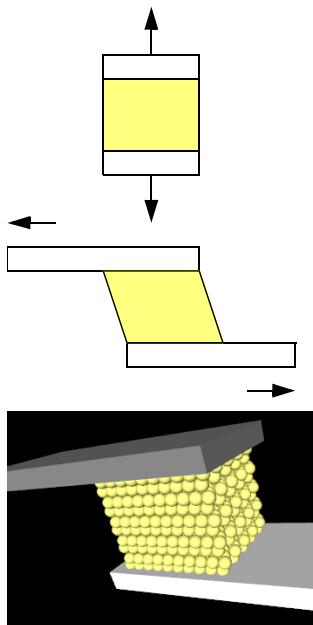
weitergeben. Dies geschieht auf völlig analoge Weise, wie in einer Kette von stossenden Kugeln, wo die Kräfte über die Oberflächenkontakte weitergeleitet werden. In beiden Fällen, bei verlinkten oder bei stossenden Objekten wird die Bewegung über virtuelle Federn von einem Objekt zum nächsten weitergeleitet. Im Prinzip ist das auch die Art und Weise wie sich die Physik solcher Vorgänge in der Wirklichkeit abspielt. Wird am Ende einer verlinkten Kette eine Kraft ausgeübt, dann dauert es eine gewisse Zeit bis das Objekt am anderen Ende der Kette etwas davon spürt. Die Fortpflanzung des Signals geschieht mit der Schallgeschwindigkeit des betreffenden Materials. Und die Schallgeschwindigkeit hat wiederum etwas zu tun mit der Steifigkeit des Mediums und der Verbindungen. Hier kommt der Elastizitätsmodul ins Spiel. Beim Setzen der Link-Konstanten muss folglich der Emodul berücksichtigt werden, dann stellt sich die Schallgeschwindigkeit der Signalfortpflanzung automatisch richtig ein. Über den Zusammenhang zwischen Link-Konstante und Emodul und wie man die Link-Konstante ggf. aus dem Emodul berechnet kommen wir an anderer Stelle zu sprechen.

Nebst den bislang besprochenen Links für Zugbeanspruchungen kennt sonar auch sog. Biege- und Torsions-Links. Kombiniert angewendet können diese drei Link-Arten alle möglichen relativen Bewegungen zwischen zwei Objekten berechnen. Jede Link-Art besitzt dabei ihre eigene Link-Konstante. Mit anderen Worten, die Steifigkeiten dieser Links können unabhängig voneinander eingestellt werden.

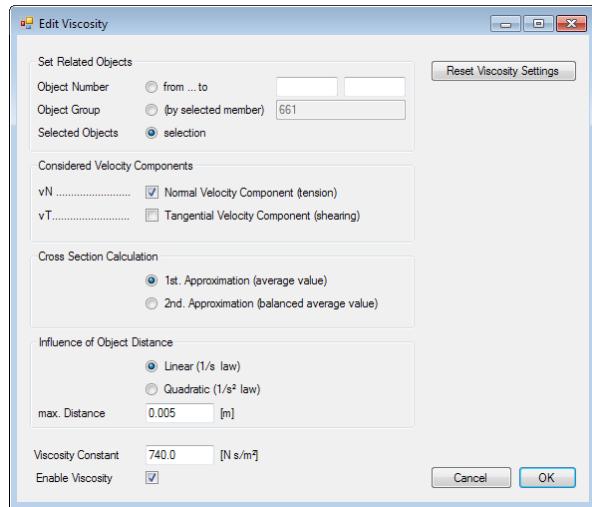
## Viskosität

---

Viskositätseigenschaften können in sonar nur dann berücksichtigt werden, wenn ein sog. Partikelhaufen erzeugt wurde. Partikel verhalten sich im Prinzip wie normale Objekte, mit dem Unterschied, dass sie eine einfachere Datenstruktur haben. Partikel sind so gesehen einfache Primitives vom Typ Kugel mit einem reduzierten Datensatz. Partikel können allein nicht auftreten. Sie werden immer als Block oder Haufen in einer grösseren Anzahl erzeugt. Partikel haben jedoch Gruppeneigenschaften wie eben die Viskosität welche die gesamte Gruppe von Partikeln betrifft. Es sind Eigenschaften die für das einzelne Partikel keine Bedeutung haben sondern immer die Gruppe als Ganzes betreffen und nur in der Gruppe auftreten. Es sind also Eigenschaften die sich über mehrere Partikel verteilen.



Eine grundlegende Messmethode der Viskosität besteht darin, die Kraft zwischen zwei parallelen Platten aufzuzeichnen welche das dazwischen eingeschlossene Medium mit dem Schereffekt deformiert. Diese Methode kann auch als sonar-Modell aufgesetzt und simuliert werden. Durch Vergleich der Scherkräfte lassen sich damit Modell und Realität direkt auf bestimmte Stoffe abgleichen. Der Zugriff auf den Dialog zum Einstellen der Viskosität erfolgt mittels:



**FIGURE 85.** Der Dialog Edit Viscosity

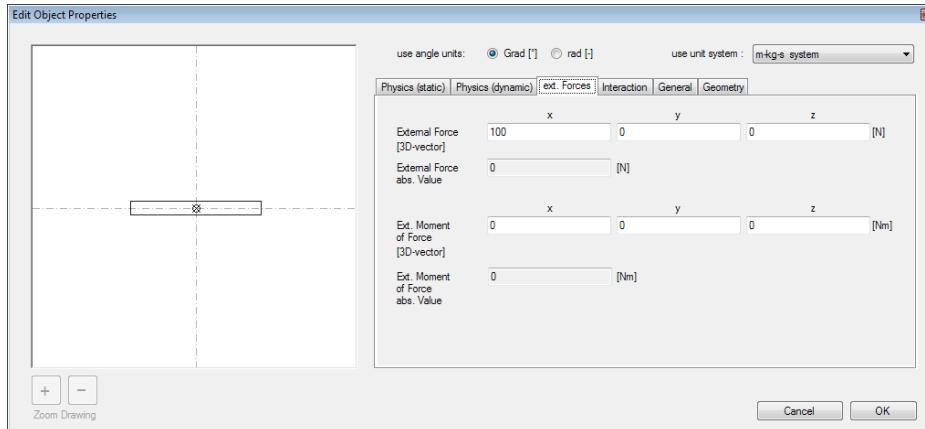
Eine tiefergehende Beschreibung aller Viskositätseinstellungen dieses Dialogs übersteigt den Zweck dieses Tutorials und wird an anderer Stelle genauer besprochen.

## Konstante Externe Kräfte

Jedem Objekt kann ggf. eine konstante externe Kraft in Form eines raumfesten Kraftvektors auferlegt werden. Es handelt sich hier also um Kräfte die sich nicht mit dem Objekt mitdrehen, sondern in Bezug zum raumfesten Koordinatensystem eine bestimmte Richtung und einen festen Wert haben. Es handelt sich um Kräfte, die ein Objekt immer in die gleiche Richtung ziehen, unabhängig davon, wie das Objekt sich bewegt und dreht.

Externe Kräfte werden im Dialog für die Objekt Eigenschaften eingegeben

Select Object / Menu / Edit / Edit Object / Properties...  
 Select Object / click right mouse button on Object / Edit / Properties...



**FIGURE 86.** Der Dialog Edit Object Properties mit dem selektierten Reiter 'ext. Forces'

Nebst translatorischen Kräften können hier auch Drehmomente definiert werden, welche wie die translatorischen Kräfte auf den Schwerpunkt wirken. Der Drehmomentvektor steht dabei senkrecht auf der Rotationsebene in der er wirkt.

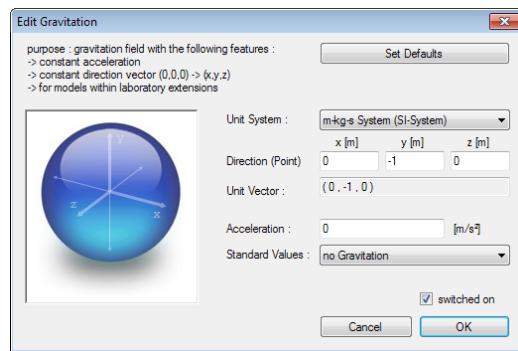
## **Gleichgerichtetes konst. Gravitationsfeld**

---

Solange wir unsere simulationstechnischen Experimente im Labormassstab durchführen, dürfen wir das Gravitationsfeld mit gutem Gewissen als konstant und gleichgerichtet betrachten. Das heisst allerdings nicht, dass wir nur mit einer bestimmten Gravitationskraft rechnen wollen. Es soll auch möglich sein, ein Experiment auf dem Mond oder dem Marsmond Deimos durchzuführen und mit den betreffenden lokalen Verhältnissen zu rechnen. Es soll sogar möglich sein, simulationstechnische Unterschiede eines Mechanismus an unterschiedlichen Positionen auf unserem Planeten zu ermitteln. Eine Pendeluhr wäre ein Beispiel eines solchen Experiments. Wie viel geht eine Pendeluhr in Washington vor oder nach gegenüber einer identischen Uhr in Berlin? Interessant sind auch Experimente in einem sog. Mikro-Gravitationsfeld wie sie z.B. an Bord des Space-Shuttle oder der ISS durchgeführt wurden.

Der Zugriff auf den Dialog Edit Gravitation geschieht wie folgt:

Menu / Ext. Forces / Gravitation (Laboratory) ...



**FIGURE 87.** Der Dialog Edit Gravitation (Labormassstab)

In diesem Dialog stellt der Benutzer ein Gravitationsfeld bezüglich Betrag und Richtung ein. Die Richtung ist ein Vektor im 3-dimensionalen Raum, der Betrag kann entweder manuell eingegeben oder aus einer Menu-Liste ausgewählt werden. Standardmässig sollten Sie die Einstellung 'Earth Standard' verwenden. Mit der Checkbox über der 'ok' -Taste wird das Gravitationsfeld ein- und ausgeschaltet.

Ein konstantes Gravitationsfeld übt auf alle Objekte eines Modells eine konstante Beschleunigung aus, unabhängig von der Masse und der Position der betreffenden Objekte.

## **Zentrales Gravitationsfeld**

---

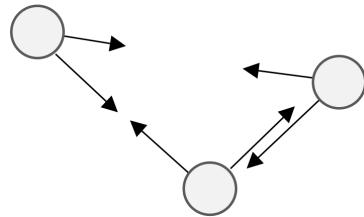
Für Simulationen im astronomischen Massstab kann das zentrale Gravitationsfeld eingeschaltet werden.

Menu / Ext. forces / Gravitation (Central) / Set (for all objects)  
Menu / Ext. forces / Gravitation (Central) / Set (for selected objects)

Die Anziehungskräfte werden bilateral nach Newton berechnet indem jedes Objekt (i) mit jedem anderen Objekt (j) wechselwirkt.

$$F_{ij} = k * (m_i \cdot m_j) / r_{ij}^2$$

(EQ 12)



**FIGURE 88.** Zentrales Gravitationsfeld

Die gesamte Gravitationskraft auf das Objekt (i) berechnet sich dann als

$$F_i = k * \Sigma ((m_i \cdot m_j) / r_{ij}^2)$$

(EQ 13)

Die zweite Variante (Set for selected objects) dient ggf. der Optimierung der Rechenleistung. In gewissen Modellen ist es nicht notwendig alle bilateralen Gravitationskräfte zu berechnen. Ein Beispiel dafür ist die Simulation der Flugbahn von 1000 Satelliten die um die Erde fliegen. In diesem Fall genügt es lediglich die zentrale Gravitationskraft der Erde einzuschalten, weil die Anziehung unter den Satelliten selbst vernachlässigt werden kann. Statt eine halbe Million müssen dann nur noch 1000 Kräfte pro Zyklus berechnet werden, was einen deutlichen Unterschied macht.

## Punktkurven

Eine Punktkurve ist im vorliegenden Kontext vorerst eine veränderliche Funktion welche durch eine Serie von Punkten definiert wird. Aufeinander folgende Punkte werden paarweise durch Strecken verbunden und beim Auslesen automatisch linear interpoliert. Die folgende Abbildung zeigt eine solche Funktion bestehend aus lediglich drei Punkten. Links oben ist die Kurve als Bild, rechts unten in der Tabelle numerisch sichtbar. Dieser Dialog verwaltet einfache Punktkurven mit bis zu 64 Punkten. Übersteigt die Punktzahl diesen Wert, dann wechseln wir auf die sog. externen Punktkurven wie sie im nächsten Abschnitt behandelt werden.

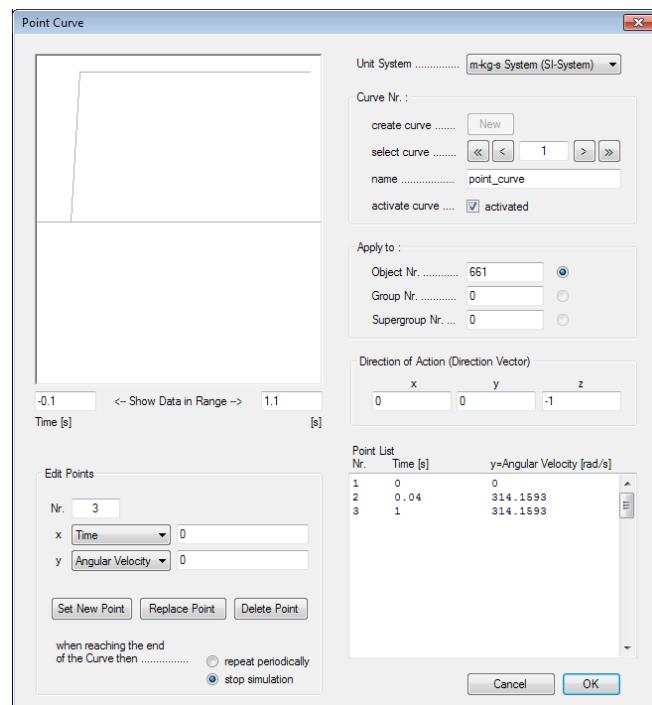


FIGURE 89. Der Dialog 'Point Curve'

Dieser Dialog ist also für einfache Punktkurven gedacht, weil die Verwaltung einer sehr grossen Punktzahl mit den einfachen Mitteln dieses Dialogs nicht mehr sinnvoll ist. Eine einfache math. Funktion, wie in der Abbildung oben dargestellt, könnte z.B. die Drehzahl bzw. die Winkelgeschwindigkeit eines Antriebsmotors eines mechanischen Systems als Funktion der Zeit sein. Der Motor fährt allmählich hoch und dreht dann mit konstanter Drehzahl. Dies wäre ein Beispiel einer sog. Zwangsbewegung eines Objektes falls die Funktion sich auf eine entsprechende Drehachse bezieht. Die Zuordnung einer Funktion geht im Allgemeinen wie folgt vor sich:

- Selektion eines Objektes im Modell (sonar-SIM)

Menu / Control Systems / New Point Curve...

- Taste 'create curve...' [New]
- Im Kopf des Dialogs das Einheitensystem auswählen, welches den zu definierenden Daten entspricht.
- die Variablen für die Abszisse und Ordinate festlegen (pulldown menus)

- Punkte für x und y eingeben und die Taste [Set New Point] drücken.
- Zum Ändern eines Punktes wird dieser in der Liste selektiert, in den Feldern x, y geändert und die Taste [Replace Point] gedrückt.
- Ein Punkt wird gelöscht mit einer Selektion in der Liste & [Delete Point]

Beachten Sie, dass Sie für eine korrekte grafische Darstellung der Kurve in der Grafik den Wertebereich in x-Richtung selbst einstellen müssen. Dazu geben Sie den minimalen und maximalen Wert des Wertebereichs in die entspr. Felder direkt unter der Grafik ein.

Alle anderen Bedienungselemente und Eingabemöglichkeiten im Dialog sind selbsterklärend. Dazu gehört die Aktivierung der Kurve, auf welche weiteren Objekte die Punktkurve ggf. angewendet und in welche Raumrichtung die Wirkung ausgeübt werden soll (Wirkungsvektor).

Die Simulations-Software sonar bestimmt während einer laufenden Simulation in jedem Rechenzyklus den aktuellen Funktionswert jeder Punktkurve und wertet diesen entsprechend aus. Liefert eine Punktkurve als Resultat eine Beschleunigung, eine Kraft, eine Winkelbeschleunigung oder ein Drehmoment, dann werden diese Werte jeweils den anderen Kräften und Momenten, die ggf. auf das betreffende Objekt wirken, überlagert. Es gilt das Superpositionsprinzip. Das heisst insbesondere, dass die Kollisionskräfte weiterhin funktionieren. Berechnet eine Punktkurve als Resultat aber eine Geschwindigkeit, eine Winkelgeschwindigkeit, eine Position oder einen Winkel, dann wird das betreffende Objekt zwangsbewegt. Andere Kräfte, die allenfalls auf das betreffende Objekt wirken, werden dann übergangen. Wenn die Punktkurve sagt, das Objekt habe jetzt diese oder jene Geschwindigkeit, dann macht es keinen Sinn gleichzeitig auch noch die Geschwindigkeit zu berechnen, die es ohne diese Punktkurve hätte. Das Objekt kann nicht beiden Berechnungen gehorchen. Die Punktkurve hat die höhere Priorität. Andere Objekte, welche nicht von dieser Punktkurve gesteuert werden, prallen am zwangsbewegten Objekt aber weiterhin physikalisch korrekt ab. Allerdings beeinflussen diese Kollisionen die Bewegung des zwangsbewegten Objektes nicht.

Bei Zwangsbewegungen bleibt der Impuls des Gesamtsystems nicht erhalten. Das gleiche gilt für die Energie. Man kann sich das so vorstellen, dass für die Zwangsbewegung ständig Kraft aufgewendet muss, um bei Kollisionen keine Bewegungsenergie zu verlieren oder zu gewinnen. Zum anderen wird die für die Zwangsbewegung notwendige Impulsänderung des zwangsweise bewegten Objektes von aussen zu- oder abgeführt. Diese Kräfte sind letztlich externe Kräfte, die in das System eingespeist werden oder auf diesem Weg abfließen. Um auf den eingangs erwähnten Antriebsmotor zurückzukommen, bezieht dieser Motor seine Energie von aussen und speist sie in das zu simulierende System ein. In die Realität übertragen wäre das z.B. die elektrische Energie die der Motor aus dem Netz aufnimmt.

## Externe Punktkurven

---

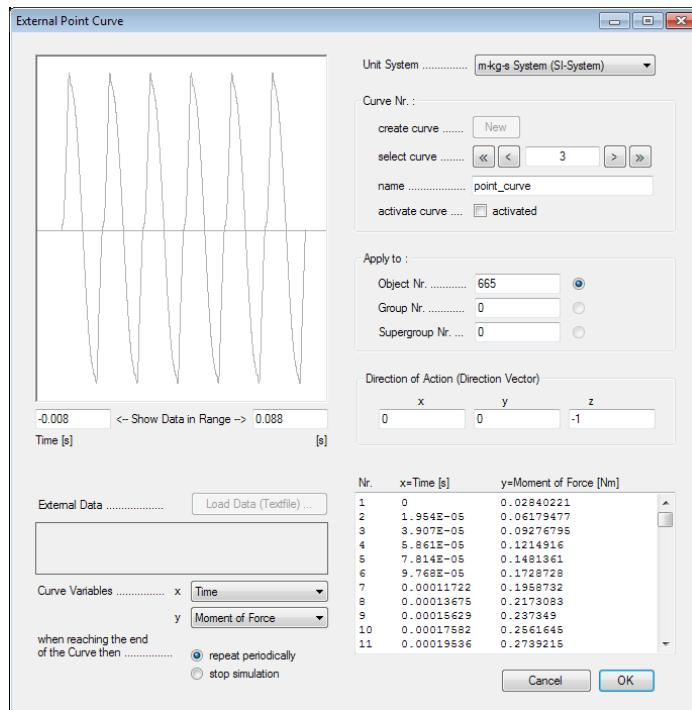
Wie im letzten Abschnitt bereits angedeutet, geht eine Punktkurve in eine externe Punktkurve über, wenn sehr viele Punkte im Spiel sind. Eine externe Punktkurve kann in der aktuellen Implementierung bis zu  $2^{12} = 4096$  Punkte aufnehmen. Die Verwaltung der Punkte geschieht in diesem Fall nicht mehr im Dialog selbst, sondern extern in einem anderen Programm. In der Regel ist dieses andere Programm ein Tabellenkalkulationsprogramm wie EXCEL. Egal um welches Programm es sich auch handelt, der Benutzer speichert die zweispaltigen Daten im Text-Format und kann dieses Textfile danach mit dem Dialog 'External Point Curve' laden. Der Ablauf mit diesem Dialog unterscheidet sich deshalb teilweise von denjenigen bei einfachen Punktkurven:

- Selektion eines Objektes im Modell (sonar-SIM)

Menu / Control Systems / New External Point Curve...

- -> der Dialog wird geöffnet
- Taste 'create curve...' [New]

- mit der Taste [Load Data (Textfile)] wird ein File geöffnet. Die Daten werden in der Folge in das Listenfeld eingefüllt und werden sichtbar. Das Programm weiss in diesem Moment allerdings noch nicht, in welchem Einheitensystem die Daten zu interpretieren sind.
- Das Einheitensystem (Unit System) festlegen.  
Die Daten können vorgängig in einem externen Programm in einem Einheitensystem definiert werden, das die sonar Software kennt ( $m\_kg\_s$ ,  $cm\_g\_\mu s$ ,  $cm\_g\_s$ ). Nach dem Laden der Daten kann der Benutzer im Dialog das zugehörige Einheitensystem mit dem 'pulldown' Menu 'Unit System' festlegen, ohne dass dabei die Daten im Listenfeld umgerechnet werden. Später, wenn das Kontrollsyste mit dem Dialog erneut geöffnet wird, bewirkt eine Änderung des Einheitensystems (Unit System) automatisch eine entsprechende Umrechnung der bestehenden Daten im Listenfeld. Eine Änderung der Beziehung (Datenwerte - Einheitensystem) ist dann also nicht mehr möglich. Man müsste dann die Daten neu laden.
- die Variablen für die Abszisse und Ordinate festlegen (pulldown menus)



**FIGURE 90.** Der Dialog External Point Curve

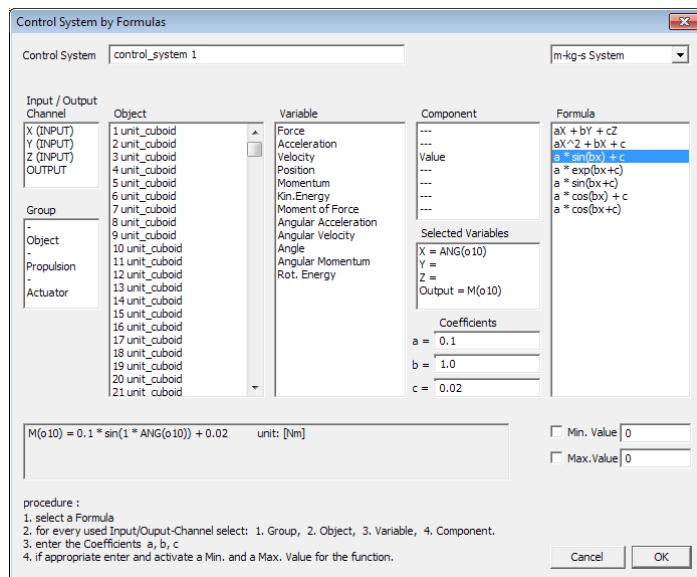
Das Löschen von ganzen Punktkurven geschieht nicht in diesen beiden Dialogen sondern ausserhalb, indem eine Punktkurve im 'Object Tool' selektiert und mit dem Menu-Befehl 'Clear' gelöscht wird.

## Kontrollsyste mit math. Formeln

Im Gegensatz zum letzten Abschnitt, wo Sie eine Funktion als Punktkurve festlegten, definieren Sie hier einen Kurvenverlauf als mathematische Funktion. Das ist eine andere Art den funktionellen Verlauf einer Zustandsgrösse einzugeben. Mit Punktkurven kann man ohne Einschränkungen jeden denkbaren Verlauf nachzeichnen. Mit mathematischen Funktionen ist man in dieser Beziehung eher eingeschränkt, dafür ist die Definition vielleicht genauer und folgt besser dem gewünschten Verlauf. Eine Sinus-Funktion lässt sich definitiv einfacher mittels einer Formel definieren als mit einer Punktkurve. Welche Methode in einem gegebenen Fall die bessere ist, hängt von der Problemstellung ab.

- sonar-SIM / Selektion eines Objektes im Modell

sonar-SIM / Menu / Controlsystem / New Control System (by Formula)  
 -> der Dialog wird geöffnet



**FIGURE 91.** Der Dialog 'Control System by Formula'

Die Bedienung dieses Dialogs geschieht nach dem folgenden Schema:

1. Man überlegt sich welche der zur Verfügung stehenden Formeln in der Liste ganz rechts am besten die Bedürfnisse abdeckt. Dazu muss man beachten, dass nicht alle Koeffizienten der angeschriebenen Formel benutzt werden müssen. Einzelne der Koeffizienten (a, b, c) dürfen auch auf null gesetzt werden. Dies erweitert die Palette der Formeln um einen Faktor.
2. Am besten versteht man den Ablauf mit einem Beispiel. Nehmen wir an, wir wollten das Drehmoment auf eine Antriebsachse so setzen, dass ein konstantes Moment plus eine winkelabhängige Abweichung Sinus-förmig wirkt. Also ein Drehmoment mit einem konstanten Wert welchem ein Sinus überlagert wird
3.  $M = a + b \cdot \sin(c \cdot t)$   
 in unserer Tabelle gibt es die Formel  $a * \sin(bx) + c$  welche im Prinzip dasselbe darstellt.
4. In der Formel  $y = a * \sin(bx) + c$  würden die Koeffizienten auf unsere Anwendung bezogen die folgende Bedeutung haben:

x : der Variablenwert. Das ist der Winkel des betr. Objektes.

y : der berechnete Funktionswert -> das Drehmoment des betr. Objektes

c : konstanter Wert = Basisdrehmoment

a : Amplitude der überlagerten Sinus-Schwingung

b : wird auf 1 gesetzt, da die Frequenz der Störung mit der Drehfrequenz der Achse übereinstimmt.

5. Jetzt wissen wir was wir wollen und selektieren im Dialog der Reihe nach:

-----  
 -> Das Meter-Kilogramm-Sekunden System (m-kg-s System)  
 -> Formula = die Formel die wir benutzen wollen

-----  
 -> Chanel = OUTPUT  
 -> Group = Object  
 -> Object = (das Objekt welches das Drehmoment bekommen soll)  
 -> Variable = Moment of Force  
 -> Component = Z

```

-> Chanel = X (Input)
-> Group = Object
-> Object = (das Objekt welches den Winkel liefern soll)
-> Variable = Angle
-> Component = Z
-----
-> Coefficients a = 0.1
-> Coefficients b = 1.0
-> Coefficients c = 0.02
-----
```

Die Formel welche Sie zusammengestellt haben wird im unteren Teil des Dialogs in ihrer ganzen Länge angezeigt. Es steht Ihnen frei die einzelnen Teile der Formel durch eine erneute Eingabe zu ändern. Die Formel kann auch jederzeit später geändert werden, falls sich ein Koeffizient als ungeeignet erweist.

Falls Sie für dasselbe Objekt zusätzlich eine weitere Funktion schreiben wollen, dann ist das kein Problem. In diesem Fall definieren Sie einfach ein weiteres Kontrollsysteem mit einer anderen Formel. Die beiden Formeln stören sich gegenseitig nicht. Sie werden während einer Simulation einzeln ausgewertet. Die Wirkungen werden einfach überlagert. Damit lassen sich letztlich auch etwas komplexere Funktionen in mehrere Formeln verpacken. Wenn diese Formel-Kontrollsysteme aber wesentlich komplexer werden, dann ist man wahrscheinlich mit einem sonar-script-Kontrollsysteem besser bedient.

### **Kontrollsysteem mit sonar script**

Wieder eine andere Art ein Kontrollsysteem zu erzeugen, ist mit der sonar script Sprache. In letzter Konsequenz sind damit recht komplexe Steuerungen möglich, weil damit praktisch beliebig lange Kontrollsysteme geschrieben werden können. Wenn wir die Formel des letzten Abschnitts mit sonar script definieren wollen, dann sieht das wie folgt aus:

```

CONTROLSYSTEM RotationSpeedControl
-- comment: Formel y = a * sin(bx) + c as sonar script code
-- (a = 0.1, b = 1.0, c = 0.02)

SET VALUE (MOMENT_FORCE_EXT.Z(o1)=1.0E-6*SIN(ANGLE.Z(o1))+2.0E-7)
-- end of control system
```

Achtung die Variablen und die Koeffizienten werden im sog. [cm g  $\mu$ s] Einheiten System eingegeben. Sie sehen die verwendete Formel im Dialog 'Control System by Formulas' z.B. in diesem Einheiten System, wenn Sie nach der Erzeugung der Formel vom [m kg s]- zum [cm g  $\mu$ s]-System wechseln.

Ein sonar-script ist ein reines Textfile welches mit jedem textverarbeitenden Programm vorbereitet werden kann. Das Windows Textprogramm 'Editor', welches mit jedem Windows System mitgeliefert wird, eignet sich gut für diesen Zweck. Ein sonar script wird direkt im Menu von sonar-SIM aufgerufen:

```
sonar-SIM / Menu / Controlsystem / Sonar Script /
Open Controlsystem...
```

```

Controlsystem
File : C:\sonar Control Systems\7. CtrlSystem_PropellerHochfahren\controlSystem-4.txt
CONTROLSYSTEM RotationSpeedControl
-11.12.14: start development

DO IF (OMG(o1)<1.0E-5)
  SET VALUE(MOMENT_FORCE_EXT.z(o1) = 1.0E-8)
END IF
DO IF (OMG(o1))>1.0E-5)
  SET VALUE(MOMENT_FORCE_EXT.z(o1) = -1.0E-8)
END IF

DO IF (T>10.0E+6)
  STOP SIMULATION
END IF
-- end of control system

```

**FIGURE 92.** Darstellung eines Kontrollsystens in einem eigenen Fenster

Das geöffnete File wird in einem eigenen Window mit dem Namen 'Controlsystem' dargestellt. Die Bedienung dieses Systems geschieht weiterhin im betr. Menu wo wir das file aufriefen. Der nächste Schritt ist die Aktivierung dieses Controlsystems mittels:

```
sonar-SIM / Menu / Controlsystem / Sonar Script / Activate
Controlsystem
```

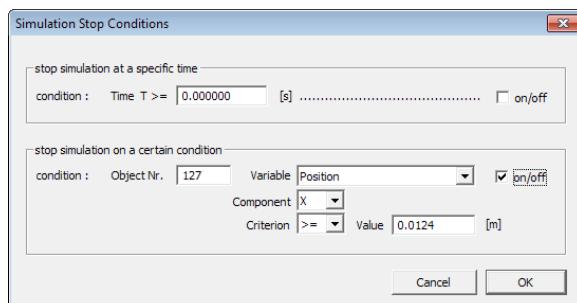
Wenn das Kontrollsysten erfolgreich aktiviert wurde, wird dieser Befehl im Menu mit einem Häkchen versehen. Eine weitere Möglichkeit die Aktivierung des Kontrollsystens zu kontrollieren ist das Fenster 'Simulations-Parameter' welches nebst vielen anderen Einstellungen auch ein aktiviertes sonar script Kontrollsysten mit einem hellen Punkt darstellt. Nur aktivierte Kontrollsysteme versehen während einer Simulation ihren Dienst.

Es ist zulässig, Änderungen an einem Kontrollsystem direkt im dargestellten Fenster vorzunehmen und diese, falls man das wünscht, anschliessend zu speichern. Der verwendete Referenztext des Kontrollsystens während einer Simulation ist aber immer das, was im Fenster 'Controlsystem' geschrieben steht und nicht was im zugehörigen File gespeichert wurde.

Die Verwendung von sonar script als Kontrollsystem-Sprache ist ein mächtiges Werkzeug, welches viele Möglichkeiten bietet auf die wir jetzt hier nicht näher eingehen wollen. Erwähnt seien lediglich die Stichworte

- Programm Ausführungs-Kontrolle und Steuerung (Simulationsstopp unter gewissen Bedingungen, Ereignissen oder Modell-Zuständen)
- dynamisches Verändern von physikalischen Variablen während einer Simulation
- dynamisches oder Ereignis-gesteuertes Verändern von geometrischen Abmessungen von Objekten oder Befestigungspunkten.
- Möglichkeiten zur gesteuerten Optimierungen von Mechanismen hinsichtlich Formgebung, Energieverbrauch, Fahrweg, usw.

## Simulationsabbruch

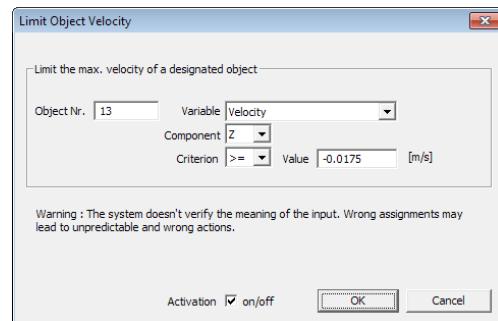


**FIGURE 93.** der Dialog 'Simulation Stop Condition'

In vielen Fällen will man einfach sagen können, dass die Simulation bei Erreichen einer bestimmten Simulationszeit anhalten soll. Für diese Aufgabe ist der obere Teil des Dialogs gedacht. Darunter hat der Benutzer die Möglichkeit, weitergehende Abbruchkriterien festzulegen. So ist es z.B. möglich anzutragen, dass die Simulation anhalten soll, wenn ein bestimmtes Objekt eine gewisse Geschwindigkeit oder Position erreicht oder überschritten hat.

## Geschwindigkeitsbeschränkung

---



**FIGURE 94.** der Dialog 'Limit Object Velocity'

Es kommt bei gewissen Simulationen vor, dass man die Geschwindigkeit oder Winkelgeschwindigkeit eines bestimmten Objektes limitieren möchte. Die Einstellungen des Dialogs greifen in diesem Sinne in die Simulation ein, wenn die definierte Bedingung verletzt wird. Solange der Wert der betreffenden Variable die Grenze nicht überschreitet hat die Einstellung keinen Einfluss.

# Interaktionssteuerung

---

## Einleitung

---

Die Kollisionskontrolle geschieht in sonar weitgehend automatisch. Man könnte alle Objekte einfach so definieren, dass man dabei bleibt, dass jedes Objekt mit jedem anderen Objekt kollidieren soll, wenn sich die betreffenden Objekte ggf. begegnen. Das funktioniert. Bei Modellen mit wenigen Objekten ist das oft auch ganz in Ordnung so. Der Grund um bei komplexeren Modellen weitergehende Kollisionsregeln einzuführen kommt allein aus dem Bedürfnis heraus, die Leistung der Ausführungsgeschwindigkeit der Simulation zu steigern. Das Potential mit diesen Mitteln die Rechenzeit zu reduzieren ist erheblich.

Die Vorgehensweise zur Leistungssteigerung der Interaktionskontrolle in einer Software wie 'sonar' geschieht zur Hauptsache immer nach dem Prinzip unnötige Berechnungen so rasch als möglich auszuschliessen. Es wird also als Erstes immer gefragt: müssen wir dies und jenes überhaupt rechnen? Können zwei bestimmte Objekte in der aktuellen Lage überhaupt kollidieren? Es geht also immer darum, mit möglichst einfachen Mitteln bzw. mit einer einfachen Rechnung eine Negativaussage zu bekommen um eine aufwendigere Rechnung zu vermeiden. Nur für Objektpaarungen für die eine offensichtliche Wahrscheinlichkeit besteht, dass sie sich berühren, wird der Aufwand einer vollen Kollisionsberechnung getrieben.

Der Benutzer kann dieses Vorgehen weitgehend unterstützen und seine Effizienz steigern, indem er selbst entsprechende Voreinstellungen durchführt. Wir wollen dies an einem Beispiel erklären:



**FIGURE 95.** Interaktionen am Beispiel eines Flipperkastens

## aktive und passive Objekte

---

Gehen wir davon aus, dass wir ein Simulationsmodell eines Flipperkastens bauen wollen. Überlegen wir uns, welche Objekte bzw. welche Teile mit welchen anderen Teilen zusammenstossen können. Konzentrieren wir uns beispielhaft auf die folgenden Teile:

- die Kugel

- die beiden Flipper am unteren Ende
- die Anprall-Inseln in der Mitte des oberen Bereichs
- die äusserste rechteckige Berandung wo nie eine Kugel hinkommt

Was kann mit was kollidieren?

- Wir stellen fest, dass die Kugel immer an einer Kollision beteiligt ist
- Die Flipper können auch kollidieren, aber nur mit der Kugel
- Das gleiche gilt für die Anprall-Inseln im oberen Bereich.
- Aber ein Flipper kann nie mit dem anderen Flipper zusammenstoßen und schon gar nicht mit einer Anprall-Insel.
- Die äusserste Berandung schliesslich kollidiert nie mit etwas anderem, auch nicht mit der Kugel.

Um dermassen unterschiedliche Interaktionsregeln in einem Simulationsmodell berücksichtigen zu können, wurden 3 Arten von Interaktionseigenschaften eingeführt. Jedes Objekt in sonar kann eine der drei folgenden Eigenschaften haben

- AKTIV
- PASSIV
- KEINE INTERAKTION

Objekte mit den beiden Eigenschaften AKTIV und PASSIV verhalten sich wie folgt:

#### AKTIV

Nur die aktiven Objekte kümmern sich 'aktiv' um eine allenfalls stattfindende Kollision. Nur sie kommen auf die Idee zu überprüfen, ob sie ev. mit einem anderen Objekt zusammengestossen sind. Sie tun das mit allen anderen Objekten, d.h. mit allen AKTIVEN und allen PASSIVEN. Ausgenommen sind die Objekte mit der Eigenschaft (KEINE INTERAKTION), denn die haben bereits angemeldet, dass sie ohnehin nie kollidieren wollen.

#### PASSIV

Die passiven Objekte verhalten sich, wie der Name sagt, 'passiv', d.h. sie kümmern sich nicht aktiv um eine mögliche Interaktion mit einem anderen Objekt. Sie stellen sich auf den Standpunkt, dass sie bereit sind allenfalls mit einem anderen Objekt zu kollidieren, wenn sie denn darum gebeten werden, aber das andere Objekt muss sich schon selbst darum kümmern, sonst passiert gar nichts.

#### Fazit

Dieses Verhalten führt dazu, dass eine Kollision immer nur dann stattfindet, wenn mindestens eines der beiden Objekte AKTIV ist. Wenn beide AKTIV sind, umso besser. Sind aber beide PASSIV, dann geschieht nichts. Und ist eines der beiden Objekte sogar vom Typ KEINE INTERAKTION dann geschieht sowieso nichts, was immer das andere Objekt auch sein mag.

Kommen wir wieder zurück zum Flipperkasten. Eine gute Wahl für diese Eigenschaften wäre wie folgt:

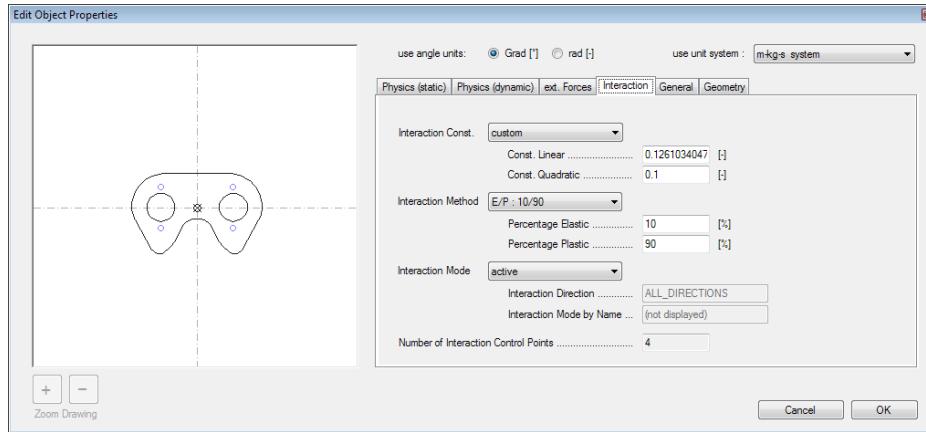
- Kugel = AKTIV
- beide Flipper = PASSIV
- Anprall Inseln = PASSIV
- äusserste Berandung = KEINE INTERAKTION

Die Simulations-Software kann ihre Kollisionskontrolle mit diesen Regeln auf ein absolutes Minimum reduzieren und nur noch diejenigen Objekt-Paarungen auf eine mögliche Kollision überprüfen, für die überhaupt eine Möglichkeit besteht, dass sie zusammentreffen. Indem der Benutzer also jedem Objekt eine dieser Interaktionseigenschaften zuweist, hilft er in einem ersten Schritt den Rechenaufwand während der Simulation zu reduzieren. Allerdings muss der Benutzer diese Interaktionseinstellungen selbst durchführen, denn standardmäßig werden Primitivkörpern beim Erzeugen immer auf AKTIV gesetzt.

Eine Interaktionseigenschaft wird für ein Objekt wie folgt gesetzt:

```
select object -> Menu / Edit / Edit Object / Properties... /  
Interaction / Interaction Mode
```

```
select object -> context menu / Edit / Properties... / Interaction  
/ Interaction Mode
```



**FIGURE 96.** Der Dialog 'Edit Object Properties' mit selektiertem Reiter 'Interaction'

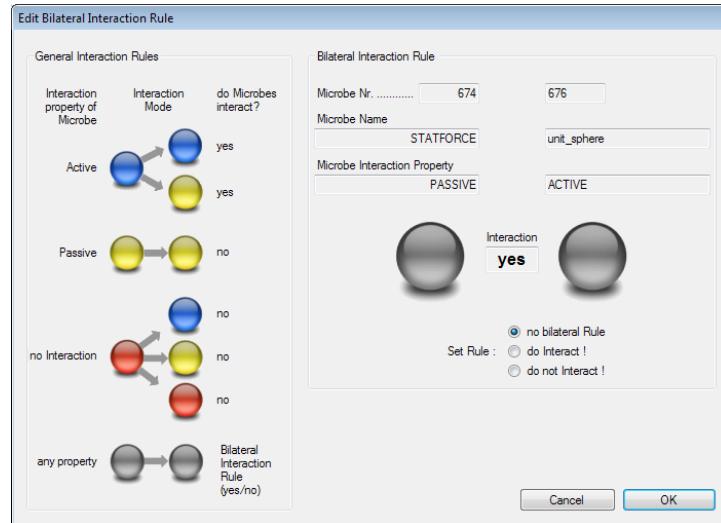
mit dem popup-menu 'Interaction Mode' kann für das selektierte Objekt das gewünschte Interaktionsverhalten (active, passive, no interaction) ausgewählt werden.

## Bilaterale Interaktionsregeln

Es kommt oft vor, dass ein Benutzer seine Einstellungen an einem Simulationsmodell nicht konsequent mit den besprochenen drei Einstellungen (ACTIVE, PASSIVE, NO INTERACTION) durchführen kann, weil es einzelne Teile gibt, welche sich nicht in dieses Schema zwingen lassen oder eine Ausnahme bilden, welche diese Einstellungen verhindern. Für diesen Zweck gibt es in sonar sog. bilaterale Regeln welche genau dieses Problem lösen. Der Benutzer kann die ursprünglichen Einstellungen wie gewünscht mit den drei Interaktionsregeln durchführen und dort wo es Ausnahmen gibt sogenannte 'bilaterale Interaktionsregeln' setzen. Eine bilaterale Regel gilt, wie der Name sagt, nur für eine einzelne Objekt-Paarung und kann individuell gesetzt werden. Die allgemeinen Interaktionsregeln bleiben in beiden Objekten weiterhin gesetzt und auch genutzt, sie werden lediglich für eine spezielle Objektbegegnung mit einer neuen bilateralen Regel mit höherer Priorität ausser Kraft gesetzt.

Das Setzen einer bilateralen Interaktionsregel geht wie folgt vor sich:

```
select 2 objects  
Menu / Functions / Interaction / Edit Bilateral Rule...
```



**FIGURE 97.** der Dialog Edit Bilateral Interaction Rule. In diesem Dialog werden stellvertretend immer Kugeln dargestellt, was aber nicht heisst, dass die betreffenden Objekte auch Kugeln sind.

der Benutzer setzt hier mit einer hohen Priorität fest, ob zwei ausgewählte Objekte letztlich miteinander interagieren sollen oder nicht. Die linke Hälfte des Dialogs ist reine Information und erklärt die Standard Interaktionsregeln. Die rechte obere Hälfte des Dialogs stellt die bisherigen Standard Einstellungen der beiden Objekte dar. Dies als Hilfestellung ob eine bilaterale Regel überhaupt nötig ist.

## Automatisch gesetzte Bilaterale Regeln

Werden zwei Objekte miteinander verlinkt, dann wird für diese beiden Objekte immer automatisch eine bilaterale Interaktionsregel gesetzt, welche besagt, dass die beiden Objekte nicht miteinander kollidieren sollen. In der Regel geht das in Ordnung, denn Objekte welche miteinander verlinkt werden überlappen sich oft gegenseitig. In der überwiegenden Anzahl aller Fälle müsste der Benutzer andernfalls diese Regeln immer manuell setzen, was umständlich wäre. Will der Benutzer umgekehrt, dass eine automatisch gesetzte Regel wieder ausser Kraft gesetzt wird, dann kann er diese einfach wieder löschen.

## Die Funktion 'überflüssige Interaktionsregeln löschen'

Oft werden beim Bau eines Modells automatisch bilaterale Regeln gesetzt, wo sie eigentlich überflüssig wären. So zum Beispiel, wenn zwei passive Objekte miteinander verlinkt werden oder wenn eines der beiden Objekte auf die Eigenschaft 'no interaction' gesetzt wurde. Überflüssige Regeln dieser Art stören meistens nicht unmittelbar und werden oft einfach belassen. In Modellen mit einer sehr grossen Anzahl von Objekten und entsprechend vielen bilateralen Interaktionsregeln, machen sich diese überflüssigen Regeln allerdings in der Rechenleistung negativ bemerkbar. Die Anwendung der folgenden Funktion entfernt alle überflüssigen bilateralen Interaktionsregeln. Die Funktion sollte mit Vorteil erst ganz am Ende eines Modellerzeugungs-Prozesses angewendet werden, wenn sichergestellt ist, dass alle Interaktions-Einstellungen ihren endgültigen Stand erreicht haben.

Menu / Interaction / Delete All Obsolete Bilateral Rules

## Interaktionsregeln mit Namen

Interaktionsregeln können auch gruppenweise erlassen werden. Der Benutzer wählt die Namensgebung seiner Objekte derart, dass er diese anschliessend nach seinen Vorstellungen miteinander in Beziehung setzen kann. In sonar-Lab öffnet sich der folgende Dialog mittels:

Menu / Interaction / Edit Rules by Name...

Im oberen Teil des Dialogs werden dem Benutzer alle bisher vergebenen Namen seines Modells zweimal angezeigt. Indem aus jeder dieser beiden Listen ein Name ausgewählt wird, definiert der Benutzer eine Namensgruppe, die untereinander eine bestimmte Interaktionseigenschaft bekommen soll. Der Dialog bietet aber noch mehr. Zusätzlich kann die gewählte Gruppe bezüglich ihrer Wirkungsrichtung eingeschränkt werden. In der Mitte des Dialogs befindet sich ein 'popup-menu' welches die Selektion der eigentlichen Interaktionsregel zulässt. Hier kann der Benutzer nicht nur Ja oder Nein sagen zu einer möglichen Interaktion der gewählten Paarungen, er hat die weiteren Möglichkeiten: 'Ja, aber nur RADIAL' oder 'Ja, aber nur AXIAL'. Die untere Hälfte des Dialogs zeigt in einer Übersicht alle definierten Einstellungen, wobei diese Regeln am linken Rand der Liste auch einzeln aktiviert oder ausgeschaltet werden können.

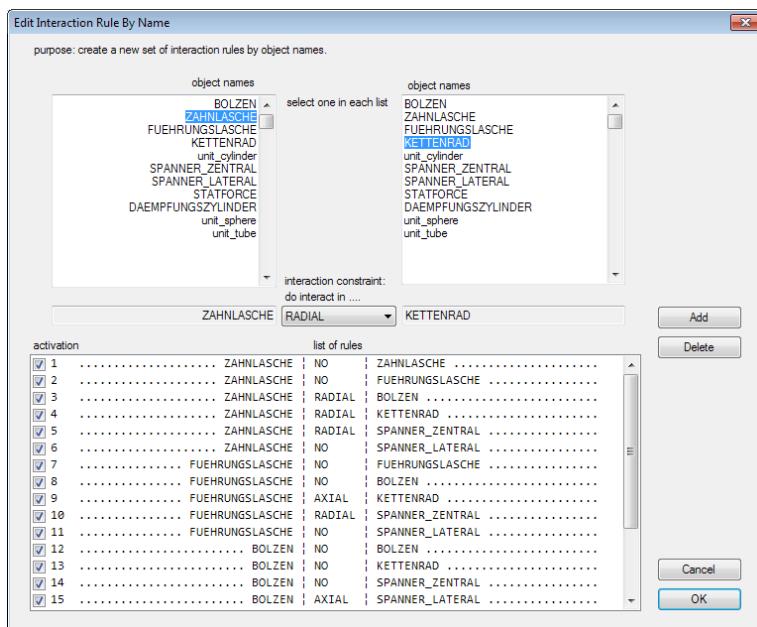


FIGURE 98. Der Dialog 'Edit Interaction Rule by Name' wie er sich in sonar-LAB darstellt

Im Programm sonar-SIM findet sich derselbe Dialog mit eingeschränkten Möglichkeiten. Regeln hinzufügen oder löschen ist dort nicht möglich, aber das Verändern oder Aktivieren bestehender Regeln ist vorgesehen.

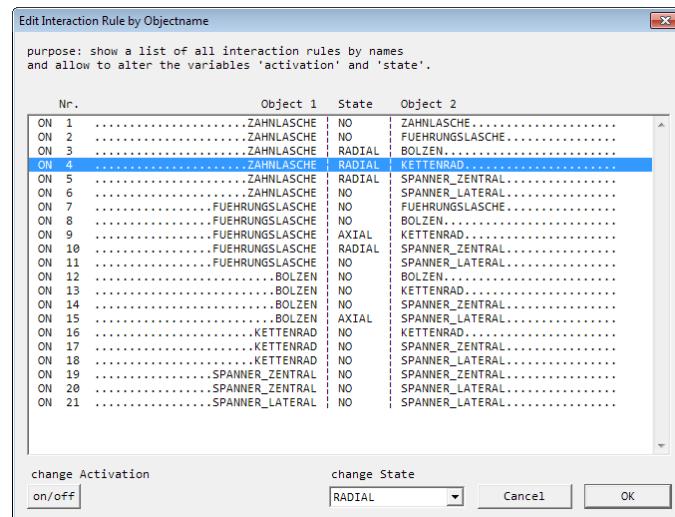


FIGURE 99. Der Dialog 'Edit Interaction Rule by Name' in sonar-SIM

## Ablauf der Interaktionssteuerung

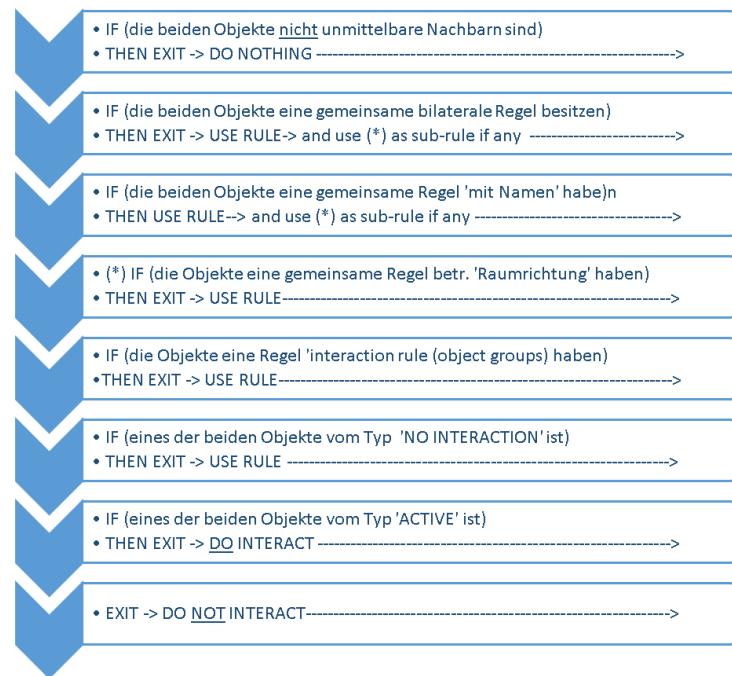


FIGURE 100. Der Ablauf des Entscheidungsprozesses zur Interaktionskontrolle

Im eigentlichen Simulationsprogramm läuft der Entscheidungsprozess, ob zwei bestimmte Objekte miteinander interagieren, nach dem dargestellten Schema ab. Der Prozess läuft dabei von oben nach unten und auf jeder Linie kann der Ablauf verlassen werden, falls auf der betreffenden Linie eine definitive Entscheidung getroffen wurde. Die obersten Fragestellungen im Schema haben die höchste Priorität. Falls bereits diese ein Ergebnis herbeigeführt haben, dann wird der Ablauf nicht weiter fortgesetzt. Andernfalls kommen weitere Tests darunter mit geringerer Priorität zum Zug. Dies setzt sich solange fort bis im äussersten Fall die unterste Linie erreicht wird, welche dann eine Interaktion als nicht gegeben einstuft.

Falls es zwischen den beiden Regeln

- Bilateral Interaction Rule
- Interaction Rule by Name

Regel Konflikte

zu einem Konflikt kommt von der Art, dass die bilaterale Regel sagt 'keine Interaktion' und die 'Interaktionsregel mit Namen' sagt z.B. 'RADIAL', dann gilt, dass die bilaterale Regel die höhere Priorität hat. Mit anderen Worten, im soeben beschriebenen Beispiel würde folglich keine Interaktion stattfinden, weil die bilaterale Regel es so festlegt.

Umgekehrt: Sagt die bilaterale Regel 'Interaktion = Ja' und die 'Interaktionsregel mit Namen' sagt 'NO DIRECTION, dann ist die erste bzw. die bilaterale Regel im Prinzip überflüssig, sie stört aber auch nicht, weil kein Widerspruch entsteht. In diesem Fall sagt die bilaterale Regel: "von meinem Gesichtspunkt aus gesehen ist eine Interaktion statthaft" und gibt die Entscheidung an die nächste Stufe darunter weiter. Diese entscheidet dann aber mit 'NO DIRECTION' dass definitiv keine Interaktion stattfindet.

# Grafische Darstellungen

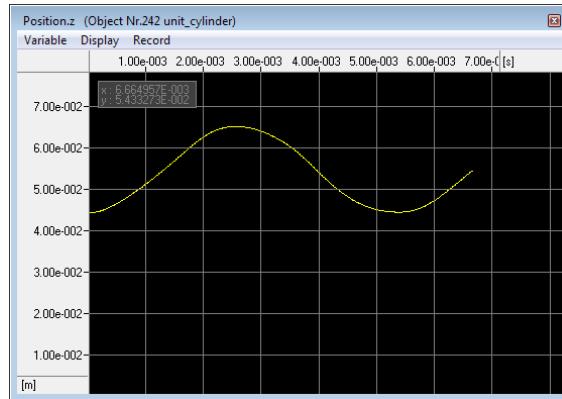
---

## Einleitung

---

Neue Grafische Darstellungen können jederzeit vor oder während einer Simulation geöffnet und eingestellt werden. Wenn während einer laufenden Simulation eine neue Grafik eingerichtet werden soll, dann empfehlen wir die Simulation zwischenzeitlich anzuhalten.

sonar-SIM / Menu / Tools / New Graphic



**FIGURE 101.** Grafik Window mit grafischer Darstellung des zeitl. Verlaufs einer Variablen

Grafische Darstellungen können im Prinzip jede geometrische und jede physikalische Variable jeder Art von Objekt als Funktion der Zeit oder wieder als Funktion einer geometrischen oder physikalischen Variablen darstellen. In diesem Sinne kann also nicht nur die zeitliche Entwicklung einer Variablen grafisch dargestellt werden, sondern es kann z.B. auch der Pfad der Bewegung eines Objektes, um nur ein Beispiel zu nennen, aufgezeichnet werden. Konkret gesprochen könnte man in X-Richtung die x-Koordinate von Objekt (i) und in Y-Richtung die y-Koordinate desselben Objektes darstellen. Dies gäbe zusammen die Projektion des Bewegungspfades von Objekt (i) auf die X-Y-Ebene. Würde man dann, um dieses Beispiel noch weiter zu führen, in einer weiteren Grafik dasselbe für die X-Z-Ebene tun, dann bekäme man zusammen eine vollständige Information über die 3-dimensionale räumliche Bewegung des Objektes.

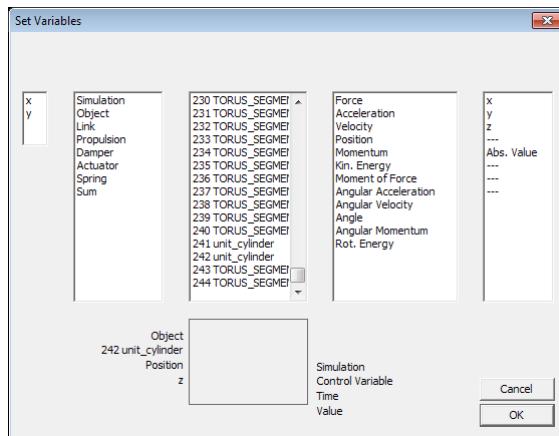
## Grafik Variablen

---

Die erste Aufgabe nach dem Öffnen einer neuen Grafik ist die Auswahl der Variablen. Was soll dargestellt werden? Die Bedienung der Grafik geschieht über das grafikeigene Menu im Kopfteil des Grafik-Windows. Das Menu besteht aus drei Namen. Der erste Name heisst 'Variable'.

Graphic Menu / Variable / Set Variables...

Dies führt uns zu folgendem Dialog:



**FIGURE 102.** Der Dialog 'Set Variable' eines Grafik Windows

In diesem Dialog wird vom Benutzer bestimmt, was in der Grafik dargestellt werden soll. Wir tun das in zwei Schritten, einmal für die X-Achse und dann dasselbe nochmals für die Y-Achse.

Im ersten Schritt wählen wir links oben im Dialog das 'x', welches stellvertretend für die x-Achse steht. Sobald wir das 'x' ausgewählt haben, wird die nachfolgende Liste 'Object Type' ausgefüllt. Wählen wir dort 'Object', dann werden uns wieder in der nächsten Liste alle Objekte angezeigt, usw. Wir tun das bis wir in der Liste ganz rechts angekommen sind. Ist das der Fall, dann ist die Variablen Wahl für die x-Achse abgeschlossen und in der symbolischen Grafik im Dialog wird all das was selektiert wurde angezeigt.

### Schritt 1

Im zweiten Schritt wiederholen wir den Vorgang für die y-Achse. Wir selektieren demzufolge das 'y' ganz links oben im Dialog und wiederholen den gesamten Vorgang bis auch die Selektion der y-Achse abgeschlossen ist. Anschliessend verlassen wir den Dialog mit 'ok'.

### Schritt 2

## Die Funktion der Grafik

Sind die darzustellenden Variablen einer Grafik selektiert, dann funktioniert die Grafik weitgehend automatisch. Die Grafik muss nicht mehr anderweitig aktiviert werden. Sobald die Simulation läuft, erwacht auch die Grafik aus dem Schlaf und stellt das dar, was der Benutzer der Grafik zugeschaut hat. Die Skalierung der Grafik geschieht automatisch. Wann immer eine Kurve während dem Zeichnen im Grafikfenster den Rand der Grafik überschreitet, wird sie neu skaliert. Die Grafik ist also immer bestrebt die gesamte Grafikkurve zur Darstellung zu bringen und umgekehrt, den ihr zur Verfügung gestellte Platz optimal zu nutzen.

Die Grafiken dürfen am Bildschirm beliebig in ihrer Grösse verändert und bezüglich ihrer Position verschoben werden. Alle Grafiken werden beim Sichern mit dem Modell gespeichert und zwar inklusive dargestellten Grafikkurven. Wird ein Modell mit Grafiken zu einem späteren Zeitpunkt erneut geöffnet, dann werden auch die gespeicherten Grafik-Windows automatisch wieder geöffnet, die Grafikvariablen sind erneut bekannt und die Darstellung der Grafikkurve im Grafikfenster wird wieder hergestellt.

## Datenaufzeichnung

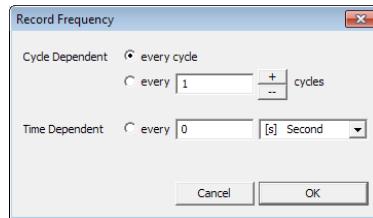
Mit dem bisherigen Vorgehen ist zwar die visuelle Darstellung der Grafikkurve im Grafikfenster gewährleistet, das heisst aber nicht, dass diese Daten dem Benutzer in Zahlenform zur Verfügung stehen. Will der Benutzer diese Zahlenreihen zwecks späterer Verwendung zur Verfügung haben,

dann muss er sie aufzeichnen lassen. Diese Aktivierung der Aufzeichnung muss im Übrigen vor der Simulation geschehen. Ein nachträgliches Ermitteln der Daten ist grundsätzlich unmöglich.

Graphic Menu / Record / Record Data...

Zwei Informationen sind dem System mitzuteilen

1. Wie oft, d.h. in welchem Zeitintervall, sollen die Daten gespeichert werden? (-> Dialog Record Frequency)
2. Unter welchem Filenamen sollen die Daten gespeichert werden. (-> Standard File Dialog zum Speichern von Files)



**FIGURE 103.** Der Dialog Record Frequency

Der Dialog 'Record Frequency' ist dazu da, die Wiederholrate der Datenaufzeichnung einzugeben. Normalerweise benutzen wir ein zeitabhängiges Kriterium und drücken entsprechend die unterste Radiotaste. Direkt daneben legen wir das Zeitintervall fest, z.B. '0.1' Millisekunde, oder was immer Sie für angebracht erachten. Sobald Sie die 'OK'-Taste drücken wird sich automatisch der zweite Dialog für die Eingabe des Filenamens öffnen.

Damit ist die Datenaufzeichnung aktiv und das System wird in der Folge in den festgelegten Zeitabständen ein Wertepaar des betreffenden Grafikfensters an das Datenfile anfügen. Das entstehende File besteht letztlich aus zwei Kolonnen, z.B. die Zeitwerte in der ersten Kolonne und die zugehörigen Geschwindigkeitswerte in der zweiten, falls dieses Wertepaar für die Grafik selektiert wurde.

## Grafik Zurücksetzen

---

Graphic Menu / Display / Reset

Die während einer Simulation im Grafik Fenster aufgezeichnete Kurve wird zurückgesetzt und die Bilddarstellung im Grafikfenster beginnt wieder von vorne. Der Einsatz dieser Funktion wäre z.B. denkbar, wenn sich die Kurve in der Grafik asymptotisch einem maximalen Wert nähert und man genau sehen möchte, wie sich die Werte in der letzten Phase annähern. Weil das Zurücksetzen der Grafik auch die Skalierung neu vornimmt, hat die zu beobachtende y-Skala nach dem 'Reset' eine bessere Auflösung.

# Filmaufzeichnungen

---

## Quicktime Filme aufzeichnen

---

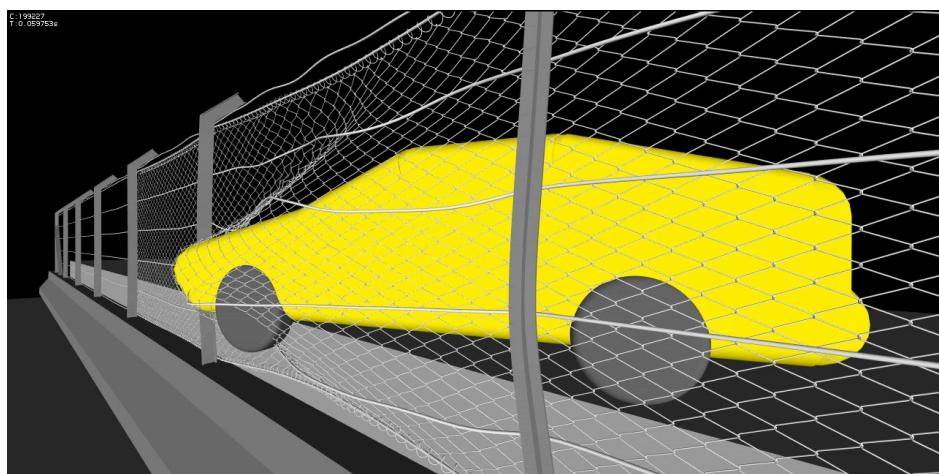
‘Quicktime’ ist ein Film- und Video-Format der Firma Apple. Es war eines der ersten Filmformate auf Personal Computer überhaupt und wird bis heute auch von Microsoft unterstützt. Quicktime liegt in zwei Versionen vor.

- Quicktime Player
- Quicktime Pro

Beide Produkte können von der Apple homepage heruntergeladen werden. Während das erste gratis ist, kostet das zweite \$30. Wir empfehlen dringend, diese zweite PRO-Version herunterzuladen, denn nur mit ihr ist das Editieren und Komprimieren der Filme möglich.

[www.apple.com](http://www.apple.com)

Quicktime wird in sonar zur Aufzeichnung von Simulationen verwendet, indem in periodischen Abständen Bilder des Simulationsgeschehens an einen Film angefügt werden. Später, wenn die Simulation abgeschlossen ist, wird der Film mit den mitgelieferten Quicktime Funktionen komprimiert und ggf. auch zusammengeschnitten. Filmaufzeichnungen sind ein sehr wichtiges Werkzeug für die Analyse des Geschehens. Die Aufzeichnungen können in beliebiger ‘slow motion’ erfolgen und sich auf kleine Ausschnitte des Modells konzentrieren. Die Filme können mit den parallel erfolgten Datenaufzeichnungen in grafischer Darstellung kombiniert werden, indem die Zeitmarken im Film mit dem zeitlichen Verlauf der Messkurven in Beziehung gesetzt werden.



**FIGURE 104.** Filmausschnitt aus einer Simulation einer Schutzanlage

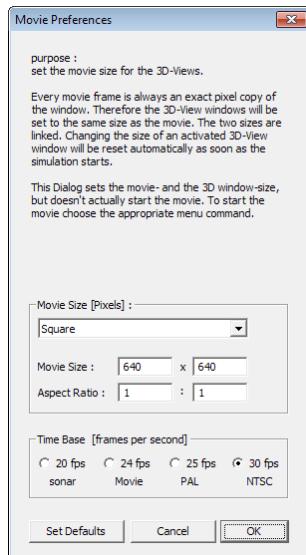
Filme werden in sonar immer so produziert, dass Bilder gleicher Grösse aus den 3D-Ansichten des Modells entnommen und an den Film übermittelt werden. Für die zu erzeugenden Filme stehen dem Benutzer mehrere Filmgrößen zur Verfügung. Hat der Benutzer am Bildschirm mehrere 3D-Ansichten offen, dann werden entsprechend auch mehrere Filme hergestellt. Die Bildrate und die Abspielgeschwindigkeit werden vom Benutzer bestimmt. Die Frequenz der Bildaufzeichnung bezieht sich nicht auf die Weltzeit bzw. die Uhr, die der Benutzer an seinem Handgelenk trägt, sondern auf die eigentliche Simulationszeit, wie sie am unteren Rand des Bildschirms im Statusfenster angezeigt wird. Dauert ein physikalischer Vorgang, wie oben im Bild dargestellt, z.B. eine Sekunde, dann würde der Benutzer als Bildrate einen Wert von 1/300 Sekunde wählen und in dieser Zeit von 1 Sekunde insgesamt 300 Bilder erzeugen, unabhängig davon wie lange die Simulation selbst dauert. Die Simulation kann bei komplexen Problemstellungen auch mal

mehrere Tage dauern. Am Ende hätte der Benutzer also einen Film von 300 Bildern welche mit einer Bildrate von 30 Bildern pro Sekunde abgespielt werden. Somit würde die Kollision des Autos mit der Schutzanlage, wie oben abgebildet, zehn Sekunden dauern, was vernünftig erscheint. Wird die Simulation zwischenzeitlich angehalten, dann hat das keinen Einfluss auf den Film, denn der Film richtet sich in diesem Sinn nach der Simulations-Uhr auf dem Bildschirm, und die bleibt bei einem Stopp auch stehen.

## Filmformat eingeben

Die Filmgrösse ist eine Eingabe welche nicht mit dem Modell gespeichert wird. Wir geben die bevorzugte Filmgrösse immer vor dem Simulationsstart mit dem folgenden Dialog ein:

sonar-SIM / Menu / File / Quicktime Movie Preferences...



**FIGURE 105.** Der Dialog 'Movie Preferences' für die Einstellungen des Filmformats

Bei diesem Vorgang werden die geöffneten 3D-Ansichten auf dem Bildschirm automatisch auch auf das eingegebene Format um-skaliert. Sie dürfen die Szenerien in den Filmen nachher nochmals nach Belieben neu einstellen, aber Sie sollten die Fenstergrösse der 3D-Ansichten nicht mehr ändern, denn die sind jetzt abgestimmt mit der Filmgrösse. Im Weiteren legen Sie bereits hier die Abspielrate des Films fest. Sie haben dazu vier Möglichkeiten die Anzahl Bilder pro Sekunde (fps: frames per second) festzulegen.

## Filmaufzeichnung starten

Damit ist die Filmaufzeichnung zwar eingestellt aber noch nicht gestartet. Die Aufzeichnung wird erst unmittelbar vor Simulationsbeginn mit dem folgenden Befehl aktiviert:

Menu / File / Start Recording Quicktime Movie

Mit diesem Menu wird Ihnen ein Standard File Dialog präsentiert wo Sie den Filenamen und den Speicherort des Films eingeben. Ist das geschehen, dann wird dieser Befehl im Menu deaktiviert und mit einem Häkchen versehen. Jetzt ist die Filmaufzeichnung quasi freigegeben und aktiviert. Solange Sie die eigentliche Simulation aber nicht starten, geschieht nichts. Erst wenn die Simulation läuft und nur solange sie läuft, werden in den vorgesehenen Zeitintervallen welche sich auf die Simulationszeit beziehen, Bilder an den Film angefügt.

## Filmaufzeichnung beenden

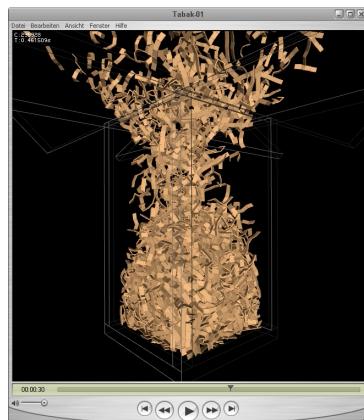
Es steht Ihnen frei die Filmaufzeichnung anzuhalten, während die Simulation ggf. noch weiter läuft. Meistens stoppt man die Filmaufzeichnung aber zusammen mit der Simulation indem der Befehl

ausgelöst wird. Dieser Befehl schliesst den Film definit ab und erklärt ihn als beendet. Der Film kann anschliessend nicht wieder gestartet werden. Will der Benutzer in einem gegebenen Fall trotzdem noch einmal eine Fortsetzung anfügen, dann startet er einen neuen Film. Später beim Komprimieren des Films kann er auch mehrere Filme aneinanderfügen und daraus einen einzigen Film produzieren.

## Quicktime Filme komprimieren

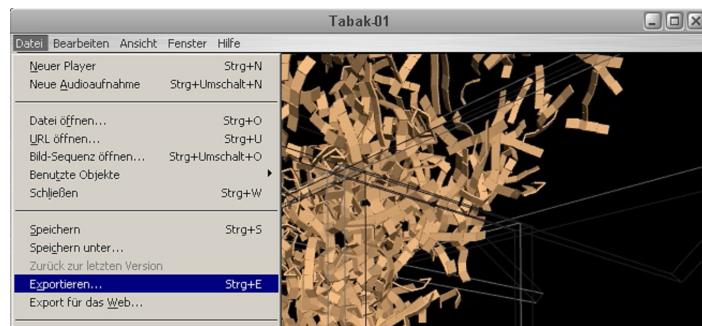
Ist ein Film wie oben beschrieben aufgezeichnet worden, dann kann er mit jedem Quicktime Player bereits geöffnet und abgespielt werden. Die Dateigrösse wurde aber noch nicht komprimiert. Der Film dürfte deshalb noch viel Platz in Anspruch nehmen auf seinem Datenträger. Zum zweiten könnte das Abspielen, aus dem gleichen Grund, ev. noch etwas holperig von statten gehen. Drittens, und das ist der wichtigste Punkt, können wir den Film noch nicht schneiden, manipulieren, zusammensetzen, usw.

Mit der PRO-Version von Quicktime ist das alles möglich. Quicktime PRO ist ein kleines minimales Filmstudio, mit dem wir unsere Filme, die wir mit unseren Simulationen erstellten, bearbeiten können. Die Anforderungen, die wir an unser kleines Filmstudio stellen sind dabei ganz andere, als für Jemanden der Filme aus Videokameras editieren will. Die Filme aus Simulationen sind im Prinzip bereits perfekt in Sachen Helligkeit, Kontrast, Auflösung usw., wir wollen Sie nur noch zusammenschneiden und vor allem komprimieren, damit sie schnell abgespielt werden können und wenig Platz in Anspruch nehmen.



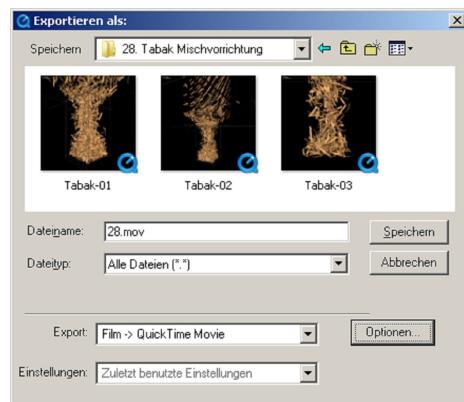
**FIGURE 106.** Ein beispielhafter Film der während einer Simulation aufgezeichnet wurde (Es handelt sich um einen Schüttvorgang mit Tabakpartikeln).

Wir spielen im Folgenden kurz durch, wie ein solcher Komprimierungs-vorgang in unserem Fall aussieht. Gehen wir davon aus, dass wir während einer Simulation einen Film wie abgebildet im Format 640 x 640 Pixel aufgenommen haben und mit der Software Apple Quicktime PRO öffnen können.



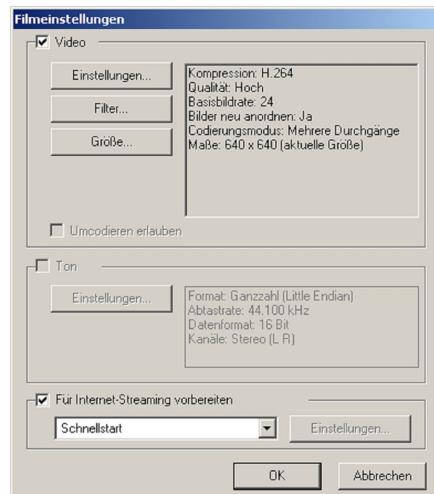
**FIGURE 107.** Einen bestehenden Film in ein anderes Format exportieren

Die Komprimierung eines Films beginnt im Quicktime-Film-Menu mit 'Exportieren As', was den folgenden Dialog auf den Bildschirm bringt



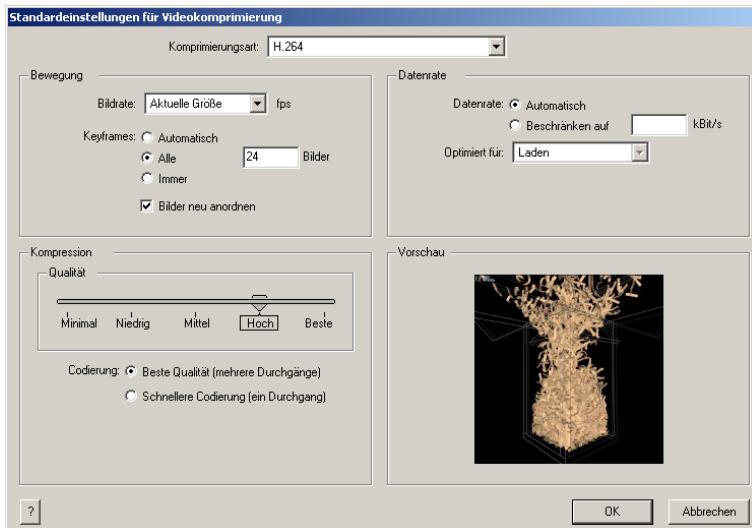
**FIGURE 108.** Auswahl eines Filmformates für den Export eines bestehenden Films

Im Dialog 'Exportieren als...' wird ein Filename mit der Endung '.mov' eingegeben, das Export-Format 'Film -> Quick Time Movie' gewählt und die Taste [Option] selektiert. Die führt uns weiter auf den folgenden Dialog:



**FIGURE 109.** Der Quicktime Dialog zur Auswahl von zusätzlichen Film Optionen

In diesem Dialog drücken wir auf die Taste [Einstellungen] was uns den nächsten Dialog auf den Bildschirm bringt:



**FIGURE 110.** Der Quicktime Dialog zur Festlegung der Komprimierungs-Methode und -Qualität

Im letzten Dialog sind Einträge angezeigt, wie sie mit sonar gut funktionieren. Es ist Ihnen natürlich freigestellt, weitere Einstellungen zu testen und sie mit den entstehenden Differenzen in den Resultaten zu vergleichen. Wichtig scheint uns hier die Komprimierungsart 'H.264' zu sein, welche für technische und grafische Filme, wie sie mit Simulationen entstehen, geeignet scheinen. Wenn wir anschliessend zum ersten Dialog zurückkehren, wo wir den Filennamen eingaben, dann wird der gesamte Kompressionsprozess in Gang gesetzt und liefert uns nach ein paar Sekunden oder Minuten einen neuen, komprimierten Film.

Achten Sie darauf, dass der Quicktime Film vor diesem ganzen Komprimierungsprozess aus zwei Files besteht, einem Film-File und einem Resourcen-File. Diese beiden Files dürfen nicht voneinander getrennt und auch nicht umbenannt werden, sondern müssen ggf. immer zusammen verschoben werden. Mit einem Doppelklick auf das Film-File greift der Film auch auf das gleichlautende Rersourcen-File zu, welches er neben sich erwartet. Ist der Film aber erst einmal komprimiert, dann entfallen diese Einschränkungen. Der Film ist anschliessend ein eigenständiges File und kann beliebig umbenannt und verschoben werden. Der Film muss einfach die Endung '.mov' behalten.

## Für Fortgeschrittene: Bewegte Kamera

Ohne weitere Vorkehrungen entstehen während den Simulationen Filme mit einer bestimmten festen Kameraposition. Alle Bewegungen spielen sich vor unserem statischen Auge ab. Vielleicht muss man bei der Einstellung der Szenerie vor der Simulation sogar darauf achten, mit der Kamera nicht zu nahe an bestimmte Teile heranzugehen, da man sie sonst während der Simulation aus dem Blickwinkel verliert. Natürlich kann man während einer Simulation Nachstellungen der Kamera vornehmen und den Bildausschnitt einer 3D-Ansicht nachstellen. Da die Filmfunktion immer die tatsächlichen Bilder der Ansichten an den Film anfügt, werden so auch immer die neusten Einstellungen der Ansichten in die Filme übernommen. Allerdings ist ein Nachstellen dieser Art nicht sehr schön, wenn wir den Film später betrachten. Sprünge in Filmen sind meistens störend. Man möchte stattdessen möglichst weiche Übergänge und Schwenks der Kamera. Das ist mit sonar möglich. Mit der Funktion 'Camera Movement' kann die gesamte Bewegung der Kamera für die bevorstehende Simulation programmiert werden. Diese Kamerafahrt zu programmieren ist zudem denkbar einfach und intuitiv zu bewerkstelligen mit dem folgenden Dialog:

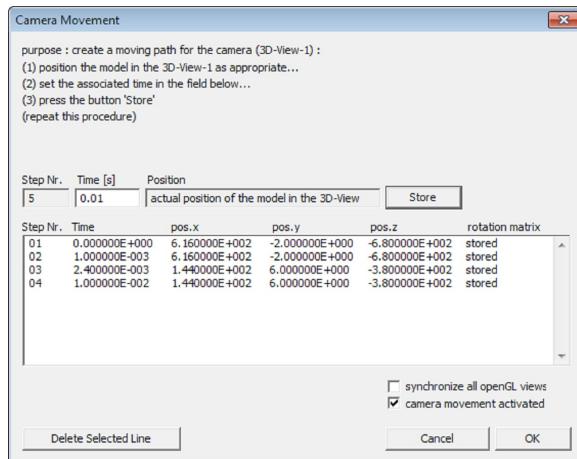


FIGURE 111. Der Dialog Camera Movement

- Der Benutzer stellt in der ersten 3D-Ansicht grafisch interaktiv eine Szenerie ein.
- Im bereits offenen Dialog 'Camera Movement' gibt der Benutzer eine zugehörige Zeit ein und drückt auf [Store]
- dieser Vorgang (Szenerie einstellen + Zeit eingeben) wird beliebig oft wiederholt mit aufsteigenden Zeitwerten
- Am Ende, wenn alle gewünschten Kamerapositionen mit den zugehörigen Zeiten eingegeben sind, setzt man noch die Checkbox 'camera movement activated' und verlässt den Dialog mit [OK].

Meistens kommt man mit ein paar wenigen Kamerapositionen aus, so wie im Dialog oben dargestellt.

Die Funktion 'Camera Movement' wird später bei der Durchführung einer Simulation zuerst an die erste Position fahren und im weiteren Verlauf der Zeit ständig die Position und Drehung der Kamera zur zweiten gespeicherten Position linear interpolieren bis diese erreicht ist. Ab der zweiten Position geht es genauso weiter bis zur dritten Position und so fort. Falls die letzte gespeicherte Position überschritten wird, dann bleibt die Kamera in dieser Position stehen und filmt weiter. Mit dieser Art die Kamerabewegung zu programmieren ist es auch möglich im Verlaufe einer Kamerafahrt für eine gewisse Zeit an einem Ort stehen zu bleiben, indem man für zwei aufeinander folgende Zeiten die gleiche Position speichert.

Die Kamerafahrt, wie mit diesem Dialog programmiert, wird beim Speichern des Modells mit in das Modellfile aufgenommen. Bei erneutem Laden des Modells muss die Kamerafahrt allerdings erneut mit der betreffenden Checkbox aktiviert werden.

# Preferences

## Programm-spezifische Einstellungen

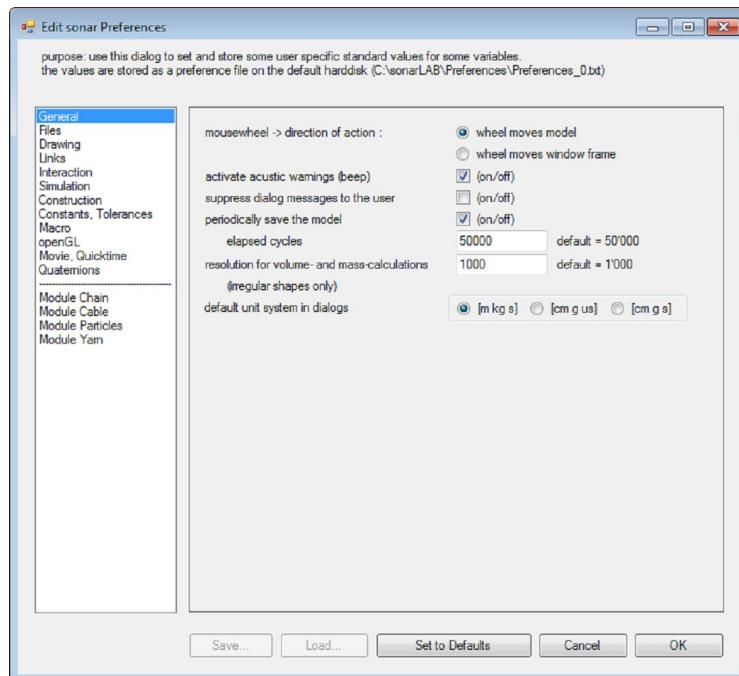
Das 'Preference-File' ist eine Datei welche die benutzerspezifischen Einstellungen der sonar-Programmumgebung an einem zentralen Ort speichert. Der Zugriff auf das Preference File erfolgt mittels:

Menu / Edit / Preferences...

Beim Öffnen des Dialogs greift das System auf folgendes File zu:

C://sonarLab/Preferences/Preferences\_0

Wird dieses File nicht gefunden, dann erzeugt das System am Ort wo das File erwartet wird ein neues Preference File mit den sog. Default-Einstellungen.



**FIGURE 112.** Der Dialog 'Edit sonar Preferences' mit dem selektierten Panel 'General'

Der Ort der Speicherung des Preference Files lässt sich nicht verändern. Die Adresse ist im Kopf des Dialogs nochmals angeschrieben. Der Ort aller anderen Files kann jedoch im Panel 'Files' (siehe Liste im Preference-Dialog) eingegeben werden.

Alle Einstellungen in diesem Dialog sind Installations-spezifisch. Sie beziehen sich auf den betreffenden Computer auf dem sonar installiert ist.

## Hinweise

Wir geben im Folgenden Hinweise zu ein paar Einstellungsmöglichkeiten des Dialogs Preferences.

## Quaternonen (Quaternions)

Wir beschäftigen uns im Programm sonar-SIM programm-intern intensiv mit Drehungen von Objekten. Die Lage eines jeden Objektes wird im Allgemeinen durch eine Drehmatrix und einen Translationsvektor bestimmt. Während einer laufenden Simulation ändern sich diese Größen kontinuierlich. In Anbetracht der Tatsache, dass eine Simulation in der Regel viele Millionen Rechenzyklen durchläuft, ergeben sich mit den einfachen 3-dimensionalen Matrizen gewisse Schwierigkeiten. Zum einen entstehen auf numerischem Weg Drifteffekte, zum anderen können Matrizen in sog. deadlock-Stellungen geraten. Diese deadlock-Situationen kann man sich anschaulich an einem kardanisch aufgehängten Kreisel veranschaulichen, bei dem die Kardanrahmen durch gewisse Drehungen in ein und dieselbe Ebene geraten und dadurch auf bestimmte Lageänderungen nicht mehr ansprechen. Dasselbe kann rein rechnerisch auch in unseren Simulationen mit Objekten und ihren Matrizen geschehen. Um diese Probleme zu umgehen und eine höhere Stabilität der Lagekontrolle der Objekte zu erreichen, bedient man sich sogenannter Quaternonen. Die Quaternonen erweitern einen 3-dimensionalen in einen 4-dimensionalen Vektorraum ähnlich wie die komplexen Zahlen eine Erweiterung der 1-dimensionalen reellen Zahlengerade in eine 2-dimensionale Zahlenebene darstellen. Tatsächlich werden in sonar alle infinitesimalen Lageänderungen der Objekte in einem 4-dimensionalen Raum berechnet und verwaltet und lediglich für die Darstellung jeweils in den 3-dimensionalen Raum zurücktransformiert. Quaternonen haben den Vorteil, dass es keine deadlock Stellungen mehr gibt und dass Drifteffekte im vierdimensionalen Raum besser unter Kontrolle gehalten werden können.

Die meisten Benutzer werden die Einstellungen zu den Quaternonen in den Preferences nie verändern. Die dargestellten Zahlenwerte und die Einstellungsmöglichkeiten waren vor allem während der Entwicklung der Software wichtig und spielen auch eine Rolle während der Weiterentwicklung und bei der Fehleranalyse von aussergewöhnlichen Problemen.

# Reports & Logfiles

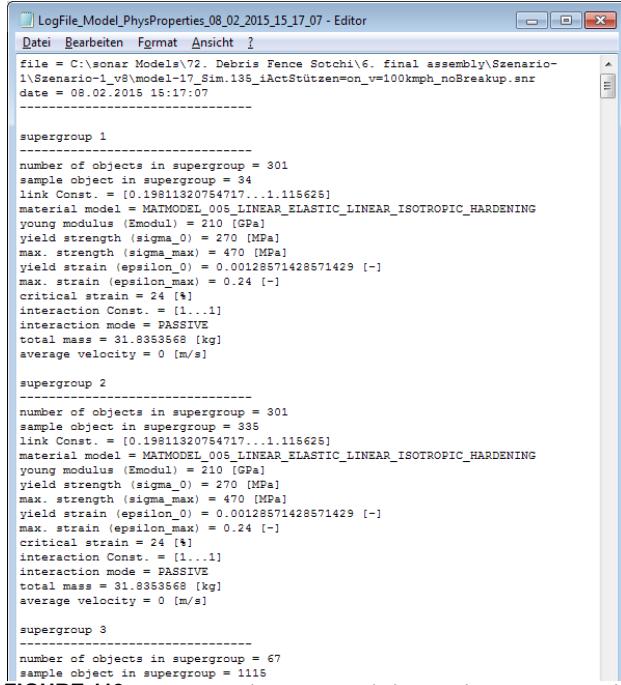
## Report (Physical Properties)

Der Report stellt die wichtigsten physikalischen Eigenschaften sämtlicher Gruppen und Supergruppen eines Modells im Textformat zusammen. Der Benutzer ist jederzeit in der Lage die Erstellung eines Reports auszulösen indem er den folgenden Menu-Befehl auslöst:

```
sonar-LAB / Menu / File / Print Report / Physical Properties
```

Der Benutzer darf diesen Report beliebig oft in Anspruch nehmen. Bei jeder Auslösung wird ein neuer Report erstellt. Der Speicherort des Reports wird im Dialog 'Preferences' festgelegt. Standardmäßig lautet die Adresse

```
C:\sonarLab\Logfiles\
```



```
LogFile_Model_PhysProperties_08_02_2015_15_17_07 - Editor
Datei Bearbeiten Format Ansicht ?
file = C:\sonar Models\72. Debris Fence Sotchi\6_final assembly\Szenario-1\Szenario-1_v8\modell-17_Sim.135_iActStützen=on_v=100kmph_noBreakup.snr
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 = 31.8355868 [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
young modulus (Emodul) = 210 [GPa]
yield strength (sigma_0) = 270 [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 = 31.8355868 [kg]
average velocity = 0 [m/s]

supergroup 3
-----
number of objects in supergroup = 67
sample object in supergroup = 1115
```

FIGURE 113. Der Output des Reports geladen mit dem Programm Editor

Jedes Mal wenn der Benutzer einen Report auslöst, wird dieser wie folgt bestätigt:

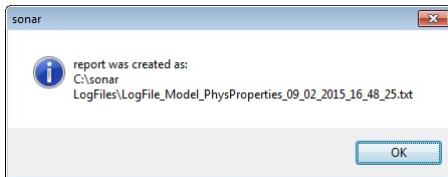


FIGURE 114. Die Bestätigung einer Reportbestellung

## LogFile Geometry (selected object)

sonar-LAB / Menu / File / Print Report / Object Geometry

Das Logfile zeigt die innere Datenstruktur eines Objektes aufgelöst in seine elementaren Konturen, Elemente, Punkte und ihre Koordinaten. Im Weiteren sind die Indizes der inneren Verknüpfungen dieser Einzelheiten aufgeführt. Es kann damit als Kontrollinstrument dienen, wenn der Benutzer die Koordinaten eines Objektes im Detail sehen will, warum auch immer. Ursprünglich während der Entwicklung der Software für diesen Zweck entwickelt, können diese Informationen in gewissen Fällen auch im praktischen Einsatz der Software einen Nutzen haben.

|    | A   | B               | C      | D           | E          | F   | G |
|----|---|-----------------|--------|-------------|------------|-----|---|
| 1  | file = C:\sonar Models\37. IWIS Motorsysteme\2014\Type 1\model-453_PointCurve.snr |                 |        |             |            |     |   |
| 2  | date = 10.2.2015 16:46:48   |                 |        |             |            |     |   |
| 3  |   |                 |        |             |            |     |   |
| 4  | Microbe   | Nr.             | 670    |             |            |     |   |
| 5  |   | nContour        | 4      |             |            |     |   |
| 6  |   | contourDataIdx  | 3205   |             |            |     |   |
| 7  |   | crossSectionIdx | 0      |             |            |     |   |
| 8  | Contour   | Nr.             | 3205   |             |            |     |   |
| 9  |   | cLength         | 14     |             |            |     |   |
| 10 |   | elemDataIdx     | 17214  |             |            |     |   |
| 11 |   | polygonDataIdx  | 17215  |             |            |     |   |
| 12 |   | prev contour    | 0      |             |            |     |   |
| 13 |   | next contour    | 3206   |             |            |     |   |
| 14 | Element   | Nr.             | 17214  |             |            |     |   |
| 15 |   | elemType        | ARC    |             |            |     |   |
| 16 |   | elemLength      | 3      |             |            |     |   |
| 17 |   | pDataIdx        | 202657 |             |            |     |   |
| 18 |   | prevElemIdx     | 0      |             |            |     |   |
| 19 |   | nextElemIdx     | 17213  |             |            |     |   |
| 20 | Data  | Index           | 202657 | -1.75987439 | -8.743388  | 0.6 |   |
| 21 |   | Index           | 202658 | -1.96266639 | -8.398608  | 0.6 |   |
| 22 |   | Index           | 202659 | -2.15417439 | -8.676098  | 0.6 |   |
| 23 | Element   | Nr.             | 17213  |             |            |     |   |
| 24 |   | elemType        | ARC    |             |            |     |   |
| 25 |   | elemLength      | 3      |             |            |     |   |
| 26 |   | pDataIdx        | 202654 |             |            |     |   |
| 27 |   | prevElemIdx     | 17214  |             |            |     |   |
| 28 |   | nextElemIdx     | 17212  |             |            |     |   |
| 29 | Data  | Index           | 202654 | 2.82385861  | -9.525618  | 0.6 |   |
| 30 |   | Index           | 202655 | -2.15417439 | -8.676098  | 0.6 |   |
| 31 |   | Index           | 202656 | -1.33434939 | -12.391248 | 0.6 |   |
| 32 | Element   | Nr.             | 17212  |             |            |     |   |

FIGURE 115. Der Output des Logfiles Geometry (selected Object)

Der Aufbau der Tabelle ist hierarchisch in eine Baumstruktur gegliedert. Zuoberst kommt das Objekt selbst. Im Anschluss kommt die erste Kontur des Objektes welche sich in mehrere Elemente aufgliedert. Jedem Element folgen anschliessend die einzelnen Punkte (Data) mit ihren Koordinaten, usw.

Hinzu kommt eine Indizierungsstruktur. Sehen Sie dazu die Abbildung oben. Im 'Microbe-Abschnitt' ganz oben steht, dass dieses Objekt vier Konturen hat und dass die erste Kontur den Index 3205 besitzt. Anschliessend folgt diese Kontur mit dieser Nummer, welche auf die gleiche Art weiter verzweigt zu den Elementen der Kontur. Die erste Kontur besitzt zudem den Index der vorangehenden und der nächsten Kontur. Auf diese Weise sind alle Einzelheiten des Objektes untereinander verknüpft.

## LogFile Interactions

Das Logfile Interactions zeichnet in chronologischer Reihenfolge die Interaktionsereignisse auf, so wie sie der Reihe nach während einer Simulation geschehen. Die Aufzeichnung geschieht in voller zeitlicher Auflösung. Gemeint ist damit, dass während eines Kontaktes zwischen zwei Objekten in jedem Zyklus ein Eintrag in das betreffende Logfile erfolgt, solange der Kontakt dauert. Jeder Eintrag hält dabei nebst anderen Informationen auch die Interaktionstiefe zwischen den beiden Objekten fest. Daraus lässt sich letztlich ein Interaktionsvorgang in seiner gesamten Entwicklung und im zeitlichen Ablauf nachvollziehen. Auch dieses Logfiles war ursprünglich ein Instrument für die Entwicklung der Software und wurde später in das eigentliche Programm sonar-SIM übernommen.

| A     | B         | C        | D           | E           | F            | G          | H       |
|-------|-----------|----------|-------------|-------------|--------------|------------|---------|
| 1     | cycle Nr. | time [s] | microbe Nr. | microbe Nr. | orientierung | depth      | Test-ID |
| 24145 |           |          |             |             |              |            |         |
| 24146 | 837       | 8.22E-06 | 176         | 189         | 189          | 0.00515702 | 2092 xx |
| 24147 | 837       | 8.22E-06 | 33          | 46          | 46           | 0.00515689 | 2092 xx |
| 24148 | 837       | 8.22E-06 | 94          | 107         | 107          | 0.00515702 | 2092 xx |
| 24149 | 837       | 8.22E-06 | 266         | 279         | 279          | 0.00515703 | 2092 xx |
| 24150 | 837       | 8.22E-06 | 184         | 197         | 197          | 0.00515702 | 2092 xx |
| 24151 | 837       | 8.22E-06 | 102         | 115         | 115          | 0.00515702 | 2092 xx |
| 24152 | 837       | 8.22E-06 | 274         | 287         | 287          | 0.00515702 | 2092 xx |
| 24153 | 837       | 8.22E-06 | 192         | 205         | 205          | 0.00515702 | 2092 xx |
| 24154 | 837       | 8.22E-06 | 49          | 62          | 62           | 0.00515702 | 2092 xx |
| 24155 | 837       | 8.22E-06 | 110         | 120         | 120          | 0.00515696 | 2092 xx |
| 24156 | 837       | 8.22E-06 | 200         | 210         | 210          | 0.00515696 | 2092 xx |
| 24157 | 837       | 8.22E-06 | 57          | 70          | 70           | 0.00515702 | 2092 xx |
| 24158 | 837       | 8.22E-06 | 123         | 136         | 136          | 0.00515689 | 2092 xx |
| 24159 | 837       | 8.22E-06 | 290         | 300         | 300          | 0.00515696 | 2092 xx |
| 24160 | 837       | 8.22E-06 | 213         | 226         | 226          | 0.00515689 | 2092 xx |
| 24161 | 837       | 8.22E-06 | 65          | 75          | 75           | 0.00515696 | 2092 xx |
| 24162 | 837       | 8.22E-06 | 131         | 144         | 144          | 0.00515702 | 2092 xx |
| 24163 | 837       | 8.22E-06 | 221         | 234         | 234          | 0.00515702 | 2092 xx |
| 24164 | 837       | 8.22E-06 | 78          | 91          | 91           | 0.00515689 | 2092 xx |
| 24165 | 837       | 8.22E-06 | 139         | 152         | 152          | 0.00515702 | 2092 xx |
| 24166 | 837       | 8.22E-06 | 229         | 242         | 242          | 0.00515702 | 2092 xx |
| 24167 | 837       | 8.22E-06 | 147         | 160         | 160          | 0.00515702 | 2092 xx |
| 24168 | 837       | 8.22E-06 | 237         | 250         | 250          | 0.00515702 | 2092 xx |
| 24169 | 837       | 8.22E-06 | 155         | 165         | 165          | 0.00515696 | 2092 xx |

**FIGURE 116.** Das Logfile Interactions

Das Logfile kann auf Wunsch in den Preferences aktiviert werden. Default-mässig ist diese Funktion ausgeschaltet. Im Übrigen muss der Einsatz dieses Logfiles mit Bedacht erfolgen, denn die Archivierung sämtlicher Interaktionen aller Objekte in jedem Rechenzyklus ist bei einer grossen Objektzahl sehr speicherintensiv.

## LogFile Events

Um bestimmte wichtige Ereignisse wie ein Überschreiten einer Streckengrenze oder eines Bruchs einer Verbindung infolge Überlastung zu dokumentieren, wurde dieses Logfile geschaffen. Das Logfile hält nicht nur fest was zwischen welchen Objekten geschah, sondern auch wo genau im Modell das Ereignis stattfand. Das Logfile wird in den Preferences aktiviert.

| 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.snr |            |               |                |               |                |          |           |             |              |             |                            |
| 3  | start: Donnerstag 20.06.2013, 21:45:15   |            |               |                |               |                |          |           |             |              |             |                            |
| 4  | computer: D440   |            |               |                |               |                |          |           |             |              |             |                            |
| 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 | 42659         | 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 | 308289   | 0.09246573 | 42644         | unit_cuboid    | 42651         | unit_cuboid    | 163571   | Normal    | 59.17693797 | 291.00031535 | 4010.292157 | link broken                |
| 19 | 309204   | 0.09274623 | 42564         | unit_cuboid    | 42573         | 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                |
| 24 | 347951   | 0.10437033 | 45221         | unit_cuboid    | 45230         | unit_cuboid    | 174464   | Normal    | 40.55923593 | 455.2401979  | 274.1086644 | link broken                |

**FIGURE 117.** Das Logfile 'Events'

## LogFile Macro / Logfile Syntax

sonar-script ist eine Sprache welche zum Zeitpunkt der Ausführung interpretiert und auf Syntaxfehler kontrolliert wird. Bei dieser Gelegenheit erstellt das System jeweils automatisch ein Logfile welches die Zerlegung des scripts in seine Bestandteile aufzeichnet. Dieses Logfile mit dem Namen 'LogFile\_Macro' oder 'LogFile\_Syntax' (je nach sonar-Programm) enthält wie abgebildet die Zerlegung jeder Programmlinie in ihre Bestandteile wie der Interpreter sie erkannt hat. Der

Buchstabe vor jedem Wort bezeichnet dabei den 'Sprachbestandteil' gemäss der folgenden Bedeutung

C: Command  
 Q: Qualifier  
 V: Name  
 K: Klammer  
 M: Komma  
 P: Parameter  
 B: Filename  
 N: Number  
 e: Element  
 o: Object

|                                |                       |                           |
|--------------------------------|-----------------------|---------------------------|
| C      BEGIN                   | C      SELECT         | M      ,                  |
| C      SCRIPT                  | Q      ELEMENT        | P      C_INTERACT_LIN     |
| V      TYPE_1                  | K      (              | M      ,                  |
|                                | e      E3             | N      0.1                |
| C      IMPORT                  | K      )              | K      )                  |
| Q      CONTOUR_LINE_ARC        |                       |                           |
| K      (                       | C      CREATE         | C      SET                |
| e      E1                      | Q      OBJECT         | Q      PROPERTY           |
| M      ,                       | K      (              | K      (                  |
| P      FILENAME                | Q      O1             | Q      O1                 |
| M      ,                       | M      ,              | M      ,                  |
| B      C:\SONAR IMPORT\LA1.TXT | P      PRISM_LINE_ARC | P      C_INTERACT_QUAD    |
| K      )                       | M      ,              | M      ,                  |
| C      CREATE                  | P      SELECTION      | N      0.1                |
| Q      ELEMENT                 | M      ,              | K      )                  |
| K      (                       | P      EXTRUDE        | C      SET                |
| e      E2                      | M      ,              | Q      PROPERTY           |
| M      ,                       | N      0.12           | K      (                  |
| P      CIRCLE                  | K      )              | Q      O1                 |
| M      ,                       | C      MOVE           | M      ,                  |
| N      0.0                     | Q      OBJECT         | P      INTERACT_DIRECTION |
| M      ,                       | K      (              | M      ,                  |
| N      0.0                     | o      O1             | P      RADIAL_ONLY        |
| M      ,                       | M      ,              | K      )                  |
| N      0.0                     | P      ABSOLUTE       | C      SET                |
| M      ,                       | M      ,              | Q      PROPERTY           |
| N      0.0                     | N      0              | K      (                  |
| M      ,                       | M      ,              | o      O1                 |
| N      0.0                     | N      0              | M      ,                  |
| M      ,                       | M      ,              | P      NAME               |
| N      1.0                     | N      0.2            | M      ,                  |
| M      ,                       | M      ,              | B      ZAHNLASCHE         |
| N      0.1605                  | N      0              | K      )                  |
| K      )                       | M      ,              |                           |

**FIGURE 118.** Ein Ausschnitt aus einem Makro/Syntax Logfile in dreispaltiger Darstellung

Dieses Logfile war ein Hilfsmittel bei der Programmierung von sonar script und wird weiterhin als Kontrolldatei für den Fall von Problemen mit der Sprache oder dem Interpreter im System unterstützt.

## Tipps

---

### Fehlersuche

Alle Reports und Logfiles helfen natürlich auch bei der Suche nach den Ursachen von Problemen. Sollte sich ein Modell in einer Simulation nicht so verhalten wie erwartet, indem z.B. gewisse Teile des Modells 'explodieren', Schwingungen unrealistischen Ausmasses entstehen oder gewisse Teile nicht in der Art interagieren, wie man es erwartet, dann sollte man den Fehler in einem ersten Schritt in den Einstellungen des Modells suchen und nicht davon ausgehen, dass die Software falsch rechnet.

Eine gute Such-Strategie verwendet die folgenden Möglichkeiten und Fragestellungen:

- Gibt es ev. von Anbeginn an Objekte die sich überschneiden und gleichzeitig entsprechend ihren Einstellungen (active, passive) miteinander interagieren?
- Wo liegt rein räumlich die Ursache des Problems?

- Lässt sich das Problem auf bestimmte Objekte eingrenzen?
- Falls nicht, dann kann die Simulation wiederholt werden und ein Film des Geschehens erstellt werden.
- Ev. können Nahaufnahmen in höherer Zeitlupe erstellt werden.
- Ev. entsteht nur schwer sichtbar irgendwo ein Problem welches sich aber sichtbar fortpflanzt und schliesslich woanders zu einem 'crash' führt?
- Sind die Link-Konstanten zu niedrig?
- Ist der Zeitschritt zu hoch?
- Ist ev. eine Dämpfung notwendig (Links, Interaktion) oder ist diese zu niedrig eingestellt.?

# Index

---

## Numerics

1-dimensional 2  
2-dimensionale 2  
2-dimensionale Konturen 16  
3-dimensionale Programme 2  
3-dimensionaler Pfad 67  
3D-Polyline 67  
3-Körperproblem 2

## A

Ablauf der Interaktionssteuerung 108

Ablauf einer Simulation 84

Abschusswahrscheinlichkeit eines Flugzeuges 8

abstossende Kräfte 85

abstraktere Funktionen 10

Achse in Bohrung 16

Actio = Reactio 82

AKTIV 104

aktive Objekte 103

Aktuator 19

Aktuatoren 57

Alauf zur Erzeugung eines Links 83

Alle Bewegungen einfrieren 77

analytisch 3

analytisch lösen 2

analytische Lösung 3

Animationsprogramme 10

Ankerpunkt 34

Antriebsstrang 11

Antriebswelle 11

Anwendungen von Objektgruppen 44

ARC 63

Assembly-Language 2

astronomische Dimensionen 87

Aufbau eines Modells 15

Aufbau eines Objektes 15

Aufbau eines sonar Modells 13

Aufprall 85

Auslenkung des Links 91

Austauschformat 11

Auswertungen 13

Autodesk 11

automatische Kollisionskontrolle 13

## B

Basismodul 12, 16

Bausteine 13

Befestigung des Aktuators 58

Beltrami-Haigh 49

Bestandteile eines sonar-Modells 19

Bewegte Kamera 117

Bewegung der Objekte einschränken 77

Bewegung Stoppen 78

Bewegungsgleichung 84

Bewegungsgleichungen 8

Biege-Link 80, 81

Bilaterale Interaktionsregeln 105

Blattfeder 53

Blattfedern 56

Blockmodell 15

Bogen 60, 64

Bohrungen 16

Bohrungsoberfläche 16

Bolzen 16, 17

Bool'sche Operation 16

Branchen 12

branchenspezifische Module 12

Bruchdehnung 48

Bruchmodell 47

## C

CAD-Welt 10

Camera Movement 118

chaotische Modelle 9

chaotisches Pendel 9

CIRCLE 63

Collective Change of Objects 44

Cone 21

constructive solid geometry 16

Contour 65

Courant-Friedrichs-Stabilitätsbedingung 5

Create Element 27, 61

CSG-Methode 16

CTR120 Link 81

CTR120-Links 81

Cuboid 21

Cylinder 21

## D

Dämpfer 57

Dämpfungseigenschaften 10

Dämpfungsfunktion 91

Dämpfungskraft 58

Datenaufzeichnung 13, 111

Daten-Import 67

Datenimport 67

Deaktivierung von Links 82

Definition einer Netzstruktur 39

Der Dialog Record Frequency 112

Details der Kontakte 11

Deterministische Modelle 9

Dialog 'Control System by Formula' 99

Dialog 'Edit Interaction Rule by Name' 107

Dialog 'Edit sonar Preferences' 119

Dialog 'Movie Preferences' 114

Dialog 'Resize Object' 72

Dialog 'Simulation Stop Condition' 101

Dialog 'Display Group Properties' 43

Dialog Camera Movement 118

Dialog Create Element 62

Dialog Create/Show Contour 29

Dialog Edit Bilateral Friction 91

Dialog Edit Bilateral Interaction Rule 106

Dialog Edit Gravitation 94

Dialog Edit Material Property 51

Dialog Edit Object Group 43

Dialog Edit Object Properties 71

Dialog External Point Curve 98

Dialog Global Friction 90

Dialog Limit Object Velocity 102

Dialog New Element 61

Dialog New Point 62  
Dialog Record Frequency 112  
Dialog Simulation Stop Condition 101  
Dialog Unilateral Friction 91  
Die Daten eines Zahn's 32  
Die Funktion der Grafik 111  
Differenzialgleichung 3  
Diskrete Simulationen 8  
Distortion Energy Theory 49  
Draht 15  
Drehachse 16, 17, 34, 80  
Drehmoment 11  
Drehmomentfunktion 11  
Drehung 23  
Drehungen Reihenfolge 25  
dreidimensionale Bewegung 25  
dreidimensionale Oberflächen 38  
Drucker-Prager 49  
Druckfedern 53, 55  
Durchgangsloch 16  
DXF 67, 69  
DXF-Daten 11  
DXF-file 11  
DXF-Schnittstelle 11  
dynamische bis hochdynamische Problemstellungen 7  
dynamische physikalische Daten 71

## E

Ebene 14  
Ebenen 34  
Edit Material Property 52  
Edit Object Properties 71  
Edit Stress/Strain 52  
EDITOR 11  
Eigenschwingung 8  
Eigenschwingungen 53  
Eigenwerte 8, 9  
Eindringtiefe 85, 88  
einfachere Punktkurven 96  
Einstein 10  
elastische Verbindung 10  
Elastische Verbindungen 17  
elastische Verbindungen 80  
Elastisch-plastische Stosskräfte 87  
elastisch-plastische Stosskräfte 87  
elastisch-plastische Verbindungen 13  
elastisch-plastische Verhalten 48  
Elastizität 10, 48  
Elastizitätsmodul 51, 92  
Element grafisch interaktiv erzeugen 60  
Element per Macro erzeugen 62  
elementare Newton'sche Bewegungsgleichungen 84  
elementare Newtonsche Bewegungsgleichungen 4  
Elemente gruppieren 64  
Elementgruppe 64  
Elementpunkte 63  
elliptischer Integrale 2  
E-Modul 48, 51  
Emodul 92  
Entscheidungsprozess zur Interaktionskontrolle 108  
Erzeugung eines 'Primitives' 22  
Erzeugung eines Primitives mit Dialogen 22  
Erzeugung von einfachen Primitivkörpern 21  
Erzeugungsmethoden 21

Euler Formalismus 8  
EXCEL 11  
EXCEL-Tabelle 38  
EXCL120 Link 81  
Exdrudierte Polygone 27  
explizite numerische Berechnung 4  
expliziter code 4, 84, 87  
expliziter Programm-code 7  
externe Dateien 18  
externe Kräfte 13, 71, 87  
externe Kräfte und Kraftfelder 20  
externe Programme 16  
Externe Punktkurven 97  
Extrude 28  
Extruded Line-Arc-Prism editieren 73  
Extrudieren 29  
extrudieren 16  
Extrudierte Linien-Bogen-Konturen 29

## F

Fadenpendel 2, 3  
Feder-Dämpfungs-Elemente 80  
Federdatenblatt 17  
Federkonstante 17, 53  
Federkraft 17  
Federn 53  
Fertigungstoleranzen 31  
fester Raumpunkt 34  
Festigkeitshypothesen 49  
Film Optionen 116  
Filmaufzeichnung beenden 114  
Filmaufzeichnung starten 114  
Filmaufzeichnungen 113  
Filme 13  
Filmformat 114  
Finite Differenzen 8  
Finite Differenzen Methode 2  
Finite Elemente 8  
Finite-Differenzen Methode 8  
Finite-Differenzen-Methode 7  
Fixed 35  
Fixed in Space 35, 77  
fixierte Objekte 35  
Fixpunkt 14  
Fixpunkte 34  
Flipperkasten 103  
Flüssigkeits- und Gasströmungen 9  
Format eines Textfiles 30  
FORTRAN 2  
Freiformfläche 16, 18  
Freiformflächen 38  
Freiheitsgrad 9  
Frustum 21  
Funktionsumfang des Basismoduls 20

## G

gedämpfte Feder 10  
Gegenuhrzeigersinn 28, 63  
Gelenke 80  
Geometrie-Informationen 11  
geometrische Elemente 60  
Geröll- und Schlammlawinen 9  
geschlossene Kontur 30  
Geschwindigkeitsbeschränkung 102

Gitterstruktur 16  
Gleichgewichtszustand 8  
Grafik Variablen 110  
Grafik Window 110  
Grafik Zurücksetzen 112  
grafische Darstellung 110  
Grafische Darstellungen 110  
grafische Darstellungen 13  
Grafisch-interaktive Erzeugung eines Primitives 24  
Gravitation 13  
Gravitationsfeld 87, 94  
Gravitationsfelder 20  
Grids 35, 38  
Gruppe von Objekten verändern 71  
Gruppen 42  
Gruppeneigenschaften 43  
Gruppennummer 42  
Gruppierungen 26  
Gummi 10

## H

Halbfabrikat 18  
Halbfabrikate 53  
Hebelsystem 11  
Hook'scher Bereich 81  
Hublänge 19  
Hysterese der Stosskräfte 88

## I

implizite Rechenmethode 8  
Implizite und explizite Rechenmethoden 9  
Import DXF File 69  
Import Grid 68  
Import Kontur 67  
Initierungsmechanismen 2  
Interaction Const. 88  
Interaction Method 89  
Interaktion von Objektgruppen anpassen 46  
Interaktionen 11  
Interaktions Eigenschaften 71  
Interaktionskonstante 88  
Interaktionskontrolle 108  
Interaktionsregeln mit Namen 107  
Interaktionssteuerung 103  
Iteration 4, 9  
iterativen Berechnungen 2

## J

John von Neumann 4  
Johnson-Cook 49  
Johnson-Cook Materialmodell 49

## K

Kamerapositionen 118  
KEINE INTERAKTION 104  
Kettenrad 32  
Klassen und Arten von Simulationsprogrammen 7  
Kleben 10  
Klebe-Verbindung 10  
Kollisionsenergie 85  
Kollisionskontrolle 103  
Kollisionskräfte 87  
Kollisionsprüfungen 11

Kollisionstiefe 85  
Kollisionsverhalten 23  
komplexe Bewegung 11  
Komplexe Linien-Bogen Konturen 32  
komplexere Modelle 18  
komplexere Strukturen 15  
komplizierte Oberflächen 11  
konkave Prismen 27  
Konstante Externe Kräfte 93  
konstantes Gravitationsfeld 94  
Kontaktkräfte 87, 89  
Kontaktstellle 90  
kontinuierliche Simulation 7, 8  
Kontinuierliche Simulationen 8  
Kontrollsysten 19  
Kontrollsysten mit math. Formeln 98  
Kontrollsysten mit sonar script 100  
Kontrollsysteme 13, 87, 96  
Kontrollsystem-Sprache 19  
Kontur 36  
Konturen 65  
konvexe Prismen 27  
Kräfte auf Objekte 87  
Kräften auf ein Objekt 87  
Kraftfelder 13, 20  
Kraftfunktion 85  
Kraftstoss 85  
Kreise 60  
kritische Faktoren einer Simulation 86  
Kugelgelenk 34, 80  
Kugelgelenke 17  
Kunst des Modellierens 10  
Kurvenscheibe 11

## L

Lagerung 16  
Lagrange Formalismus 8  
Laschen 16  
Lawrence Livermore 2  
Leistungssteigerung 103  
LINE 63  
Linien 60  
Link 10, 15  
Link erzeugen 82  
Link Konstante 52  
Link-Kombinationen 81  
Link-Kraft 80, 82  
Link-Kräfte 87, 91  
Link-Paket 81  
Links 13, 17, 80  
Links kollektiv ändern 45  
Links manuell löschen 82  
Loch 16  
Löcher 29  
LogFile Events 123  
LogFile Geometry 122  
LogFile Interactions 122  
LogFile Macro 123  
LogFile Syntax 123  
Los Alamos 2  
Lösungsmethoden 3  
Luftströmung 9

## M

Macro 17, 21  
Macro Befehle für Objekgruppen 46  
Macro-Bibliothek 18  
Macros 18  
Macrosprache 19  
Manhattan Projekt 4  
Marsmond Deimos 94  
Maschinenkette 17  
Maschinensprache 2  
Maschinenteile 10, 15  
Masse 10  
Material Modelle 47  
Material von Objektgruppen ändern 45  
Materialmodell 80  
mathematische Druckfeder 55  
mathematische Feder 14  
mathematischer Aktuator 15  
mathematischer Dämpfer 15  
mathematisches Fadenpendel 3  
Matrizenrechnung 9  
maximal zulässiger Zeitschritt 86  
Maximum Principal Strain Theory 49  
Maximum Principal Stress Theory 49  
Maximum Shear Stress Theory 49  
Mehrfachselektion 64  
Mehrkörperdynamik Simulation 8  
Mehrkörperdynamik-Simulationsprogramm 7  
Merge 67, 70  
Merge Model 70  
Merkmale des Objektes 23  
Messmethode der Viskosität 92  
Milliarden-Körper-Probleme 2  
Modell 9  
Modell Beruhigen 78  
Modell fliegt auseinander 85  
Modellbestandteile 13  
Modellerzeugung 19  
Modell-Erzeugungsprogramme 11  
Modellermöglichkeiten 13  
Modellier-Philosophie 16  
Modellierung 9, 13, 16  
Modellierungsphilosophie 16  
modulares Programm Paket 11  
Möglichkeiten der Modellerzeugung 18  
Mohr-Coulomb 49  
Monte Carlo Simulation 7  
Muren 9

## N

Nachbearbeitung von extrudierten Konturen 31  
NBT Link 81  
negative Konturen 16  
Netzlinien 38  
Netzstruktur 8, 39  
Netzstruktur (Typ 1) 39  
Netzstruktur (Typ 2) 39  
Netzstrukturen 38  
New Element 61  
Newton 3, 94  
newtonsche 84  
Nieten 10  
n-Körperproblem 2  
NNB Link 81

NNN Link 81  
NNNN Link 81  
NO INTERACTION 105  
non relocatable 77  
nonrelocatable 35  
Normalen-Vektoren 61  
Normaler Link 80  
Normalspannung 52  
Normteile 18  
Nuklearwaffen 2  
Nulllage 25  
Nulllage der Primitivkörper 26  
Nulllage eines Objektes 25  
numerisch 3  
numerische Lösung 3  
Numerische Simulation 2  
numerische Stabilität 84

## O

Oberflächenhärte 10  
Oberflächenhärte der Objekte 86, 88  
Object Properties 52  
Object Tool 44  
Objekt 14  
Objekt ändern 71  
Objekt Eigenschaften 23  
Objekte 17  
Objekte Modifizieren 71  
Objekte Spiegeln 73  
Objekte verbinden 80  
Objekt-Eigenschaften ändern 71  
Objektgrösse ändern 72  
Objektgruppen 42  
Objekt-Rotation mit einem Dialog 25  
Objekt-Verbindungen 10  
OpenGL 28  
OpenGL-Grafiksoftware 28  
Orientierung 63  
Orientierungsrichtung 63

## P

parametrisieren 15  
Partikel 92  
Partikelhaufen 92  
PASSIV 104  
passive Objekte 103  
Pendel 2  
Pendelbewegung 4  
Philosophie hinter 'sonar' 11  
physikalische Feder 14  
physikalische Litzen und Seile 15  
Physikalische Zugfedern 54  
plastisch kollidieren 10  
POINT 63  
POLYGON 63  
Polygon 27, 61  
POLYLINE 63  
Polyline 67  
Positionierung 23  
positive Orientierung 27  
Preferences 119  
Primitives 24  
Primitives mit Macros 21  
Primitivkörper 13, 14, 16, 17, 21

Primitivkörper verbinden 17  
Prismatische Körper 27  
prismatischer Körper 16  
Prismen 27  
Programm-spezifische Einstellungen 119  
PRO-Version von Quicktime 115  
prozentuale Dehnung 51  
Punktkurven 87, 96  
Punktschweissen 10  
Punktspiegelungen 73  
Punktverbindungen 16

## **Q**

QUAD\_STRIP 63  
Quadstrip 27  
Querschnittsfläche 36  
queues 8  
Quicktime Filme aufzeichnen 113  
Quicktime Filme komprimieren 115  
Quicktime Player 113  
Quicktime PRO 115  
Quicktime Pro 113

## **R**

Randbedingungen 13  
Rankine 49, 50  
raumfesten Kraftvektors 93  
Raumlage eines Objektes 25  
räumliche Bewegung 11  
Rechenaufwand 86  
rechenintensivste Simulationsmethode 2  
Rechenzyklen 4  
rechnerische Nachbildung einer Kollision 85  
rechnerische Simulationsverfahren 10  
reflektierter Energie beim Stoss 88  
Reibung 13  
Reibungskoeffizient 10, 88  
Reibungskräfte 87, 90  
Reibwerte 11  
rein explizite Rechenmethoden 8  
Report (Physical Properties) 121  
Reports & Logfiles 121  
Resize 73  
Resonanzverhalten 53  
Rohdaten 14, 60  
Rohdaten Import 67  
Rotation und Position 23  
Rotationsachsen einfrieren 77  
Rotationskörper 16, 36  
rotieren 16  
Rückstellung der Bewegungen 78  
Ruhelänge 10

## **S**

Schallgeschwindigkeit 92  
Schnittstellen 16  
Schrauben 10  
Schraubverbindung 10  
Schweissen 10  
Schweisspunkte 17  
Schwingungsverhalten 53  
Seil (Blockmodell) 15  
Serie von Punkten 96  
Signalfortpflanzung 53, 92

Simulation 84  
Simulation einer Schutzanlage 113  
Simulationsabbruch 101  
Simulationsprogramm 11  
Simulationsprogramme und Methoden 7  
Simulationszyklus 84  
Simulieren 9  
solid-modeling-Programm 16  
sonar Files zusammenführen 70  
sonar Programm Module 11  
sonar samples 1  
sonar script 19, 87, 100  
sonar-Bestandteile 13  
sonar-Format 11  
sonar-LAB 12  
sonar-LAB Umgebung 16  
sonar-Modell 19  
sonar-SIM 12  
Spannungs- Dehnungs-Kurve 47  
Spannungs/Dehnungs-Funktionen 47  
Spannungs/Dehnungs-Variablen 47  
Spannungs-Dehnungskurve 81  
Spannungs-Dehnungs-Variablen 52  
Spannungsverteilung 8  
Sphere 21  
Spiel 16  
Sprengstoffen 2  
St. Venant 49  
Standard Objekte 19  
Standard-Maschinenteile 15  
Standardobjekte 18  
Starrkörper 85, 89  
statische physikalische Daten 71  
Steifigkeit der Objekte 17  
Steifigkeit des Links 91  
Steuerung des Aktuators 58  
Steuerungen 13, 19  
Stossdauer 85  
Stossvektoren 84  
Strich-Hebel 10  
Strukturanalyse 8  
Supergruppen 42  
Supergruppen-Nummer 42  
Sweep'-Funktion 67

## **T**

Teile-Bibliothek 18  
TEXT Format 67  
Texteditor 11  
Textfile 30, 67  
Textformat 11  
Top-500 2  
Torsions-Link 80, 81  
Torus 21  
TorusSegment 21  
Torussegment 53  
Total Strain Energy Theory 49  
Trägheitsmoment 10  
Tresca 49  
Tube 21  
TubeSegment 21

## **U**

Ueberlast 80

Umweltbedingungen 11  
Unverrückbare Objekte 34  
U-Profil 18

## V

vektorielle Geschwindigkeitsdifferenz 90  
Verändern der Geometrie eines Objektes 32  
Verformbarkeit 17  
Vergleichsspannung 49  
Vergleichsspannungs-Hypothese 49  
Vergleichsspannungshypothese 50  
Verwaltung von Objektgruppen 42  
Vibrationen 35  
vibrieren 35  
virtuelle physikalische Welt 11  
virtuelle Steuerung 19  
virtueller Windkanal 9  
Viskosität 13, 87, 92  
Viskositätseigenschaften 92  
Volumenkörper 10  
vom Groben zum Feinen 10, 13  
von Mises 49, 50  
Von-Mises 51  
Vorteile von Macros 18

## W

Wahrscheinlichkeit 8  
Wetter und seine Vorhersage 9  
Wie modellieren? 10  
Windkanalmodell 10  
Winkelbeschleunigung (Fadenpendel) 3  
Wirkung der Link-Kraft 82  
WORD 11

## Y

Yield Criteria 50  
Yield Criterion 49  
young modulus 51

## Z

Zahndaten 33  
Zahnkettenrad 33  
Zahnkontur 33  
Zahnrad\_Z38 33  
Zeichenrate 86  
Zeitschritt 5, 84  
Zeitschritt erhöhen 84  
Zeitschritterhöhung 86  
zentrales Basisprogramm 11  
Zentrales Gravitationsfeld 94  
Zugbelastung 52  
Zugfeder 15  
Zugfedern 53, 54  
Zustand des Systems 4, 9  
Zwangsbewegung eines Objektes 96  
Zyklen 9  
Zyklus 4, 84  
Zylinder 57  
zylindrischer Bolzen 16

---