

Julius-Echter-Gymnasium
8751 Elsenfeld

Kollegstufenjahrgang 1988/90

F A C H A R B E I T

aus dem Fach

Mathematik

Mathematische Erfassung von rotationssymmetrischen
biologischen Körpern und deren computergestützte
Verarbeitung

Verfasser: Marcus Bäckmann

Leistungskurs: Mathematik

Kursleiter: OStR Bruno Göb

Abgabetermin: 01.02.1990

Erzielte Note: in Worten:

Erzielte Punkte: in Worten:
(einfache Wertung)

Eintrag des Ergebnisses:

.....
(Unterschrift des Kursleiters)

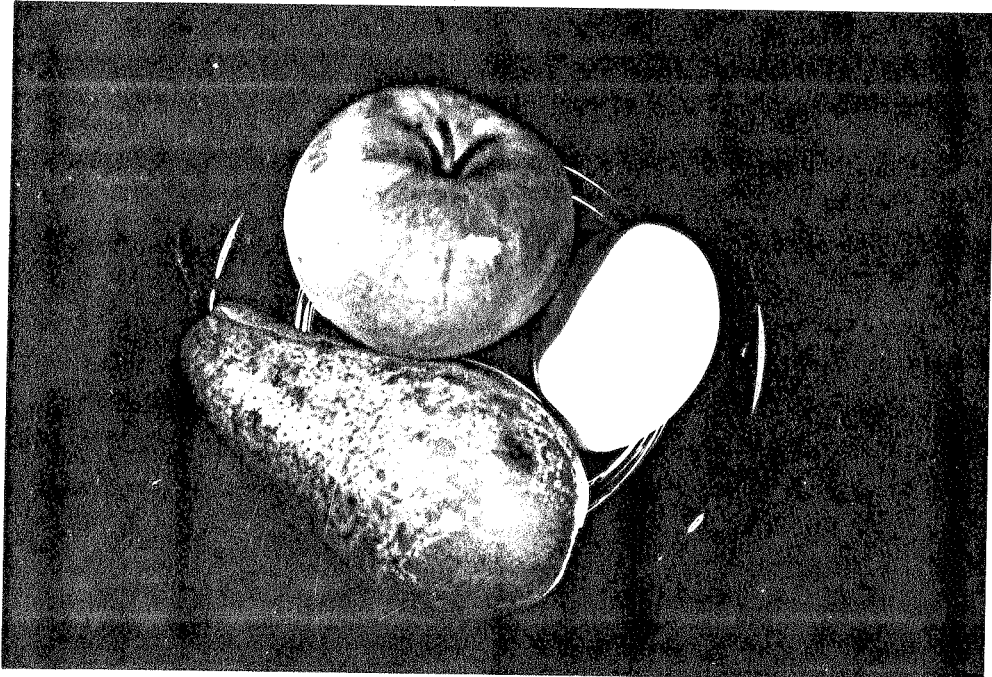
F A C H A R B E I T

aus dem Fach

Mathematik

Apfel - Birne - Ei

Mathematische Erfassung von rotationssymmetrischen
biologischen Körpern und deren computergestützte
Verarbeitung



Verfasser: Marcus Bäckmann
Hauptstraße 151a
8751 Heimbuchenthal

Abgabetermin: 01.02.1990

Inhaltsverzeichnis

1	Elementare Methoden.....	3
2	Infinitesimale Methoden.....	4
2.1	Volumen unter dem Graph von $f(x)$	5
2.1.1	Zerlegung in Schichten.....	5
2.1.2	Drehung des Flächenschwerpunktes (Guldin'sche Regel).....	6
2.2	Volumen unter einer Funktion in Parameter- darstellung.....	8
2.2.1	Die Parameterform einer Funktion.....	9
2.2.2	Volumenbestimmung unter einer Funktion in Parameterdarstellung.....	10
3	Computerunterstützung.....	13
3.1	Modula2 auf dem Amiga.....	13
3.2	Plotten eines Funktionsgraphen.....	14
3.2.1	Explizite Form $f(x)$ [Programm Drehefx.MOD].....	14
3.2.2	Parameterform [Programm DreheParam.MOD].....	15
3.3	Drehung um die X-Achse.....	16
3.3.1	Explizite Form [Programm Drehefx.MOD].....	18
3.3.2	Parameterform [Programm DreheParam].....	19
3.4	Die Perspektivdarstellung am Bildschirm.....	19
3.5	Die Bedienung der Programme.....	22
3.6	Die numerische Integration nach Gauß.....	23
4	Praktische Anwendungen.....	24
4.1	Die Birne.....	24
4.2	Das Ei.....	30
4.3	Der Apfel.....	36
5	Ausblicke.....	43
6	Anhang: Programme.....	44
6.1	Programm Drehe Funktion $f(x)$	44
6.2	Programm Drehe Parameterform.....	51
6.3	Programm Gaußintegration.....	59
7	Literaturverzeichnis.....	61
8	Erklärung.....	63

1 Elementare Methoden

Im allgemeinen ist der römische Staatsmann, Redner und Geschichtsschreiber Marcus Tullius Cicero meist nur als Schöpfer und Verfasser von Senats- und Gerichtsreden und philosophischen Schriften bekannt. Daß er einen Beitrag zur Geschichte der Mathematik leistete, erscheint vielen als nebensächlich.

Mehr als 200 Jahre vor Ciceros Amtszeit in Sizilien lebte der wohl bedeutendste Mathematiker der Antike, Archimedes von Syrakus. Die wichtigsten Arbeiten von Archimedes betrafen Grundprobleme der Stereometrie, die der Ausarbeitung von Methoden zur Bestimmung von Flächen- und Rauminhalten der klassischen Figuren und Körper Zylinder, Kegel, Kugel, Quader und Pyramiden dienten.

Er experimentierte dazu mit dem Hebelgesetz, um die Formel des Kugelvolumens zu erhalten, und er fand auf Grund von Annahmen, daß das Volumen eines Zylinders und der von ihm umschlossenen Kugel im Verhältnis 3 : 2 stehen. Diese Erkenntnis hatte bei Archimedes anscheinend besonders große Begeisterung ausgelöst, denn er wünschte sich auf seinem Grab einen Zylinder und eine Kugel aufgestellt.

Syrakus wurde 212 v.Chr. nach zweijähriger Belagerung von den Römern eingenommen, bei der üblichen Plünderung der Stadt kam auch der greise Archimedes ums Leben. Als Cicero nun 75 v.Chr. als Quaestor auf Sizilien residierte, gelang es ihm, das damals bereits unbekannte und verfallene Grab des Archimedes anhand dieser Säule zu identifizieren und der Nachwelt zu beschreiben ([1], S. 33ff).

Allerdings hatte schon die Mathematik der vorgriechischen Zeit Flächen- und Raumberechnungen gekannt, die fast immer aus Aufgaben der Landvermessung und der Bautechnik resultierten. Archimedes jedoch erweiterte diese Berechnung auf Drehkörper zweiter Ordnung, deren Begrenzungsflächen nun auch gekrümmt sein konnten.

Spätere Mathematiker wie Pappos von Alexandria stellten im wesentlichen Sammelwerke zusammen, worunter besonders die Bemerkungen über die Extremwertbehandlung und der sogenannte Guldin'sche Schwerpunktsatz an Drehkörpern hervorzuheben sind. Die meisten dieser Schriften wurden zu Ansatzpunkten für die weiterführenden Methoden des 16. und 17. Jahrhunderts. Hier treten die Visierbücher mit Anleitungen für die Küfer zum Eichen von Fässern und zur Volumenberechnung auf. Vor allem auf die Anwendung (z.B. im Gelände oder zur Raumbestimmung von Fässern) kam es auch in den zahlreichen geometrischen Schriften dieser Zeit an (vgl. [1]).

2 Infinitesimale Methoden

Systematisiert wurde die Volumenberechnung von Johannes Kepler. Er griff dabei auf die aus der Antike stammenden Methoden vor allem des Archimedes zurück. Und dennoch ging Kepler methodisch weit über Archimedes hinaus, indem er den Begriff vom unendlich Kleinen in mathematische Rechenmethoden einbezog.

Infinitesimale Zerlegungsgedanken sprach Kepler für Kugel und Zylinder aus. Zum Beispiel schrieb er: »Die Kugel besteht aus unendlich vielen Kegeln, deren Scheitel im Mittelpunkt zusammenreffen, und deren auf der Oberfläche gelegene Grundflächen durch Punkte ersetzt sind.«

Aus Kugeln, Zylindern und Kegelstümpfen setzte Kepler komplizierte Körper näherungsweise zusammen, zum Beispiel das Faß aus Zylinder und zwei Kegelstümpfen. Insgesamt war Kepler so imstande, weit über Archimedes hinaus zu gehen, auch hinsichtlich der Anzahl der Körper, die - zumindest näherungsweise - dem Volumen nach berechenbar wurden. Kepler gab unter anderem auch Verfahren für *Torus*, *zitronenförmige*, *apfelförmige*, *birnen- und pflaumenförmige Körper*, *Quitte*, *Kürbis*, *Olive*, *Spindel* und andere mehr an. Keplers Vermählung war dann der Anlaß der berühmten »Faßrechnung« im Jahre 1615. Kepler selbst berichtete

»Als ich im November des letzten Jahres (1613) meine Wiedervermählung feierte ... da war es die Pflicht des neugebackenen Gatten ..., für sein Haus den nötigen Hastrunk zu besorgen. Als einige Fässer eingekellert waren, kam am vierten Tag der Verkäufer mit einer Meßrute, mit der er alle Fässer ohne Rücksicht auf ihre Form, ohne jede weitere Überlegung oder Rechnung, ihrem Weininhalt nach bestimmte. Die Visierrute wurde mit ihrer metallenen Spitze durch das oben befindliche Spundloch quer bis zu den Rändern der beiden Faßböden eingeführt, und als Längen gleich gefunden worden waren, ergab die Marke am Spundloch die Zahl der Eimer im Faß« ([2], S. 15).

Kepler bezweifelte zu Recht die Richtigkeit dieser Meßmethode. Er beschloß, die Rauminhaltsbestimmung = Stereometrie selbst zu behandeln. Er veröffentlichte deshalb zuerst 1615 einen Theorieband und anschließend 1616 einen Praxisband, bekannt unter dem Namen »Österreichisches Wein-Visier-Büchlein«.

Zwar läßt man Keplers Schlußweisen nur in Bezug auf ihren praktischen Wert gelten, trotzdem wurden diese über Cavalieri, Pascal, Fermat, Newton und besonders Leibniz wegberreitend für die moderne Integralrechnung. Denn mit dieser ist die stereometrische Erfassung rotationssymmetrischer und darüberhinaus auch beliebiger Körper, sei es die Berechnung der Oberfläche oder des Volumens, jeder technischen wie praktischen Problematik gewachsen (vgl. [2]).

Die Infinitesimalrechnung bietet zur Volumenberechnung von Körpern die Integration und Differentiation von Funktionen als Hilfsmittel an. Insbesondere für Rotationskörper werden durch die Integration die Beschränkungen der einfachen Stereometrie aufgehoben. Unter Rotationskörpern sind dreidimensionale Gebilde zu verstehen, die durch Rotation eines oder mehrerer Kurvenzüge um eine Drehachse entstehen.

2.1 Volumen unter dem Graph von $f(x)$

Rotiert also der Graph einer Funktion $f(x)$ um die X-Achse, so wird der Mantel eines Rotationskörpers überstrichen (vgl. [3], S. 279). Um das einbeschriebene Volumen des erzeugten Drehkörpers zu bestimmen, existieren zwei unterschiedliche Verfahren, die hier zunächst erläutert werden sollen.

2.1.1 Zerlegung in Schichten

Ein Rotationskörper, der durch die Drehung des Flächenintegrals um die X-Achse entsteht, kann in einen "Stapel" von verschiedenen Kreisen, deren Mittelpunkte auf der X-Achse liegen, zerlegt werden. Der Radius jedes einzelnen Kreises entspricht dem Funktionswert $f(x)$ an der Stelle x . Somit ergibt sich für den Flächeninhalt einer kreisförmigen Schicht die Beziehung

$$A = r^2 \pi$$

und aus $r = f(x)$ folgt damit

$$A(x) = f(x)^2 \pi \quad . \quad (1)$$

Das Volumen wird nun durch die Bildung des Integrals über die einzelnen Flächenelemente $A(x)$ von x_1 bis x_2 berechnet mit

$$V = \int_{x_1}^{x_2} (f(x)^2 \cdot \pi) dx = \pi \int_{x_1}^{x_2} f(x)^2 dx \quad (2)$$

(mit $x_1, x_2 \in \mathbb{R}, x_1 < x_2$). (vgl. [4], S. 239)

Die rotierende Funktion muß im Drehintervall $[x_1; x_2]$ stetig integrierbar sein, außerdem muß gelten $f(x) \geq 0$ mit $x \in [x_1; x_2]$.

Beispiel:

$$f(x) = \frac{x}{3} \sqrt{6x - x^2}$$

Der Graph wird um die X-Achse gedreht (Bild 2-1), wobei das Volumen des entstehenden Körpers von der Y-Achse bis zur nächsten Nullstelle bestimmt werden soll.

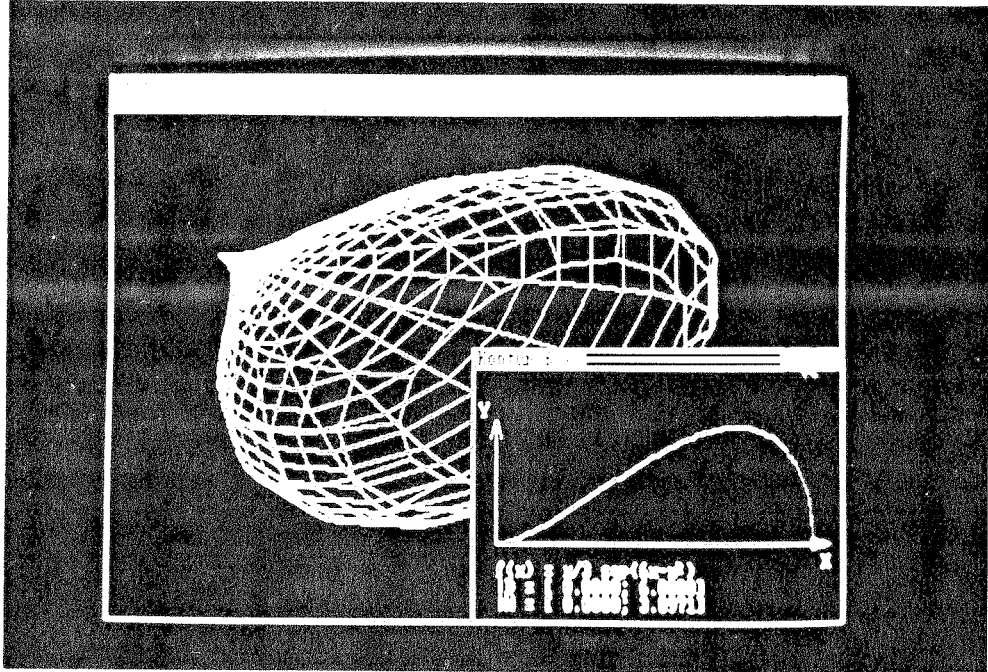


Bild 2-1: Der Graph und der Rotationskörper der Funktion

Voraussetzungsgemäß ist $x_1 = 0$, für $f(x_2) = 0$ ergibt sich $x_2 = 6$.

$$\begin{aligned} V_f(x) &= \pi \int_0^6 \left(\frac{x}{3} \sqrt{6x - x^2} \right)^2 dx = \pi \int_0^6 \frac{x^2}{9} (6x - x^2) dx \\ &= \frac{\pi}{9} \int_0^6 (6x^3 - x^4) dx = \frac{\pi}{9} [1,5x^4 - 0,2x^5]_0^6 \\ &= \pi \frac{216}{5} = \pi \cdot 43,2 \approx 135,72 \text{ (Volumeneinheiten VE)} \end{aligned}$$

2.1.2 Drehung des Flächenschwerpunktes (Guldin'sche Regel)

Eine zweite Methode, um den Rauminhalt eines Rotationskörpers zu bestimmen, resultiert aus der 1. Guldin'schen Regel.

Diese besagt, daß das Volumen eines Rotationskörpers das Produkt aus dem Inhalt der erzeugenden Fläche und dem Umfang des von dem Flächenschwerpunktes beschriebenen Kreises ist ([3], S. 281 und [5], S. 145) mit

$$V = 2\pi S_y A \quad (3)$$

A Flächeninhalt der erzeugenden Fläche

S_y Y-Koordinate des Flächenschwerpunktes.

Die Fläche A ergibt sich aus dem Integral über die Funktion $f(x)$ von x_1 bis x_2 . Die Schwerpunktskoordinaten eines homogenen Flächenstücks, das von der Kurve $y = f(x)$, der X-Achse und den Geraden $x = x_1$ und $x = x_2$ begrenzt wird, sind

$$S_x = \frac{\int_{x_1}^{x_2} x f(x) dx}{\int_{x_1}^{x_2} f(x) dx} \quad (4) \quad S_y = \frac{\int_{x_1}^{x_2} f(x)^2 dx}{2 \int_{x_1}^{x_2} f(x) dx} \quad (5)$$

(vgl. [5], S. 346 und [3], S. 292)

Zur Berechnung des Volumens wird allerdings nur die Koordinate S_y benötigt, die Formel zur Berechnung von S_x sei nur der Vollständigkeit halber erwähnt.

Auch hierzu ein Beispiel bei der Rotation der Funktion $f(x) = \sin x$, im Intervall $[0; \pi]$ (Bild 2-2).

Der Inhalt der erzeugenden Fläche ist

$$A = \int_0^{\pi} \sin x dx = [-\cos x]_0^{\pi} = 2 \text{ (Flächeneinheiten FE).}$$

Gleichung (5) führt zu

$$S_y = \frac{\int_0^{\pi} \sin^2 x dx}{2 \int_0^{\pi} \sin x dx} = \frac{\left[\frac{1}{2} \left(x - \frac{1}{2} \sin 2x \right) \right]_0^{\pi}}{2 [-\cos x]_0^{\pi}} = \frac{\pi}{8}$$

Einsetzen in (3) ergibt

$$V = 2\pi \cdot \frac{\pi}{8} \cdot 2 = \frac{\pi^2}{2} \approx 4,93 \text{ (VE).}$$

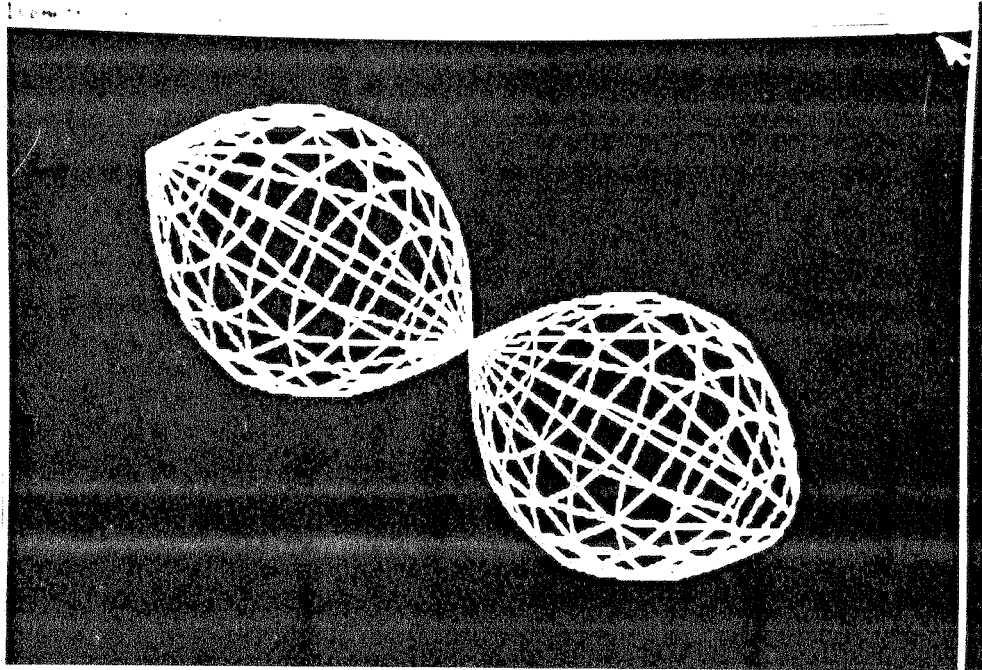


Bild 2-2: Der Rotationskörper der Funktion $f(x) = \sin x$ für $x \in [0; 2\pi]$.

Ein Nachteil dieser Berechnungsmethode soll nicht verschwiegen werden, denn bei der Volumenberechnung mit (2) wird nur einmal ein Integral berechnet, bei der Schwerpunktbestimmung müssen jedoch zweimal Integrale berechnet werden, was die Berechnung häufig unnötig verkompliziert. Der Vorteil von (3) kommt dann zum Tragen, wenn, anders als im vorhergehenden Beispiel, der Schwerpunkt bereits bekannt ist, weil dann zwei gleichwertige Alternativen bestehen, die jeweils verschiedene Vorteile haben. So wäre, immer vorausgesetzt, daß der Schwerpunkt bekannt ist, bei einer Funktion mit quadratischen Wurzelgliedern wegen der Elimination der Wurzel durch das Quadrat Formel (2) zu bevorzugen. Bei Polynomen oder bei trigonometrischen Funktionen ist Methode (3) besser, da hier der höchste Grad nicht noch einmal erhöht wird.

2.2 Volumen unter einer Funktion in Parameterdarstellung

Nicht nur die explizite Funktionsform $y = f(x)$ kann um die X-Achse gedreht werden, dies geht auch bei einer weiteren Art, der Parameterform einer Funktion. Doch bevor man die

Parameterform verwendet, sollte man wissen, wie diese aufgebaut ist.

2.2.1 Die Parameterform einer Funktion

Häufig ist die Darstellungsweise von Funktionen und vor allem von Kurven in Form von Ausdrücken der Art $y = f(x)$ nicht mehr ausreichend. Bei der expliziten Form der Funktionsgleichung wird jedem Wert x höchstens ein Wert $y = f(x)$ zugeordnet. Tritt der Fall auf, daß einem x -Wert zwei oder mehr y -Werte zugeordnet werden, läßt sich dieser Graph nicht mehr in expliziter Form darstellen. Aus diesem Grund ist für viele Anwendungsfälle die Darstellung der Funktion in Parameterform vorteilhaft bis notwendig ([6], Kapitel 2).

Ist bei einer Funktion $y = f(x)$ sowohl x als auch y wieder eine Funktion eines gemeinsamen Parameters t , dann ist

$$x = \phi(t) = x(t)$$

$$y = \delta(t) = y(t)$$

eine Parameterdarstellung der Funktion $y = f(x)$ (vgl. [7], S. 280).

Die Parameterform kann erzeugt werden, indem man $x = x(t)$ beliebig wählt und $y = y(t)$ durch das Einsetzen von $x = x(t)$ in $y = f(x)$ bestimmt. Die Wahl von $x = x(t)$ sollte zweckmäßiger Weise so sein, daß die Funktionen $x(t)$ und $y(t)$ einfache und vor allem eindeutige Funktionen von t sind (vgl. [8], S. 118f).

Beispiel:

$$f(x) = \sqrt{x}$$

Man wählt $x(t) = t^2$ und setzt ein in $y(t) = f(x(t))$, dabei wird $y(t) = \sqrt{t^2} = t$.

Somit sind die beiden Ausdrücke $x(t) = t^2$; $y(t) = t$ die Parameterform der Funktion $f(x) = \sqrt{x}$.

Umgekehrt ist aber auch wieder die Elimination des Parameters t und die Rückführung in die explizite Form möglich. Dies wird erreicht, indem entweder $x(t)$ oder $y(t)$ nach t umgestellt und in die andere Funktion eingesetzt wird. Ein erneutes Umstellen, diesmal nach y , ergibt die explizite Form.

Eine Einschränkung muß gemacht werden: zu jeder expliziten Form gibt es sehr viele verschiedene Parameterdarstellungen, aber nicht jede Parameterdarstellung ergibt zwangsläufig eine explizite Form, da oft ein Umstellen nach t nicht möglich ist.

Hier ein Beispiel für die Rückführung in die explizite Form.

Die Funktionen $x(t) = 2t + 3$ und $y(t) = 3t + 2$ sollen parameterfrei dargestellt werden ([6], Beispiel 2.5),

wobei aus $x(t)$ folgt $t = 0,5 \cdot (x - 3)$

wird dies eingesetzt in $y(t)$, wird $y = 3 \cdot 0,5 \cdot (x - 3) + 2$
 $= 1,5x - 2,5$

Der Ausdruck $f(x) = 1,5x - 2,5$ stellt nun die parameterfreie Form dar.

2.2.2 Volumenbestimmung unter einer Funktion in Parameterdarstellung

Da die Parameterdarstellung einer Funktion aus der expliziten Form hervorgeht, läßt sich die Funktion auch in dieser Form um die X-Achse rotieren. Voraussetzung ist aber, daß die Rotationsfunktion im Rotationsintervall $[t_1; t_2]$ nur im positiven Bereich des Koordinatensystems liegt, daß also gilt $x(t) \geq 0$ mit $t \in [t_1; t_2]$. Die Volumenformel ergibt sich nun durch Umformung von (2).

Bei der Integration bestimmter Integrale gilt die 2. Substitutionsregel

$$\int_{\Phi(a)}^{\Phi(b)} f(u) \, du = \int_a^b f(\Phi(x)) \Phi'(x) \, dx \quad (6)$$

([9], S. 493 und [10], S. 465)

in diese wird die Parameterform einer Funktion eingesetzt

$$x(t) = \Phi(t)$$

$$y(t) = f(x(t)) = f(\Phi(t))$$

$$\int_{x_1}^{x_2} f(x)^2 dx = \int_{t_1}^{t_2} f(\Phi(t))^2 \Phi'(t) dt$$

$$x_1 = x(t_1) = \Phi(t_1)$$

$$x_2 = x(t_2) = \Phi(t_2)$$

daraus ergibt sich die Volumenformel für Funktionen in Parameterform durch das Zurückeinsetzen

$$V = \pi \int_{t_1}^{t_2} y^2(t) x'(t) dt \quad . \quad (7)$$

Beispiel:

Der Graph der Kardioide mit

$$x(t) = 2\cos t - \cos 2t$$

$$y(t) = 2\sin t - \sin 2t$$

wird für $t_1 = \pi$; $t_2 = 0$ um die X-Achse rotiert (Bild 2-3).

Der Grund, warum von π bis 0 integriert wird, und nicht umgekehrt, liegt darin, daß Formel (7) aus Formel (2) entstand. Bei (2) wird von links nach rechts integriert. Bei (7) entsteht ein kleineres x aus einem größerem t , ein größeres x aus einem kleinerem t , da der Umlaufsinn der Kurve gegen den Uhrzeigersinn gerichtet ist. Also wird auch hier - bildlich gesprochen - von links nach rechts integriert.

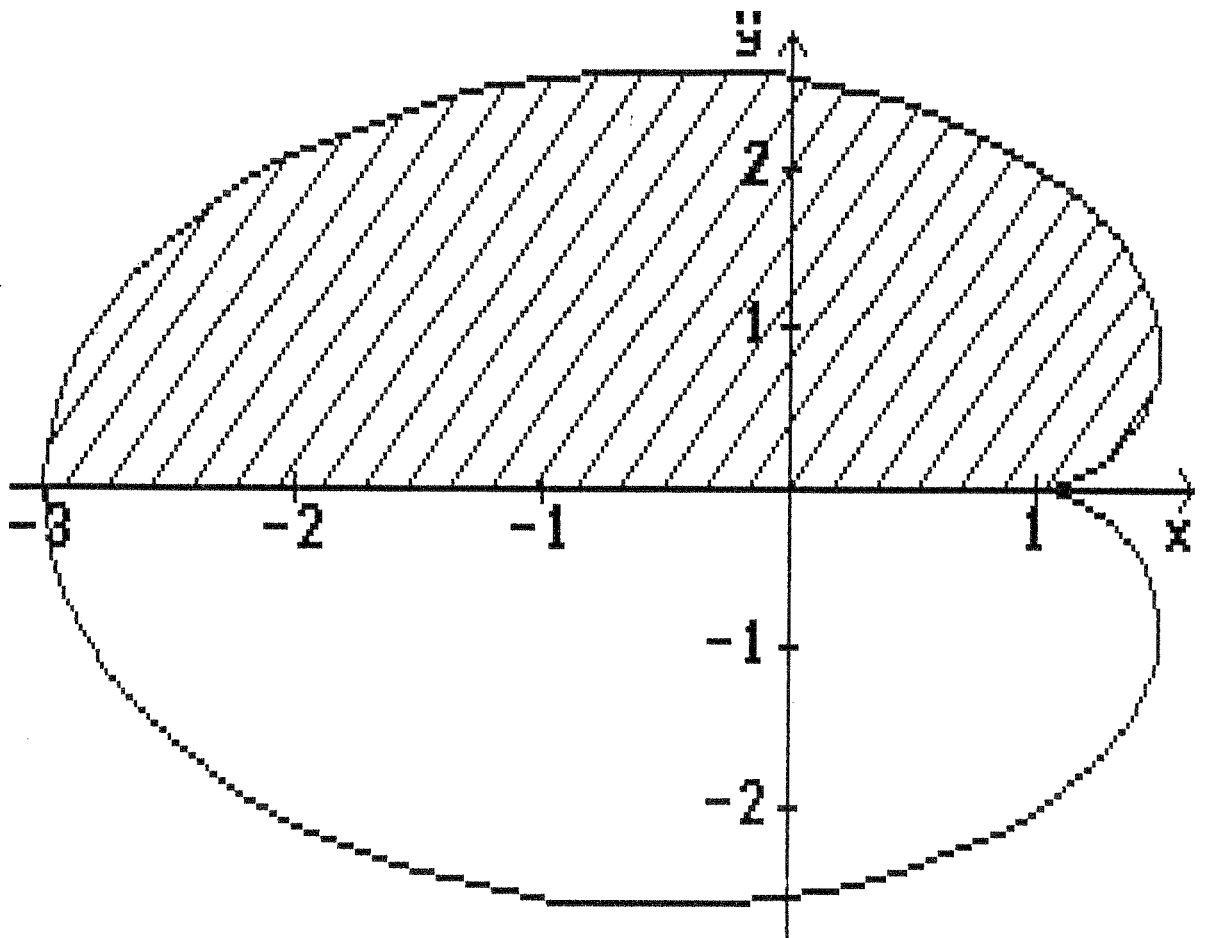


Bild 2-3: Der Graph der Kardioide; der Bereich $t \in [0; \pi]$ ist schraffiert dargestellt.

$$\begin{aligned}
 V &= \pi \int_{\pi}^0 (2\sin t - \sin 2t)^2 (-2\sin 2t + 2\sin 2t) dt \\
 &= \pi \int_{\pi}^0 (4\sin^2 t + \sin^2 2t - 4\sin t \sin 2t) (2\sin 2t - 2\sin t) dt \\
 &= \pi \int_{\pi}^0 (-8\sin^3 t + 2\sin^3 2t - 10\sin t \sin^2 2t + 16\sin^2 t \sin 2t) dt \\
 &= \pi \int_{\pi}^0 (-8\sin^3 t + 16\sin^3 t \cos^3 t - 40\sin^3 t \cos^2 t + 32\sin^3 t \cos t) dt \\
 &= \pi \left[8\cos t + \frac{8}{3}\cos^3 t - \frac{16}{6}\sin^2 t \cos^4 t - 2\cos^4 t + 8\sin^2 t \cos^3 t + 8\sin^4 t \right]_{\pi}^0 \\
 &= \pi \left(8 - \frac{8}{3} - 2 + \frac{16}{3} + 8 - \frac{8}{3} + 2 + \frac{16}{3} \right) = \pi \frac{64}{3} \\
 &\approx 67,02 \text{ (VE)}
 \end{aligned}$$

3 Computerunterstützung

Seit der Erfindung der ersten integrierten Schaltungen hat sich die Mikrocomputertechnik überall durchgesetzt. In einem großen Teil der Haushalte stehen sogenannte Heimcomputer, kaum ein Büro kommt heute noch ohne einen Personalcomputer aus. Nicht nur die Anzahl der Geräte hat sich lawinenartig erhöht, auch die Ausstattung der Computer wurde, vor allem was Geschwindigkeit und Speicherkapazität betrifft, um ein Vielfaches verbessert. Im Jahr 1969 hatte der Bordcomputer der Mondlandefähre "Eagle" einen Speicher von 4 KByte, was 4096 Zeichen entspricht. Ein mittlerer Heimcomputer hat heute gut seine 1024 KByte, was eine Speicherkapazität von stattlichen 1.048.576 Zeichen bedeutet. Geradezu prädestiniert sind diese Geräte für mathematische Anwendungen - letztendlich ist auch der Computer ein Kind der Mathematik, die Digitaltechnik basiert auf der Bool'schen Algebra, einem Teilgebiet der Mathematik.

Daher bietet es sich an, den Computer Tätigkeiten verrichten zu lassen, die er genauer und schneller erledigt als der Mensch, zum Beispiel das Skizzieren von Funktionen oder das Zeichnen von Körpern. Wird heute ein Funktionsgraph mühselig mit Hand Punkt für Punkt gezeichnet, so scheint es sich im Zeitalter der Mikrocomputer dabei um einen Anachronismus zu handeln. Daher soll unser Computer die Bilder zeichnen, wir verschwenden unsere Zeit nicht mehr damit, die Probleme darzustellen, sondern verwenden sie für etwas, das wir besser können: für das Lösen der Probleme.

3.1 Modula2 auf dem Amiga

Alle Computerprogramme, die sich im Anhang befinden, sind auf einem Amiga 500 in der Sprache Modula2 entwickelt worden, als Compiler wurde das M2Amiga-Entwicklungssystem Version 3.3d von A+L eingesetzt. Im Gegensatz zu den meisten Programmiersprachen ist Modula2 noch sehr neu. Sie wurde 1980 von Niklaus Wirth, der bereits 1970 die Sprache Pascal entwickelte, als Verbesserung der Sprachen Pascal und Modula vorgestellt. Wirth ist als Professor an der Eidgenössischen Technischen Hochschule Zürich (ETHZ) tätig.

In Modula2 werden Variablen, genau wie in Pascal, vor ihrer ersten Verwendung bereits definiert, wobei ihnen ein bestimmter Typ zugewiesen wird (Bsp: `i : INTEGER` für eine Ganzzahl im Bereich `[-32768;32767]`). Bei den Variablen wird zwischen lokalen und globalen Variablen unterschieden, das

heißt, daß in einem Unterprogramm, PROCEDURE genannt, eine Variable definiert werden kann, die nur in diesem Unterprogramm verfügbar ist. Anders als in Basic muß der Programmierer sich nicht bei jeder neu angelegten Variablen vergewissern, ob diese nicht schon einmal verwendet wurde. Unterschiede zu Pascal bestehen darin, daß der Befehl GOTO entfernt wurde, und daß der Compiler noch genauer Variablenzuweisungen verschiedener Typen prüft ([11], S. 409ff und [12], S. 11f).

Das Betriebssystem vom Commodore Amiga bietet eine sehr leistungsfähige grafische Benutzeroberfläche, genannt Intuition, an. Zu ihren Vorzügen gehören verschiedene Bildschirmfenster und Schalter, die mit der Maus bedient werden. Selbstverständlich wird dieser Teil des Betriebssystems von den beiliegenden Programmen voll unterstützt, verschiedene Grafiken werden auch in verschiedene Fenster gezeichnet, die Menüsteuerung erfolgt über die Maus.

3.2 Plotten eines Funktionsgraphen

Da die Zeichnung eines Funktionsgraphen am Bildschirm nichts grundlegend Neues ist, soll das Vorgehen nur grob erläutert werden. Zuerst wird der Wertebereich und der Definitionsbereich der Funktion untersucht. Danach wird ein Skalierungsfaktor berechnet, um den Graphen der Funktion in das vorgegebene Fenster einzupassen. Jetzt werden die Achsen des Koordinatensystems gezeichnet, schließlich der Graph der Funktion.

3.2.1 Explizite Form $f(x)$ [Programm Drehefx.MOD]

Im Programm Drehefx ist der Definitionsbereich der jeweiligen Funktion aus Vereinfachungsgründen bereits vorgegeben (Unterprogramm InitFunktionen). Der Wertebereich der Funktion wird im Unterprogramm Wertebereich bestimmt. Dabei wird folgendermaßen vorgegangen: eine Variable x durchläuft den Definitionsbereich mit einer vorgegebenen Schrittweite. Das Maximum im gegebenen Intervall wird bestimmt, indem geprüft wird, ob der Wert $y = f(x)$ größer ist als das bisherige Maximum. Trifft dies zu, so wird dieser Wert das neue Maximum. Analog erfolgt die Bestimmung des Minimums. Das Unterprogramm ZeichneGraph berechnet nun die Skalierungsfaktoren aus diesen Werten. Nun wird zu jedem x im gegebenen Intervall der Funktionswert $f(x)$ berechnet, danach werden diese Zahlen

mit Hilfe der Skalierungswerte in Bildschirmkoordinaten umgerechnet (Bild 3-1).

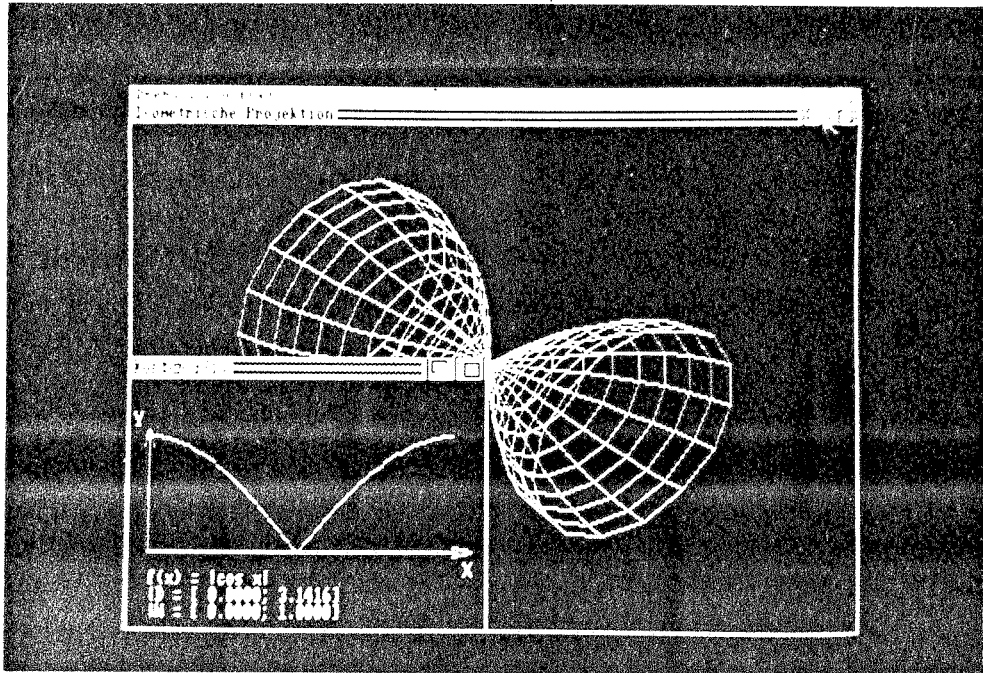


Bild 3-1: Der Graph der Funktion $f(x) = |\cos x|$ für $x \in [0;\pi]$ wird im Bildschirmfenster links-unten gezeichnet.

3.2.2 Parameterform [Programm DreheParam.MOD]

Prinzipiell weist dieses Programm bei der Bereichsgrenzenbestimmung keine Unterschiede auf. Nur wird hier das Unterprogramm Wertebereich, da das Intervall in X-Richtung nicht mehr vorgegeben ist, zweimal angewendet, einmal auf die Funktion $x(t)$, zum anderen auf die Funktion $y(t)$. Mit den errechneten Werten wird erneut eine Skalierung bestimmt, anschließend wird die Funktion gezeichnet. Als Laufvariable wird dabei nicht x genommen, da x diesmal auch abhängig von einer Veränderlichen ist, sondern t (Bild 3-2).

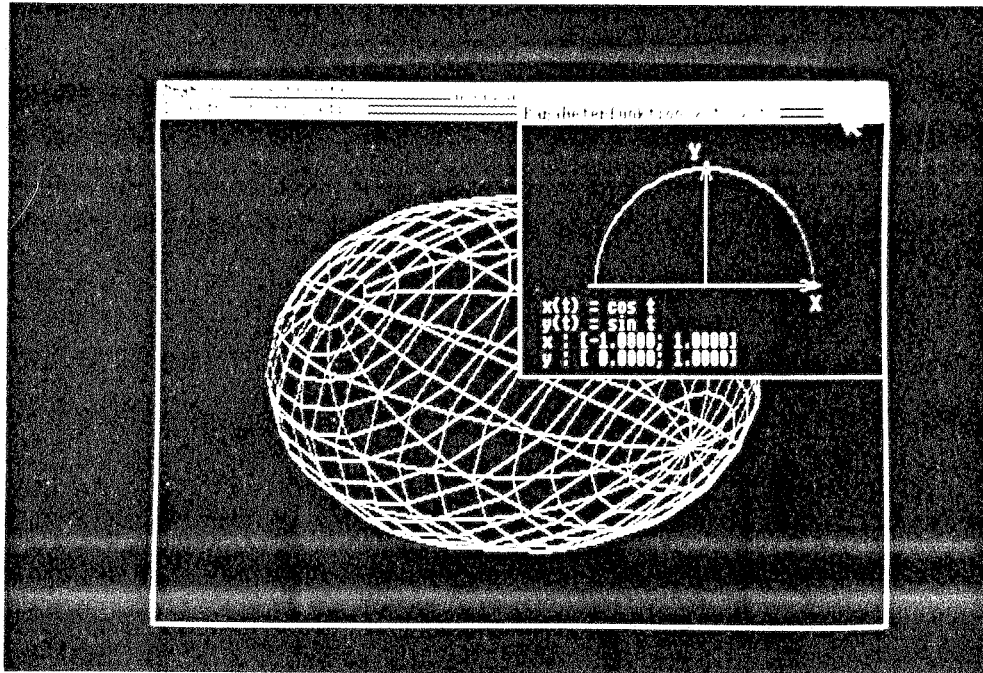


Bild 3-2: Der Graph eines Kreises in Parameterdarstellung.

3.3 Drehung um die X-Achse

Bei diesem Unterpunkt handelt es sich um das Kernstück der Computerdarstellung von Rotationsfunktionen. Noch einmal zur Erinnerung: eine Rotationsfunktion ist eine Funktion, deren Graph um die X-Achse rotiert, so daß ein dreidimensionaler Körper entsteht. Diese Operation läßt sich zwar sehr leicht gedanklich ausführen, doch wie wird sie mathematisch beschrieben, daß ein Computer etwas damit anfangen kann?

Man zerlegt einen Rotationskörper, ähnlich wie bei der Volumenberechnung in Punkt 2.1.1, in Schichten. Eine solche Schicht hat, wenn man sie von oben betrachtet und der Mittelpunkt auf der X-Achse liegt, eine Kreisform. Der Radius des Kreises wird durch den Funktionswert $f(x)$ an der Stelle x bestimmt.

Die Parameterdarstellung der Kreisfunktion lautet

$$\begin{aligned}x(t) &= r \cdot \cos t \\y(t) &= r \cdot \sin t\end{aligned}$$

mit r = Radius des Kreises.

Man erkennt, daß sich eine Schicht des Rotationskörpers durch diese Funktion beschreiben läßt, denn die Z-Achse entspricht der X-Achse der zweidimensionalen Darstellung, und die Y-Achse bleibt die Y-Achse. Somit läßt sich ein Punkt $P(X;Y;Z)$, der auf der Oberfläche des Rotationskörpers liegt, berechnen mit

$$X_P = x \tag{8a}$$

$$Y_P = f(x) \cdot \sin \phi \tag{8b}$$

$$Z_P = f(x) \cdot \cos \phi \tag{8c}$$

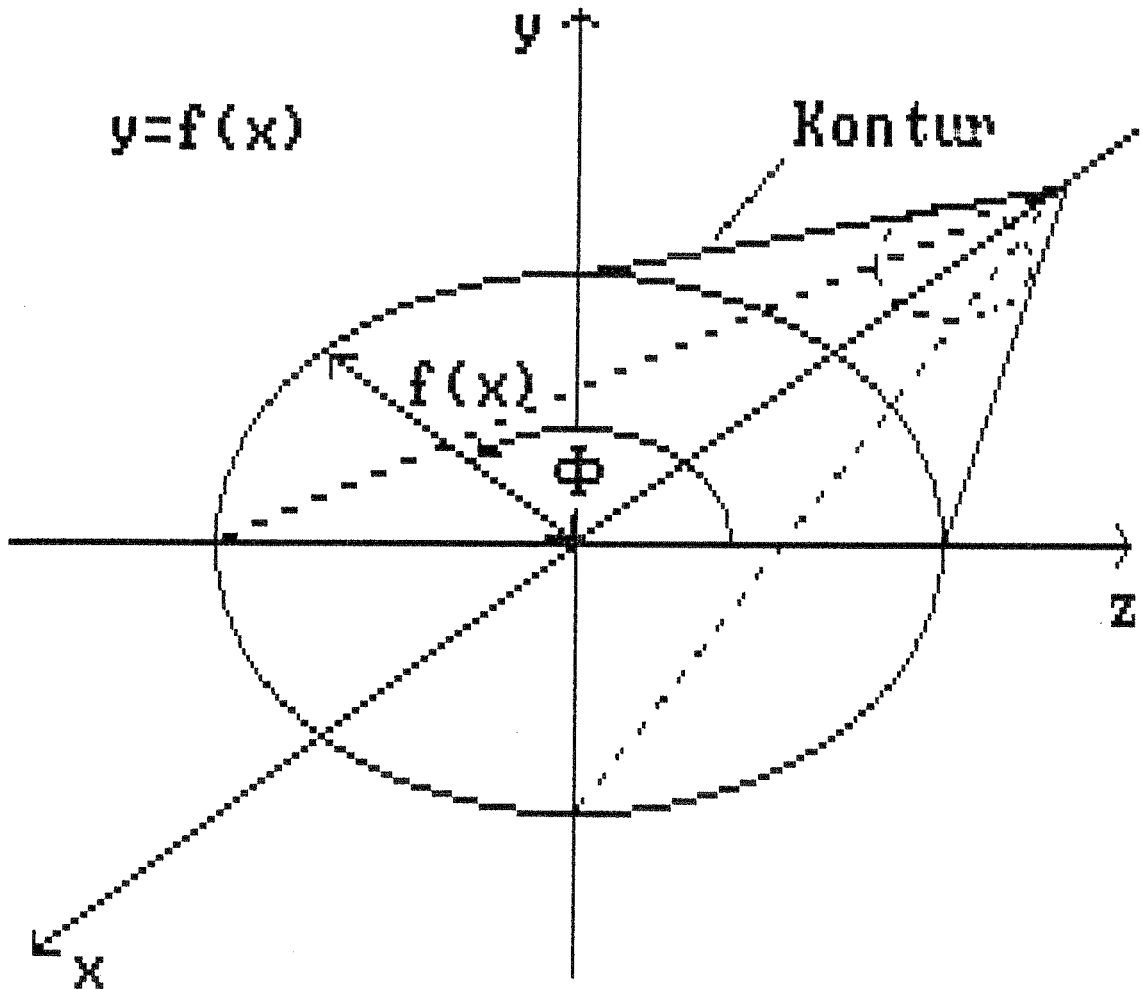


Bild 3-3: Ein kreisförmiger Schnitt durch einen Rotationskörper.

Für einen Rotationskörper, der durch Drehung einer Funktion in Parameterform entstand, müssen die Formeln etwas abgeändert werden, da nun auch x eine veränderlich Größe ist, es ergibt sich

$$X_P = x(t) \tag{9a}$$

$$Y_P = y(t) \cdot \sin \phi \tag{9b}$$

$$Z_P = y(t) \cdot \cos \phi \tag{9c}$$

Bei (8) und (9) ist ϕ immer der Winkel zwischen der Z-Achse und der Gerade zwischen dem Kurvenpunkt und dem Drehmittelpunkt, also der X-Achse (Bild 3-3).

3.3.1 Explizite Form [Programm Drehefx.MOD]

Die Drehung einer Funktion erledigt das Unterprogramm Drehefunktion. Erwähnt sei noch, daß die Variable phi ($=\phi$) nicht im Bereich $[0;2\pi]$, sondern im Bereich $[0^\circ;360^\circ]$ liegt. Dies ist eine rein willkürliche Festlegung, weil die Entstehung des Körpers durch eine volle Umdrehung mit 360° besser zum Ausdruck kommt. Zur Beschreibung des Unterprogramms sei auf das Struktogramm (Bild 3-4) verwiesen, das den Programmfluß darstellt.

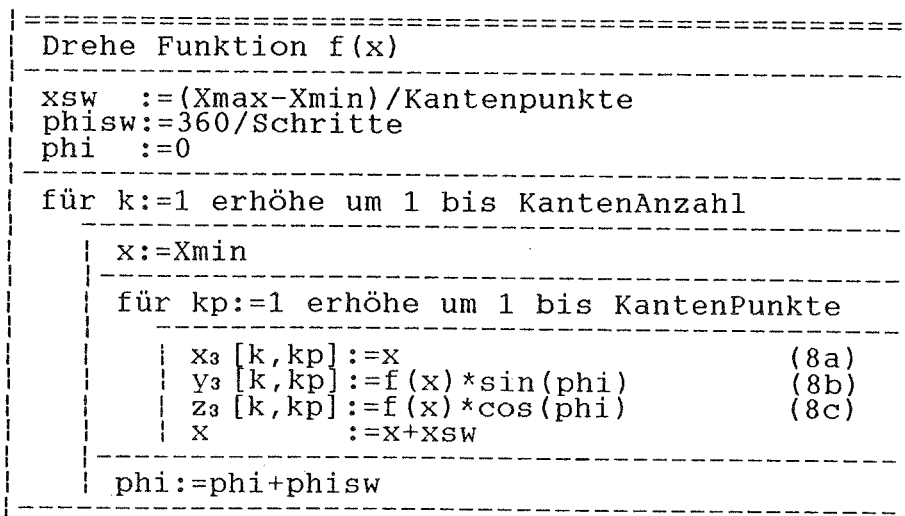


Bild 3-4: Nassy-Schneidermann Struktogramm zur Drehung einer Funktion $f(x)$ um die X-Achse.

3.3.2 Parameterform [Programm DreheParam]

Auch hier statt einer ausführlichen Beschreibung ein Struktogramm (Bild 3-5). Das Programm verwendet hier zur Berechnung die Formeln (9).

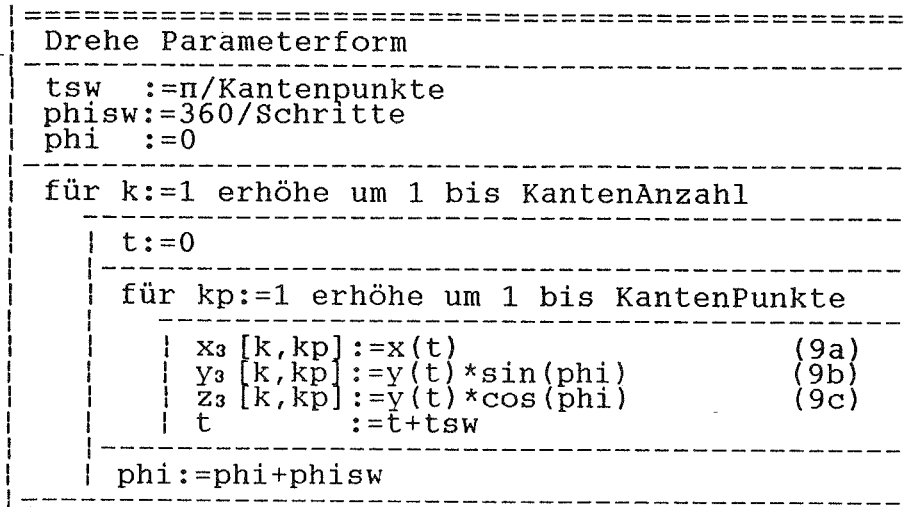


Bild 3-5: Nassy-Schneidermann Struktogramm zur Drehung einer Funktion in Parameterdarstellung $x(t);y(t)$ um die X-Achse.

3.4 Die Perspektivdarstellung am Bildschirm

Die Funktion ist nun gedreht, die Punkte der Oberfläche liegen nun alle in der Form $(X;Y;Z)$ vor. Doch bis jetzt ist noch keine Linie des Körpers am Bildschirm erschienen, die dreidimensionalen Koordinaten müssen erst in zweidimensionale transformiert werden, damit sie gezeichnet werden können.

Bei der Darstellung von Körpern auf einer Ebene gibt es verschiedene Projektionsmöglichkeiten, zum einen die Gruppe der Parallelprojektionen, und die Gruppe der Fluchpunktprojektionen. Für eine Darstellung eines Körpers, bei dem mathematische und nicht künstlerische Aspekte im Vordergrund stehen, kommt nur eine Parallelprojektion in Frage, weil in dieser Darstellung alle Parallelen erhalten bleiben.

Bei der Kavalierverspektive bleiben alle X- und Y-Werte längentreu erhalten, die Z-Achse wird in einem Winkel von 45° gegen die Bildebene gedreht und um die Hälfte gekürzt.

Eine dazu sehr ähnliche Darstellung ist die isometrische Projektion. Bei ihr haben die Achsen das Verhältnis $1 : 1 : 1$, die Achsen stehen in der Bildebene in einem Win-

kel von 120° aufeinander (Bild 3-6). Die Projektionsgleichungen für diese Darstellung lauten

$$x' = x \cdot \cos 30^\circ - z \cdot \cos 30^\circ \quad (10a)$$

$$y' = y - x \cdot \cos 60^\circ - z \cdot \cos 60^\circ \quad (10b)$$

(Vgl. [13], S. 30f und S. 35f)

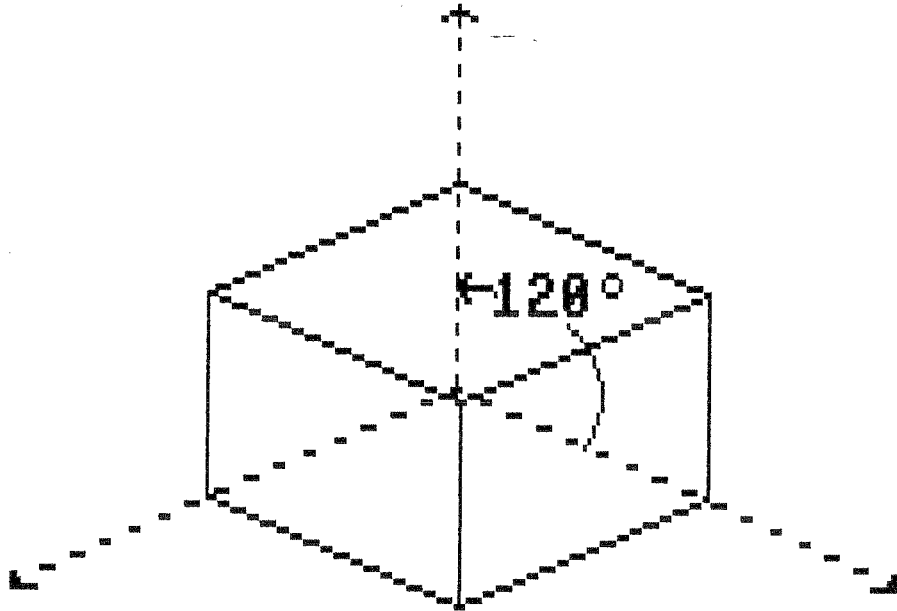


Bild 3-6: Eine isometrische Projektion eines Quaders.

Diese Umformung erledigt das Unterprogramm Mache3Dzu2D in den beiden Programmen. Danach werden die Koordinaten skaliert, damit der Körper am Bildschirm immer mittig gezeichnet wird. Der Körper wird nun gezeichnet, indem die einzelnen Punkte, die jetzt in der Form (X;Y) vorliegen, in bestimmter Reihenfolge miteinander verbunden werden. Dies erledigt das Unterprogramm ZeichneRotKoerper bei beiden Programmen. Als erstes werden die Kanten gezeichnet, also die Funktionskurve zu jedem Drehwinkel. Danach werden die Kantenpunkte um den Körper herum miteinander verbunden, es werden sozusagen die Umfänge der Schnittkreise gezeichnet (Bild 3-7).

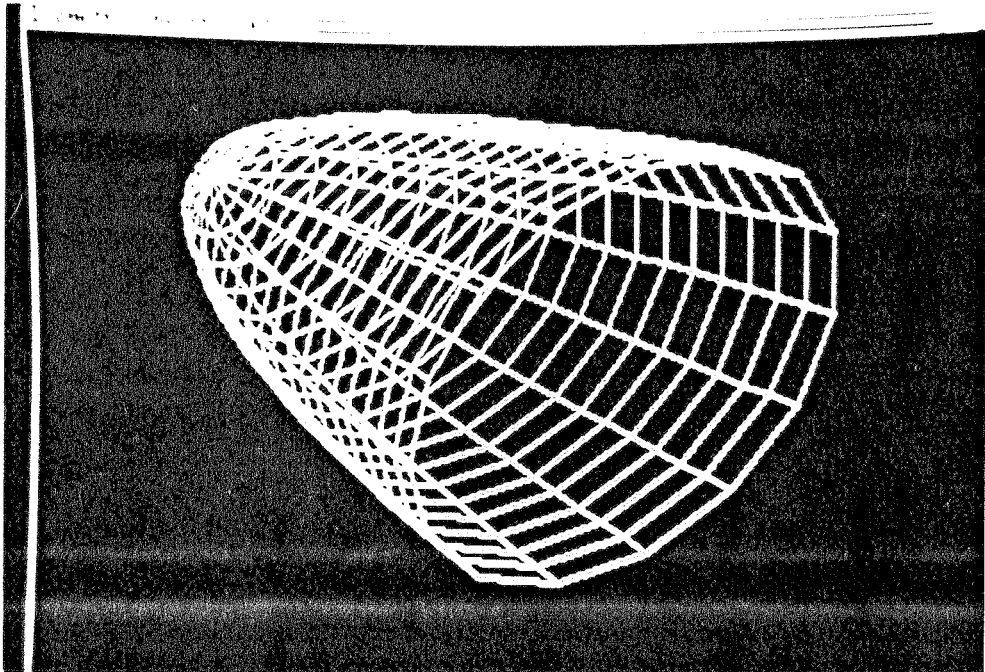


Bild 3-7: Das Drahtgittermodell des Rotationskörpers von \sqrt{x} verdeutlicht den Aufbau aus Längs- und Querlinien.

Betrachtet man die Drahtgitterdarstellung am Bildschirm, so fallen noch einige kleinere Mängel auf: die Linien, die im Bildhintergrund liegen, sind sichtbar, obwohl dies nicht der Realität entspricht. Auch ist das Bild statisch, es wäre denkbar, daß der Körper auf Tastendruck um die verschiedenen Achsen gedreht wird oder dergleichen. Zudem könnte das Bild noch an Plastizität gewinnen, indem es als Vollflächenmodell dargestellt wird. Eine Methode dazu wäre, die ganze Oberfläche in Dreiecke zu zerlegen. Die Dreiecke werden nun nach ihrem Abstand zum Betrachter sortiert und von hinten nach vorne gezeichnet. Die Farbe der Dreiecksfläche wird durch die Neigung der Dreiecksfläche gegen eine Lichtquelle im Raum bestimmt. Ist die Fläche der Lichtquelle zugewandt, so ist sie heller, ist sie abgewandt, so ist sie dunkler. Das Ergebnis ist ein Vollflächenmodell, daß sogar Schatten aufweist (dazu: [14], [15]).

Doch würde diese sehr genaue, aber auch aufwendige Darstellung den Rahmen dieser Facharbeit bei weitem sprengen, sie ist nur erwähnt, um aufzuzeigen, welche programmtechnischen Möglichkeiten und Erweiterungen existieren.

3.5 Die Bedienung der Programme

Nach der Beschreibung der Struktur der Programme folgt nun noch eine Anleitung zur Programmbedienung. Die Bedienung unterscheidet sich bei Drehefx und DreheParam nicht, nur die Menütexe sind unterschiedlich (Bild 3-8).

Am Anfang erscheint ein Menü, in dem alle Funktionen, die gezeichnet werden können, dargestellt werden. Mit dem Mauszeiger wird nun eine bestimmte Funktion angeklickt, die sofort am Bildschirm in einem kleinerem Fenster mit Angabe der Definitions- und Wertebereiche gezeichnet wird. Sekunden danach erscheint auch das Drahtgittermodell des Rotationskörpers in einem weiteren Fenster. Das Menü erscheint nach circa 5 Sekunden erneut, es kann nun eine neue Funktion gewählt oder das Programm verlassen werden. Da sich einzelne Fenster am Bildschirm überschneiden, kann zu jedem Zeitpunkt ein Fenster durch Anklicken des "WindowDepth"-Schalters (das Symbol in der Ecke rechts-oben) in den Vordergrund gebracht oder durch Anklicken des "WindowDrag"-Schalters (die Titellezeile des Fensters) verschoben werden.

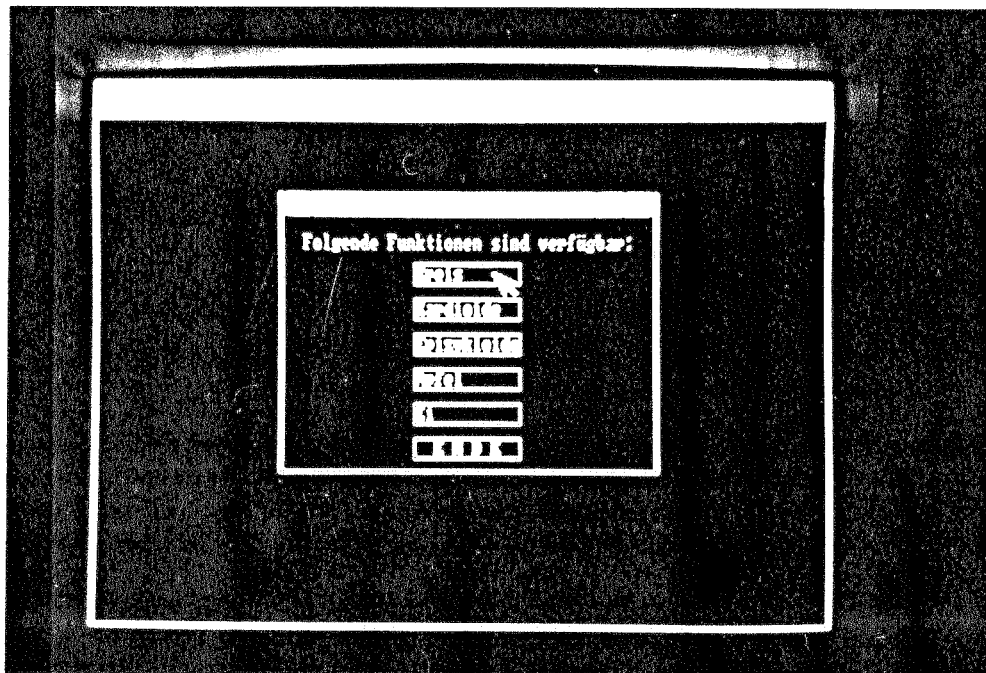


Bild 3-8: Das Auswahlmenü und der Mauszeiger.

3.6 Die numerische Integration nach Gauß

Bei der Arbeit in Abschnitt 4.3 tauchte das Problem auf, daß sich bestimmte algebraische und transzendente Funktionen nicht mehr analytisch integrieren lassen. Eine Näherungslösung, zum Beispiel durch eine Reihenentwicklung oder durch Polynome höherer Ordnung, erwies sich in der praktischen Anwendung als zu ungenau.

Daher entstand als Kompromiß das Ausweichen auf die numerische Integration, was aber auch in einer Mathematikfacharbeit vertretbar ist, zum einen, weil diese Methode nur einmal benötigt wurde, zum anderen, weil das dahintersteckende mathematische Verfahren doch relativ komplex ist.

Das Programm Integration.MOD (im Anhang) arbeitet nach dem Gauß-Algorithmus, und ist eine 1:1 Umsetzung eines Basic-Programmes für den Schneider CPC (aus: [16], S. 64). Zur genaueren Funktionsweise sei daher auf die entsprechende Programmsammlung verwiesen. Zur Arbeitsweise sei nur soviel gesagt, daß beim Gauß-Algorithmus die Fläche unter der Funktion in Teilstücke zerlegt wird, die einzeln berechnet und dann addiert werden (vgl.: [17], S. 86ff).

4 Praktische Anwendungen

Nach dieser theoretischen Vorarbeit zu den Rotationsfunktionen sollen nun einige praktischen Anwendungen zur Volumenberechnung erläutert werden. Anhand dreier grundlegend unterschiedlicher biologischer Körper, einer Birne, einem Ei und einem Apfel, soll die praktische Seite der Volumenberechnung von Rotationskörpern dargestellt werden. Hierbei wird die mathematische Fiktion vorausgesetzt, daß bei diesen Körpern näherungsweise die Rotationssymmetrie vorhanden ist - was im allgemeinen durch die Einführung eines Querschnittes durch die Körper leicht bewiesen werden kann.

»Drei Körper - drei Methoden« könnte die Überschrift zu diesem Kapitel auch heißen, denn die drei Körper werden auf verschiedene Art und Weise durch Funktionen angenähert. Zum einen wird die Kontur durch abschnittsweise Definition von Funktionen, zum anderen durch Umwandlung von Funktionen in andere Darstellungsweisen, und schließlich durch Verzerrung bestehender Funktionen mathematisch beschrieben.

4.1 Die Birne

Birne [lat.] *Pirus*, Gattung der Rosengewächse, rund 25 Arten, die Kulturbirne wurde unter anderem aus der dornigen Wildbirne (Misch- und Auwälder) gezüchtet (aus [18], S. 430).

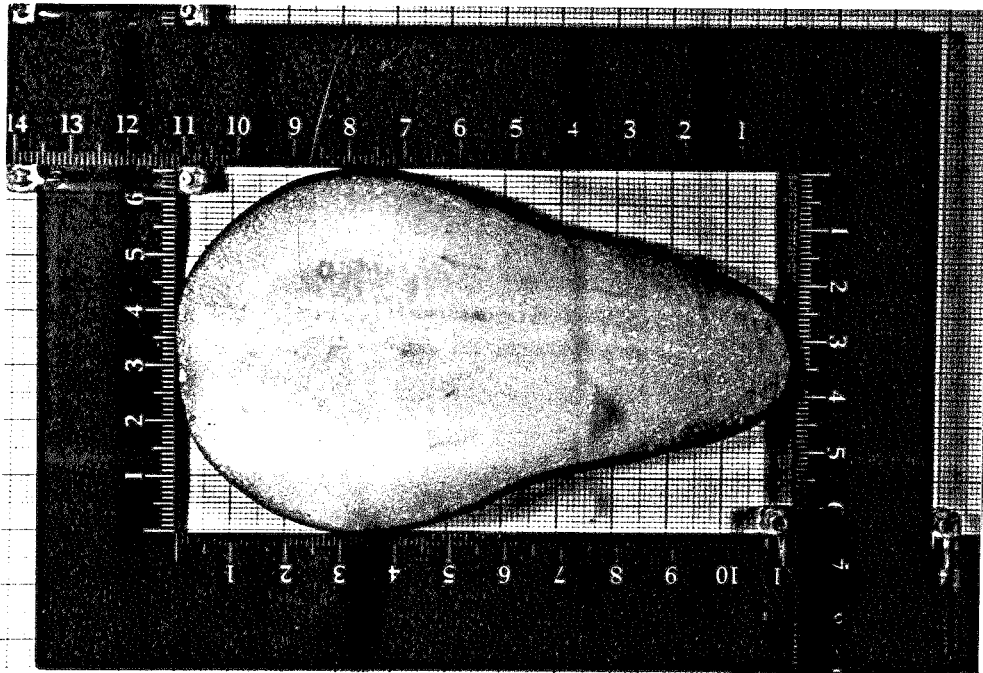


Bild 4-1: Ein Schnitt durch eine handelsübliche Birne, Sorte "Williams Christ"

Betrachtet man den Längsschnitt durch eine Birne (Bild 4-1), so läßt sich die Konturlinie in drei Abschnitte einteilen (Bild 4-2). Der erste und der dritte Abschnitt können durch Teile einer Ellipse, die durch einen glatten Kurvenzug verbunden sind, beschrieben werden.

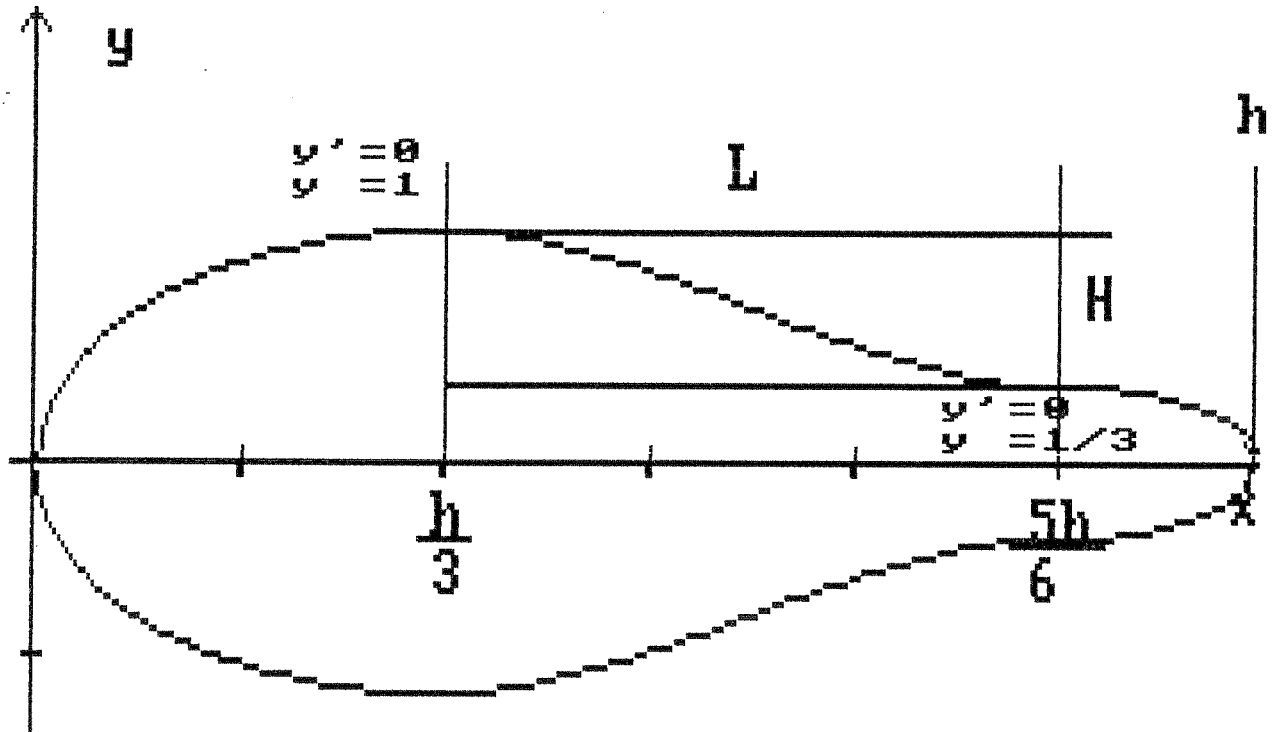


Bild 4-2: Das Profil einer Birne als Funktion

Der Graph einer Ellipse hat die implizite Darstellung

$$\frac{x^2}{R^2} + \frac{y^2}{r^2} = 1 \quad \text{mit} \quad (11)$$

R = Radius in X-Richtung und
r = Radius in Y-Richtung.

Aufgelöst nach y ergibt sich

$$f(x) = \pm \frac{r}{R} \sqrt{R^2 - x^2} \quad (12)$$

Für den ersten Abschnitt mit $x \in [0; h/3]$ ist der Radius in Y-Richtung $r = 1$. Dies ist eine Festlegung, durch die die Streckung des Graphen in Y-Richtung vereinfacht wird. Der Radius in X-Richtung wird $R = h/3$. Zudem liegt der Mittelpunkt der Ellipse auf der Y-Achse, die Funktion muß noch um

R nach rechts verschoben werden. Wird dies alles in (12) eingesetzt, so gilt für den positiven Kurvenzug

$$\begin{aligned} f(x) &= \frac{3}{h} \sqrt{(h/3)^2 - (x-h/3)^2} \\ &= \frac{1}{h} \sqrt{-9x^2 + 6hx} \end{aligned} \quad (13)$$

Der zweite Abschnitt wird durch ein Polynom angenähert. Der Graph eines Polynoms zwischen den Punkten (0;H) und (L;0) (Bild 4-2) muß folgende Nebenbedingungen erfüllen damit die Kurve an den Nahtstellen einen glatten Übergang aufweist

$$\begin{aligned} \text{(I)} \quad f(0) &= H; & \text{(II)} \quad f'(0) &= 0 \\ \text{(III)} \quad f(L) &= 0; & \text{(IV)} \quad f'(L) &= 0 \end{aligned} \quad .$$

Das sind insgesamt 4 Bedingungen, die den Polynom-Ansatz

$$y = a x^3 + b x^2 + c x + d \quad (14)$$

ergeben. Es wird nun noch die erste Ableitung der rationalen Funktion gebildet,

$$y' = 3a x^2 + 2b x + c$$

danach wird das lineare Gleichungssystem erstellt

$$\begin{aligned} \text{(I)} \quad a \cdot 0^3 + b \cdot 0^2 + c \cdot 0 + d &= H \\ \text{(II)} \quad 3a \cdot 0^2 + 2b \cdot 0 + c &= 0 \\ \text{(III)} \quad a L^3 + b L^2 + c L + d &= 0 \\ \text{(IV)} \quad 3a L^2 + 2b L + c &= 0 \end{aligned} \quad .$$

Die Lösungen

$$a = 2 \frac{H}{L^3}, \quad b = -3 \frac{H}{L^2}, \quad c = 0, \quad d = H$$

werden in (14) eingesetzt und ergeben

$$y = 2 \frac{H}{L^3} x^3 - 3 \frac{H}{L^2} x^2 + H \quad (15)$$

(vgl. hierzu [19], S. 68f)

Das Teilstück muß nun noch an die geforderten Maße angepasst werden. Es ist ersichtlich (Bild 4-2), daß

$$L = \frac{1}{2} h \quad , \quad H = \frac{2}{3} \quad \text{wird,}$$

zudem muß der Ast um $1/3$ angehoben und um $h/3$ nach rechts verschoben werden. Dies in (15) eingesetzt ergibt für das Polynomzwischenstück

$$y = \frac{32}{3h^3} x^3 - \frac{56}{3h^2} x^2 + \frac{80}{9h} x - \frac{23}{81} \quad . \quad (16)$$

Der dritte Abschnitt für $x \in [5h/6; h]$ wird wieder durch eine Viertelellipse angenähert. Hier gilt, daß der Radius in X-Richtung $R = h/6$ und der Radius in Y-Richtung $r = 1/3$ ist. Die normale Ellipse in Nulllage muß noch um $5h/6$ nach rechts verschoben werden. Daraus folgt mit (12) für den positiven Zweig

$$\begin{aligned} y &= \frac{6}{3h} \sqrt{h^2 - 36 (x - 5h/6)^2} \\ &= \frac{2}{3h} \sqrt{-9 x^2 + 15 xh - 6 h^2} \quad . \quad (17) \end{aligned}$$

Die Funktionsabschnitte wurden so normiert, daß der größte Wert der Funktion 1 ist. Eine Streckung in Y-Richtung wird durch eine Multiplikation mit dem halben Durchmesser erreicht. Werden die Ergebnisse (13), (16) und (17) zusammengefaßt, so ist das Ergebnis eine abschnittsweise Definition einer Funktion $f(x)$, die der Oberflächenkontur einer Birne angenähert ist

$$f(x) = \begin{cases} \frac{d}{2h} \sqrt{-9 x^2 + 6 hx} & ; 0 \leq x \leq \frac{h}{3} \\ \frac{d}{2} \left(\frac{32}{3h^3} x^3 - \frac{56}{3h^2} x^2 + \frac{80}{9h} x - \frac{23}{81} \right) & ; \frac{h}{3} \leq x \leq \frac{5}{6}h \\ \frac{d}{3h} \sqrt{-9 x^2 + 15 xh - 6 h^2} & ; \frac{5}{6}h \leq x \leq h \quad . \end{cases} \quad (18)$$

Das Volumen des Rotationskörpers wird nun berechnet, indem die Integralformel (2) auf die einzelnen Abschnitte angewandt wird.

$$\begin{aligned} V &= \pi \int_0^h f(x)^2 dx \\ &= \pi \left(\int_0^{h/3} f(x)^2 dx + \int_{h/3}^{5h/6} f(x)^2 dx + \int_{5h/6}^h f(x)^2 dx \right) \\ &= \pi (J_1 + J_2 + J_3) \quad (J_n = \text{Teilintegral}) \end{aligned}$$

$$\begin{aligned} J_1 &= \int_0^{h/3} \left(\frac{d}{2h} \sqrt{-9x^2 + 6hx} \right)^2 dx \\ &= \frac{3d^2}{4h^2} \left[-x^3 + hx^2 \right]_0^{h/3} \\ &= \frac{1}{18} d^2 h \end{aligned}$$

$$\begin{aligned} J_2 &= \int_{h/3}^{5h/6} \left(\frac{d}{2} \left(\frac{32}{3h^2} x^3 - \frac{56}{3h^2} x^2 + \frac{80}{9h} x - \frac{23}{81} \right) \right)^2 dx \\ &= \frac{d^2}{4} \left[\frac{1024}{63h^6} x^7 - \frac{3584}{54h^5} x^6 + \frac{14528}{135h^4} x^5 - \frac{82112}{972h^3} x^4 + \frac{21776}{729h^2} x^3 - \frac{3680}{1458h} x^2 + \frac{529}{6561} x \right]_{h/3}^{5h/6} \\ &\approx \frac{hd^2}{4} \cdot \frac{1}{4} = \frac{hd^2}{16} \end{aligned}$$

$$\begin{aligned} J_3 &= \int_{5h/6}^h \left(\frac{d}{3h} \sqrt{-9x^2 + 15xh - 6h^2} \right)^2 dx \\ &= \frac{d^2}{9h^2} \left[-3x^3 + 7,5hx^2 - 6h^2x \right]_{5h/6}^h \\ &= \frac{1}{324} d^2 h \end{aligned}$$

$$V = \left(\frac{1}{18} + \frac{1}{16} + \frac{1}{324} \right) \pi d^2 h$$

$$V_{\text{Birne}} = \frac{157}{1296} \pi d^2 h \quad . \quad (19)$$

Das Volumen einer Birne kann demnach unter Beachtung der Voraussetzungen durch eine einfache Formel berechnet werden. »Doch grau, lieber Freund, ist jede Theorie« sagte schon Goethe, diese Volumenformel muß natürlich noch auf Übereinstimmung mit der Realität geprüft werden. Am einfachsten werden dazu Durchmesser und Höhe von einigen Test-

birnen mit einer Art Schieblehre gemessen. Das Volumen wird dann über die Wasserverdrängung der Birnen in einem Gefäß nach der klassischen Methode von Archimedes bestimmt. Das Ergebnis der Messungen und die Abweichungen sind in einer Tabelle (Bild 4-3) dargestellt und geben einen gute Eindruck von der Präzision der Formel.

d gemessen	(in cm)	6,5	8,4	7,3	7,6	
h gemessen	(in cm)	11,1	13,6	12,1	11,8	
V _{Exp}	(in cm ³)	187	351	250	263	
V _{Birne}	(in cm ³)	178,5	365,2	245,4	259,4	
Abweichung*	(in %)	4,5	4,0	1,2	1,4	

* mit [5], S. 448

Bild 4-3: Die Tabelle der Meßwerte von Birnen

4.2 Das Ei

Lexikalisch ist das Ei (lat. Ovus) die biologische Fortpflanzungszelle von Mensch, Tier und Pflanze. Das tierische Ei besteht aus Eikern, Ei-plasma (mit Dotter) und fester Eihülle. Die Formenmannigfaltigkeit der Vogeleier hat zu einem bes. Wissenschaftszweig, der *Oologie* (Eierkunde), die vor allem für die Systematik von Bedeutung ist, geführt (aus [20], S. 913).

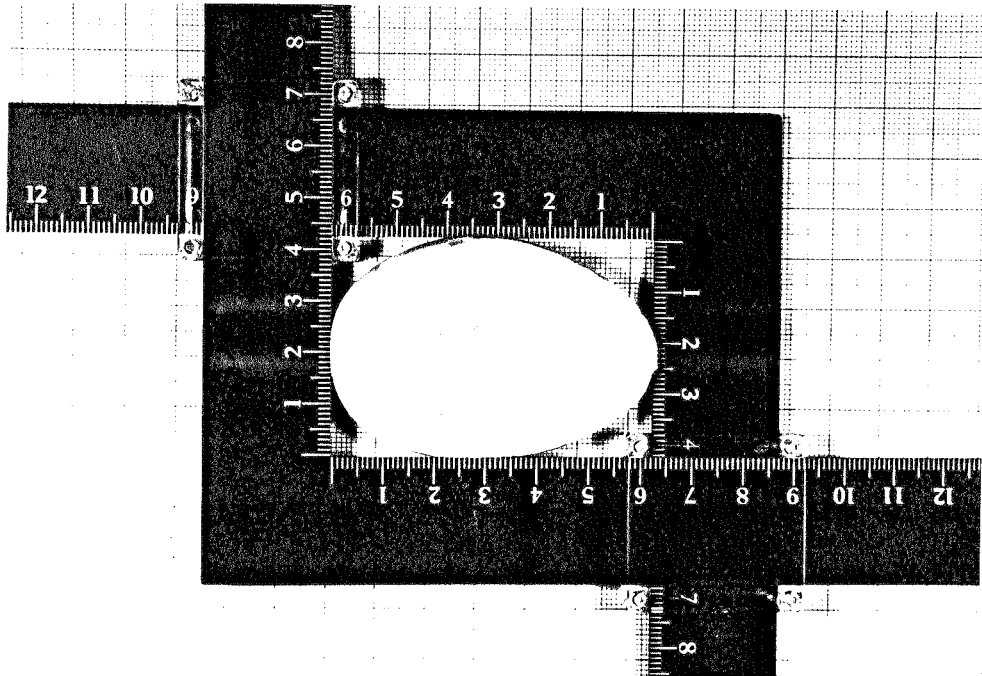


Bild 4-4: Ein Schnitt durch ein Hühnerei

In der Geometrie und der Mechanik gibt es verschiedene Näherungskonstruktionen mittels Kreisbögen für Eiprofile. Ein Eiprofil läßt sich aber auch durch ein Parabelsegment an einem Kreis darstellen ([21], Abb. 70). Dies ist wieder eine abschnittsweise Definition. Die Mathematik bietet zu diesen Methoden, wie sooft, Alternativen an. Eine solche Alternative wird hier gewählt, das Ei soll nämlich durch Anpassung einer bestehenden algebraischen, geschlossenen Funktion beschrieben werden.

Untersucht man die Relation $f(x) = \pm\sqrt{x^3} - x$ (Bild 4-5), so stellt man fest, daß die Relation im Bereich der Nullstellen $N_1(-1;0)$ und $N_2(0;0)$ ungefähr ein Eiprofil (Bild 4-4) aufweist. Leider hat diese Relation auch die unangenehme Eigenschaft, daß sie in den Bereichen $]-\infty;-1[$ und $]0;1[$ nicht definiert ist. Außerdem paßt der Ast im Bereich $[1;+\infty[$ nicht recht zu der landläufigen Vorstellung von der Figur eines Eies. Will man das Intervall, in dem der Graph

der Relation einem Ei gleicht, vom Rest der Relation isoliert erhalten, so bietet sich die schon in 2.2.1 angesprochene Transformation des eiförmigen Teiles der Relation in Parameterdarstellung an.

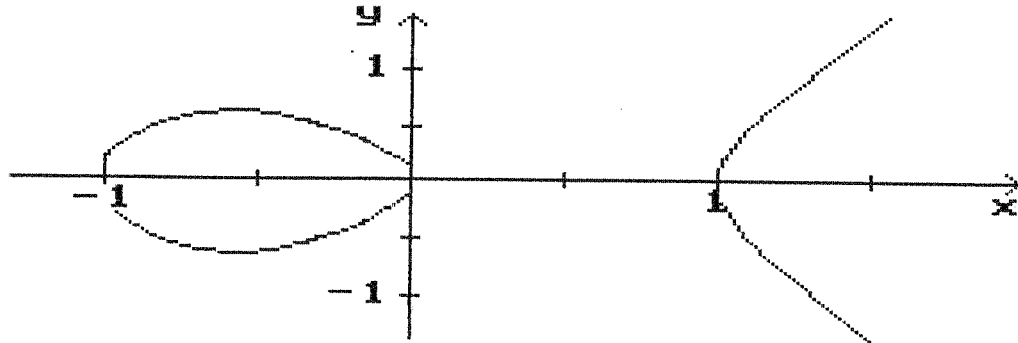


Bild 4-5: Die Funktion $f(x) = \pm\sqrt{x^3-x}$

Zur Umwandlung in die Parameterform wird zuerst x in Abhängigkeit von t gesetzt. Für ein Eiprofil, das für $t \in [0; \pi]$ genau den halben Umfang umläuft, darf x nur im Intervall $[-1; 0]$ liegen (der Parameter t darf übrigens keinesfalls als Winkel zwischen einer Koordinatenachse und dem Funktionsstrahl interpretiert werden). Eine Funktion, die also diese Bedingung erfüllt, ist eine in X -Richtung gestreckte und gekippte Sinusfunktion. Für x gilt also

$$x(t) = -\sin 0,5t \quad .$$

Wird dies nun in die Funktion $f(x)$ eingesetzt, so erhält man y in Abhängigkeit von t mit

$$y(t) = \sqrt{-\sin^3 0,5t + \sin 0,5t} \quad .$$

Die Richtigkeit dieser Überlegung beweist der Graph der Parameterform der Funktion (Bild 4-6).

Es ergibt sich vorteilhafterweise eine elegante Darstellung eines Eiprofils, das ähnlich Kreis und Ellipse in einfacher Parameterdarstellung darstellbar und berechenbar ist.

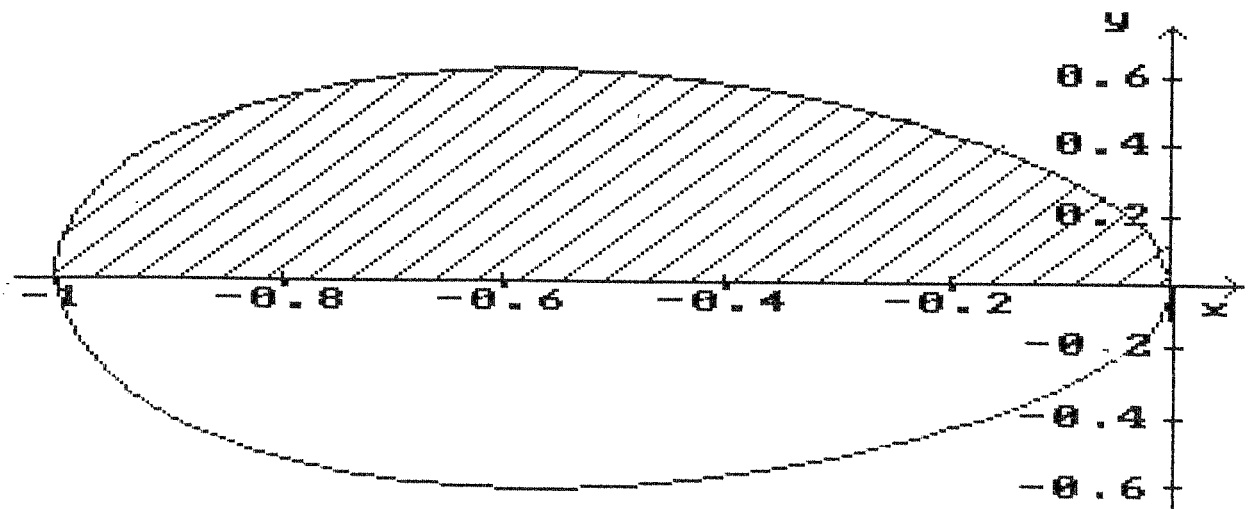


Bild 4-6: Der Graph der Parameterfunktionen $x(t) = -\sin 0,5t$ und $y(t) = \sqrt{-\sin^3 0,5t + \sin 0,5t}$

Eine Streckung in X-Richtung ergibt sich durch Multiplikation mit einem Parameter r_x , eine Streckung in Y-Richtung durch Multiplikation mit einem Parameter r_y . Der Graph der Funktion läßt sich also wesentlich einfacher als bei einer expliziten Darstellung in Höhe und Breite verändern. Werden nun die beiden Parameter auf 1 normiert, so entsteht quasi ein Einheitsei mit einer Konturformel, in die direkt Längenangaben in cm einfließen können. Es ergibt sich für

$$r_x = h \quad , \quad (20)$$

da x nur im Bereich $[-1;0]$ liegt und daher $X_{\max} - X_{\min} = 1$ ist.

Wird das Maximum von $y(t)$ mit Hilfe von $y'(t)$ bestimmt, so gilt

$$y'(t) = 0$$

$$\text{für } t = 2 \cdot \arcsin \sqrt{\frac{1}{3}}$$

$$\approx 1,2310$$

der Funktionswert an dieser Stelle ist

$$y(2 \arcsin \sqrt{\frac{1}{3}}) = \frac{\sqrt{2}}{4\sqrt{27}} \approx 0,6204 \quad .$$

Der Korrekturfaktor für die Verzerrung läßt sich mit einem einfachen Dreisatz berechnen, der gesamte Durchmesser in cm in Abhängigkeit vom Radius r_y ist

$$2 \frac{\sqrt{2}}{\sqrt[4]{27}} \text{ cm für } r_y = 1, \text{ somit ist}$$

$$\text{für } 1 \text{ cm } r_y = \frac{\sqrt[4]{27}}{2 \sqrt{2}} .$$

Es ergibt sich daher der Zusammenhang

$$r_y = \frac{\sqrt[4]{27}}{2 \sqrt{2}} d \quad . \quad (21)$$

Wird nun das Volumen des Rotationskörpers (Bild 4-7) dieser Funktion in Parameterform bestimmt, so ist das Volumen eingesetzt in (7)

$$V = \pi \int_0^{\pi} (r_y \sqrt{-\sin^3 0,5t + \sin 0,5t})^2 (r_x (-\sin 0,5t))' dt \quad .$$

Die Konstanten werden aus dem Integral herausgezogen, da dies die Schreibweise vereinfacht

$$k := r_y^2 \cdot r_x \cdot \pi$$

$$V = k \int_0^{\pi} (\sqrt{-\sin^3 0,5t + \sin 0,5t})^2 (-0,5 \cos 0,5t) dt$$

$$= -0,5 k \int_0^{\pi} (-\cos 0,5t \cdot \sin^3 0,5t + \cos 0,5t \cdot \sin 0,5t) dt$$

$$= -0,5 k \left[-\frac{1}{2} \sin^4 0,5t - \cos^2 0,5t \right]_0^{\pi}$$

$$= -0,5 k \left(-\frac{1}{2} \sin^4 0 - \cos^2 0 + \frac{1}{2} \sin^4 \frac{\pi}{2} + \cos^2 \frac{\pi}{2} \right)$$

$$= \frac{k}{4} \quad (\text{Stammfunktionen aus [3], S. 232})$$

nun wird k wieder eingesetzt und es entsteht die Volumenformel des Eies mit

$$V_{Ei} = \frac{\pi}{4} \cdot r_y^2 \cdot r_x \quad . \quad (22)$$

Werden nun die Normierungsformeln (20) und (21) angewandt, ergibt sich eine einfache endgültige Volumenformel

$$V_{E1} = \frac{3 \sqrt{3}}{32} \pi d^2 h \quad . \quad (23)$$

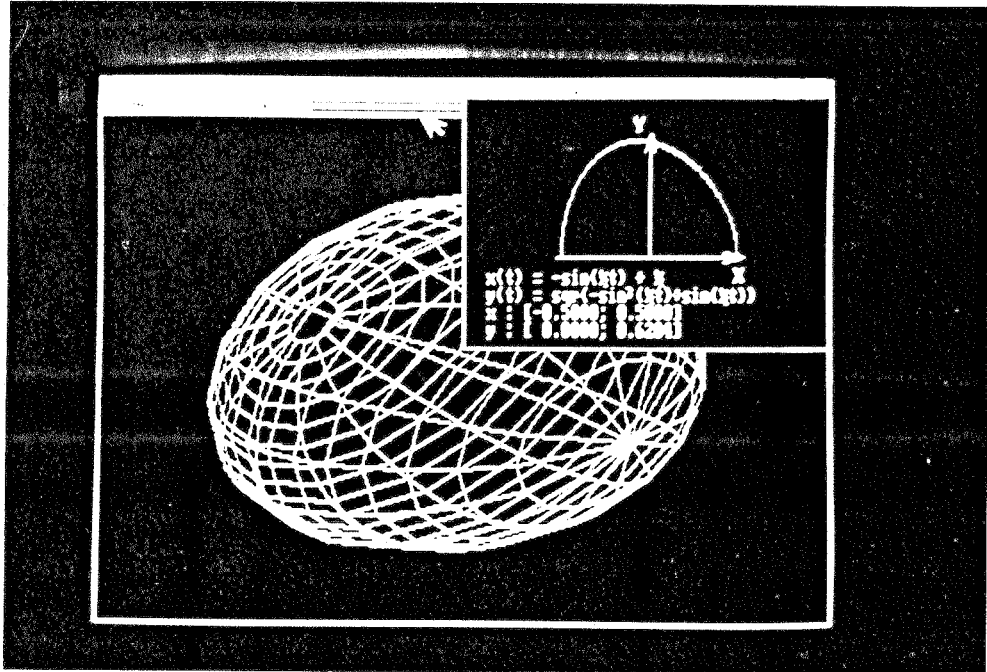


Bild 4-7: Das Ei als Rotationskörper

Auch bei dieser Volumenformel wurden in einer Meßreihe die Abweichungen der erwarteten Volumen vom gemessenen Volumen bestimmt. Zuerst wurden Höhe und Durchmesser der Eier, danach das Volumen mit Hilfe der Wasserverdrängung bestimmt. Die Abweichungen (Tabelle in Bild 4-8) sind mit einem mittleren prozentualen Fehler von 1,5% relativ klein, eine Annäherung des Eies durch ein Ellipsoid, die ebenfalls gerechnet wurde, ergab einen mittleren prozentualen Fehler von 3,1%. Bei der Annäherung durch ein Ellipsoid war zudem die Differenz zwischen dem größten und dem kleinsten Fehler mit 5,1% doppelt so groß wie bei der Annäherung mit Formel (23). Es wird daher deutlich, daß eine an sich naheliegende Annäherung durch ein Ellipsoid zu grob ist.

d gemessen	(in cm)	4,1	4,0	3,9	4,1	4,5	
h gemessen	(in cm)	5,4	5,5	5,6	5,5	5,9	
V _{Exp}	(in cm ³)	45,7	44,8	44,7	46,9	59,3	
V _{Ei} mit (23)	(in cm ³)	46,3	44,9	43,5	47,2	60,9	
Abweichung	(in %)	1,3	0,2	2,7	0,6	2,7	
V _{Ellipsoid}	(in cm ³)	47,5	46,1	44,6	48,4	62,6	
Abweichung*	(in %)	3,9	2,9	0,2	3,2	5,3	

* mit [5], S. 448

Bild 4-8: Die Tabelle der Meßwerte des Eies

Darüberhinaus könnte die hier vorgeschlagene Eiformel durch Einführung der Achsenlage des Durchmessers im Verhältnis zur Eihöhe zu einer weiteren Verbesserung führen. Auf diese Berechnung muß hier aus Platzgründen leider verzichtet werden.

4.3 Der Apfel

Apfel [lat.] *Malus*, Kernobst, Familie Rosengewächse; aus wilden Arten (z.B. *Malus silvestris*, einheimisch, bereits von Pfahlbauern verwendet) wurde unser Apfel in unzähligen Edelsorten gezüchtet (aus [18], S. 183).

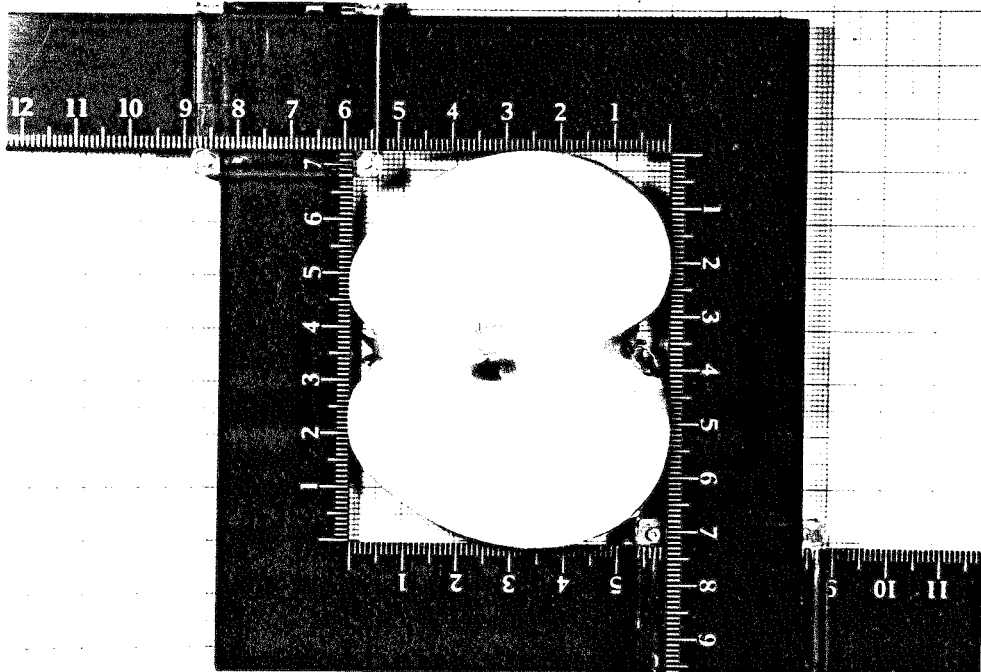


Bild 4-9: Ein der Länge nach aufgeschnittener Apfel

Die sich ergebende Bahnkurve eines Punktes P , wenn ein Kreis k mit dem Radius r auf der Außenseite eines festen Kreises K mit dem Radius R abrollt, heißt gemeine Epizykloide. Beträgt das Verhältnis von $R : r = 2$, so entsteht eine Kardioide ([10], S. 465f). Der Längsschnitt durch einen Apfel (Bild 4-9) ähnelt sehr stark dem Funktionsgraphen der Kardioide (Bild 2-3). Es ist daher zu vermuten, daß sich die Konturkurve des Apfels durch eine Kardioide annähern läßt.

Die Parameterform der Kardioide mit

$$\begin{aligned}x(t) &= 2\cos t - \cos 2t \\y(t) &= 2\sin t - \sin 2t\end{aligned}\tag{24}$$

muß zuerst wieder auf 1 normiert werden.

Dazu werden die Extremstellen der beiden Funktionen mit Hilfe der Ableitungen

$$x'(t) = 0$$

$$y'(t) = 0$$

bestimmt. Das Maximum der Funktion $x(t)$ liegt bei

$$t = \frac{\pi}{3} \quad ; \quad x\left(\frac{\pi}{3}\right) = 1,5 \quad ,$$

das Minimum bei

$$t = \pi \quad ; \quad x(\pi) = -3 \quad .$$

Die maximale Ausdehnung in X-Richtung beträgt daher 4,5. Auch hier wird über einen Dreisatz der Skalierungsfaktor berechnet.

$h = 4,5$ cm für $r_x = 1$, das bedeutet

$$\text{für } h = 1 \text{ cm ist } r_x = \frac{2}{9} \quad . \quad (25)$$

Die gleiche Rechnung wird auch mit der Funktion $y(t)$ durchgeführt. Das Maximum ist dabei

$$t = \frac{2}{3}\pi \quad ; \quad x\left(\frac{2}{3}\pi\right) = \frac{3}{2}\sqrt{3} \quad .$$

Daraus folgt, daß die maximale Ausdehnung in Y-Richtung $1,5\sqrt{3}$ beträgt. Der Skalierungsfaktor wird analog berechnet mit

$d = 2 \cdot 1,5\sqrt{3}$ cm für $r_y = 1$, also ist

$$\text{für } d = 1 \text{ cm ist } r_y = \frac{1}{3\sqrt{3}} \quad . \quad (26)$$

Das Volumen eines "Kardioids" mit den Radien $r_x = 1$ und $r_y = 1$ wurde schon in Punkt 2.2.2 als Beispiel berechnet. Aus dem Ergebnis folgt die allgemeine Volumenformel

$$V_{\text{Kardioid}} = \frac{64}{3} \pi r_y^2 r_x \quad , \quad (27)$$

durch Einsetzen der Zusammenhänge (25) und (26) entsteht dann die normierte Form der Volumenformel

$$V = \frac{64}{3} \cdot \pi \cdot \left(\frac{1}{3\sqrt{3}} d\right)^2 \cdot \left(\frac{2}{9} h\right)$$

zusammengefasst ist das Resultat die Volumenformel eines Apfels mit

$$V_{\text{Apfel}} = \frac{128}{729} \pi d^2 h \quad . \quad (28)$$

Der zugehörige Rotationskörper ist in Bild 4-10 abgebildet.

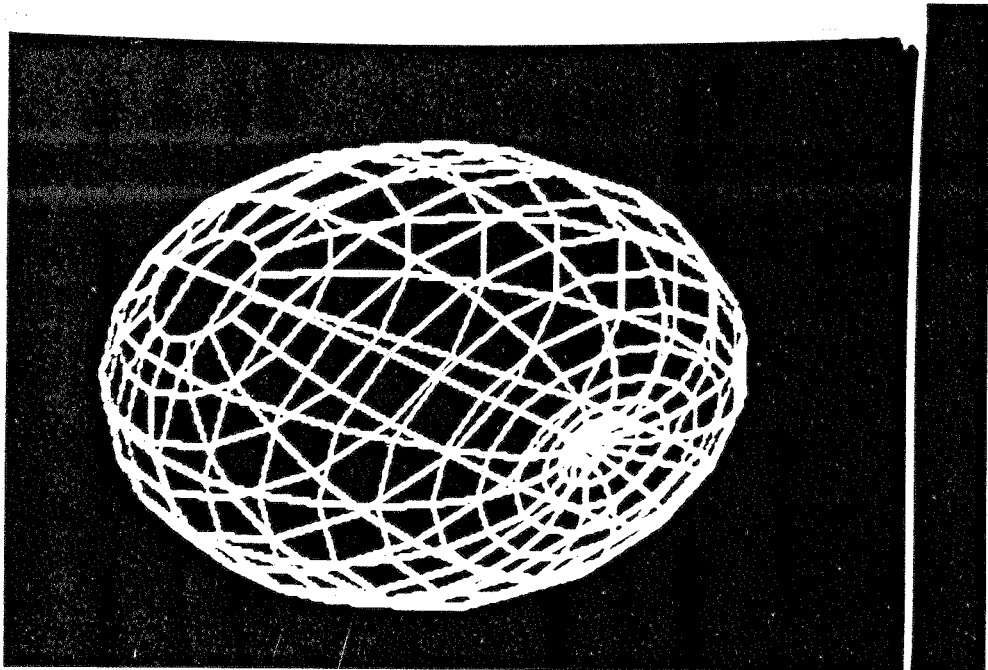


Bild 4-10: Die Rotationsfigur der Kardioide

Ruft man sich noch einmal den Apfelquerschnitt (Bild 4-9) in Erinnerung, sieht man, daß der Apfel auch an der Unterseite einen, wenn auch kleineren, Nabel, bedingt durch den Blütenansatz, aufweist. Die Näherung (24) ist darum offenbar noch verbesserungsfähig, wenn der Funktionsgraph einen zweiten Einschnitt aufweist. Aus dem Bildungsgesetz der Epizykloiden läßt sich die Parameterdarstellung einer solchen Kurve mit

$$\begin{aligned} x(t) &= 3\cos t - \cos 3t \\ y(t) &= 3\sin t - \sin 3t = 4 \sin^3 t \end{aligned} \quad (29)$$

herleiten (Graph und Rotationskörper in Bild 4-11).

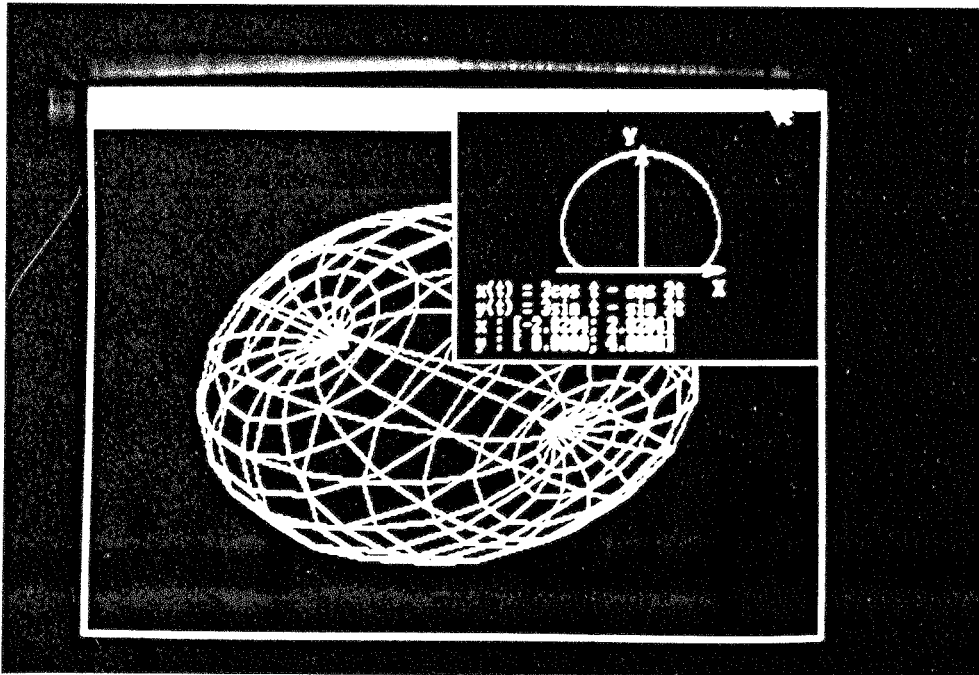


Bild 4-11: Eine Epizykloide mit $R : r = 3$

Der Graph dieser Funktion ist aber immer noch zu symmetrisch, denn bei einem Apfel ist der untere Einschnitt kleiner als der obere, außerdem liegt die dickste Stelle des Apfels auch nicht in der Mitte des Apfels. Die Funktion (29), die zu der Geraden $x = 0$ symmetrisch ist, kann durch Multiplikation mit einem Linearfaktor, der mit kleinerem x größer wird, wie gewünscht verzerrt werden. Da die Funktion gegen den Uhrzeigersinn läuft, wird ein kleineres x durch ein größeres t bedingt. Folglich wird die Verzerrung durch eine Multiplikation mit dem Laufparameter t erreicht.

Die Lösung

$$\begin{aligned} x(t) &= t (3\cos t - \cos 3t) \\ y(t) &= t (3\sin t - \sin 3t) \end{aligned} \tag{30}$$

ist die neue Parameterform (Graph in Bild 4-12). Die Darstellung sieht schon viel besser aus, doch hat die Formel (30) einen großen Haken: durch die Quadrierung von $y(t)$ und der Multiplikation mit der Ableitung von $x(t)$ entsteht ein Term mit Gliedern der Form $t \cdot \sin^m x \cdot \cos^n x$ mit $m, n > 5$, der sich analytisch nicht mehr integrieren läßt.

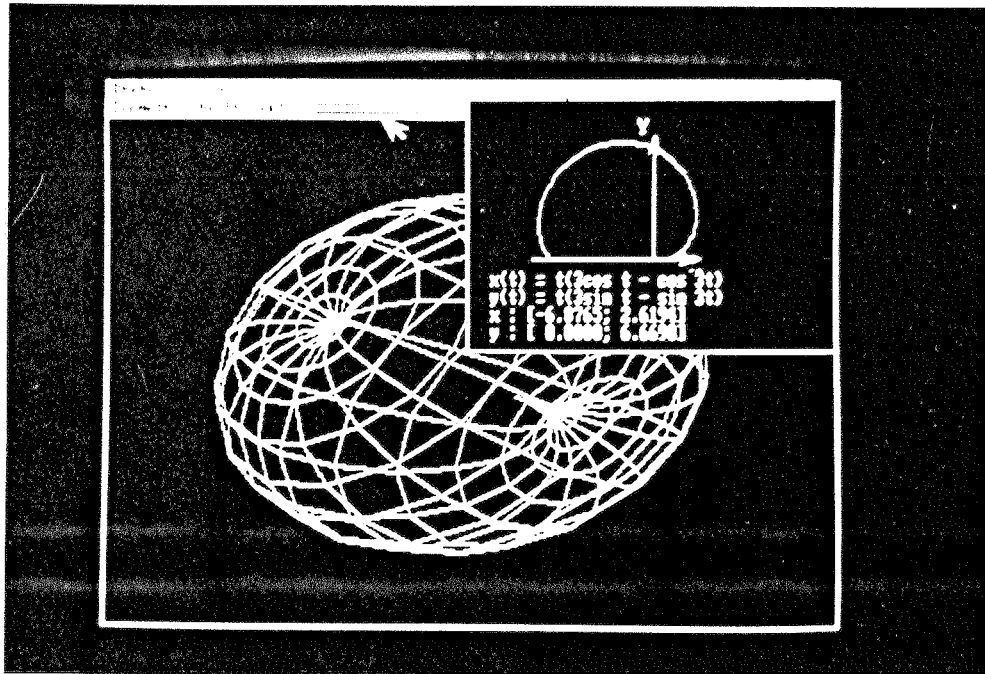


Bild 4-12: Der Funktionsgraph von (30), im Hintergrund die gedrehte Funktion

Wie man bereits sehen konnte, lassen sich bei der Volumenberechnung jedesmal die Radien in X- und Y-Richtung herausziehen, so daß nur noch ein Term mit einem Parameter integriert werden muß. Diese numerische Integration wird nun mit dem Programm Integration (siehe 3.6) durchgeführt. Das Ergebnis der Rechnung ist

$$\begin{aligned} V &= \pi \int_0^0 y^2(t) x'(t) dt \\ &= 317,7 \pi r_y^2 r_x \end{aligned} \quad (31)$$

Das Maximum von $x(t)$ liegt bei

$$t \approx 1,03 \quad ; \quad x(1,03) \approx 2,62 \quad ,$$

das Minimum liegt bei

$$t \approx 2,51 \quad ; \quad x(2,51) \approx -6,88 \quad .$$

Daraus ergibt sich der Zusammenhang zwischen r_x und h mit

$$r_x = \frac{1}{9,5} h = \frac{2}{19} h \quad . \quad (32)$$

Von $y(t)$ ist das Maximum

$$t \approx 1,76 \quad ; \quad y(1,76) \approx 6,67 \quad .$$

Für r_y und h gilt daher

$$r_y = \frac{1}{13,34} d \quad . \quad (33)$$

Werden die Formeln (31) bis (33) zusammengefaßt, ist das Resultat die normierte Volumenformel

$$\begin{aligned} V_{\text{Apfel 2}} &\approx 0,1875 \pi d^2 h \\ &= \frac{3}{16} \pi d^2 h \end{aligned} \quad (34)$$

(Rotationskörper in Bild 4-13).

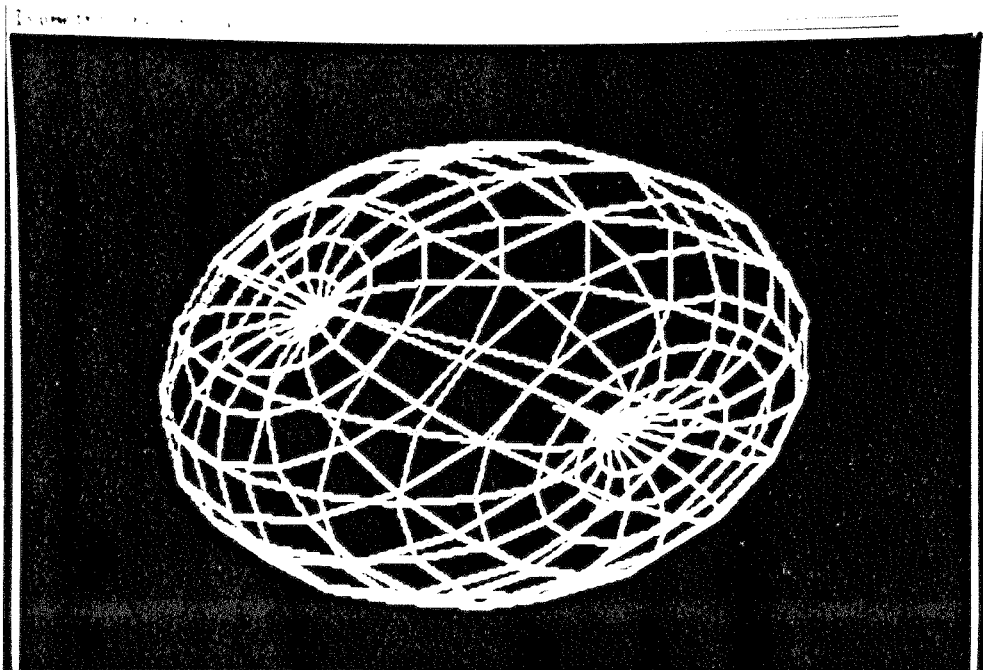


Bild 4-13: Die asymmetrische und damit verbesserte Form des Apfels

Auch an Äpfeln wurden Volumenmessungen zur Überprüfung der Formeln durchgeführt, was besonders interessant ist, da hier zwei verschiedene Lösungen miteinander konkurrieren (Tabelle in Bild 4-14).

d gemessen	(in cm)	7,9	7,2	7,8	9,0	
h gemessen	(in cm)	6,1	5,9	6,6	8,9	
V _{Exp}	(in cm ³)	200	160	220	400	
V _{Apfel} (28)	(in cm ³)	210,0	168,7	221,5	397,7	
Abweichung	(in %)	5,0	5,4	0,7	0,6	
V _{Apfel} (34)	(in cm ³)	224,3	180,2	236,5	424,6	
Abweichung*	(in %)	7,2	12,6	7,5	6,2	

* mit [5], S. 448

Bild 4-14: Die Tabelle der Meßwerte von Äpfeln

Überraschenderweise ist die Volumenberechnung mit (34) ungenauer als mit (28), da hier der mittlere prozentuale Fehler kleiner ist. Jedoch ist bei (34) die geometrische Ähnlichkeit mit einem Apfellängsschnitt wesentlich größer, so daß im praktischen Gebrauch durch Ansatz eines Korrekturgliedes durch Hinzunahme der unterschiedlichen Tiefen der Nabelpunkte eine wesentliche Verbesserung erreicht werden könnte. Dies würde jedoch zuerst eine umfangreiche Meßreihe verschiedener Apfelsorten voraussetzen, sowie eine Anpassung der Meßwerte an die Parameterformen durch die sogenannte Regressionsrechnung.

Auch bei der Birne und dem Ei wäre hiermit eine Verbesserung möglich, doch da kein Ei dem anderen gleicht - hier hat der Volksmund unrecht - muß wie immer ein Kompromiß zwischen Ergebnis und Aufwand fürs erste genügen.

5 Ausblicke

Von der Antike bis zum Computerzeitalter spannt sich ein weiter Bogen in der Entwicklung stereometrischer Berechnungen, insbesondere der Volumenberechnung. Dabei ist die Infinitesimalrechnung ein wichtiges Hilfsmittel geworden. Der Vollständigkeit halber sei erwähnt, daß sich Volumina auch in Polarkoordinaten, mit Gebietsintegralen, Raumintegralen und allgemeinen Volumenintegralen berechnen lassen. Grundsätzlich bleibt jedoch immer die Bestimmung der dem Körper zugrundeliegenden Funktion in einer oder mehreren Ebenen im Vordergrund stehen. Wenn nicht als klares geometrisches Gesetz, dann zumindest als Annahme und Arbeitshypothese. Hierzu ist zu erwähnen, daß sogar umfangreiche Hand- und Lehrbücher der Mathematik sich in der Berechnung elementarer Körper erschöpfen, allerhöchstens werden Kegel, Kugel, Paraboloid oder Ellipsoid mit ihren stereometrischen Größen bestimmt. Eine Birne ist aber kein Zylinder, ein Apfel keine Kugel, und ein Ei kein Ellipsoid, denn die Natur zeigt uns wesentlich komplexere und kompliziertere Formen auf. Der natürlichen Umwelt und Form gegenüber bleibt die elementare Stereometrie eine theoretische Lehre, denn sie beschreibt etwas, das es nur in Annäherung gibt. Die Beschreibung der Natur und ihrer Formen, man denke nur an Berge, Bäume und deren Früchte, steckt immer noch in den Kinderschuhen. Erste Schritte auf diesem neuen Forschungsgebiet wurden erst mit Benoit Mandelbrot und der Entdeckung der sogenannten »Apfelmännchen« oder auch »Mandelbrotmengen« unternommen. Dieser neue Zweig der Mathematik, die fraktale Geometrie, spielt sich aber hauptsächlich in der Ebene der komplexen Zahlen ab, doch auch die normale Mathematik ist - wie im letzten Kapitel gezeigt wurde - zur Beschreibung von natürlichen Körpern in der Lage [22].

Die hier vorgestellten Methoden versuchen die praktische Realität abzubilden, reale biologische Körper sind aber nur mit einer Vielzahl von Einflußgrößen zu erfassen, dementsprechend schwierig wird die rechnerische und analytische Behandlung. Dazu ist mehr mathematische Theorie als üblich notwendig - dies bedeutet aber auch ein Mehr an Realität und mehr Wert für die Praxis.

6 Anhang: Programme

6.1 Programm Drehe Funktion f(x)

```
(* Programm   Drehefx.MOD                                     *)
(* Mathematik Facharbeit 1988/1990, Julius-Echter-Gymnasium Elsenfeld *)
(* Autor:     Marcus Bäckmann, Hauptstr. 151a, 8751 Heimbuchenthal *)
(* Datum:     13.01.1990                                       *)

MODULE Drehefx;

FROM IntuiSupport IMPORT SetScreen,SetWindow,GetIMes;
FROM IntGadget IMPORT SetBooleanGadget,FreeBooleanGadget;
FROM PrintTools IMPORT Print,PtoG,Locate,CLS,Pen,PrintFFP,AutoLF,Using,OFF;
FROM GfxTools IMPORT MoveTo,DrawTo,Line,Origin,OrigoModes;
FROM GfxRTools IMPORT MoveR,DrawR,LineR,GPrintR;
FROM SYSTEM IMPORT ADR,ADDRESS,FFP;
FROM Arts IMPORT TermProcedure;
FROM Intuition IMPORT ScreenPtr,Gadget,WindowFlags,WindowFlagSet,
                    ActivationFlags,ActivationFlagSet,IDCMPFlags,
                    IDCMPFlagSet,WindowPtr,WindowToFront,CloseWindow,
                    CloseScreen;
FROM Graphics IMPORT RastPortPtr,SetRGB4,ViewPortPtr,SetAPen;
FROM Dos IMPORT Delay;
FROM MathTrans IMPORT Cos,Sin,Sqrt,Tieeee,Fieeee;

CONST PI = 3.1415926536;
      Zeit = 200;
      schwarz = 0;
      weiss = 1;
      gelb = 2;
      gruen = 3;
      MaxFunk = 5; (* Anzahl der Funktionen *)
      Kanten = 15; (* Anzahl der Kanten des Rot-Körpers *)
      KantPunkte = 20; (* Anzahl der Kantepunkte des Körpers *)

TYPE FFPFunktion = PROCEDURE(FFP) : FFP;
  Funktionsblock = RECORD
    Term : ARRAY [0..20] OF CHAR;
    f : FFPFunktion;
    Xmin,Xmax : FFP;
  END;
  Punkt3dim = RECORD
    x,y,z : FFP;
  END;
  Punkt2dim = RECORD
    x,y : FFP;
  END;
  PunktPix = RECORD
    x,y : INTEGER;
  END;
  PFeld3dim = ARRAY [1..Kanten],[1..KantPunkte] OF Punkt3dim;
  PFeld2dim = ARRAY [1..Kanten],[1..KantPunkte] OF Punkt2dim;
  PFeldPix = ARRAY [1..Kanten],[1..KantPunkte] OF PunktPix;

VAR BildPtr : ScreenPtr;
    Komm,
    GraFenster,
    Kon : WindowPtr;
    Krp,Grp,Frp : RastPortPtr;
    Gad : ARRAY[0..MaxFunk] OF Gadget;
    i : INTEGER; (* i und c sind Dummy-Variablen *)
    c : CARDINAL;
    Aktiv : CARDINAL; (* aktive/dargestellte Funktion *)
    Funktion : ARRAY [1..MaxFunk] OF Funktionsblock;
```

```
Ymin,Ymax : FFP;
Koord3d    : PFeld3dim;
KoordPix   : PFeldPix;

(* Hier werden alle Funktionen definiert *)
PROCEDURE f1(x : FFP) : FFP;
BEGIN
  RETURN Sqrt(x);
END f1;

PROCEDURE f2(x : FFP) : FFP;
BEGIN
  RETURN ABS(Sin(x));
END f2;

PROCEDURE f3(x : FFP) : FFP;
BEGIN
  RETURN ABS(Cos(x));
END f3;

PROCEDURE f4(x : FFP) : FFP;
BEGIN
  RETURN -(x*x)+4.0*x; (* verschobene Normalparabel der Form *)
END f4;                (*      -(x-2)2+4      *)

PROCEDURE f5(x : FFP) : FFP;
BEGIN
  RETURN (x/3.0)*Sqrt(6.0*x-x*x); (* "Zwiebel" *)
END f5;

PROCEDURE InitFunktionen;
BEGIN
  Funktion[1].Term:="sqr x      ";
  Funktion[1].f    :=f1;
  Funktion[1].Xmin:=0.0;
  Funktion[1].Xmax:=4.0;

  Funktion[2].Term:="|sin x|    ";
  Funktion[2].f    :=f2;
  Funktion[2].Xmin:=0.0;
  Funktion[2].Xmax:=2.0 * PI;

  Funktion[3].Term:="|cos x|    ";
  Funktion[3].f    :=f3;
  Funktion[3].Xmin:=0.0;
  Funktion[3].Xmax:=PI;

  Funktion[4].Term:="-(x-2)2+4  ";
  Funktion[4].f    :=f4;
  Funktion[4].Xmin:=0.0;
  Funktion[4].Xmax:=4.0;

  Funktion[5].Term:="x/3 sqr(6x-x2)";
  Funktion[5].f    :=f5;
  Funktion[5].Xmin:=0.0;
  Funktion[5].Xmax:=6.0;
END InitFunktionen;

PROCEDURE degTOrad(grad : FFP) : FFP;
BEGIN (* wandelt Grad in Bogenmaß um (für Winkelfunktionen !) *)
  RETURN (grad * PI / 180.0);
END degTOrad;

PROCEDURE BildschirmInit;
CONST wflags = WindowFlagSet{windowDrag,windowDepth};
VAR vp : ViewPortPtr;
BEGIN
  BildPtr:=SetScreen(ADR("Drehung von f(x)"),640,256,2);
  vp:=ADR(BildPtr.viewPort);
```

```
SetRGB4(vp,schwarz, 0, 0, 0);
SetRGB4(vp,weiss, 15,15,15);
SetRGB4(vp,gelb, 15,15, 0);
SetRGB4(vp,gruen, 0,15, 0);
Komm:=SetWindow(0,20,320,128,ADR("Menü"),
               WindowFlagSet{windowClose,windowDrag,windowDepth,
                             activate},
               IDCMPFlagSet{closeWindow,gadgetUp},BildPtr);
Kon:=SetWindow(320,20,320,128,ADR("Kontur f(x)"),wflags,
               IDCMPFlagSet{},BildPtr);
GraFenster:=SetWindow(0,10,640,246,ADR("Isometrische Projektion"),wflags,
                      IDCMPFlagSet{},BildPtr);
Krp:=Komm^.rPort;
Frp:=Kon^.rPort;
Grp:=GraFenster^.rPort;
END BildschirmInit;

PROCEDURE HauptMenueAnzeigen;
VAR i : CARDINAL;
    x,y : INTEGER;
BEGIN
  CLS(Komm);
  Pen(Komm,gruen);
  Locate(Komm,3,2);
  Print(Komm,"Folgende Funktionen sind verfügbar:");
  FOR i:=1 TO MaxFunk DO
    PtoG(Komm,13,1+i+i,x,y);
    SetBooleanGadget(Gad[i],Komm,NIL,ADR(Funktion[i].Term),x,y,
                    ActivationFlagSet{gadgImmediate,relVerify});
  END;
  PtoG(Komm,13,13,x,y);
  SetBooleanGadget(Gad[0],Komm,NIL,ADR(" E N D E "),x,y,
                  ActivationFlagSet{gadgImmediate,relVerify});
END HauptMenueAnzeigen;

PROCEDURE HauptMenueAbfragen(VAR gadget : CARDINAL);
CONST NotUsed = 9999;
VAR code : CARDINAL;
    adr : ADDRESS;
    class : IDCMPFlagSet;
    i : CARDINAL;
BEGIN
  WindowToFront(Komm);
  gadget:=NotUsed;
  WHILE gadget=NotUsed DO
    GetIMes(Komm,class,code,adr);
    IF (closeWindow IN class) THEN
      gadget:=0;
    END;
    IF (gadgetUp IN class) THEN
      FOR i:=0 TO MaxFunk DO
        IF (ADR(Gad[i])=adr) THEN
          gadget:=i;
        END;
      END;
    END;
  END;
  FOR i:=0 TO MaxFunk DO
    FreeBooleanGadget(Gad[i],Komm);
  END;
  CLS(Komm);
END HauptMenueAbfragen;

PROCEDURE Wertebereich(FunkData : Funktionsblock; VAR Min : FFP;
                      VAR Max : FFP);
CONST Schritte = 400.0;
VAR f : FFPfunktion;
    x,y: FFP;
    sw : FFP;
```

```
BEGIN
  f:=FunkData.f;
  x:=FunkData.Xmin;
  Min:=10.OE9;
  Max:=-10.OE9;
  sw:=(FunkData.Xmax-FunkData.Xmin)/Schritte;
  WHILE (x <= FunkData.Xmax) DO
    y:=f(x);
    IF y < Min THEN
      Min:=y;
    ELSIF y > Max THEN
      Max:=y;
    END;
    x:=x + sw;
  END;
END Wertebereich;

PROCEDURE ZeichneGraph(FunkData : Funktionsblock; Min,Max : FFP);
CONST XBreite = 300.0;
     YHoehe = 60.0;
     Schritte = 100.0;
VAR f      : FFPFunktion;
    x,y,
    sw,
    Skalx,
    Skaly : FFP;
    XPix,
    YPix  : INTEGER;

PROCEDURE ZeichneAchsen;
VAR x,y      : FFP;
    XPix,YPix : INTEGER;
BEGIN
  SetAPen(Frp,weiss);
  x:=FunkData.Xmax * 1.05;
  XPix:=TRUNC(Tieeee(x * Skalx));
  Line(Frp,0,0,XPix,0); (* Die X-Achse *)
  DrawR(Frp,-16,-2); (* Der Pfeil *)
  LineR(Frp,16,2,-16,2);
  GPrintR(Frp,9,-13,ADR("X "),1);

  y:=Max * 1.05;
  YPix:=TRUNC(Tieeee(y * Skaly));
  Line(Frp,0,0,YPix,0); (* Die Y-Achse *)
  DrawR(Frp,-4,-8); (* Der Pfeil *)
  LineR(Frp,4,8,4,-8);
  GPrintR(Frp,-18,10,ADR("Y "),1);
END ZeichneAchsen;

BEGIN
  CLS(Kon);
  Origin(Frp,20,90,Up);
  f:=FunkData.f;
  Skalx:=XBreite/((FunkData.Xmax-FunkData.Xmin)*1.1);
  Skaly:=YHoehe/((Max-Min)*1.1);
  ZeichneAchsen;
  x:=FunkData.Xmin;
  sw:=(FunkData.Xmax-FunkData.Xmin)/Schritte;
  XPix:=TRUNC(Tieeee(x * Skalx));
  YPix:=TRUNC(Tieeee(f(x) * Skaly));
  MoveTo(Frp,XPix,YPix);
  SetAPen(Frp,gelb);
  WHILE (x < FunkData.Xmax) DO
    y:=f(x);
    XPix:=TRUNC(Tieeee(x * Skalx));
    YPix:=TRUNC(Tieeee(y * Skaly));
    DrawTo(Frp,XPix,YPix);
    x:=x+sw;
  END;
```



```
WindowToFront (Kon);
END ZeichneGraph;

PROCEDURE WertebereichZeigen(FunkData : Funktionsblock; Min,Max : FFP);
BEGIN
  Pen(Kon, gelb);
  AutoLF(Kon, OFF);
  Using(Kon, 2, 4);
  Locate(Kon, 3, 12);
  Print(Kon, "f(x) = ");
  Print(Kon, FunkData.Term);
  Locate(Kon, 3, 13);
  Print(Kon, "|D = [");
  PrintFFP(Kon, FunkData.Xmin);
  Print(Kon, ";");
  PrintFFP(Kon, FunkData.Xmax);
  Print(Kon, "]");
  Locate(Kon, 3, 14);
  Print(Kon, "|W = [");
  PrintFFP(Kon, Min);
  Print(Kon, ";");
  PrintFFP(Kon, Max);
  Print(Kon, "]");
END WertebereichZeigen;

PROCEDURE DreheFunktion(FunkData : Funktionsblock; VAR K3d : PFeld3dim);
CONST Grad = 360.0;
VAR x, y, phi,
    xsw, phisw,
    sinphi,
    cosphi : FFP;
    k, kp : CARDINAL;
BEGIN
  xsw:=(FunkData.Xmax-FunkData.Xmin)/Fieee(FLOAT(KantPunkte));
  phisw:=Grad/Fieee(FLOAT(Kanten));
  phi:=0.0;
  FOR k:=1 TO Kanten BY 1 DO
    x:=FunkData.Xmin;
    sinphi:=Sin(degTORad(phi)); (* Werte bleiben in der Schleife *)
    cosphi:=Cos(degTORad(phi)); (* gleich ==> Geschwindigkeit *)
    FOR kp:=1 TO KantPunkte BY 1 DO
      K3d[k, kp].x:=x;
      y:=FunkData.f(x);
      K3d[k, kp].y:=y * sinphi;
      K3d[k, kp].z:=y * cosphi;
      x:=x + xsw;
    END
    (* FOR *);
    phi:=phi + phisw;
  END;
END DreheFunktion;

PROCEDURE KoerperAufFlaeche(VAR K3d : PFeld3dim; VAR KPix : PFeldPix);
VAR K2d : PFeld2dim;
    Xmin, Xmax,
    Ymin, Ymax : FFP;
PROCEDURE Mache3Dzu2D;
CONST Alpha = 30.0; (* Betrachterwinkel *)
    Beta = 60.0;
VAR k, p : CARDINAL;
    cosalpha,
    cosbeta : FFP;
BEGIN
  cosalpha:=Cos(degTORad(Alpha)); (* mehr Geschwindigkeit *)
  cosbeta :=Cos(degTORad(Beta));
  FOR k:=1 TO Kanten BY 1 DO
    FOR p:=1 TO KantPunkte BY 1 DO
      WITH K3d[k, p] DO
        K2d[k, p].x:=x * cosalpha - z * cosalpha;
```

```
                K2d[k,p].y:=y - x * cosbeta - z * cosbeta;
            END;
        END;
    END Mache3Dzu2D;

PROCEDURE FindeAussenpunkte;
    VAR k,p : CARDINAL;
        v   : FFP;
    BEGIN
        Xmin:=10.OE9; (* kleinster X-Wert *)
        Ymin:=Xmin;  (* kleinster Y-Wert *)
        Xmax:=-10.OE9; (* größter X-Wert *)
        Ymax:=Xmax;  (* größter Y-Wert *)
        FOR k:=1 TO Kanten BY 1 DO
            FOR p:=1 TO KantPunkte BY 1 DO
                v:=K2d[k,p].x; (* Geschwindigkeit *)
                IF v < Xmin THEN
                    Xmin:=v;
                ELSIF v > Xmax THEN
                    Xmax:=v;
                END;
                v:=K2d[k,p].y; (* dito *)
                IF v < Ymin THEN
                    Ymin:=v;
                ELSIF v > Ymax THEN
                    Ymax:=v;
                END;
            END;
        END;
    END FindeAussenpunkte;

PROCEDURE InBildschirmkoordinaten;
    CONST XBreite = 640.0;
          YHoehe  = 248.0; (* Bildschirmauflösung *)
    VAR k,p      : CARDINAL;
        Skalx,Skaly : FFP;
        OffSetX,
        OffSetY    : INTEGER;
    BEGIN
        Skalx:=XBreite/((Xmax-Xmin)*1.5);
        Skaly:=YHoehe/((Ymax-Ymin)*1.5);
        OffSetX:=TRUNC(Xmin * Skalx);
        OffSetY:=TRUNC(Ymin * Skaly);
        FOR k:=1 TO Kanten BY 1 DO
            FOR p:=1 TO KantPunkte BY 1 DO
                KPix[k,p].x:=TRUNC(K2d[k,p].x * Skalx) - OffSetX;
                KPix[k,p].y:=TRUNC(K2d[k,p].y * Skaly) - OffSetY;
            END;
        END;
    END InBildschirmkoordinaten;
BEGIN
    Mache3Dzu2D;
    FindeAussenpunkte;
    InBildschirmkoordinaten;
END KoerperAufFlaeche;

PROCEDURE ZeichneRotKoerper(rp : RastPortPtr; VAR KPix : PFeldPix);
    VAR k,p : CARDINAL;
    BEGIN
        CLS(GraFenster);
        Origin(Grp,100,200,Up);
        SetAPen(rp,gelb);
        (* Erst werden die Kanten gezeichnet *)
        FOR k:=1 TO Kanten BY 1 DO
            MoveTo(rp,KPix[k,1].x,KPix[k,1].y);
            FOR p:=2 TO KantPunkte BY 1 DO
                DrawTo(rp,KPix[k,p].x,KPix[k,p].y);
            END;
        END;
    END;
END;
```

```
END;
(* und dann, um den Körper herum, die Verbindungslinien zwischen *)
(* den Kantenpunkten *)
FOR p:=1 TO KantPunkte BY 1 DO
  MoveTo(rp,KPix[1,p].x,KPix[1,p].y);
  FOR k:=2 TO Kanten BY 1 DO
    DrawTo(rp,KPix[k,p].x,KPix[k,p].y);
  END;
  DrawTo(rp,KPix[1,p].x,KPix[1,p].y);
END;
WindowToFront(GraFenster);
END ZeichneRotKoerper;

PROCEDURE BildschirmSchluss;
BEGIN
  CloseWindow(Komm);
  CloseWindow(GraFenster);
  CloseWindow(Kon);
  CloseScreen(BildPtr);
END BildschirmSchluss;

(* Hauptprogramm *)

BEGIN
  (* wird bei Fehler oder Programmende aufgerufen *)
  TermProcedure(BildschirmSchluss);

  (* Funktionen initialisieren *)
  InitFunktionen;

  (* Bildschirm und Fenster öffnen *)
  BildschirmInit;

  LOOP

    (* Hauptmenü anzeigen und auf Eingabe warten *)
    CLS(Komm);
    HauptMenueAnzeigen;
    HauptMenueAbfragen(Aktiv);

    IF (Aktiv=0) THEN
      EXIT;
    END;

    (* Wertebereich der gewählten Funktion bestimmen und Graph zeichnen *)
    Wertebereich(Funktion[Aktiv],Ymin,Ymax);

    ZeichneGraph(Funktion[Aktiv],Ymin,Ymax);
    WertebereichZeigen(Funktion[Aktiv],Ymin,Ymax);

    (* Jetzt wird die Funktion um die X-Achse gedreht *)
    DreheFunktion(Funktion[Aktiv],Koord3d);

    (* Die Koordinaten der Form p(x,y,z) werden in Bildschirmkoordinaten *)
    (* der Form pix(x,y) umgerechnet. Perspektive = Schrägriß *)
    KoerperAufFlaeche(Koord3d,KoordPix);

    (* Der Rotationskörper der Funktion f(x) wird als Drahtgittermodell *)
    (* gezeichnet. *)
    ZeichneRotKoerper(Grp,KoordPix);

    (* Jetzt noch einen Moment warten... *)
    Delay(Zeit);

    (* ... dann geht es wieder von vorne los *)

  END;

END Drehefx.
```

6.2 Programm Drehe Parameterform

```
(* Programm   DreheParam.MOD *)
(* Mathematik Facharbeit 1988/1990, Julius-Echter-Gymnasium Elsenfeld *)
(* Autor:     Marcus Bäckmann, Hauptstr. 151a, 8751 Heimbuchenthal *)
(* Datum:    13.01.1990 *)

MODULE DreheParam;

FROM IntuiSupport IMPORT SetScreen,SetWindow,GetIMes;
FROM IntGadget IMPORT SetBooleanGadget,FreeBooleanGadget;
FROM PrintTools IMPORT Print,PtoG,Locate,CLS,Pen,PrintFFP,AutoLF,
                    Using,OFF;
FROM GfxTools IMPORT MoveTo,DrawTo,Line,Origin,OrigoModes;
FROM GfxRTools IMPORT MoveR,DrawR,LineR,GPrintR;
FROM SYSTEM IMPORT ADR,ADDRESS,FFP;
FROM Arts IMPORT TermProcedure;
FROM Intuition IMPORT ScreenPtr,Gadget,WindowFlags,WindowFlagSet,
                    ActivationFlags,ActivationFlagSet,IDCMPFlags,
                    IDCMPFlagSet,WindowPtr,WindowToFront,CloseWindow,
                    CloseScreen;
FROM Graphics IMPORT RastPortPtr,SetRGB4,ViewPortPtr,SetAPen;
FROM Dos IMPORT Delay;
FROM MathTrans IMPORT Cos,Sin,Sqrt,Tieeee,Fieeee;

CONST PI = 3.1415926536;
      Zeit = 300;
      schwarz = 0;
      weiss = 1;
      gelb = 2;
      gruen = 3;
      MaxFunk = 5; (* Anzahl der Funktionen *)
      Kanten = 15; (* Anzahl der Kanten des Rot-Körpers *)
      KantPunkte = 20; (* Anzahl der Kantenpunkte des Körpers *)

TYPE FFPFunktion = PROCEDURE(FFP) : FFP;
   Funktionsblock = RECORD
       Name,
       Termx,
       Termy : ARRAY [0..30] OF CHAR;
       x,y : FFPfunktion;
   END;
   Punkt3dim = RECORD
       x,y,z : FFP;
   END;
   Punkt2dim = RECORD
       x,y : FFP;
   END;
   PunktPix = RECORD
       x,y : INTEGER;
   END;
   PFeld3dim = ARRAY [1..Kanten],[1..KantPunkte] OF Punkt3dim;
   PFeld2dim = ARRAY [1..Kanten],[1..KantPunkte] OF Punkt2dim;
   PFeldPix = ARRAY [1..Kanten],[1..KantPunkte] OF PunktPix;

VAR BildPtr : ScreenPtr;
    Komm,
    GraFenster,
    Kon : WindowPtr;
    Krp,Grp,Frp : RastPortPtr;
    Gad : ARRAY[0..MaxFunk] OF Gadget;
    i : INTEGER; (* i und c sind Dummy-Variablen *)
    c : CARDINAL;
    Aktiv : CARDINAL; (* aktive/dargestellte Funktion *)
    Funktion : ARRAY [1..MaxFunk] OF Funktionsblock;
    Xmin,Xmax,
    Ymin,Ymax : FFP;
    Koord3d : PFeld3dim;
```

```
KoordPix      : PFeldPix;

(* Hier werden alle Funktionen definiert *)

PROCEDURE x1(t : FFP) : FFP;
  BEGIN
    RETURN Cos(t);
  END x1;

PROCEDURE y1(t : FFP) : FFP;
  BEGIN
    RETURN Sin(t);
  END y1;

PROCEDURE x2(t : FFP) : FFP;
  BEGIN
    RETURN (2.0*Cos(t)-Cos(2.0*t));
  END x2;

PROCEDURE y2(t : FFP) : FFP;
  BEGIN
    RETURN (2.0*Sin(t)-Sin(2.0*t));
  END y2;

PROCEDURE x3(t : FFP) : FFP;
  BEGIN
    RETURN (3.0*Cos(t)-Cos(3.0*t));
  END x3;

PROCEDURE y3(t : FFP) : FFP;
  BEGIN
    RETURN (3.0*Sin(t)-Sin(3.0*t));
  END y3;

PROCEDURE x4(t : FFP) : FFP;
  BEGIN
    RETURN t*(3.0*Cos(t)-Cos(3.0*t));
  END x4;

PROCEDURE y4(t : FFP) : FFP;
  BEGIN
    RETURN t*(3.0*Sin(t)-Sin(3.0*t));
  END y4;

PROCEDURE x5(t : FFP) : FFP;
  BEGIN
    RETURN -Sin(t/2.0)+0.5;
  END x5;

PROCEDURE y5(t : FFP) : FFP;
  VAR Hilf : FFP;
  BEGIN
    Hilf:=Sin(t/2.0);
    RETURN Sqrt(Hilf-Hilf*Hilf*Hilf);
  END y5;

PROCEDURE InitFunktionen;
  BEGIN
    Funktion[1].Name := "Kreis      ";
    Funktion[1].Termx := "cos t";
    Funktion[1].Termy := "sin t";
    Funktion[1].x     := x1;
    Funktion[1].y     := y1;

    Funktion[2].Name := "Kardioiden ";
    Funktion[2].Termx := "2cos t - cos 2t";
    Funktion[2].Termy := "2sin t - sin 2t";
    Funktion[2].x     := x2;
    Funktion[2].y     := y2;
  END;
```

```
Funktion[3].Name := "Epizykloide";
Funktion[3].Termx := "3cos t - cos 3t";
Funktion[3].Termy := "3sin t - sin 3t";
Funktion[3].x := x3;
Funktion[3].y := y3;

Funktion[4].Name := "Apfel ";
Funktion[4].Termx := "t(3cos t - cos 3t)";
Funktion[4].Termy := "t(3sin t - sin 3t)";
Funktion[4].x := x4;
Funktion[4].y := y4;

Funktion[5].Name := "Ei ";
Funktion[5].Termx := "-sin(0,5t) + 0,5";
Funktion[5].Termy := "sqr(-sin3(0,5t)+sin(0,5t))";
Funktion[5].x := x5;
Funktion[5].y := y5;
END InitFunktionen;

PROCEDURE degTOrad(grad : FFP) : FFP;
BEGIN (* wandelt Grad in Bogenmaß um (für Winkelfunktionen !) *)
RETURN (grad * PI / 180.0);
END degTOrad;

PROCEDURE Min(f,g : FFP) : FFP;
BEGIN (* gibt die kleinere Zahl von f und g zurück *)
IF f<=g THEN
RETURN f;
ELSE
RETURN g;
END;
END Min;

PROCEDURE Max(f,g : FFP) : FFP;
BEGIN (* gibt die größere Zahl von f und g zurück *)
IF f>=g THEN
RETURN f;
ELSE
RETURN g;
END;
END Max;

PROCEDURE BildschirmInit;
CONST wflags = WindowFlagSet{windowDrag,windowDepth};
VAR vp : ViewPortPtr;
BEGIN
BildPtr:=SetScreen(ADR("Drehung von x(t);y(t)"),640,256,2);
vp:=ADR(BildPtr^.viewPort);
SetRGB4(vp,schwarz, 1, 1, 1);
SetRGB4(vp,weiss, 15,15,15);
SetRGB4(vp,gelb, 15,15, 0);
SetRGB4(vp,gruen, 0,15, 0);
Komm:=SetWindow(0,20,320,128,ADR("Menü"),
WindowFlagSet{windowClose,windowDrag,windowDepth,
activate},
IDCMPFlagSet{closeWindow,gadgetUp},BildPtr);
Kon:=SetWindow(320,20,320,128,ADR("Parameterfunktion x(t);y(t)"),
wflags,IDCMPFlagSet{},BildPtr);
GraFenster:=SetWindow(0,10,640,246,ADR("Isometrische Projektion"),
wflags,IDCMPFlagSet{},BildPtr);
Krp:=Komm^.rPort;
Frp:=Kon^.rPort;
Grp:=GraFenster^.rPort;
END BildschirmInit;

PROCEDURE HauptMenueAnzeigen;
VAR i : CARDINAL;
x,y : INTEGER;
```

```
BEGIN
  CLS(Komm);
  Pen(Komm,gruen);
  Locate(Komm,3,2);
  Print(Komm,"Folgende Funktionen sind verfügbar:");
  FOR i:=1 TO MaxFunk DO
    PtoG(Komm,15,1+i+i,x,y);
    SetBooleanGadget(Gad[i],Komm,NIL,ADR(Funktion[i].Name),x,y,
      ActivationFlagSet{gadgImmediate,relVerify});
  END;
  PtoG(Komm,15,13,x,y);
  SetBooleanGadget(Gad[0],Komm,NIL,ADR(" E N D E "),x,y,
    ActivationFlagSet{gadgImmediate,relVerify});
END HauptMenueAnzeigen;

PROCEDURE HauptMenueAbfragen(VAR gadget : CARDINAL);
CONST NotUsed = 9999;
VAR code : CARDINAL;
    adr : ADDRESS;
    class : IDCMPFlagSet;
    i : CARDINAL;
BEGIN
  WindowToFront(Komm);
  gadget:=NotUsed;
  WHILE gadget=NotUsed DO
    GetIMes(Komm,class,code,adr);
    IF (closeWindow IN class) THEN
      gadget:=0;
    END;
    IF (gadgetUp IN class) THEN
      FOR i:=0 TO MaxFunk DO
        IF (ADR(Gad[i])=adr) THEN
          gadget:=i;
        END;
      END;
    END;
  END;
  FOR i:=0 TO MaxFunk DO
    FreeBooleanGadget(Gad[i],Komm);
  END;
  CLS(Komm);
END HauptMenueAbfragen;

PROCEDURE Wertebereich(f : FFPFunktion; VAR Min,Max : FFP);
CONST Schritte = 400.0;
VAR t,ft: FFP;
    sw : FFP;
BEGIN
  t:=0.0;
  sw:=PI/Schritte;
  Min:=f(0.0);
  Max:=f(0.0);
  WHILE (t <= PI) DO
    ft:=f(t);
    IF ft < Min THEN
      Min:=ft;
    ELSIF ft > Max THEN
      Max:=ft;
    END;
    t:=t + sw;
  END;
END Wertebereich;

PROCEDURE ZeichneGraph(FunkData : Funktionsblock; xmin,xmax,ymin,ymax : FFP);
CONST XBreite = 300.0;
    YHoehe = 60.0;
    Schritte = 100.0;
VAR xt,yt : FFPFunktion;
    t,x,y,
```

```
sw,  
Skalx,  
Skaly : FFP;  
XPix,  
YPix,  
xo    : INTEGER;  
  
PROCEDURE ZeichneAchsen;  
  VAR x,y    : FFP;  
      Pix1,Pix2 : INTEGER;  
  BEGIN  
    SetAPen(Frp,weiss);  
    x:=xmin * 1.05;  
    Pix1:=TRUNC(Tieeee(x * Skalx));  
    x:=xmax * 1.05;  
    Pix2:=TRUNC(Tieeee(x * Skalx));  
    Line(Frp,Pix1,0,Pix2,0);    (* Die X-Achse *)  
    DrawR(Frp,-16,-2);        (* Der Pfeil *)  
    LineR(Frp,16,2,-16,2);  
    GPrintR(Frp,9,-13,ADR("X "),1);  
  
    y:=ymax * 1.05;  
    YPix:=TRUNC(Tieeee(y * Skaly));  
    Line(Frp,0,0,0,YPix);    (* Die Y-Achse *)  
    DrawR(Frp,-4,-8);        (* Der Pfeil *)  
    LineR(Frp,4,8,4,-8);  
    GPrintR(Frp,-18,10,ADR("Y "),1);  
  END ZeichneAchsen;  
  
BEGIN  
  CLS(Kon);  
  xo:=TRUNC(Tieeee((-xmax/xmin)*160.0));  
  Origin(Frp,160,85,Up);  
  xt:=FunkData.x;  
  yt:=FunkData.y;  
  Skalx:=XBreite/((xmax-xmin)*1.1);  
  Skaly:=YHoehe/((ymax-ymin)*1.1);  
  Skaly:=Min(Skalx,Skaly);  
  Skalx:=Skaly*1.75; (* <-- Korrekturfaktor *)  
  ZeichneAchsen;  
  t:=0.0;  
  sw:=PI/Schritte;  
  XPix:=TRUNC(Tieeee(xt(t) * Skalx));  
  YPix:=TRUNC(Tieeee(yt(t) * Skaly));  
  MoveTo(Frp,XPix,YPix);  
  SetAPen(Frp,gelb);  
  WHILE (t <= PI) DO  
    x:=xt(t);  
    y:=yt(t);  
    XPix:=TRUNC(Tieeee(x * Skalx));  
    YPix:=TRUNC(Tieeee(y * Skaly));  
    DrawTo(Frp,XPix,YPix);  
    t:=t+sw;  
  END;  
  WindowToFront(Kon);  
END ZeichneGraph;  
  
PROCEDURE WertebereichZeigen(FunkData : Funktionsblock;  
                             xmin,xmax,ymin,ymax : FFP);  
  BEGIN  
    Pen(Kon,gelb);  
    AutoLF(Kon,OFF);  
    Using(Kon,2,4);  
    Locate(Kon,3,11);  
    Print(Kon,"x(t) = ");  
    Print(Kon,FunkData.Termx);  
    Locate(Kon,3,12);  
    Print(Kon,"y(t) = ");  
    Print(Kon,FunkData.Termy);
```



```
Locate (Kon, 3, 13);
Print (Kon, "x : ["");
PrintFFP (Kon, xmin);
Print (Kon, "];");
PrintFFP (Kon, xmax);
Print (Kon, "];");
Locate (Kon, 3, 14);
Print (Kon, "y : ["");
PrintFFP (Kon, ymin);
Print (Kon, "];");
PrintFFP (Kon, ymax);
Print (Kon, "];");
END WertebereichZeigen;

PROCEDURE DreheFunktion(FunkData : Funktionsblock; VAR K3d : PFeld3dim);
CONST Grad = 360.0;
VAR t, y, tsw,
    phi, phisw,
    sinphi,
    cosphi : FFP;
    k, kp : CARDINAL;
BEGIN
    tsw:=PI/Fieee(FLOAT(KantPunkte));
    phisw:=Grad/Fieee(FLOAT(Kanten));
    phi:=0.0;
    FOR k:=1 TO Kanten BY 1 DO
        t:=0.0;
        sinphi:=Sin(degTOrad(phi)); (* Werte bleiben in der Schleife *)
        cosphi:=Cos(degTOrad(phi)); (* gleich ==> Geschwindigkeit *)
        FOR kp:=1 TO KantPunkte BY 1 DO
            K3d[k, kp].x:=FunkData.x(t);
            y:=FunkData.y(t);
            K3d[k, kp].y:=y * sinphi;
            K3d[k, kp].z:=y * cosphi;
            t:=t+tsw;
        END;
        phi:=phi+phisw;
    END;
END DreheFunktion;

PROCEDURE KoerperAufFlaeche(VAR K3d : PFeld3dim; VAR KPix : PFeldPix);
VAR K2d : PFeld2dim;
    Xmin, Xmax,
    Ymin, Ymax : FFP;
PROCEDURE Mache3Dzu2D;
CONST Alpha = 30.0; (* Betrachterwinkel *)
    Beta = 60.0;
VAR k, p : CARDINAL;
    cosalpha,
    cosbeta : FFP;
BEGIN
    cosalpha:=Cos(degTOrad(Alpha)); (* mehr Geschwindigkeit *)
    cosbeta :=Cos(degTOrad(Beta));
    FOR k:=1 TO Kanten BY 1 DO
        FOR p:=1 TO KantPunkte BY 1 DO
            WITH K3d[k, p] DO
                K2d[k, p].x:=x * cosalpha - z * cosalpha;
                K2d[k, p].y:=y - x * cosbeta - z * cosbeta;
            END;
        END;
    END;
END Mache3Dzu2D;

PROCEDURE FindeAussenpunkte;
VAR k, p : CARDINAL;
    v : FFP;
BEGIN
    Xmin:=10.0E9; (* kleinster X-Wert *)
    Ymin:=Xmin; (* kleinster Y-Wert *)
```

```
Xmax:=-10.0E9; (* größter X-Wert *)
Ymax:=Xmax; (* größter X-Wert *)
FOR k:=1 TO Kanten BY 1 DO
  FOR p:=1 TO KantPunkte BY 1 DO
    v:=K2d[k,p].x; (* Geschwindigkeit *)
    IF v < Xmin THEN
      Xmin:=v;
    ELSIF v > Xmax THEN
      Xmax:=v;
    END;
    v:=K2d[k,p].y; (* dito *)
    IF v < Ymin THEN
      Ymin:=v;
    ELSIF v > Ymax THEN
      Ymax:=v;
    END;
  END;
END;
END FindeAussenpunkte;

PROCEDURE InBildschirmkoordinaten;
CONST XBreite = 640.0;
      YHoehe = 248.0; (* Bildschirmauflösung *)
VAR k,p : CARDINAL;
    Skalx,Skaly : FFP;
    OffSetX,
    OffSetY : INTEGER;
BEGIN
  Skalx:=XBreite/((Xmax-Xmin)*1.5);
  Skaly:=YHoehe/((Ymax-Ymin)*1.5);
  OffSetX:=TRUNC(Xmin * Skalx);
  OffSetY:=TRUNC(Ymin * Skaly);
  FOR k:=1 TO Kanten BY 1 DO
    FOR p:=1 TO KantPunkte BY 1 DO
      KPix[k,p].x:=TRUNC(K2d[k,p].x * Skalx) - OffSetX;
      KPix[k,p].y:=TRUNC(K2d[k,p].y * Skaly) - OffSetY;
    END;
  END;
END InBildschirmkoordinaten;
BEGIN
  Mache3Dzu2D;
  FindeAussenpunkte;
  InBildschirmkoordinaten;
END KoerperAufflaeche;

PROCEDURE ZeichneRotKoerper(rp : RastPortPtr; VAR KPix : PFeldPix);
VAR k,p : CARDINAL;
BEGIN
  CLS(GraFenster);
  Origin(Grp,100,210,Up);
  SetAPen(rp,gelb);
  (* Erst werden die Kanten gezeichnet *)
  FOR k:=1 TO Kanten BY 1 DO
    MoveTo(rp,KPix[k,1].x,KPix[k,1].y);
    FOR p:=2 TO KantPunkte BY 1 DO
      DrawTo(rp,KPix[k,p].x,KPix[k,p].y);
    END;
  END;
  (* und dann, um den Körper herum, die Verbindungslinien zwischen *)
  (* den Kantenpunkten *)
  FOR p:=1 TO KantPunkte BY 1 DO
    MoveTo(rp,KPix[1,p].x,KPix[1,p].y);
    FOR k:=2 TO Kanten BY 1 DO
      DrawTo(rp,KPix[k,p].x,KPix[k,p].y);
    END;
    DrawTo(rp,KPix[1,p].x,KPix[1,p].y);
  END;
  WindowToFront(GraFenster);
END ZeichneRotKoerper;
```

```
PROCEDURE BildschirmSchluss;
  BEGIN
    CloseWindow(Komm);
    CloseWindow(GraFenster);
    CloseWindow(Kon);
    CloseScreen(BildPtr);
  END BildschirmSchluss;

(* Hauptprogramm *)

BEGIN
  (* wird bei Fehler oder Programmende aufgerufen *)
  TermProcedure(BildschirmSchluss);

  (* Funktionen initialisieren *)
  InitFunktionen;

  (* Bildschirm und Fenster öffnen *)
  BildschirmInit;

  LOOP

    (* Hauptmenü anzeigen und auf Eingabe warten *)
    CLS(Komm);
    HauptMenueAnzeigen;
    HauptMenueAbfragen(Aktiv);

    IF (Aktiv=0) THEN
      EXIT;
    END;

    (* Wertebereiche der gewählten Funktion bestimmen und Graph zeichnen *)
    Wertebereich(Funktion[Aktiv].x,Xmin,Xmax);
    Wertebereich(Funktion[Aktiv].y,Ymin,Ymax);

    ZeichneGraph(Funktion[Aktiv],Xmin,Xmax,Ymin,Ymax);
    WertebereichZeigen(Funktion[Aktiv],Xmin,Xmax,Ymin,Ymax);

    (* Jetzt wird die Funktion um die X-Achse gedreht *)
    DreheFunktion(Funktion[Aktiv],Koord3d);

    (* Die Koordinaten der Form p(x,y,z) werden in Bildschirmkoordinaten *)
    (* der Form pix(x,y) umgerechnet. Perspektive = Schrägriß *)
    KoerperAufFlaeche(Koord3d,KoordPix);

    (* Der Rotationskörper der Funktion f(x) wird als Drahtgittermodell *)
    (* gezeichnet. *)
    ZeichneRotKoerper(Grp,KoordPix);

    (* Jetzt noch einen Moment warten... *)
    Delay(Zeit);

    (* ... dann geht es wieder von vorne los *)

  END;

  (* Programmende *)

END DreheParam.
```

6.3 Programm Gaußintegration

```
(* Programm      Integration.MOD                               *)
(* Mathematik   Facharbeit 1988/1990, Julius-Echter-Gymnasium Elsenfeld *)
(* Autor:       Marcus Bäckmann, Hauptstr. 151a, 8751 Heimbuchenthal   *)
(* Datum:       13.01.1990                                             *)

MODULE Integration;

FROM Terminal IMPORT WriteLn,WriteString;
FROM LongRealInOut IMPORT WriteReal,ReadReal;
FROM MathLibLong IMPORT pi,sqrt,cos,sin;

TYPE LRFunktion = PROCEDURE(LONGREAL) : LONGREAL;

VAR a,b,
    I   : LONGREAL;

PROCEDURE fx(x : LONGREAL) : LONGREAL;
BEGIN
  RETURN (x*(3*cos(x)-cos(3*x)))^2*(-3*sin(x)+3*sin(3*x))
END fx;

PROCEDURE Integral(f : LRFunktion; a,b : LONGREAL) : LONGREAL;
VAR n,i : CARDINAL;
    j    : INTEGER;
    A,fx,
    d,h,jf,nf,
    w,x,y,z : LONGREAL;
BEGIN
  n:=1;
  y:=0.0;
  w:=sqrt(0.6);
  LOOP
    INC(n);
    nf:=FLOAT(n);
    h:=(b-a)/2.0/nf;
    A:=0.0;
    d:=w*h;
    z:=a-h;
    FOR i:=1 TO n DO
      z:=z+2.0*h;
      FOR j:=-1 TO 1 DO
        jf:=FLOAT(j);
        x:=z+jf*d;
        fx:=f(x);
        A:=A+(8.0-jf*jf*3.0)*fx;
      END;
    END;
    A:=A*h/9.0;
    IF ABS(y-A)>(1.0E-8) THEN
      y:=A;
    ELSE
      EXIT;
    END;
  END;
  RETURN A;
END Integral;

BEGIN
  WriteLn;
  WriteString("Integralberechnung nach Gauß-Verfahren"); WriteLn;
  WriteString("====="); WriteLn;
  WriteLn;
  WriteString("Intervallgrenzen:"); WriteLn;
  WriteString("      a:= ");
  ReadReal(a);
  IF (a=3.14) THEN a:=pi; END;
```

```
WriteString("      b:= ");
ReadReal(b);
IF (b=3.14) THEN b:=pi; END;
WriteLn;
I:=Integral(fx,a,b);
WriteString("Näherungswert I[a;b]= ");
WriteReal(I,20,10);
WriteLn;
WriteLn;
END Integration.
```

7 Literaturverzeichnis:

- [1] **Biographien bedeutender Mathematiker.** Eine Sammlung von Biographien. Wußing, Hans und Arnold, Wolfgang [Hrsg.]. Berlin: Volk und Wissen Volkseigener Verlag, 1975
- [2] **Blätter für Technikgeschichte.** 41./42./43. Heft. Niederhuemer, Rolf [Hrsg.]. Wien: Springer; 1983
- [3] **Netz, Heinrich:** Formeln der Mathematik. völlig neu bearbeitet von J. Rast. München; Wien: Carl Hanser, 1986⁶
- [4] **Metzger, R.:** Vom Bruch zum Integral. Band II. Höhere Mathematik. Kurze allgemeinverständliche Mathematik für Fortgeschrittene. Trechtingshausen: R. Metzger & Söhne, 1988²
- [5] **Bartsch, Hans-Jochen:** Mathematische Formeln. Leipzig: VEB Fachbuchverlag Leipzig, 1988¹⁵
- [6] **Kreul, Marianne:** Mathematik in Beispielen. Bd. 2: Funktionen und Kurven. Thun; Frankfurt/Main: Harri Deutsch, 1989
- [7] **Netz, Heinrich:** Formeln der Mathematik. Braunschweig: Georg Westermann, 1965
- [8] **Spoerel, Johannes:** Mathematik von der Schule bis zur Hochschule. Berlin: Walter de Gruyter & Co, 1966³
- [9] **Bosch, Karl:** Mathematik Taschenbuch. München; Wien: Oldenbourg, 1989
- [10] **Mathematik Ratgeber:** für Lehrer, Schüler, Eltern und zum Selbststudium. Gekürzte Sonderausgabe d. kleinen Enzyklopädie Mathematik. Gottwald, S. [Hrsg.]. Frankfurt/Main: Harri Deutsch, 1988²
- [11] **Feichtinger, Herwig:** Arbeitsbuch Mikrocomputer: Funktion und Anwendung von Mikrocomputern, Peripherie und Software. München: Franzis, 1985
- [12] **Krüger, Ingolf:** AMIGA, Programmieren mit MODULA-2. Haar bei München: Markt & Technik, 1988
- [13] **Vieweg Programmbibliothek Mikrocomputer** Bd. 10. PC-1500 Sammlung III. Schummy, Harald [Hrsg.]. Braunschweig; Wiesbaden: Friedrich Vieweg & Sohn, 1984

- [14] Glaeser, Georg: 3-D-Programmierung mit BASIC. Stuttgart: Teubner, 1986
- [15] Bielig-Schulz, Gisela: 3D-Graphik in PASCAL. Stuttgart: Teubner, 1987
- [16] Merziger, Thomas; Wirth, Thomas: Basic Programme zur Höheren Mathematik. Hannover: C. Feldmann, 1986
- [17] Herrmann, Dietmar: Numerische Mathematik. 40 BASIC-Programme. Braunschweig; Wiesbaden: Friedrich Vieweg& Sohn, 1983
- [18] Das neue Duden-Lexikon: in 10 Bänden. Band 1: A-Bise. [Hrsg.] Meyers Lexikonredaktion. Mannheim; Wien; Zürich: Dudenverlag, 1989²
- [19] Tönnies, Christian: Praktische und angewandte Mathematik. Band 2: Differentialrechnung. Essen: Girardet, 1965
- [20] Das neue Duden-Lexikon: in 10 Bänden. Band 3: Drai-Frid. [Hrsg.] Meyers Lexikonredaktion. Mannheim; Wien; Zürich: Dudenverlag, 1989²
- [21] Kühlein, Theo: Mentor-Repetitorien. Band 34. Differentialrechnung II. Berlin-Schöneberg: Mentor, 1966⁶
- [22] Mandelbrot, Benoit B.: Die fraktale Geometrie der Natur. Basel; Boston: Birkhäuser, 1987

Bildnachweis: alle Fotos und Zeichnungen von
Marcus Bäckmann, Heimbuchenthal

Amiga ist eine Produktbezeichnung der Commodore-Amiga Inc., USA
M2Amiga ist ein eingetragenes Warenzeichen der Firma Meier-Vogt, CH

8 Erklärung

Ich erkläre, daß ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

Heimbuchenthal, den 30.01.1990

Marcus Bäckmann
