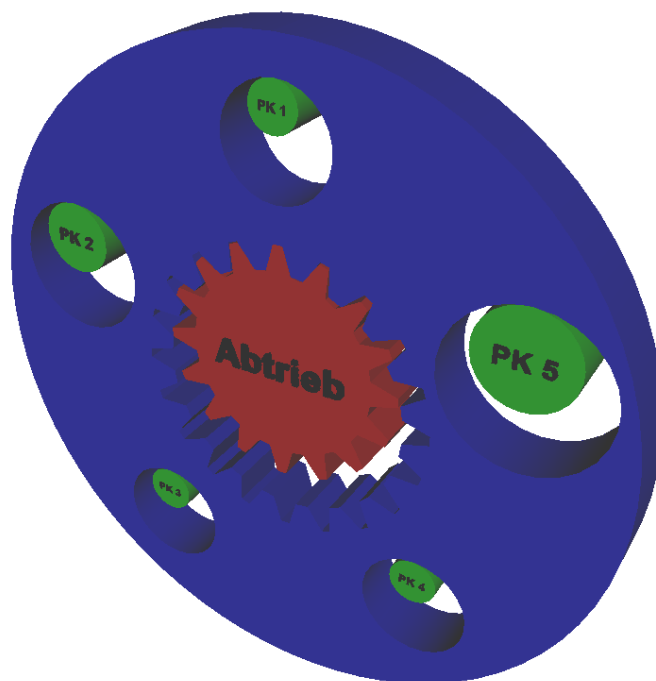


# Beleg

–

## Grafiksysteme: Java3D

Thema:  
Animation eines parametrisierbaren  
Kreisschubgetriebes  
als Applikation und Applet



Student : Stefan Pohl  
Professor : Prof. Dr. Gaudlitz  
Datum : 12.01.2003



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>3</b>
<b>2</b>	<b>Beweggrund.....</b>	<b>3</b>
<b>3</b>	<b>Überblick.....</b>	<b>3</b>
<b>4</b>	<b>Bedienungsanleitung .....</b>	<b>4</b>
4.1	Allgemeines.....	4
4.2	Kreisschubgetriebeeigenschaften .....	4
4.3	Darstellung der Eigenschaften mittels GUI.....	4
4.3.1	Abtrieb (Stirnradparameter) .....	5
4.3.2	Antrieb (Hohlradparameter).....	5
4.3.3	Allgemein.....	5
4.3.4	Beobachterpositionierung.....	5
4.4	Constraints (Einschränkungen) .....	5
<b>5</b>	<b>Technische Referenz.....</b>	<b>6</b>
5.5	Allgemeines.....	6
5.6	Knotenbeschreibung.....	6
5.6.1	Background .....	6
5.6.2	Viewpoints .....	6
5.6.3	Lights.....	6
5.6.4	Shapes.....	6
5.6.5	Solide Körperbeschreibung.....	6
5.7	Szenenhierarchie.....	7
5.8	Dokumentation ausgewählter Klassen .....	8
5.8.1	Stirnrad.....	8
5.8.2	Hohlrad.....	8
5.8.3	Animation.....	8
<b>6</b>	<b>Mathematischer Hintergrund .....</b>	<b>8</b>
<b>7</b>	<b>Quellenangabe.....</b>	<b>19</b>
<b>8</b>	<b>Bemerkungen .....</b>	<b>19</b>

## 1 Einleitung

Java3D ist als API zur dreidimensionalen Programmierung von Welten in Java entwickelt worden. Da eine Vielzahl der zur realitätsnahen Darstellung notwendigen Algorithmen NP-vollständig sind, muß bei der Realisierung Abstriche gemacht werden, so daß eine praktische Nutzung selbst in der Zeit der Gigahertz-Prozessoren erst möglich wird. Selbst bei Einsatz von modernen VR-Systemen begrenzt das den überhaupt erreichbaren Immersionsgrad.

Um eine Welt beschreiben zu können, muß man sich vorher über die Beschreibung von Objekten und deren Beziehungen zueinander Gedanken machen. Anstatt Objekte durch feste Werte zu beschreiben, ist es effizienter, sozusagen die Meta-Informationen der Objekte direkt zu hinterlegen. Da Objekte Eigenschaften haben, kann man bei vorhandener mathematischer Beschreibung leicht eine Parametrisierung der Objekte und damit der gesamten Welt zulassen. Den Vorteil dieses Vorgehens erkennt man leicht, wenn man bei komplexeren Welten elementare Eigenschaften, sei es auch nur die Detailtreue, nachträglich ändern will.

## 2 Beweggrund

Animationen lohnen sich besonders bei statisch schwer beschreibbaren Eigenschaften. Insofern liegt es nahe, etwas zu simulieren, das nur durch Worte oder Bilder schwer vorstellbar erscheint: Ein Kreisschubgetriebe.

Ein solches Getriebe besteht aus einem Hohlrad mit einer Innenverzahnung, einem Stirnrad mit einer Außenverzahnung und mehreren Parallelkurbeln.

Um die Parametrisierung der hier aufgeführten Eigenschaften zu ermöglichen, mußte zuvor eine mathematische Beschreibung erstellt werden, die die gewünschten Attribute validiert und bei Erfolg sonstige damit eng verbundene Eigenschaften berechnet. Diese kann im Abschnitt „Mathematischer Hintergrund“ und im Programmcode gefunden werden.

Der Einsatz von Java3D im Rahmen eines Beleges begünstigt die Überlegung, die Grenzen des in Java3D Machbaren auszureizen, um bei späteren Problemstellungen die Vor- und Nachteile der Realisierung in den unterschiedlichen heute zur Verfügung stehenden 3D-Beschreibungs- (VRML, X3D) und Programmiersprachen (diverse OpenGL language mappings) besser ausloten zu können. Diese Ausreizung betrifft natürlich das große Manko von Java und anderen interpretierten Sprachen, die Ausführungsgeschwindigkeit.

Daher fiel die Wahl des Themas für diesen Beleg auf eine kompliziertere Animation eines funktionierenden Kreisschubgetriebes, das mittels einer grafischen Oberfläche parametrisierbar und immanent funktionsfähig sein soll.

Die Nutzung von Java legt die Implementierung einer dualen Anwendung nahe, die standalone als Applikation sowie als Applet innerhalb einer HTML-Oberfläche lauffähig ist.

## 3 Überblick

Die beiliegende CD enthält folgende Verzeichnisse und Dateien:

<u>Verzeichnis/Datei</u>	<u>Beschreibung</u>
/	JBuilder – Projektdateien
classes/	im Bytecode vorliegende .class – Dateien
doc/	Dokumentation
doc/javadoc	Per Javadoc generierte Klassenbeschreibungen
runimage/internet	Per Browser zu startende Online-Version
runimage/standalone	sofort startbare Version: „Startme.bat“
src/	Java-Quellcode

## 4 Bedienungsanleitung

### 4.1 Allgemeines

Durch den Einsatz von Java ist sichergestellt, daß sich die Bedienung der Anwendung nicht in Abhängigkeit von der Ausführungsumgebung unterscheidet.

Im folgenden sollen die Hintergründe eines Kreisschubgetriebes beleuchtet und dessen Parameter beschrieben werden.

### 4.2 Kreisschubgetriebeeigenschaften

Die Animation unterstützt Getriebe bestehend aus:

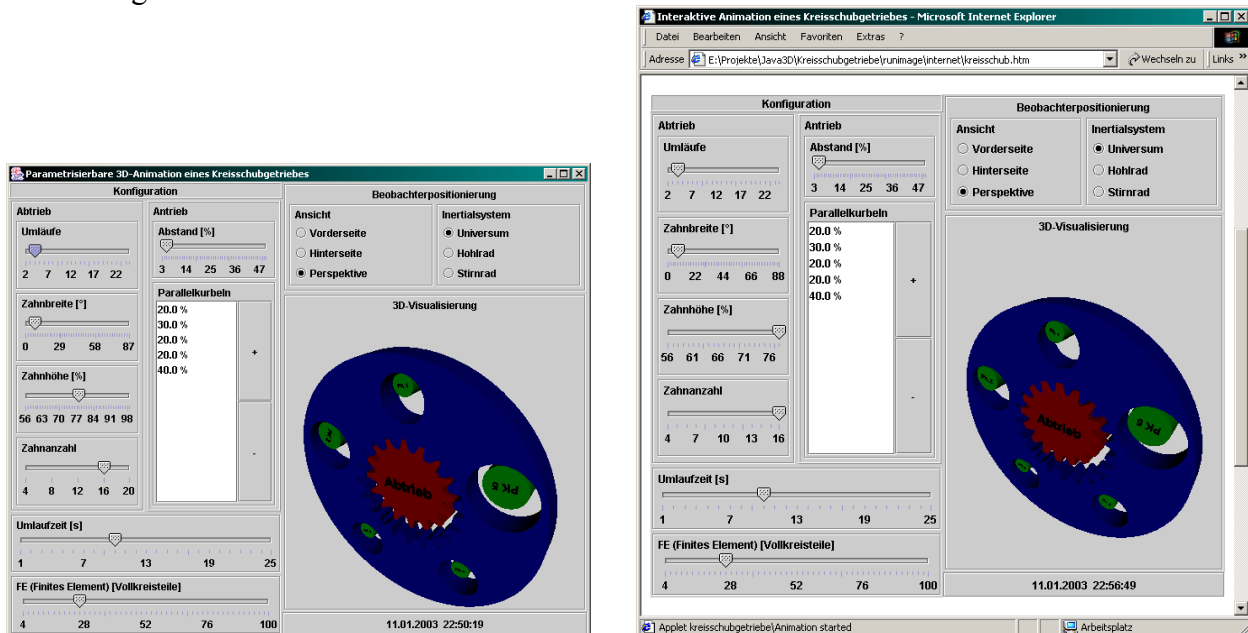
- einem Hohlrad mit einer Innenverzahnung
- einem Stirnrad mit einer Außenverzahnung
- beliebig vielen Parallelkurbeln

Diese stehen so in Beziehung, daß sich ein Kreisschubprinzip ergibt.

Zur Funktion müssen einige Verhältnisse sichergestellt werden, wodurch eines der Zahnräder durch die Parallelkurbeln erst eine drehungsfreie kreisförmige Schiebebewegung erfährt. Das andere Zahnrad ist um eine ortsfeste Achse drehbar, steht mit dem ersten Zahnrad in Eingriff und ist dadurch gekennzeichnet, daß die Innen- und Außenverzahnung trapezförmige oder ein spitzwinkliges Dreieck bildende Zähne mit ebenen Zahnflanken aufweisen.

### 4.3 Darstellung der Eigenschaften mittels GUI

Damit man sich ein Bild von den Interfaces zum Benutzer machen kann, seien selbige hier einmal abgebildet:



Die Oberfläche gliedert sich in Konfigurationsfenster links und oberhalb und ein Visualisierungsfenster rechts. Hier läuft die Animation und die sofortige Aktualisierung der Parameter.

Die Parameter sind nach ihrer Zugehörigkeit geordnet, so gibt es Konfigurationen zum Abtrieb (dem Stirnrad), zum Antrieb (den Parallelkurbeln), allgemeine Einstellungen und zur Ansicht.

**4.3.1 Abtrieb (Stirnradparameter)**

Umläufe	gibt an, wieviel volle Umläufe das Hohlrad vollführen muß, damit sich das Stirnrad einmal um die volle Achse dreht
Zahnbreite [°]	gibt an, wie breit die „Zahnspitze“ der Zähne des Stirnrades ist Die Zahnbreiten innen und die des Hohlrades werden aus den Parametern maximal berechnet
Zahnhöhe [%]	gibt prozentual an, wieviel volle Kreisfläche dem Stirnrad bleibt bzw. ab wo die Zähne beginnen, sich einzuschneiden
Zahnanzahl	gibt an, wieviel Zähne das Stirnrad besitzen soll (immer Vielfache von Umläufen, siehe Zykloiden)

**4.3.2 Antrieb (Hohlradparameter)**

Abstand [%]	gibt an, wieviel Abstand zwischen dem Hohlradrand und der größten Parallelkurbel vorhanden bleibt (relativ zum Stirnradradius)
Parallelkurbeln	hier sind die Größen der Parallelkurbeln prozentual zum Stirnrad in der Darstellungsreihenfolge im Uhrzeigersinn aufgelistet. Es können mittels „+“ und „-“ neue Parallelkurbeln hinzugefügt oder gelöscht werden

**4.3.3 Allgemein**

Umlaufzeit [s]	gibt die Zeit in Sekunden an, die für die gesamte Animation benötigt wird (volle Umdrehung des Stirnrades)
FE [Vollkreisteile]	gibt die Größe für ein „Finites Element“ an Dies ist wichtig für die Genauigkeit von Rundungen wie bei wenigen, schmalen Zähnen, aber auch für die Berechnungsgeschwindigkeit

**4.3.4 Beobachterpositionierung**

Die Ansichts-Einstellungen dienen der initialen Positionierung des Beobachters der Szene. Man hat die Auswahl zwischen den gängigen Ansichten: „Vorderseite“, „Hinterseite“ und „Perspektive“. Es obliegt dem Betrachter, durch Manipulation der Maus und gleichzeitiger Betätigung der Maustasten die Szene genauer zu inspizieren.

Die Auswahl des Inertialsystems ermöglicht dem Betrachter in die Rolle der Elemente des Kreisschubgetriebes zu schlüpfen, d.h. man kann festlegen, zu welchem Element die relative Positionierung konstant bleiben soll.

**4.4 Constraints (Einschränkungen)**

Für die korrekte, praktische Funktion sind aufgrund der Beziehungen zueinander diverse Einschränkungen auf die Wertebereiche anderer Parameter auszuüben. Je nachdem, welche Parameter man zuerst verändert, gibt es unterschiedliche Auswirkungen auf mehrere andere Parameter. Deshalb wurde im GUI eine von oben absteigende Anordnung der Parameter gewählt, so daß Änderungen eines Parameters sich auf die Wertebereiche der darunter (noch) zu spezifizierenden Parameter auswirken. Dem hier vorliegenden Getriebe ist es geschuldet, daß es dennoch dazu kommen kann, daß für Parameter kein Wertebereich zur Verfügung steht. Diesen Fall löst die Applikation mittels einer Rückmeldung und Rücksetzung der vom Nutzer gewollten, unzulässigen Aktion. Ein Label, das standardmäßig die aktuelle Zeit anzeigt, gibt dem Anwender in Ausnahmesituationen Hinweise.

Diese Fälle lassen sich nicht vermeiden ohne die Parametrisierung um ein zu hohes Maß einzuschränken.

## 5 Technische Referenz

Dieser zweite Teil der Dokumentation beschäftigt sich mit den Implementierungsdetails.

### 5.5 Allgemeines

Es wurde von Anfang an eine Funktionalitätskapselung durch „packages“ angestrebt. Als Resultat stehen die Packages „kreisschubgetriebe.math“, „kreisschubgetriebe.primitives“ und letztlich das Programm selbst, „kreisschubgetriebe“, zur (Wieder-)Verwendung zur Verfügung. Die Formatierung des Quelltextes und aller Kommentare orientiert sich an Javadoc, um Lesbarkeit zu garantieren, Mißverständnisse zu minimieren und Dokumentation automatisch zu generieren.

Bei der Verwendung von Swing wurde darauf geachtet, eine starke Modulbindung und minimale Kopplung zu erreichen. Dazu ist die konfigurierende Erstellung jedes „Jpanel“ in eine eigene Prozedur gekapselt. Somit wird das Reengineering der Quellen optimal unterstützt.

### 5.6 Knotenbeschreibung

#### 5.6.1 Background

Um eine nahtlose Integration in die umgebende Anwendung zu gewährleisten, bietet sich der Background-Knoten an, um den Hintergrund des Canvas3D festzulegen. Damit dient die Anwendung auch als Beispiel für eine Java-Anwendung, die durch 3D-Elemente visuell aufgewertet wird.

#### 5.6.2 Viewpoints

Es sind die gängigen Ansichten vorhanden: Vorder-, Hinteransicht und Perspektive.

Um eine Vorstellung von den zueinander in Beziehung stehenden relativen Translationen zu bekommen, existieren mehrere „ViewingPlatforms“, die sich in den selben Inertialsystemen befinden wie die Elemente des Kreisschubgetriebes.

#### 5.6.3 Lights

Die Hauptanwendung der Szene liegt bei der Inspektion, daher sollte eine Ausleuchtung der Szene aus jedem Blickwinkel sichergestellt sein. Aus diesem Grund bietet sich ein „Headlight“ an, d.h. Licht, das vom Betrachter aus auf das Zentrum des Interesses gerichtet ist.

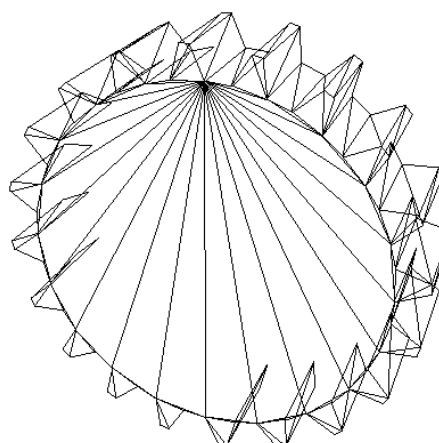
#### 5.6.4 Shapes

In der gesamten Szene werden zur Modellierung neben „Text3D“ ausschließlich eigene Shape3D-Knoten definiert, die über „GeometryInfo“-Objekte in Form von Polygon-Arrays ihre Gestalt erhalten.

Die dynamische Erstellung gibt eine bessere Kontrolle über das Genauigkeit-Geschwindigkeits-Verhältnis der Szene.

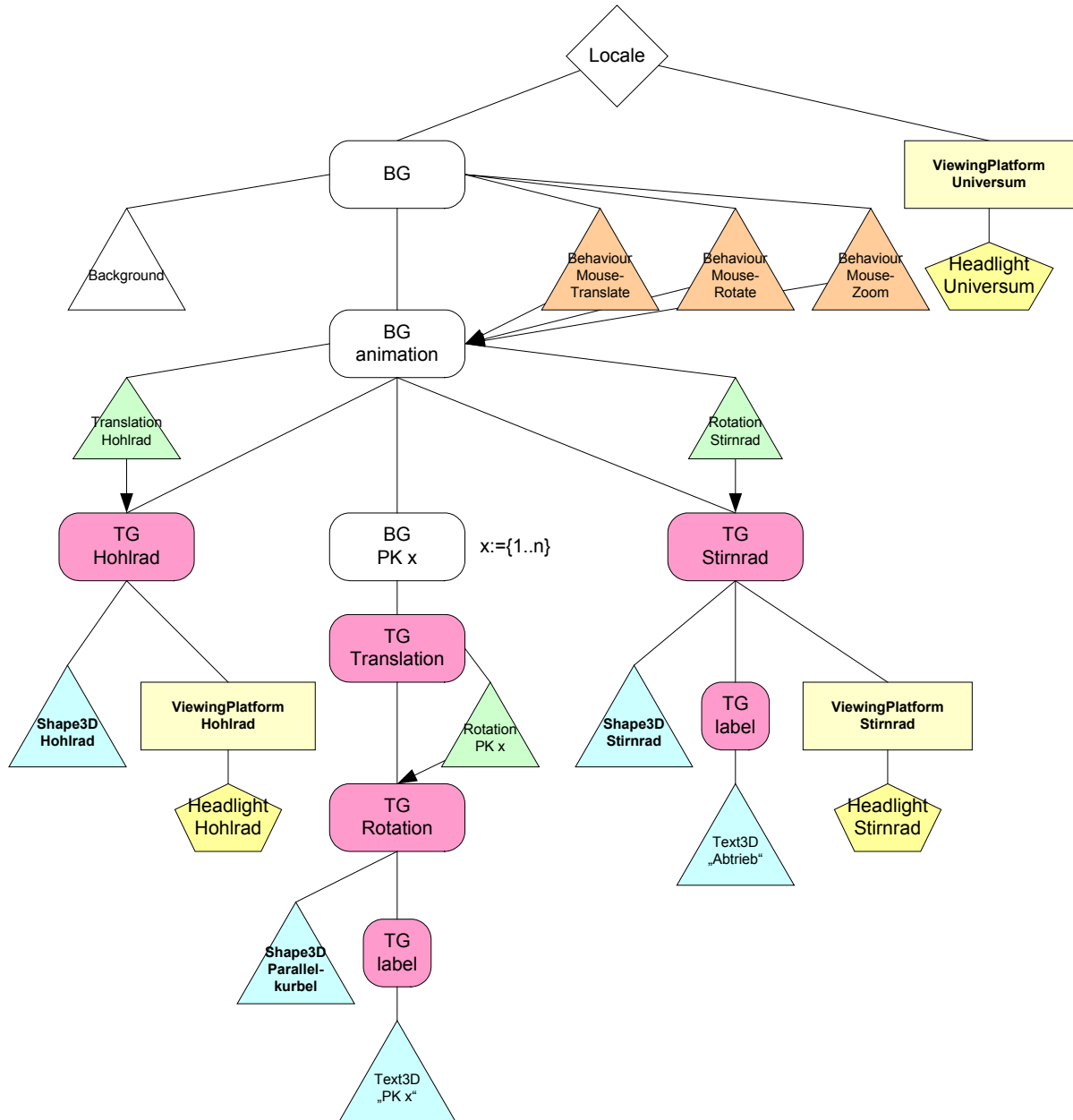
#### 5.6.5 Solide Körperbeschreibung

Um geschlossene Körper zu erhalten, ist eine rechtsdrehende Beschreibung aller Objektflächen erforderlich. Somit haben in Java3D Flächen auch eine „Richtung“. Das ermöglicht, daß nur die Flächen in die Rendering-Phase eintreten, die überhaupt sichtbar bzw. außen am Körper befindlich sind. Durch die Berechnung von zumeist nur der Hälfte aller Flächen



läßt sich die Verarbeitungsgeschwindigkeit drastisch steigern!  
 Das läßt sich leicht anhand von nebenstehendem Drahtgittermodell verdeutlichen.  
 Durch Überschneidungen von Polygonkanten aufgrund von Beschreibungsungenauigkeiten (in Anwendung einstellbar), kann es zur Umkehr der Außenrichtung bei Flächen kommen. Aus diesem Grund wurden Vorder- und Hinterseite des Hohlrades doppelt jeweils in umgekehrter Richtung definiert.

**5.7 Szenenhierarchie**



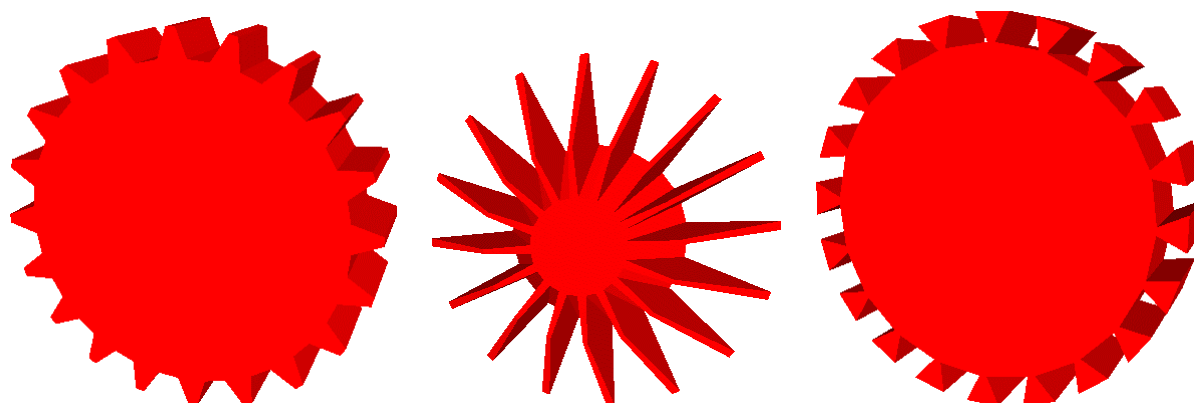


## 5.8 Dokumentation ausgewählter Klassen

### 5.8.1 Stirnrad

Bei der Nutzung dieser Klasse kann jede Kombination der Parameter dem Konstruktor übergeben werden, da hier wie in C davon ausgegangen wird, daß der Nutzer weiß, was er tut. Außerdem gestaltet es sich einfacher, die Validierung der Parameter an einer zentralen Stelle vorzunehmen.

Nichtsdestotrotz kann man mittels dieser Klasse folgende Objekte erzeugen:



### 5.8.2 Hohlrad

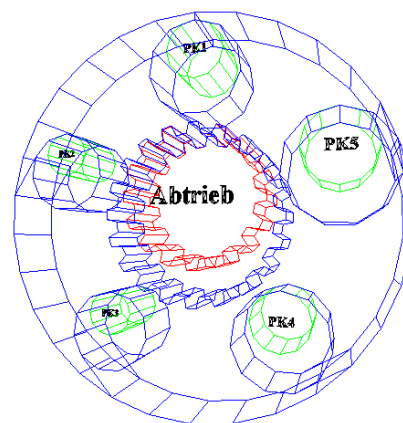
Diese Klasse stellt ebenso wie die Klasse Stirnrad eine Geometrie-Beschreibung dar. Die Parameter dieses Konstruktors sind aufgrund der ungleich größeren Komplexität komplizierter. Ein Studium des mathematischen Hintergrunds und der dort vorgenommenen Begriffseinführungen läßt es zu, beliebige Hohlräder zu erstellen.

### 5.8.3 Animation

Hier wird die gesamte Szenenbeschreibung generiert. Diese Klasse ist die Regie, unter der alle anderen durch Objekte repräsentierten Programmmodule aufgerufen und Berechnungen verwertet werden.

Das Ergebnis ist eine komplette dynamische Java3D-Szenenbeschreibung.

Diese Klasse ist selbst ein Applet und kann somit in HTML-Seiten referenziert werden, implementiert aber zugleich eine „main“-Methode, die als Einstiegspunkt zur Ausführung der Anwendung als Applikation dienen kann.



## 6 Mathematischer Hintergrund

Zur korrekten Beschreibung und Funktionsweise des Getriebes war einige mathematische Vorarbeit notwendig. Folgende aufeinander aufbauende Formeln sind so auch im Quelltext wiederzufinden. Sie stellen das Fundament dar, das zum Verständnis der Verhältnisse in einem solchen Getriebe nötig ist und somit die Voraussetzung für eine parametrisierbare Animation dieser Verhältnisse.

**Bewegung eines Punktes des Kreisumfangs eines Kreises, der auf dem Kreisumfang eines größeren Kreises abläuft**

Anzahl voller Umdrehungen des Innenkreises:  $n := 4$

Radien: innerer Kreis:  $r1 := 1$

äußerer Kreis:  $r2 := r1 \cdot \left(1 + \frac{1}{n}\right)$   $r2 = 1.25$

Winkel vom MP großer Kreis zum MP des kleinen Kreises:

$$\beta := 0, \frac{(2 \cdot \pi) \cdot n}{325} .. (2 \cdot \pi) \cdot n$$

Translation:  $xt(\beta) := \sin(\beta) \cdot (r2 - r1)$       Rotation:  $\alpha(\beta) := \beta \cdot \frac{r2 - r1}{r1} + 0.0$

$yt(\beta) := \cos(\beta) \cdot (r2 - r1)$        $xr(\beta) := -\sin(\beta) \cdot r1$

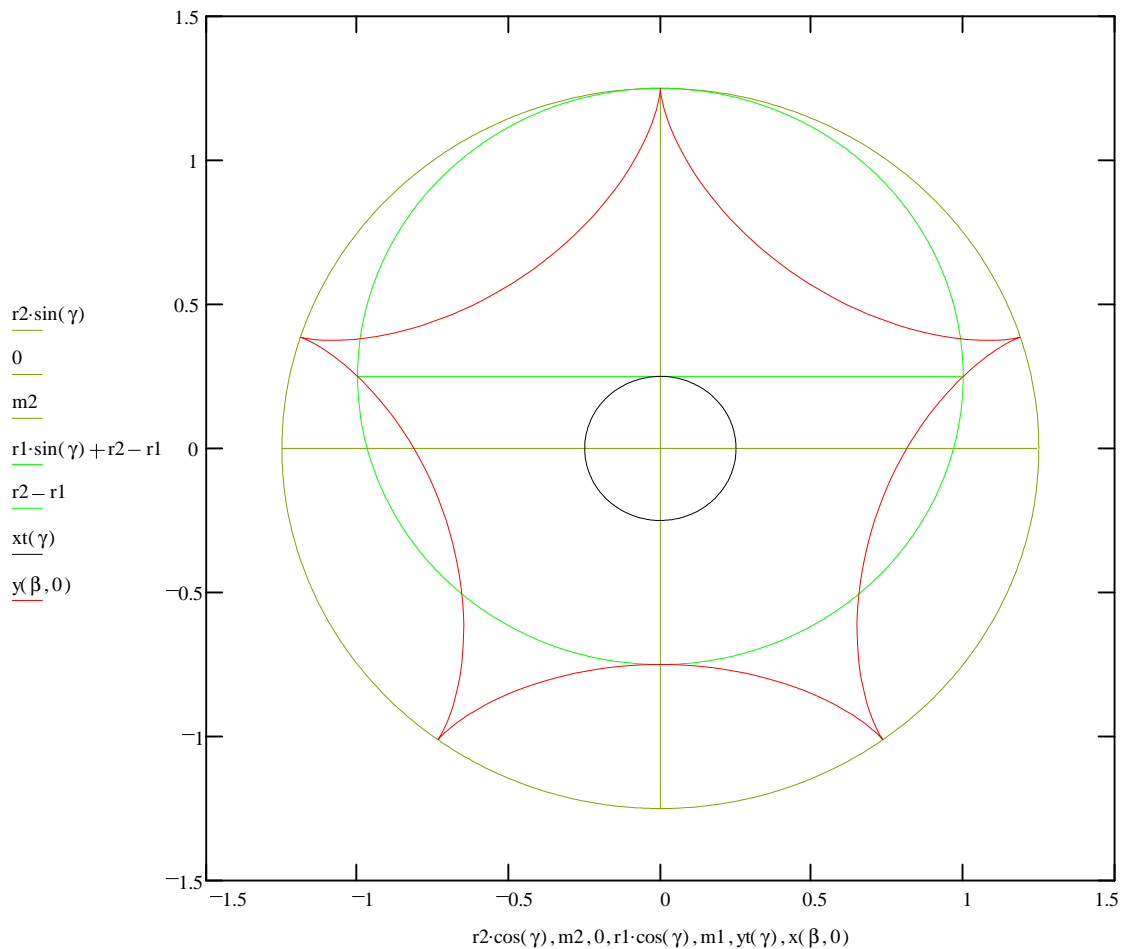
$yr(\beta) := \cos(\beta) \cdot r1$

Superposition:  $x(\beta, i) := xt(\beta) + xr(\alpha(\beta) + i)$

$y(\beta, i) := yt(\beta) + yr(\alpha(\beta) + i)$

Grafik:  $m1 := -r1, -\frac{r1}{360} .. r1$   $m2 := -r2, -\frac{r2}{360} .. r2$

$\gamma := 0, 0.01 .. 2 \cdot \pi$



**Erweiterung der Betrachtung auf Punkte, die fest mit dem Innenkreis verbunden sind:**

Ausgangspositionen:

Punkt:

$$P0(x0, y0) := \begin{pmatrix} x0 \\ y0 \end{pmatrix}$$

Referenz für Translation:  
(MP Innenkreis)

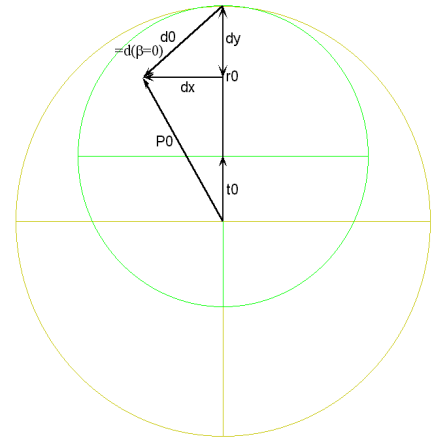
$$t0 := \begin{pmatrix} 0 \\ r2 - r1 \end{pmatrix}$$

Referenz für Rotation:  
(bzgl. Innenkreis)

$$r0 := \begin{pmatrix} 0 \\ r1 \end{pmatrix}$$

relative Position des Punktes  
zur Referenz des kl. Kreis:

$$d0(x0, y0) := P0(x0, y0) - (t0 + r0)$$



Aktuell:

Translation:  
(MP Innenkreis)

$$t(\beta) := \begin{pmatrix} xt(\beta) \\ yt(\beta) \end{pmatrix}$$

Rotation:  
(bzgl. Innenkreis)

$$r(\beta) := \begin{pmatrix} xr(\alpha(\beta)) \\ yr(\alpha(\beta)) \end{pmatrix}$$

Richtungsänderung der relativen Position des Punktes durch Rotation:

"y"-Komponente:  $dy(\beta, x0, y0) := \frac{r(\beta)_1}{|r(\beta)|} \cdot d0(x0, y0)_1$

Normalenvektor:  $rn(\beta) := \begin{pmatrix} r(\beta)_1 \\ -r(\beta)_0 \end{pmatrix}$

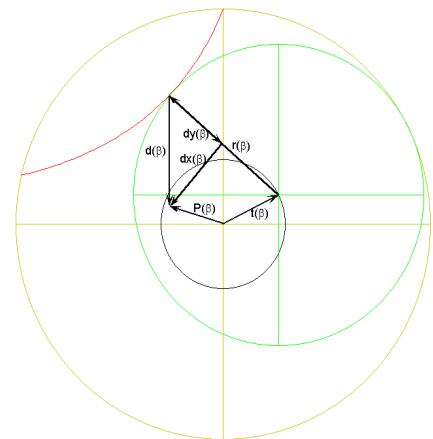
"x"-Komponente:  $dx(\beta, x0, y0) := \frac{rn(\beta)_0}{|rn(\beta)|} \cdot d0(x0, y0)_0$

akt. relative Position  
des Punktes:

$$d(\beta, x0, y0) := dx(\beta, x0, y0) + dy(\beta, x0, y0)$$

aktuelle Punktposition:

$$P(\beta, x0, y0) := t(\beta) + r(\beta) + d(\beta, x0, y0)$$



Grafik:

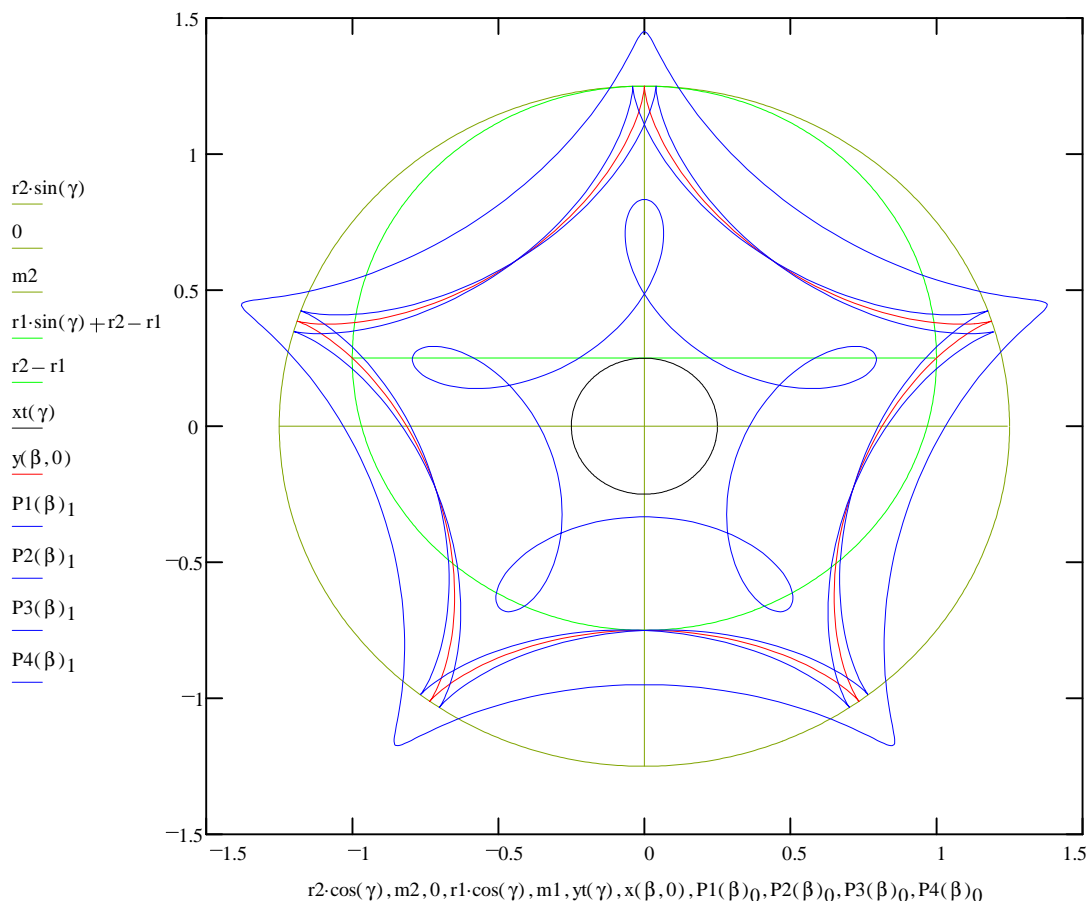
y-Wert auf Kreisumfang:  $yu(x) := \sqrt{r1^2 - x^2} + r2 - r1$

Punkte

auf Kreisumfang:  $P1(\beta) := P(\beta, -0.04, yu(-0.04))$        $P2(\beta) := P(\beta, 0.04, yu(0.04))$

innerhalb:  $P3(\beta) := P\left(\beta, 0, r2 \cdot \frac{2}{3}\right)$

außerhalb:  $P4(\beta) := P(\beta, 0, r2 + 0.2)$



**Anwendungsbeispiel: Kreisschubgetriebe**

Zusatzeigenschaften:

relative Zahnhöhe:

für kleines Rad:  $h := 0.85$

für großes Rad:  $h2 := 1 - \left[ (1 - h) \cdot \frac{r1}{r2} \right]$   $h2 = 0.88$

Zahnbreite außen:

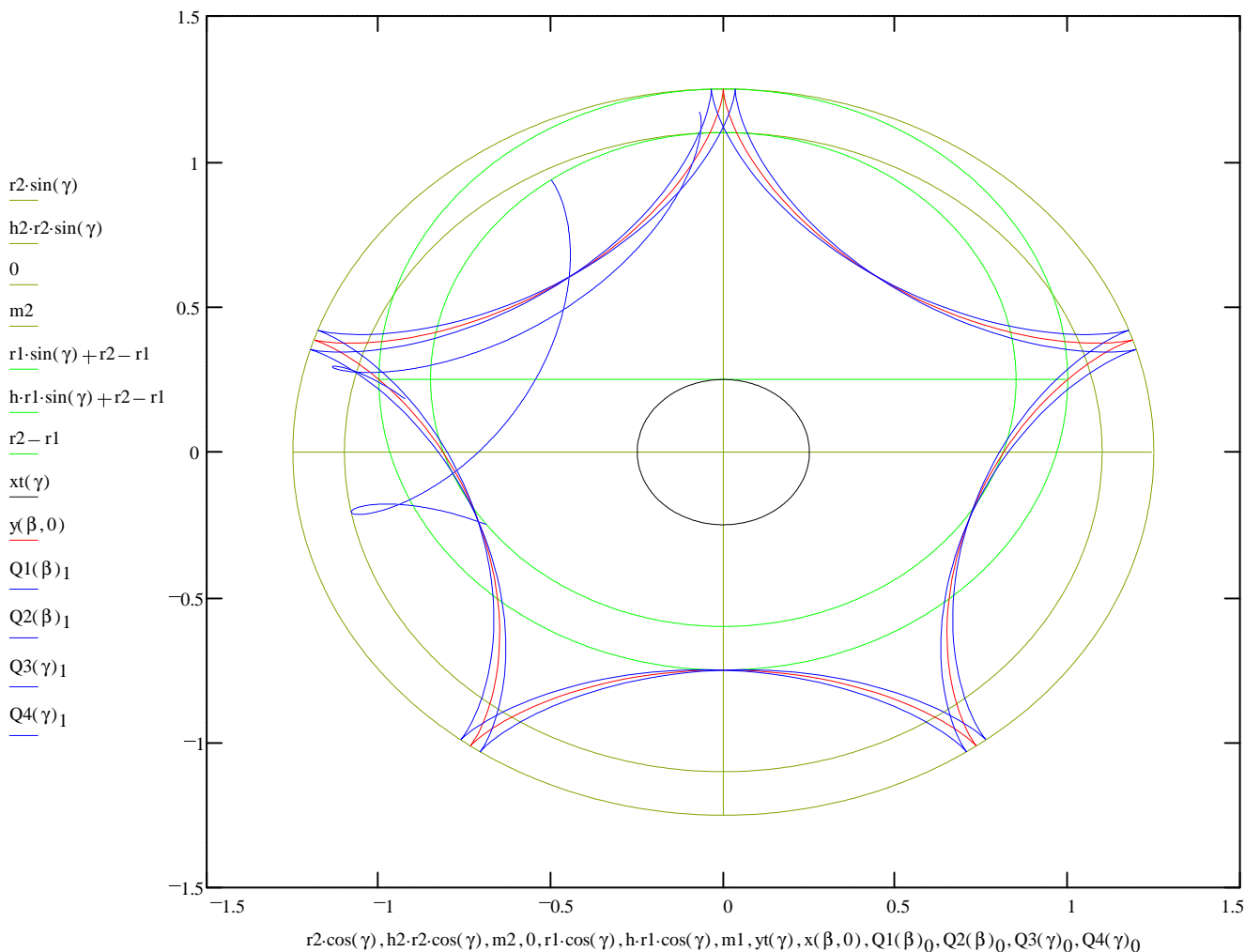
für kleines Rad:  $ba1 := \frac{2 \cdot \pi}{90}$   $ba1 = 0.0698131701$

Übertragung der Zahnbreite:  
(auf Koordinaten)  $Ax := -\sin\left(\frac{ba1}{2}\right) \cdot r1$

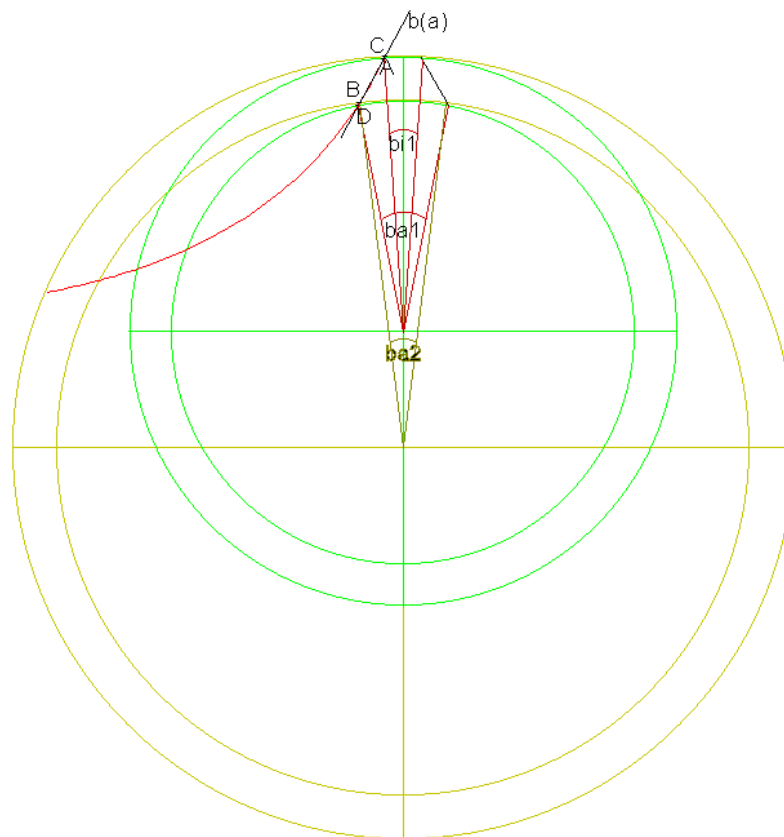
Punkt des Zahnes außen im kleinen Rad:  $A := \begin{pmatrix} Ax \\ yu(Ax) \end{pmatrix}$   $A = \begin{pmatrix} -0.0348994967 \\ 1.249390827 \end{pmatrix}$

Grafik:  $Q1(\beta) := P(\beta, A_0, A_1)$   $Q2(\beta) := P(\beta, -A_0, A_1)$

$Q3(\beta) := P\left[\beta, -\frac{r1}{2}, \sqrt{h^2 - \left(-\frac{r1}{2}\right)^2} + r2 - r1\right]$   $Q4(\beta) := P(\beta, -0.07, 1.17)$



minimale Zahnbreite innen entspricht  
 Schnittpunkt der Bewegung der oberen Zahnbegrenzungen mit großem Rad innen:



Gesucht: Schnittpunkt B

Lösung liegt im Intervall:  $\beta := 0, 0.1 \dots \frac{2 \cdot \pi \cdot n}{n + 1}$

Versuch:  $\beta := \frac{2 \cdot \pi \cdot n}{n + 1} \cdot 2$   $\beta = 1.2566370614$

vorgabe

$$(Q1(\beta)_0)^2 + (Q1(\beta)_1)^2 = (h_2 \cdot r_2)^2$$

Näherungslösung:  $\eta := \text{minfehl}(\beta)$   $\eta = 0.9891473785$

Probe:

$$(Q1(\eta)_0)^2 + (Q1(\eta)_1)^2 = 1.21 \quad (h_2 \cdot r_2)^2 = 1.21$$

Punkt des Zahnes innen im großen Rad:  $B := Q1(\eta)$   $B = \begin{pmatrix} -0.0695735271 \\ 1.0977975789 \end{pmatrix}$

Übertragung der Koordinaten:  $bi2 := 2 \cdot \text{atan}\left(\frac{-B_0}{B_1}\right)$   $bi2 = 0.126581814$

Zahnbreiten für großes Rad müssen mindestens so groß wie die des kleineren Rades sein

Zahnflankengeradengleichung:  $b(a) := \frac{A_1 - B_1}{A_0 - B_0} \cdot (a - B_0) + B_1$

Schnittpunkte mit Kreisen:

$a := 0 \quad b := 1$

vorgabe

$$b = \frac{A_1 - B_1}{A_0 - B_0} \cdot (a - B_0) + B_1 \quad a^2 + b^2 = r^2$$

$C := \text{suchen}(a, b)$  Punkt des Zahnes außen im großen Rad:  $C = \begin{pmatrix} -0.034871438 \\ 1.2495134985 \end{pmatrix}$

vorgabe

$$b = \frac{A_1 - B_1}{A_0 - B_0} \cdot (a - B_0) + B_1 \quad a^2 + (b - (r_2 - r_1))^2 = (h \cdot r_1)^2$$

$D := \text{suchen}(a, b)$  Punkt des Zahnes innen im kleinen Rad:  $D = \begin{pmatrix} -0.0697249822 \\ 1.0971354242 \end{pmatrix}$

Übertragung der Koordinaten:  
(Zahnbreite außen großes Rad)  $ba_2 := 2 \cdot \text{atan} \left( \frac{-C_0}{C_1} \right) \quad ba_2 = 0.0558015403$

Übertragung der Koordinaten:  
(Zahnbreite innen kleines Rad)  $bi_1 := 2 \cdot \text{atan} \left[ \frac{-D_0}{D_1 - (r_2 - r_1)} \right] \quad bi_1 = 0.1642433281$

Zusammenfassung der Berechnungen:

kleines Rad:

Zahnbreite außen:  $ba_1 = 0.0698131701$

Zahnbreite innen:  $bi_1 = 0.1642433281$

großes Rad:

Zahnbreite innen:  $bi_2 = 0.126581814$

Zahnbreite außen:  $ba_2 = 0.0558015403$

**Berechnungen zur Zahnanzahl:**

Aus dem Verhältnis  $r_2/r_1$  resultiert die ganze Zahl der Zähne des äußeren Rades als Vielfaches  $k$  von:

$$n + 1 = 5$$

äußeres Zahnrad:  $k := 4$

$$z_2 := k \cdot (n + 1) \quad z_2 = 20 \quad (\text{Hier muß eine ganze Zahl stehen}) \rightarrow k \geq 1$$

Weg von einem Zahn zum nächsten:  
(bei beiden Rädern gleich)

$$s := \frac{2 \cdot \pi}{z_2} \cdot r_2 \quad s = 0.3926991$$

Daraus folgt:

inneres Zahnrad:  $z_1 := \frac{r_1}{r_2} \cdot z_2$  oder:  $z_1 := z_2 - k$  oder:  $z_1 := k \cdot n$

$$z_1 = 16 \quad (\text{Hier steht zwangsläufig eine ganze Zahl})$$

**Berechnungen zu Winkeln:**

Winkel vom MP großer Kreis zum MP des kleinen Kreises:

Eigenrotation des inneren Rades:

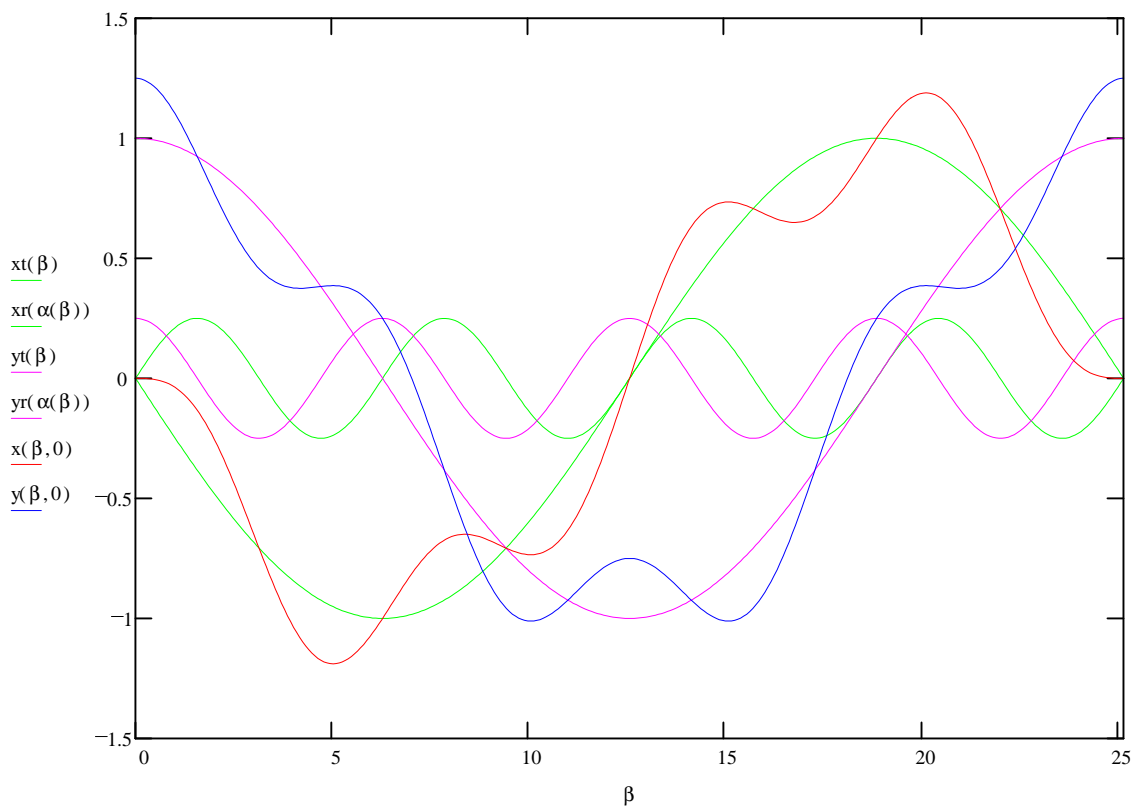
$$\alpha(\beta) := \beta \cdot \frac{r_2 - r_1}{r_1} \quad \alpha(2 \cdot \pi) = 1.5707963$$

Periodendauer:

$$T := 2 \cdot \pi \cdot n \quad T = 25.1327412$$

Grafik:

$$\beta := 0, \frac{(2 \cdot \pi) \cdot n}{325} .. (2 \cdot \pi) \cdot n$$





**Constraints:**

1. volle Umdrehungen:

$$n \geq 2$$

2. relative Zahnhöhe kleines Rad:

minimaler Außenpunkt-  
abstand:

$$h \geq \frac{|H|}{r1} \quad \frac{|H|}{r1} = 0.5765642497$$

$$H := Q1 \left( \frac{2 \cdot \pi \cdot n}{n+1} \right) - \begin{pmatrix} 0 \\ r2 - r1 \end{pmatrix} \quad H = \begin{pmatrix} -0.4687151623 \\ 0.335756505 \end{pmatrix} \quad |H| = 0.5765642497$$

3. Zahnanzahl kleines Rad:

$$z1 \leq \frac{2 \cdot \pi}{bi1} \quad \frac{2 \cdot \pi}{bi1} = 38.2553457805$$

Grafik: Geradengleichung,  
durch die Schnittpkt geht:

$$d(x) := \frac{-x}{\tan\left(\frac{\pi}{n+1}\right)}$$

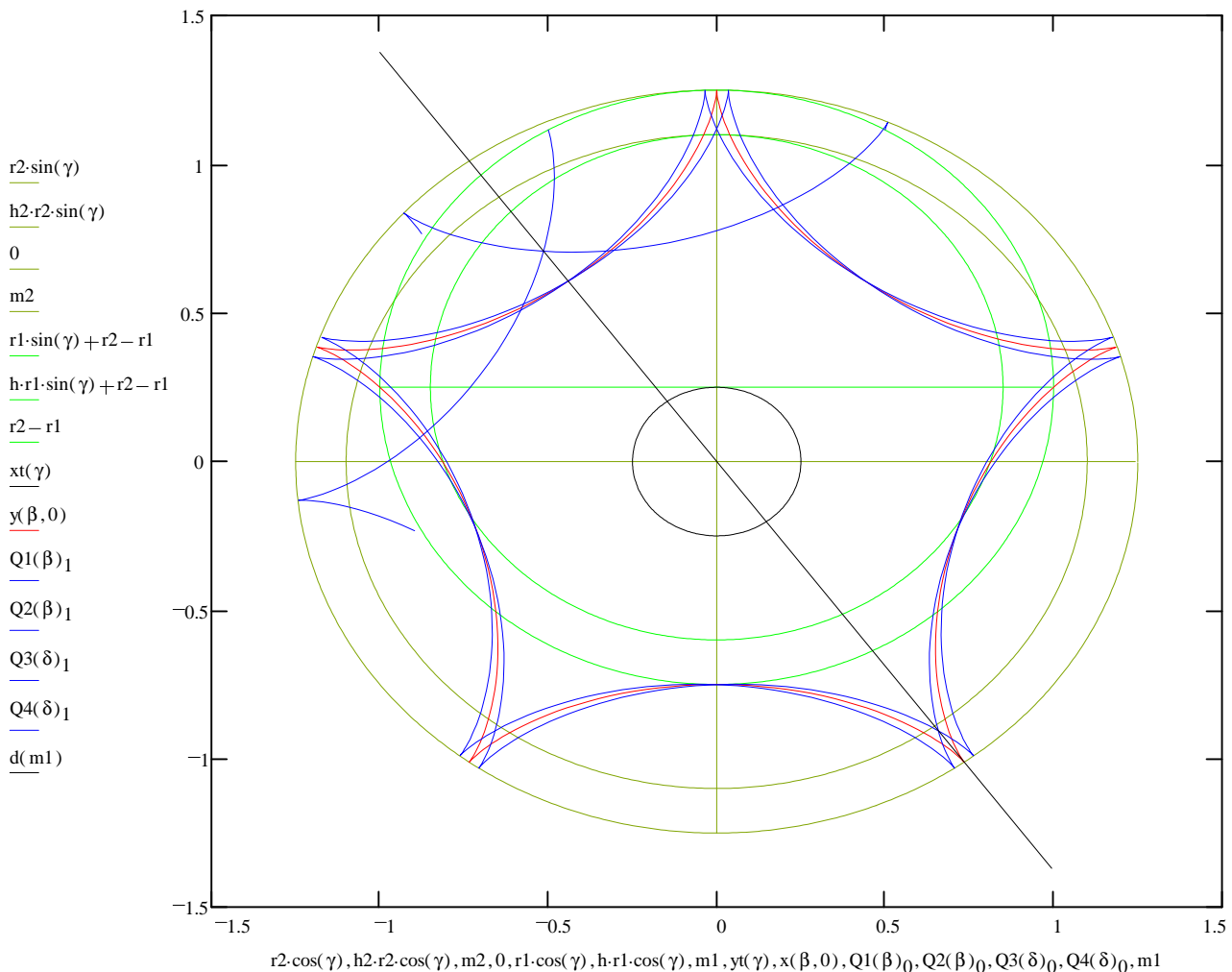
$$Q1(\beta) := P(\beta, A_0, A_1)$$

$$Q2(\beta) := P(\beta, -A_0, A_1)$$

$$Q3(\beta) := P(\beta, -0.5, yu(-0.5))$$

$$Q4(\beta) := P(\beta, 0.5, yu(0.5))$$

$$\delta := 0, 0.1 \dots \frac{2 \cdot \pi \cdot n}{n}$$



4. Zahnbreite außen kleines Rad (ba1):

Zahnbreite Stirnrad außen darf nicht größer sein als die Bewegung desselben,  
sonst dürfte das umgebende Hohlrad keine Zähne haben!!!

$$\text{Winkel der Schnittgeraden: } w_2 := \frac{\pi}{n+1} \quad w_2 = 0.6283185307$$

Hilfsgröße a,

$$\text{Weg vom MP } r_2 \text{ bis zum Umfang von } r_1: \quad a^2 - 2 \cdot (r_2 - r_1) \cdot \cos(w_2) \cdot a + (r_2 - r_1)^2 - r_1^2 = 0$$

Lösung für a:

$$L := \begin{bmatrix} \cos(w_2) \cdot r_2 - \cos(w_2) \cdot r_1 + \frac{1}{2} \cdot \sqrt{4 \cdot \cos(w_2)^2 \cdot r_2^2 - 8 \cdot \cos(w_2)^2 \cdot r_2 \cdot r_1 + 4 \cdot \cos(w_2)^2 \cdot r_1^2 - 4 \cdot r_2^2 + 8 \cdot r_2 \cdot r_1} \\ \cos(w_2) \cdot r_2 - \cos(w_2) \cdot r_1 - \frac{1}{2} \cdot \sqrt{4 \cdot \cos(w_2)^2 \cdot r_2^2 - 8 \cdot \cos(w_2)^2 \cdot r_2 \cdot r_1 + 4 \cdot \cos(w_2)^2 \cdot r_1^2 - 4 \cdot r_2^2 + 8 \cdot r_2 \cdot r_1} \end{bmatrix}$$

$$a \geq 0 \quad a := L_0 \quad a = 1.1913987179 \quad L = \begin{pmatrix} 1.1913987179 \\ -0.7868902207 \end{pmatrix}$$

$$\text{Ermittlung des maximalen } ba_1: \quad w_1 := \text{asin}\left(\frac{a \cdot \sin(w_2)}{r_1}\right)$$

$$ba_1 < 2 \cdot w_1 \quad 2 \cdot w_1 = 1.5515977804$$

2. Constraint für ba1,

da bei Überschreitung die Zahnflanke außen bereits innerhalb des Hohlrades liegt:

$$w_1 := \text{acos}\left[\frac{(h_2 \cdot r_2)^2 + r_2^2 + \frac{r_2}{r_2 - r_1} \cdot [r_1^2 - (h_2 \cdot r_2)^2 - (r_2 - r_1)^2] - 2 \cdot r_1^2}{-2 \cdot r_1}\right]$$

$$w_1 := \text{acos}\left[\frac{1}{2} \cdot \frac{(-h_2^2 \cdot r_2^2 + r_2^2 - 2 \cdot r_2 \cdot r_1 + 2 \cdot r_1^2)}{(-r_2 + r_1)}\right]$$

$$ba_1 < 2 \cdot w_1 \quad 2 \cdot w_1 = 2.5426816162$$

5. relative Höhe kleines Rad:

h-Constraint = Schnittpunkt der Wege der Zahnflanken eines Zahnes außen (Q1(β)=Q2(β))

Lösung liegt im Intervall:  $\beta := 0,01.. \frac{2 \cdot \pi \cdot \frac{n}{n+1}}{2}$  Versuch:  $\beta := \frac{2 \cdot \pi \cdot \frac{n}{n+1}}{2 \cdot 2}$   $\beta = 1.2566370614$   $dx := 0$   
 vorgabe

$$Q1(\beta)_0 = dx$$

$$Q1(\beta)_1 = d(dx)$$

Lösung:  $\eta := \text{suchen}(\beta, dx)$   $\eta = \begin{pmatrix} 2.4434343423 \\ -0.4411077116 \end{pmatrix}$

$$H := \begin{pmatrix} \eta_1 \\ d(\eta_1) \end{pmatrix} - \begin{pmatrix} 0 \\ r2 - r1 \end{pmatrix} \quad H = \begin{pmatrix} -0.4411077116 \\ 0.3571326792 \end{pmatrix} \quad |H| = 0.5675559565$$

$$h \geq \frac{|H|}{r1} \quad \frac{|H|}{r1} = 0.5675559565$$

2. Constraint für h1:

$$ba1 = 2 \cdot \arccos \left[ \frac{1}{2} \cdot \frac{(-h^2 \cdot r2^2 + r2^2 - 2 \cdot r2 \cdot r1 + 2 \cdot r1^2)}{(-r2 + r1)} \right]$$

$$h2_{max} := \frac{\sqrt{2 \cdot \cos\left(\frac{1}{2} \cdot ba1\right) \cdot r2 - 2 \cdot \cos\left(\frac{1}{2} \cdot ba1\right) \cdot r1 + r2^2 - 2 \cdot r2 \cdot r1 + 2 \cdot r1^2}}{r2} \quad h2_{max} = 0.9999025276$$

$$h1_{max} := 1 - \left[ (1 - h2_{max}) \cdot \frac{r2}{r1} \right]$$

$$h1 < h1_{max} \quad h1_{max} = 0.9998781595$$

## **7 Quellenangabe**

- Java3D – API
- David Flanagan – Java in a nutshell
- [https://www.htwm.de/martina/intranet/grafik/index\\_grafik.html](https://www.htwm.de/martina/intranet/grafik/index_grafik.html)

## **8 Bemerkungen**

Hiermit versichere ich, den Beleg selbständig, nur unter Zuhilfenahme oben genannter Quellen, erstellt zu haben.

Unterschrift

Mittweida, den 12.01.2003

Stefan Pohl