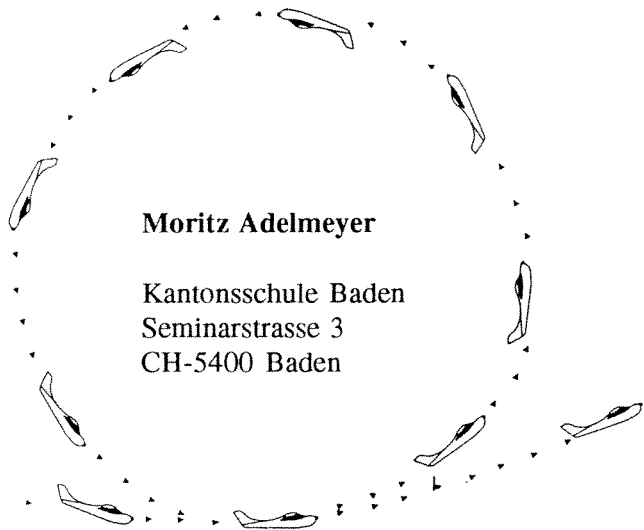




KS Flight Simulator

Eine Unterrichtsprojekt zur Raumgeometrie

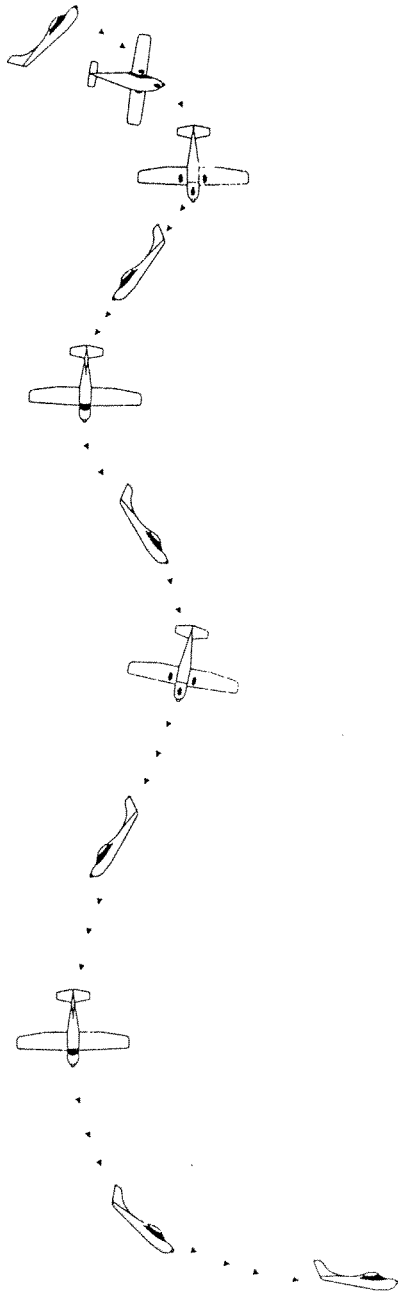


Moritz Adelmeyer

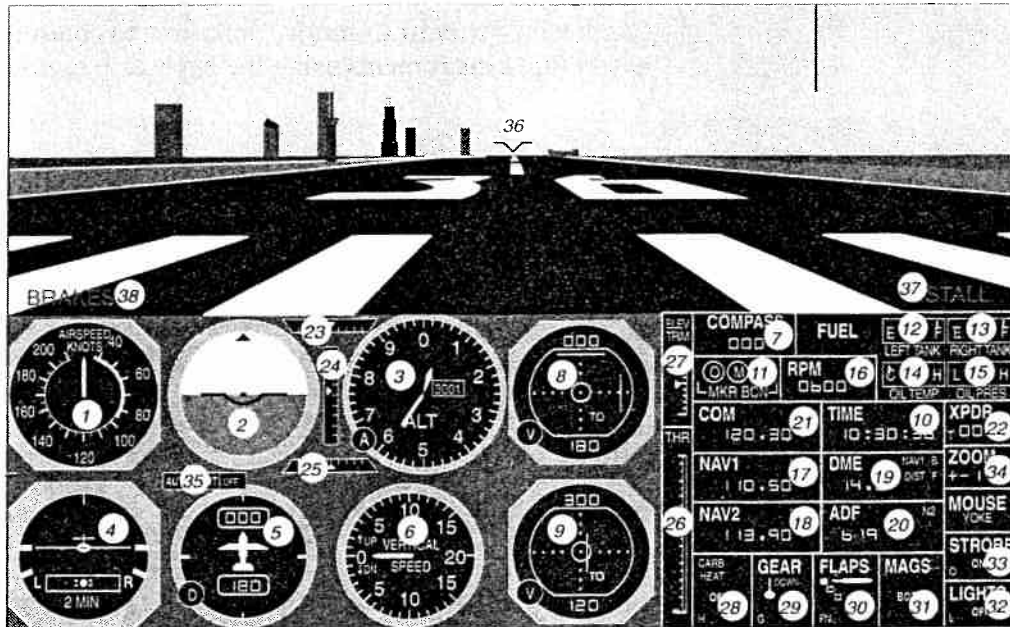
Kantonsschule Baden
Seminarstrasse 3
CH-5400 Baden

Inhaltsverzeichnis

Einstimmung	2
Projekt KS Flight Simulator	4
1. Grundlagen des Fliegens	6
2. Komponenten des KS Flight Simulator	9
3. Geometrische Berechnungen	13
4. Programmierung	15
5. Ablauf des Projekts	25
Sechs gute Gründe	28
Aufgaben	30
Prüfung	37
Quellenverzeichnis	42



Einstimmung



Ein glücklicher Start

Der Motor brummte im Leerlauf. Nervös studierte ich die Instrumententafel der Cessna. War ich so verrückt, einen Alleinflug zu wagen, ohne vorher eine einzige Flugstunde genommen zu haben? Oder sollte ich nicht lieber nach einem erfahrenen Piloten Ausschau halten, der mich sicher in die Lüfte geleitet?

Draussen, auf dem kahlen Gelände von Meigs Field in Chigago war keine Menschenseele zu sehen. Mein einziger Gefährte bei diesem selbstmörderischen Unternehmen war und blieb das Flughandbuch auf meinem Schoß. Ich hatte zehn Minuten darin gelesen.

Ich glaube, mit einem Steuerknüppel in der Hand wäre ich zuversichtlicher gewesen. Ich hatte mir mehrere Male «Top Gun» angeschaut, so dass ich immerhin in Grundzügen über das Fliegen Bescheid wusste. Aber dieses Flugzeug war supermodern. Es wurde ausschliesslich über eine Tastatur gesteuert.

Ich seufzte und schlug Seite 28 des Handbuchs auf. In einem Anfall von Mut drückte ich versuchsweise die Taste 9. Das Dröhnen des Motors wurde höher und lauter. Die Rollbahn begann langsam auf mich zuzukommen. Plötzlich fuhr mir der Schreck in die Glieder. Nur 300 Meter vor mir glitzerte das tiefblaue Wasser des Michigan-Sees. Ich brauchte mehr Tempo. Ich schlug wiederholt die Taste 9 an. Der Motor dröhnte voller, die Rollbahn zog immer schneller an mir vorbei, aber das Flugzeug hob nicht ab. Panik erfasste mich.

Die Höhenruder! Fieberhaft schlug ich im Handbuch nach. Ich musste die Höhenruder hochklappen. Das würde die Flugzeugnase nach oben bringen. Ich drückte die Taste 2 gerade noch rechtzeitig, um dem nassen Grab zu entinnen. Die Konturen von Wasser und Land wichen langsam zurück. Geschafft! Ich war in der Luft! Ohne mein Zutun hielt das Flugzeug Kurs Nord und stieg an.

Von der Linkskurve zum Looping

Sie werden es längst erraten haben: Das Cockpit war in Wahrheit mein Arbeitszimmer zu Hause, und die Bedienungstasten waren die meines Computers, auf dem ein Flugsimulationsprogramm lief. Hier konnte ich Dinge ausprobieren, die ich mich in einem wirklichen Flugzeug nie trauen würde. Auch ein Absturz würde schliesslich nicht Kopf und Kragen kosten.

Ich betätigte mit der Taste 4 die Querruder und zog so das Flugzeug in einer scharfen Linkskurve nach Westen. Der Horizont vor mir kippte nach rechts, die ganze Landschaft schien langsam die schiefe Ebene hinunterzugleiten. Vor mir tauchte Chicago auf. Nach einer Weile erspähte ich den Sears Tower. Ich war entzückt!

Berauscht von meinen ersten Erfolgen wollte ich unbedingt noch ein waghalsiges Manöver probieren. Dazu flog ich erst einmal hoch hinaus über den Michigan-See. Runter, hoch und rum – so kannte ich den Looping aus Filmen.

Ich tippte mehrmals die Taste 8. Die Nase des Flugzeugs senkte sich. Vor mir hob sich der Horizont und füllte schliesslich den ganzen Monitor. Die Maschine ging auf Sturzflug. Der Zeiger des Fahrtmessers kletterte auf 140 Knoten. Es war Zeit, die Maschine aufzufangen. Ich legte den Finger auf die Taste 2. Die Maschine kam dröhnend aus dem Sturzflug in die Horizontale und stieg dann hoch in den blauen Himmel. Ich drückte auf Taste 9 und gab Gas. Ich sah den Horizont unter mir verschwinden und wartete. Ich hatte nicht die leiseste Ahnung, ob ich es schaffen würde. Plötzlich tauchte vor mir eine auf dem Kopf stehende Landschaft auf. Ich war zu verwirrt, um klar denken zu können. Entschlossen machte ich einfach weiter. Zunächst war alles vor mir Wasser, dann schob sich von oben der Horizont ins Bild. Hastig versuchte ich die Nase nach unten zu drücken, um nicht noch einen zweiten Looping zu machen. Der Horizont schwankte und stabilisierte sich schliesslich. Ich hatte es geschafft!

Eine misslungene Landung

Die Landebahn von Meigs Field erstreckte sich von Süden nach Norden. Aus meiner Position südöstlich des Flughafens wollte ich zunächst nach Westen fliegen und sobald ich auf einer Linie mit der Piste war, nach Norden abdrehen. Aber irgendwie klappte der Plan nicht. Ich kam einfach nicht genau über die Landebahn und näherte mich dem Flughafen auf viel zu grosser Höhe. Es war ein Albtraum. Ich drosselte das Tempo und drückte die Nase der Cessna nach unten. Rasch verlor ich an Höhe, während ich mich weiter nach rechts zu schieben versuchte. Aber die Piste geriet aus dem Monitor, und ich verlor den Horizont. Der Boden raste heran. Das Programm stoppte. Verdammt, ich war abgestürzt!

Es folgte noch so mancher Absturz, was mein Selbstbewusstsein als Flieger zwar ankratzte, meiner Begeisterung für das Fliegen aber nichts anhaben konnte.

Projekt KS Flight Simulator

Das Projektziel und der Projektrahmen

Dieser Bericht dokumentiert ein Unterrichtsprojekt zur Raumgeometrie am Gymnasium. Das Projektziel ist die **Erstellung eines eigenen einfachen Flugsimulationsprogramms**. Der Projektrahmen ist wie folgt abgesteckt:

- Der Flugsimulator soll **so einfach wie möglich** und doch **so realistisch wie nötig** konzipiert werden.
- Der Flugsimulator soll weitgehendst mit den **vorhandenen** Kenntnissen und Fertigkeiten der Schüler aus dem Physik-, Mathematik- und Informatikunterricht realisiert werden.

Der Projektname

Das populärste kommerzielle Flugsimulationsprogramm für PCs ist «MS Flight Simulator». Dieses Programm diente als Orientierung. Es stand auch Pate beim Titel des Projekts. MS steht für die Herstellerfirma Microsoft. KS steht für Kantonsschule.

Die Projektteilnehmer

Das Projekt KS Flight Simulator ist an der Kantonsschule Baden zweimal durchgeführt worden, einmal von Marco Bettinaglio und einmal vom Autor dieses Berichts. In beiden Fällen waren die Rahmenbedingungen nahezu gleich:

- Die beteiligten Klassen waren 3. Klassen¹ des mathematisch-naturwissenschaftlichen Gymnasialtypus.
- Die Projekte fanden im Rahmen des Fachs Angewandte Mathematik statt. Sie erstreckten sich über ein Quartal.
- Die Schüler² brachten aus dem Mathematikunterricht umfassende Vorkenntnisse aus der Vektorgeometrie mit, aus dem Unterricht in Angewandter Mathematik Fertigkeiten in der zeichnerischen Darstellung räumlicher Objekte und im Umgang mit einer höheren Programmiersprache.

Die Raumgeometrie

Im Laufe des Projekts müssen **unzählige räumliche Situationen** wie zum Beispiel Landschaften oder Flugfiguren **gedacht, gezeichnet und gerechnet** werden. Eine spezielle raumgeometrische Herausforderung stellen die Darstellung der Cockpit-Sicht des Piloten und die Ausführung der Steuerbefehle des Piloten dar:

- Zur Darstellung eines Flugs wird die Fluglandschaft **zentralprojiziert**. Das Projektionszentrum ist der Kopf des Piloten, die Projektionsebene ist das Cockpitfenster.
- Die Steuerung eines Flugzeugs erfolgt über drei sog. Ruder. Sie bewirken, dass das Flugzeug um die Längs-, Quer-, bzw. Hochachse **rotiert**.

¹ 12. Schuljahr, eineinhalb Jahre vor der Matur.

² Schülerinnen gab es in beiden Klassen keine.

Merkmale von projektartigem Unterricht

Das Projekt KS Flight Simulator weist einige typische Merkmale von projektartigem Unterricht auf.

- Das Projektthema entstammt dem Erfahrungsbereich der Projektteilnehmer. Fast alle beteiligten Schüler haben schon mit Flugsimulationsprogrammen gespielt.
- Das Projektziel ist ein vorzeigbares Produkt. Am Ende steht ein lauffähiges Computerprogramm.
- Die Projektarbeit ist teilnehmerzentriert organisiert. Der Lehrer als Projektleiter gibt die grosse Linie vor, leitet die Arbeiten ein, ist Ansprechpartner bei Problemen. Die Schüler als Projektteilnehmer erledigen die Arbeiten selbständig in kleinen Teams.
- Die Projektteilnehmer können ihre Neigungen und Interessen ins Projekt einbringen.

Der Aufbau dieses Berichts

Dieser Bericht ist in fünf Kapitel und einen Anhang unterteilt.

- Das **Kapitel 1** legt in aller Kürze die Grundlagen des Fliegens dar, soweit sie nötig sind zum Verständnis des KS Flight Simulator. Dazu gehören insbesondere die Wirkungsweise der drei Steuer- ruder und die Funktion der Cockpitinstrumente.
- Das **Kapitel 2** zeigt, wie die vier wesentlichen Komponenten eines Flugsimulationsprogramms – die Landschaft, das Flugzeug, die Cockpit-Sicht und die Steuerung – im KS Flight Simulator modelliert werden.
- Das **Kapitel 3** präsentiert in prägnanter Form die geometrischen Berechnungen zur Darstellung der Cockpit-Sicht und zur Steuerung.
- Das **Kapitel 4** dokumentiert die Programmierarbeit. Die Programme sind in der Sprache TrueBASIC geschrieben.
- Das **Kapitel 5** beschreibt, wie das Unterrichtsprojekt in einer der beiden oben erwähnten Klassen abgelaufen ist.
- Im **Anhang** werden zunächst sechs gute Gründe für das Thema Flugsimulation im gymnasialen Mathematikunterricht angeführt. Dann sind einige Aufgaben und eine Prüfung über das Projekt KS Flight Simulator abgedruckt. Zum Schluss sind die Quellen zusammengestellt.

Dank

Ich möchte allen danken, welche direkt oder indirekt an der Ausarbeitung und/oder Durchführung des Unterrichtsprojekts beteiligt waren. Mein besonderer Dank gilt

- den Schülern der Klasse 3bR der Kantonsschule Baden des Schuljahrs 1996/97, welche das Projekt mit viel Engagement durchgeführt haben;
- Marco Bettinaglio, der das Thema Flugsimulation an der Kantonsschule Baden als erster aufgegriffen hat und dessen Vorarbeiten mir eine wertvolle Starthilfe waren;
- Prof. Dr. Urs Kirchgraber und der ETH Zürich, welche das Projekt unterstützt haben.

Kapitel 1

Grundlagen des Fliegens

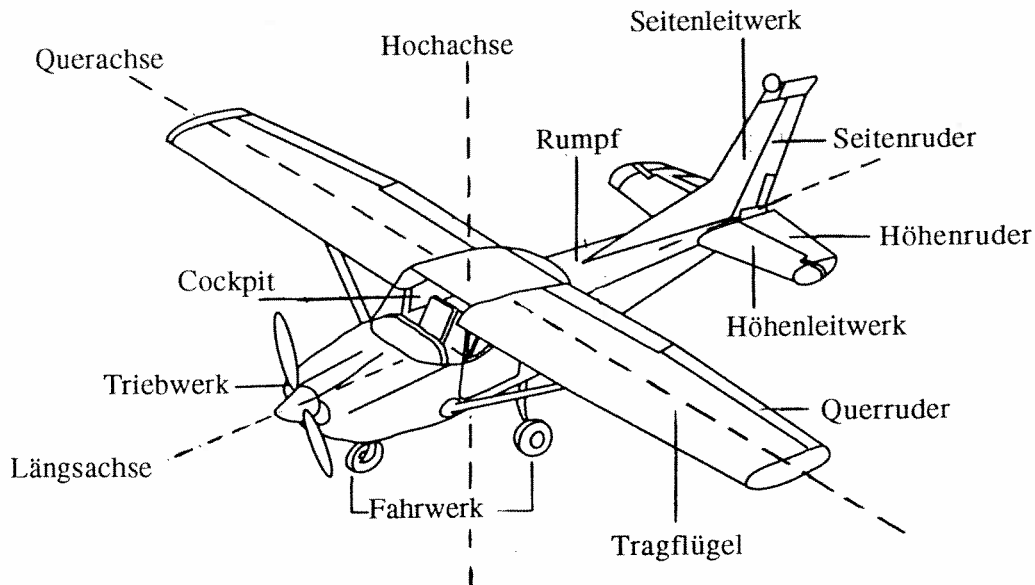


Abb. 1 Aufbau eines Flugzeugs

Ein Flugzeug besteht aus Rumpf, Tragflügel, Leitwerk, Triebwerk und Fahrwerk (Abb. 1).

Der Tragflügel. Der wichtigste Teil eines Flugzeugs ist der Tragflügel. Er wird während des Flugs von der Luft umströmt und erzeugt eine dem Flugzeuggewicht entgegengesetzt wirkende Kraft, den Auftrieb. Die Stärke des Auftriebs hängt von Fläche, Form und Stellung des Tragflügels und von der Geschwindigkeit der anströmenden Luft ab.

Die Ruder. An der Rückseite des Tragflügels sind die Querruder montiert. Sie ermöglichen es, das Flugzeug um die Längsachse zu drehen (zu «rollen») (Abb. 2).

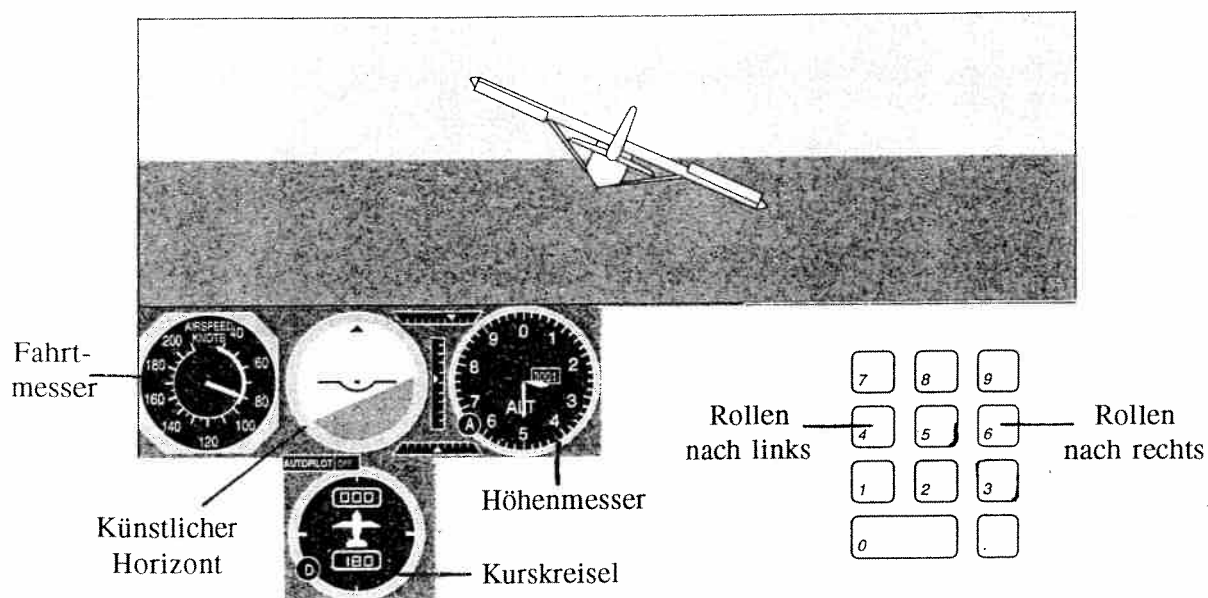


Abb. 2 Instrumente zur Flugüberwachung, Wirkung der Querruder und Tastaturbelegung im KS Flight Simulator

Hinten am Rumpf befindet sich das Leitwerk. Es besteht aus Höhen- und Seitenleitwerk. Das Leitwerk stabilisiert einerseits die Fluglage und erlaubt andererseits, das Flugzeug mit Hilfe des Höhen- und Seitenruders zu manövrieren:

- Das Höhenruder ermöglicht es, das Flugzeug um die Querachse zu drehen (zu «nicken») (Abb. 3).
- Das Seitenruder erlaubt, das Flugzeug um die Hochachse zu drehen (zu «gieren») (Abb. 4).

Das Seitenruder dient in erster Linie der Richtungshaltung. Kurven werden zur Hauptsache mit den Querrudern durchgeführt.

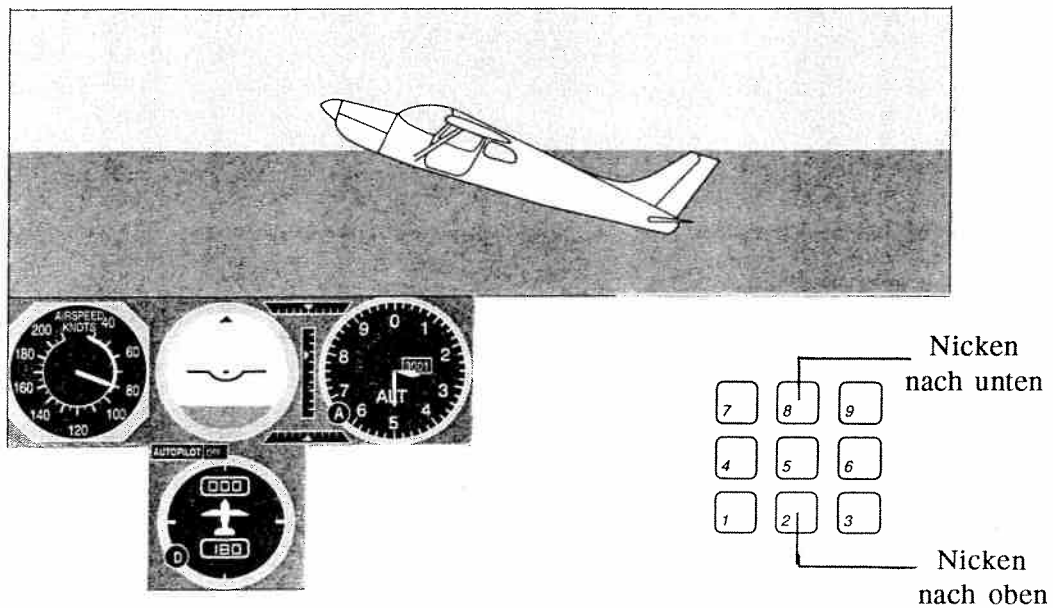


Abb. 3 Wirkung des Höhenruders

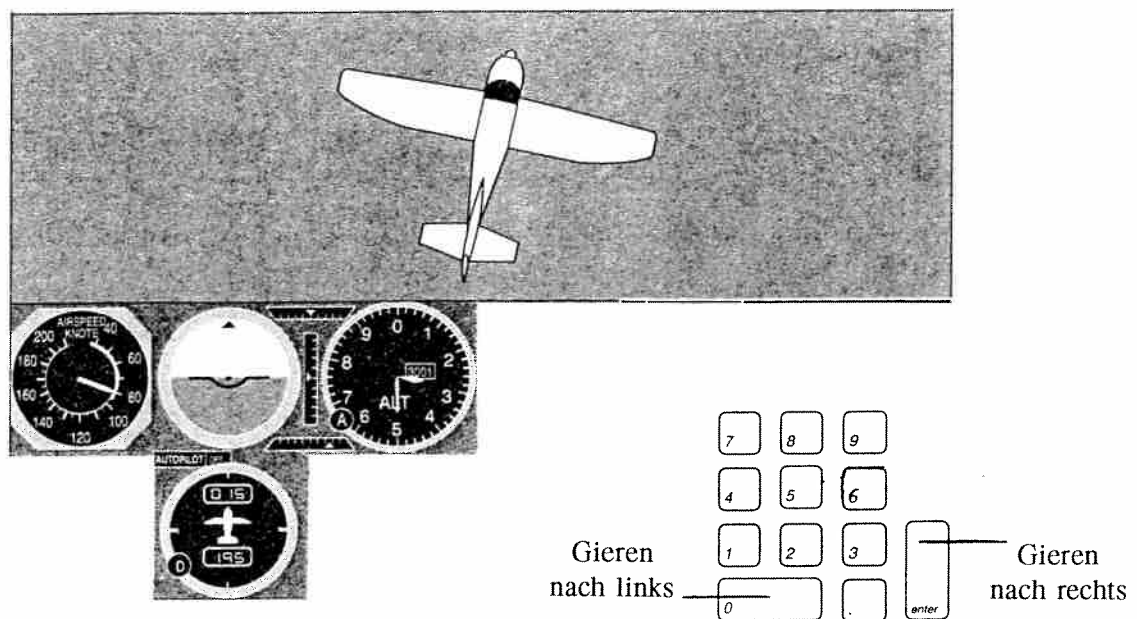


Abb. 4 Wirkung des Seitenruders

Der Pilot im Cockpit steuert das Quer- und das Höhenruder mit dem Steuerknüppel, das Seitenruder mit den Fusspedalen. Der Steuerausschlag erfolgt jeweils in Richtung der beabsichtigten Flugbewegung:

- Drücken des Steuerknüppels nach links/rechts – Rollen nach links/rechts,
- Drücken des Steuerknüppels nach vorne/hinten – Nicken nach unten/oben,
- Treten der linken/rechten Pedale – Gieren nach links/rechts.

Im KS Flight Simulator werden die Ruder über die Tastatur gesteuert (Abb. 2 – 4).

Die Hauptinstrumente. Der Pilot im Cockpit verfügt über eine Vielzahl von Instrumenten zur Flugüberwachung. Die wichtigsten sind der Fahrtmesser, der Höhenmesser, der künstliche Horizont und der Kurskreisel (Abb. 2).

Der Fahrtmesser gibt die Geschwindigkeit des Flugzeugs gegenüber der umgebenden Luft in Knoten¹ an, der Höhenmesser die Höhe des Flugzeugs über Meer in Fuss². Fahrtmesser und Höhenmesser funktionieren als Barometer. Ersterer misst den Druck der anströmenden Luft, letzterer den Druck der ruhenden Luft. Der eine setzt die Drücke in Geschwindigkeiten um, der andere in Höhen.

Der künstliche Horizont heisst auch Fluglageanzeiger. Zwei Striche und ein Halbkreis mit Punkt symbolisieren Tragflügel und Rumpf des Flugzeugs. Der helle Teil der Anzeige stellt den Himmel dar, der dunkle Teil die Erde. Die Trennlinie dazwischen heisst künstlicher Horizont. Die Lage des symbolischen Flugzeugs zum künstlichen Horizont entspricht stets genau der Lage des richtigen Flugzeugs zum natürlichen Horizont. Die Abb. 2 – 4 zeigen den künstlichen Horizont bei Kurvenflug, bei Steigflug und bei Horizontalflug.

Der Kurskreisel gibt den Kurs des Flugzeugs in Grad an. 0 Grad entspricht Kurs Nord, 90 Grad Kurs Ost, 180 Grad Kurs Süd, 270 Grad Kurs West.

Künstlicher Horizont und Kurskreisel funktionieren als schnell rotierende, kardanisch gelagerte Kreisel, die ihre Lage im Raum beibehalten. Die Kreiselachse von ersterem ist senkrecht zum Horizont ausgerichtet, die von letzterem parallel zur Erdachse.

Das Triebwerk. Das Triebwerk sorgt für den Vortrieb des Flugzeugs. Es besteht aus Motor und Propeller. Der Propeller wirkt wie ein sich drehender Tragflügel mit «Auftrieb» nach vorne. Der Motor arbeitet wie ein Automotor. Der Pilot reguliert das Triebwerk mit einem Gashebel, der dem Gaspedal im Auto vergleichbar ist. Im KS Flight Simulator wird das Triebwerk über die Tastatur reguliert (Abb. 5).

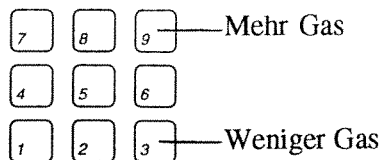


Abb. 5 Regulierung des Triebwerks

Das Fahrwerk. Das Fahrwerk besteht aus dem unter dem Tragflügel montierten Hauptfahrwerk und dem vorne am Rumpf angebrachten Bugrad. Über die Fusspedale kann der Pilot das Bugrad steuern und die am Hauptfahrwerk angebrachten Bremsen betätigen.

¹ 1 Knoten (kn) = 1.85 km/h

² 1 Fuss (ft) = 0.30 m

Kapitel 2

Komponenten des KS Flight Simulator

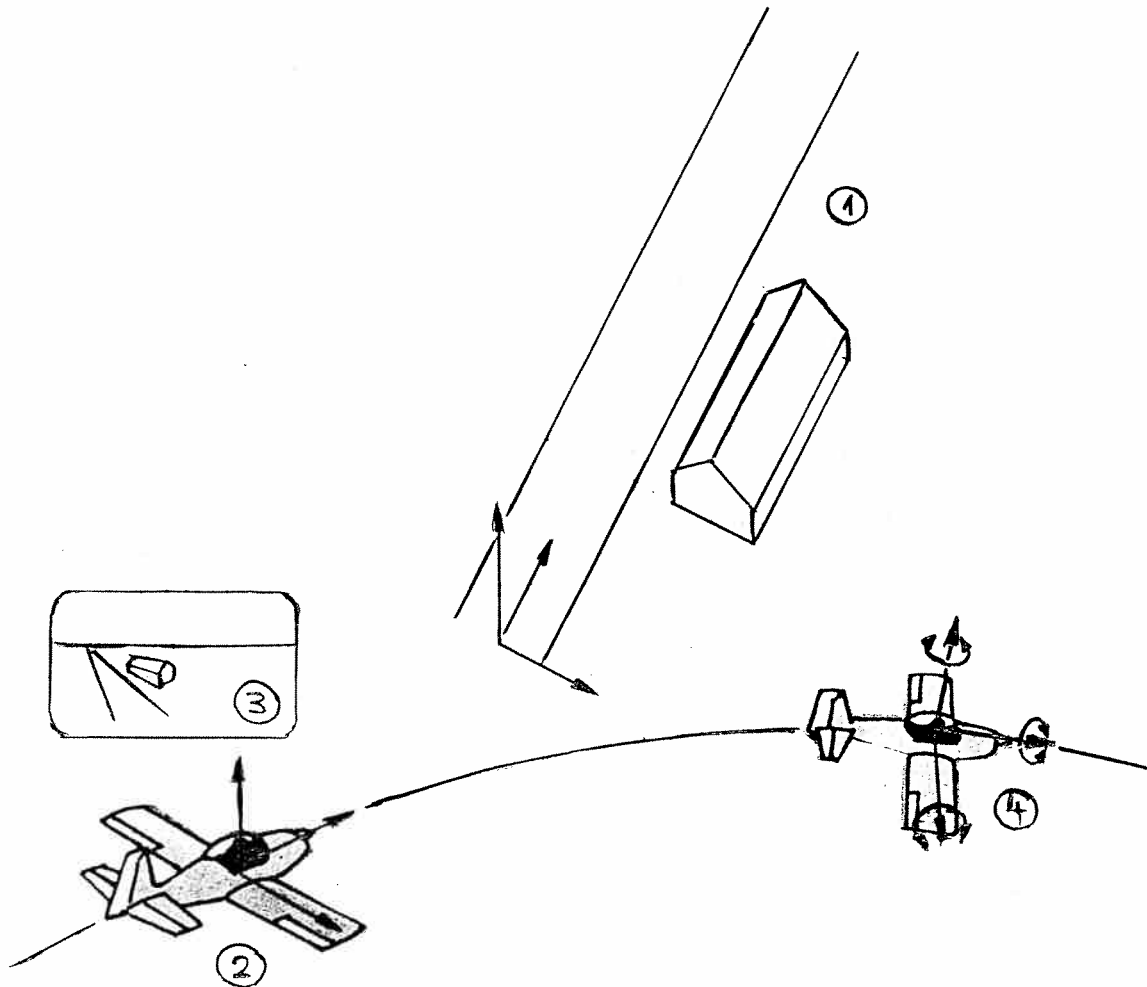


Abb. 6 Die vier Komponenten des KS Flight Simulator:
Landschaft (①), Flugzeug (②), Cockpit-Sicht (③) und Steuerung (④).

Der KS Flight Simulator besteht im wesentlichen aus vier Komponenten (Abb. 6). Sie werden im folgenden in ihrer minimalen Form vorgestellt gemäss dem Motto «so einfach wie möglich, so realistisch wie nötig».

Komponente 1: Die Landschaft

Die Landschaft besteht aus nur zwei Objekten, einer Piste und einem Hangar (Abb. 7). Diese werden durch 14 Punkte und 17 Verbindungskanten mathematisch modelliert. Zur Festlegung der Punkte $P_1, \dots, P_{14}$ werden Weltkoordinaten x, y, z eingeführt.

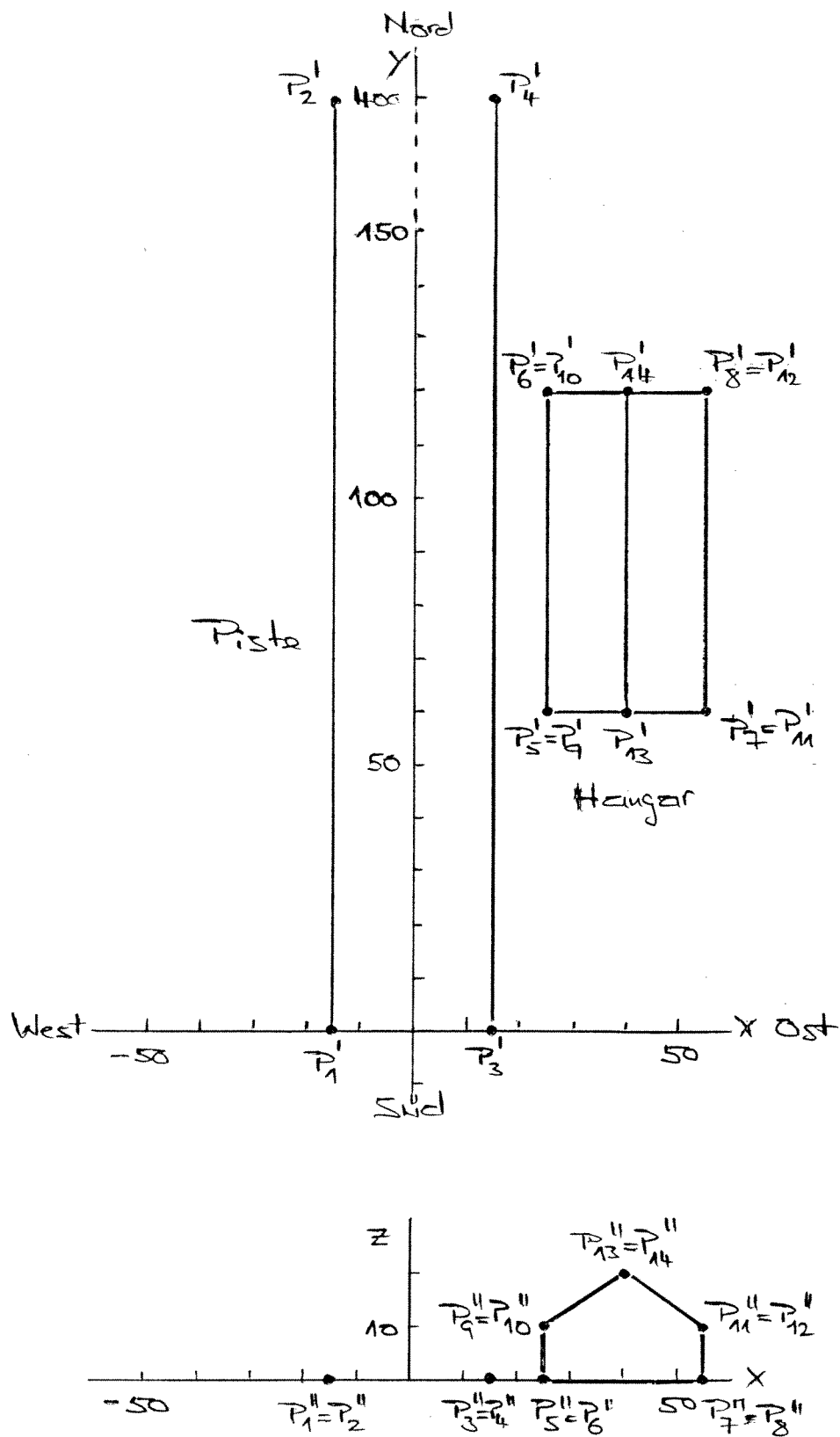


Abb. 7 Die Landschaft im Grundriss (oben) und Aufriss (unten)

Komponente 2: Das Flugzeug

Das Flugzeug besteht aus einem Punkt und drei paarweise senkrecht stehenden Einheitsvektoren (Abb. 8):

- Der Punkt M repräsentiert den Mittelpunkt des Cockpitfensters.
- Der Vektor $\vec{n}$ zeigt in Richtung der Längsachse vom Piloten aus gesehen nach vorne, der Vektor $\vec{r}$ in Richtung Querachse nach rechts und der Vektor $\vec{s}$ in Richtung Hochachse nach oben.

M legt die Position des Flugzeugs fest. Zwei der drei Vektoren $\vec{n}$, $\vec{r}$, $\vec{s}$ legen dessen Lage fest.

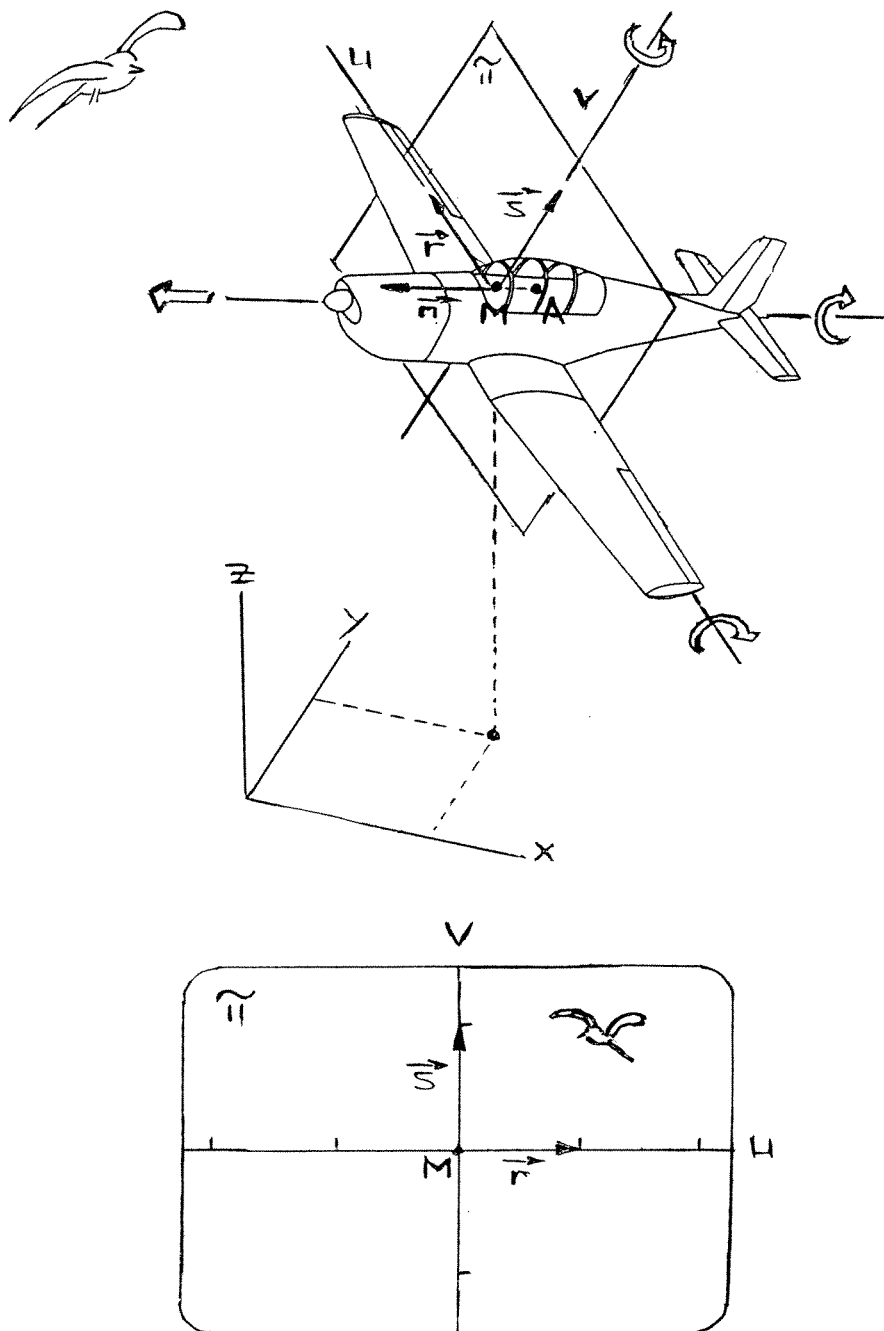


Abb. 8 Position und Lage des Flugzeugs (oben) und Cockpit-Sicht (unten)

Komponente 3: Die Cockpit-Sicht

Zur Darstellung der Cockpit-Sicht wird die Landschaft zentralprojiziert (Abb. 8):

- Die Projektionsebene π ist das Cockpitfenster. Sie wird durch $M, \vec{r}, \vec{s}$ aufgespannt.
- Das Projektionszentrum A ist der Augpunkt des Piloten. Es liegt im Abstand $d = 0.5$ m zur Projektionsebene hinter M .

Zur Festlegung der Bildpunkte dienen die Bildkoordinaten u, v .

Komponente 4: Die Steuerung

Das Flugzeug wird durch Translation in Richtung der Längsachse und durch Rotation um die Längs-Quer- und Hochachse gesteuert. Jede der drei Rotationen kann unabhängig voneinander ausgeführt werden.

Der KS Flight Simulator erlaubt, eine Kurve ganz allein mit dem Seitenruder durch eine reine Drehung um die Hochachse zu fliegen. In der Realität ist dies unmöglich. Einen Kurvenflug wird stets mit dem Querruder und einer Drehung um die Längsachse eingeleitet. Die Steuerung ist also beim KS Flight Simulator geometrisch und nicht physikalisch orientiert.

Kapitel 3

Geometrische Berechnungen

Problem 1: Berechnung der Cockpitsicht

Gegeben: Position M des Flugzeugs,
 Lage $\vec{n}, \vec{r}, \vec{s}$ des Flugzeugs (wobei $\vec{n}, \vec{r}, \vec{s}$ paarweise senkrecht stehen und Länge 1 haben),
 Distanz d zwischen Pilot und Cockpittfenster,
 Punkt P der Landschaft.
 Gesucht: Bildkoordinaten u, v des Bildpunkts B zu P .

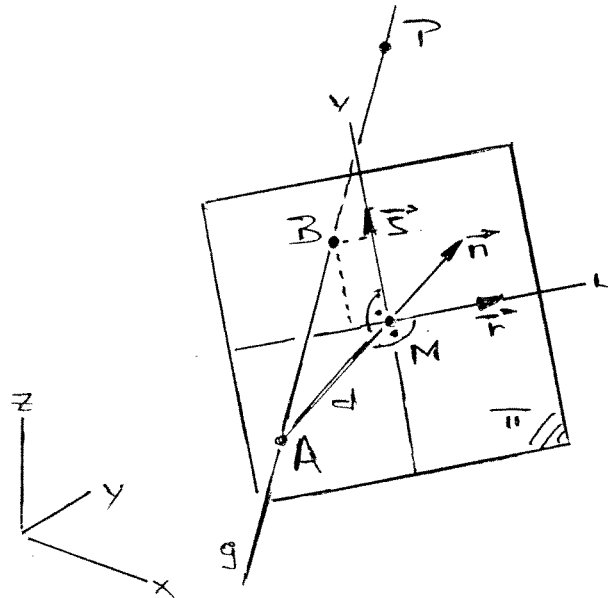


Abb. 9 Berechnung der Cockpitsicht

Nach Abb. 9 hat die Parameterdarstellung der Sehgeraden g die Form

$$\vec{OX} = \vec{OA} + t \cdot \vec{AP} = \vec{OM} - d \cdot \vec{n} + t \cdot (d \cdot \vec{n} + \vec{MP}). \quad (1)$$

Die Parameterdarstellung der Bildebene π lautet

$$\vec{OX} = \vec{OM} + u \cdot \vec{r} + v \cdot \vec{s}. \quad (2)$$

Der Bildpunkt B ist der Schnittpunkt von g mit π . Setzt man (1) und (2) gleich, so erhält man

$$t \cdot (d \cdot \vec{n} + \vec{MP}) - u \cdot \vec{r} - v \cdot \vec{s} = d \cdot \vec{n}. \quad (3)$$

Multipliziert man (3) skalar mit dem Vektor $\vec{n}$, wendet man die Distributivität des Skalarprodukts an, nutzt man die paarweise Orthogonalität und die Normiertheit der Vektoren $\vec{n}, \vec{r}, \vec{s}$ aus, so erhält man

$$t = \frac{d}{d + \vec{MP} \circ \vec{n}}. \quad (4)$$

Multipliziert man (3) skalar mit dem Vektor $\vec{r}$, bzw. $\vec{s}$, so erhält man auf die gleiche Weise

$$u = \frac{d \cdot (\vec{MP} \circ \vec{r})}{d + \vec{MP} \circ \vec{n}}, \quad \text{bzw.} \quad v = \frac{d \cdot (\vec{MP} \circ \vec{s})}{d + \vec{MP} \circ \vec{n}}. \quad (5), \text{ bzw. } (6)$$

Problem 2: Berechnung der Lage nach Steuerung

Gegeben: Momentane Lage $\vec{n}_1, \vec{r}_1, \vec{s}_1$ des Flugzeugs, (wobei $\vec{n}_1, \vec{r}_1, \vec{s}_1$ paarweise senkrecht stehen und Länge 1 haben),

Winkel α .

Gesucht: Neue Lage $\vec{n}_2, \vec{r}_2, \vec{s}_2$ des Flugzeugs, nachdem es sich um die Längsachse um den Winkel α nach rechts gedreht hat (vom Pilot aus gesehen).

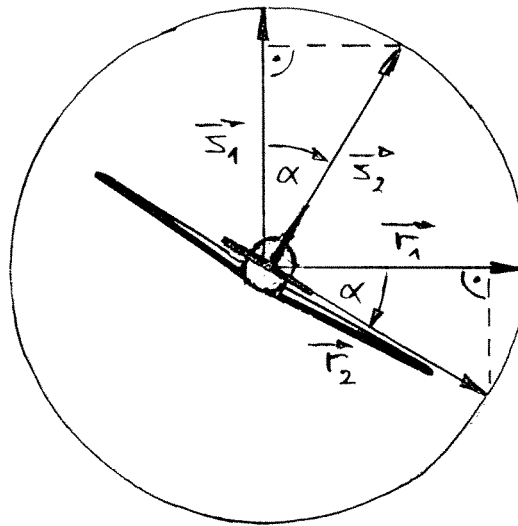


Abb. 10 Berechnung der Lage nach Drehung um Längsachse

Der gesuchte Zusammenhang zwischen $\vec{n}_1, \vec{r}_1, \vec{s}_1$ und $\vec{n}_2, \vec{r}_2, \vec{s}_2$ kann aus Abb. 10 herausgelesen werden:

$$\vec{n}_2 = \vec{n}_1 \quad (7)$$

$$\vec{r}_2 = \cos \alpha \cdot \vec{r}_1 - \sin \alpha \cdot \vec{s}_1 \quad (8)$$

$$\vec{s}_2 = \sin \alpha \cdot \vec{r}_1 + \cos \alpha \cdot \vec{s}_1 \quad (9)$$

Gleichartige Zusammenhänge lassen sich für die neue Lage nach Drehung um die Quer-, bzw. Hochachse aufstellen.

Kapitel 4

Programmierung

Die Grobstruktur des Programms

Der KS Flight Simulator besteht aus fünf Unterprogrammen:

- **Unterprogramm Landschaft:** Eingabe der Punkte und Kanten der Landschaft.
- **Unterprogramm Start:** Eingabe der Startposition und der Startlage des Flugzeugs.
- **Unterprogramm Bildberechnung:** Berechnung der Bildpunkte (vgl. Kap. 3).
- **Unterprogramm Bildanzeige:** Darstellung der Bildkanten auf dem Monitor.
- **Unterprogramm Steuerung:** Abfrage, ob via Tastatur ein Steuerbefehl eingegeben wurde. Wenn ja, Berechnung der neuen Position und der neuen Lage des Flugzeugs (vgl. Kap. 3).

Das Hauptprogramm des KS Flight Simulator hat in Pseudocode die folgende Struktur:

```
CALL Landschaft
CALL Start
DO
    CALL Bildberechnung
    CALL Bildanzeige
    CALL Steuerung
LOOP
```

Die globalen Variablen

Im KS Flight Simulator sind alle wichtigen Variablen global vereinbart. Dies ermöglicht den einzelnen Schülergruppen, ihre Unterprogramme unabhängig voneinander und doch kompatibel zueinander zu entwickeln. Die globalen Variablen sind im folgenden nach Datentyp geordnet aufgelistet.

- **Zahlen:**

D	Distanz (= Abstand Augpunkt zu Bildebene)
Alpha, Beta, Gamma	Drehwinkel um Längs-, Quer-, Hochachse
Vor	Verschiebung in Richtung Längsachse

- **Eindimensionale Arrays:**

M	Mittelpunkt der Bildebene; $M(1), M(2), M(3) = x, y, z$ -Koordinate von M
N, R, S	Einheitsvektoren in Richtung Längs-, Quer-, Hochachse

- **Zweidimensionale Arrays:**

P	Punkte der Landschaft; $P(I, 1), P(I, 2), P(I, 3) = x, y, z$ -Koordinate von Punkt Nr. I, $I = 1, \dots, 14$
K	Kanten der Landschaft; $K(J, 1), K(J, 2) =$ Nummern der beiden Punkte, welche die Kante Nr. J bilden, $J = 1, \dots, 17$
B	Bildpunkte; $B(I, 1), B(I, 2) = u, v$ -Koordinate vom Bild von Punkt Nr. I, $I = 1, \dots, 14$

Das Rahmenprogramm

Der Autor dieses Berichts entwarf als Projektleiter ein lauffähiges Rahmenprogramm, welches den Schülern als Projektteilnehmer als Ausgangspunkt für die eigene Arbeit diene.

Die globalen Variablen sind – soweit nötig – festgelegt, die Landschaft- und Startdaten sind eingegeben. Alle anderen Teile sind noch nicht erstellt.

Als Programmiersprache diene TrueBASIC. Im folgenden ist der gesamte Code des Rahmenprogramms abgedruckt.

```

REM *****
REM
REM KS Flight Simulator
REM
REM Kantonsschule Baden/Klasse 3bR/ADM
REM 2. Semester Schuljahr 1996/97
REM
REM Rahmenprogramm
REM
REM*****

REM **** Einstellungen ****

OPTION ANGLE DEGREES

REM **** Variablen ****

DIM M(1 TO 3)
DIM N(1 TO 3)
DIM R(1 TO 3)
DIM S(1 TO 3)
DIM A(1 TO 3)
DIM P(1 TO 14, 1 TO 3)
DIM K(1 TO 17, 1 TO 2)
DIM B(1 TO 14, 1 TO 2)

REM **** Konstanten ****

LET D = 0.5
SET WINDOW -D, D, -D, D/2

REM **** Unterprog. Landschaft ****

SUB Landschaft

    REM Punkte Piste

    LET P(1, 1) = -15
    LET P(1, 2) = 0
    LET P(1, 3) = 0

```

```

LET P(2, 1) = -15
LET P(2, 2) = 400
LET P(2, 3) = 0

```

```

LET P(3, 1) = 15
LET P(3, 2) = 0
LET P(3, 3) = 0

```

```

LET P(4, 1) = 15
LET P(4, 2) = 400
LET P(4, 3) = 0

```

```

REM Punkte Hangar

```

```

LET P(5, 1) = 25
LET P(5, 2) = 60
LET P(5, 3) = 0

```

```

LET P(6, 1) = 25
LET P(6, 2) = 120
LET P(6, 3) = 0

```

```

LET P(7, 1) = 55
LET P(7, 2) = 60
LET P(7, 3) = 0

```

```

LET P(8, 1) = 55
LET P(8, 2) = 120
LET P(8, 3) = 0

```

```

LET P(9, 1) = 25
LET P(9, 2) = 60
LET P(9, 3) = 10

```

```

LET P(10, 1) = 25
LET P(10, 2) = 120
LET P(10, 3) = 10

```

```

LET P(11, 1) = 55
LET P(11, 2) = 60
LET P(11, 3) = 10

```

```

LET P(12, 1) = 55
LET P(12, 2) = 120
LET P(12, 3) = 10

```

```

LET P(13, 1) = 40
LET P(13, 2) = 60
LET P(13, 3) = 20

```

```

LET P(14, 1) = 40
LET P(14, 2) = 120
LET P(14, 3) = 20

```

```

REM Kanten Piste

```

```

LET K(1, 1) = 1
LET K(1, 2) = 2

```

```
LET K(2, 1) = 3
LET K(2, 2) = 4

REM Kanten Hangar

LET K(3, 1) = 5
LET K(3, 2) = 6

LET K(4, 1) = 7
LET K(4, 2) = 8

LET K(5, 1) = 5
LET K(5, 2) = 7

LET K(6, 1) = 6
LET K(6, 2) = 8

LET K(7, 1) = 5
LET K(7, 2) = 9

LET K(8, 1) = 6
LET K(8, 2) = 10

LET K(9, 1) = 7
LET K(9, 2) = 11

LET K(10, 1) = 8
LET K(10, 2) = 12

LET K(11, 1) = 9
LET K(11, 2) = 10

LET K(12, 1) = 11
LET K(12, 2) = 12

LET K(13, 1) = 9
LET K(13, 2) = 13

LET K(14, 1) = 11
LET K(14, 2) = 13

LET K(15, 1) = 10
LET K(15, 2) = 14

LET K(16, 1) = 12
LET K(16, 2) = 14

LET K(17, 1) = 13
LET K(17, 2) = 14

END SUB

REM **** Unterprogramm Start ****

SUB Start

LET M(1) = 0
LET M(2) = -10
LET M(3) = 2

LET N(1) = 0
LET N(2) = 1
LET N(3) = 0

LET R(1) = 1
LET R(2) = 0
LET R(3) = 0

LET S(1) = 0
LET S(2) = 0
LET S(3) = 1

END SUB

REM **** Unterprog. Bildberechnung ****

SUB Bildberechnung

END SUB

REM **** Unterprog. Bildanzeige ****

SUB Bildanzeige

CLEAR

REM Anzeige Koordinatenachsen
PLOT LINES: -D, 0; D, 0
PLOT LINES: 0, -D/2; 0, D/2

REM Anzeige Instrumentenbrett
PLOT AREA: -D, -D; D, -D; D, -D/2; -D, -D/2

END SUB

REM **** Unterprogramm Steuerung ****

SUB Steuerung

END SUB

REM **** Hauptprogramm ****

CALL Landschaft
CALL Start

DO
    CALL Bildberechnung
    CALL Bildanzeige
    CALL Steuerung
LOOP

END
```

Die Unterprogramme Bildberechnung und Bildanzeige – Version 1

Der erste Schritt in der Programmierarbeit der Schüler bestand darin, die Unterprogramme Bildberechnung und Bildanzeige in TrueBASIC zu implementieren. Die Bildberechnung erfolgt mit den Formeln aus Kap. 3. Abb. 11 zeigt das Ergebnis dieser Programmierarbeit: Die Cockpit-Sicht des Piloten vor dem Start.

```

REM **** Unterprogramm Bildberechnung ****

SUB Bildberechnung
  REM Berechnung Bildpunkte
  DIM Q(1 TO 3)
  FOR I = 1 TO 14
    LET Q(1) = P(I, 1) - M(1)
    LET Q(2) = P(I, 2) - M(2)
    LET Q(3) = P(I, 3) - M(3)
    LET T = D / ( D + Q(1)*N(1) + Q(2)*N(2) + Q(3)*N(3) )
    LET B(I, 1) = T * ( Q(1)*R(1) + Q(2)*R(2) + Q(3)*R(3) )
    LET B(I, 2) = T * ( Q(1)*S(1) + Q(2)*S(2) + Q(3)*S(3) )
  NEXT I
END SUB

REM **** Unterprogramm Bildanzeige ****

SUB Bildanzeige
  CLEAR
  PLOT LINES: -D, 0; D, 0
  PLOT LINES: 0, -D/2; 0, D/2
  PLOT AREA: -D, -D; D, -D; D, -D/2; -D, -D/2
  REM Anzeige Bildkanten
  FOR J = 1 TO 17
    LET I1 = K(J, 1)
    LET I2 = K(J, 2)
    PLOT LINES: B(I1, 1), B(I1, 2); B(I2, 1), B(I2, 2)
  NEXT J
END SUB

```

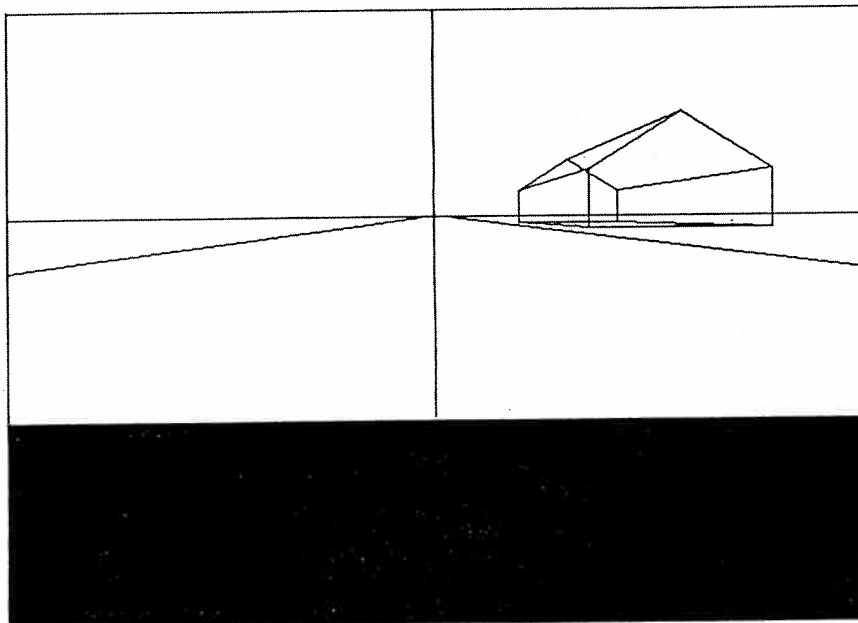


Abb. 11 Erstes Ergebnis der Programmierarbeit

Probleme mit der Darstellung der Kanten

Mit dem bisher entwickelten Programm unternahmen die Schüler erste Erkundungsflüge. Indem sie die Daten im Unterprogramm Start änderten, konnten sie Standbilder der Landschaft aus jeder beliebigen Flugposition und Fluglage schießen. Dabei stiessen sie bald auf Ungereimtheiten. Beispielsweise tauchten bereits nach 70 m Fahrt auf der Piste Linien über dem Horizont auf, die nie und nimmer dahingehörten (Abb. 12). Was ist schiefgelaufen?

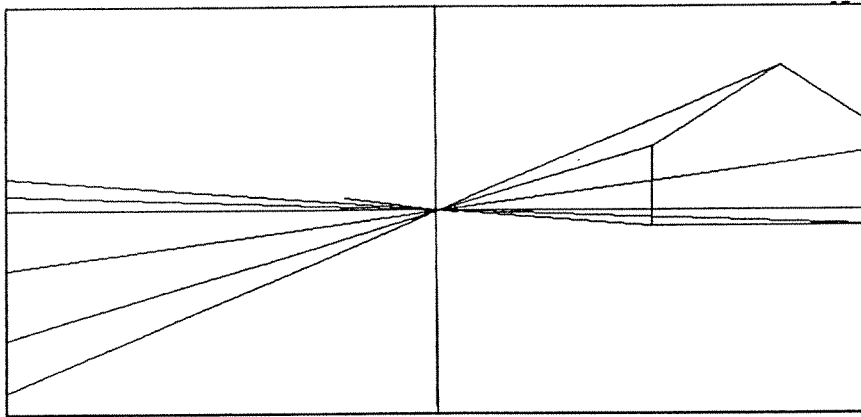


Abb. 12 Was ist schiefgelaufen?

Die Ungereimtheiten stecken im Unterprogramm Bildanzeige: Es zeigt nicht nur diejenigen Teile der Landschaft an, die **vor** dem Piloten liegen, sondern auch die **hinter** ihm liegenden!

Abb. 13 zeigt die Problematik. Der Punkt P_1 liegt **vor** dem Augpunkt A, der Punkt P_2 liegt **hinter** dem Augpunkt A. P_1^* und P_2^* sind die zugehörigen Bildpunkte. Q ist derjenige Punkt auf der Kante P_1P_2 , der genau auf Augenhöhe liegt.

Das Unterprogramm Bildanzeige stellt die Kante P_1P_2 als Strecke $P_1^*P_2^*$ dar:

$$\overrightarrow{OP_1^*} + w \cdot \overrightarrow{P_1^*P_2^*}, 0 \leq w \leq 1.$$

Abb. 13 zeigt, dass diese Darstellung falsch ist. Von A aus gesehen ist von der Kante P_1P_2 nur der Teil P_1Q sichtbar. Das Bild von P_1Q ist die von P_1^* aus nach rechts laufende Halbgerade. Die korrekte Darstellung der Kante P_1P_2 lautet daher:

$$\overrightarrow{OP_1^*} + w \cdot \overrightarrow{P_2^*P_1^*}, w \geq 0.$$

Ob ein Objektpunkt P vor oder hinter dem Aufpunkt liegt, kann wie folgt bestimmt werden: Im Unterprogramm Bildberechnung wird für jeden Objektpunkt P der zugehörige Parameterwert t berechnet, so dass

$$\overrightarrow{OP^*} = \overrightarrow{OA} + t \cdot \overrightarrow{AP}.$$

- Ist $t > 0$, so liegt der Objektpunkt P vor dem Augpunkt A.
- Ist $t < 0$, so liegt der Objektpunkt P hinter dem Augpunkt A.

Ist t nicht definiert, so liegt der Objektpunkt P genau auf Höhe des Augpunkts A.

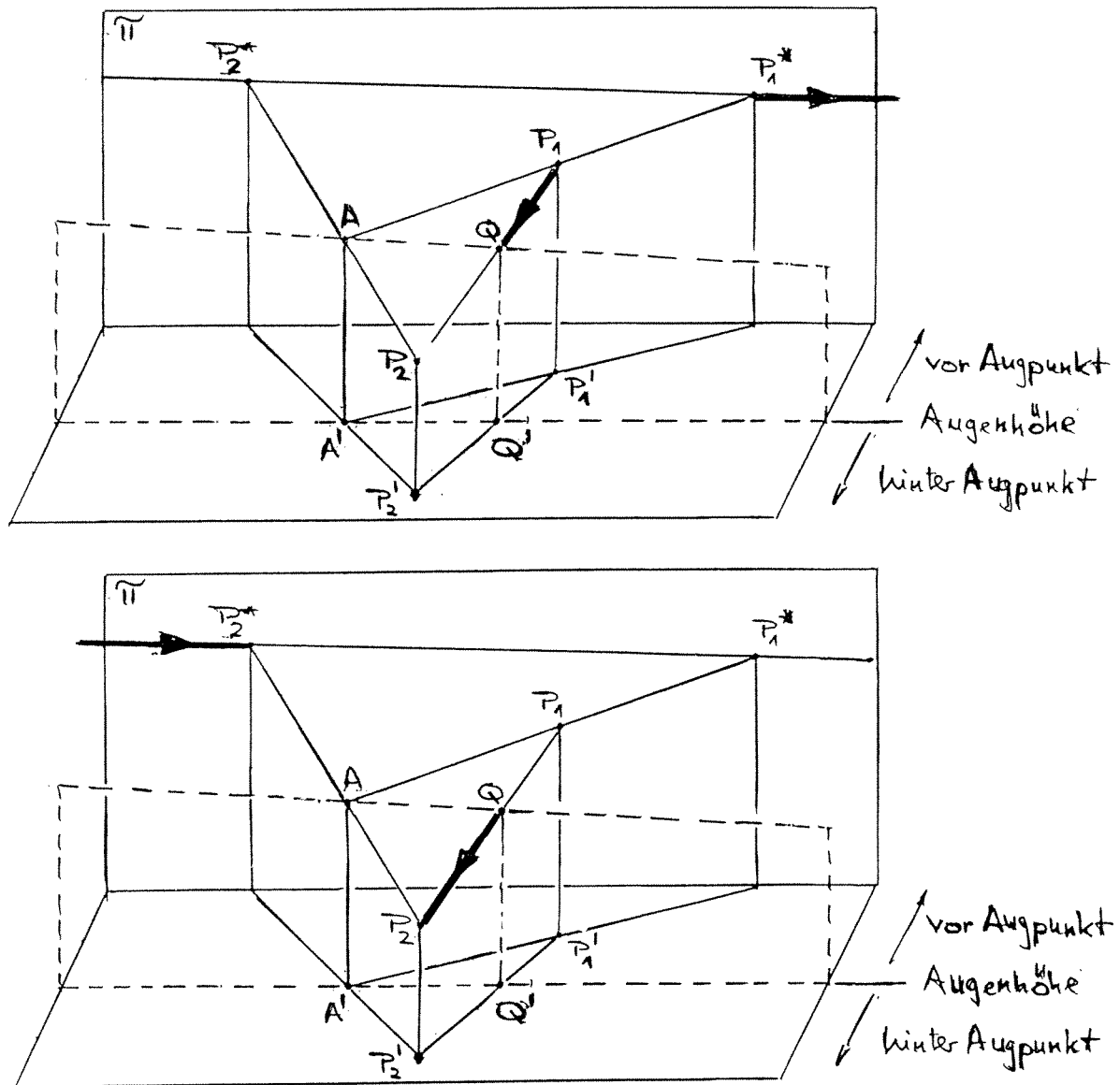


Abb. 13 Projektion der Kante P_1P_2 , die teils vor und teils hinter dem Augpunkt A liegt. Der von A aus sichtbare Teil P_1Q wird auf die rechts von P_1^* liegende Halbgerade abgebildet (oben), der nicht sichtbare Teil QP_2 wird auf die links von P_2^* liegende Halbgerade abgebildet (unten).

Die Unterprogramme Bildberechnung und Bildanzeige – Version 2

In Version 2 sind die Mängel bei der Bilddarstellung aus Version 1 behoben.

- In Version 2 wird ein eindimensionaler Array L eingeführt, welcher die Lage der Punkte in bezug auf den Augpunkt festhält. $L(I)$ ist 1, -1 oder 0, je nachdem ob der Punkt Nr. I vor, hinter, oder auf gleicher Höhe wie der Augpunkt liegt. $L(I)$ wird im Unterprogramm Bildberechnung ermittelt.
- In Version 2 werden die Kanten mit Hilfe von L und den obigen Überlegungen durch das Unterprogramm Bildanzeige korrekt gezeichnet.

Abb. 14 wurde mit Version 2 erstellt. Abb. 14 gibt die Cockpitsicht von Abb. 12 korrekt wieder.

```
REM **** Unterprogramm Bildberechnung ****
```

```
SUB Bildberechnung
```

```
    DIM Q(1 TO 3)
```

```
    DIM L(1 TO 17)
```

```
    FOR I = 1 TO 14
```

```
        LET Q(1) = P(I, 1) - M(1)
```

```
        LET Q(2) = P(I, 2) - M(2)
```

```
        LET Q(3) = P(I, 3) - M(3)
```

```
        IF D + Q(1)*N(1) + Q(2)*N(2) + Q(3)*N(3) = 0 THEN
```

```
            LET L(I) = 0
```

```
        ELSE
```

```
            LET T = D / ( D + Q(1)*N(1) + Q(2)*N(2) + Q(3)*N(3) )
```

```
            LET B(I, 1) = T * ( Q(1)*R(1) + Q(2)*R(2) + Q(3)*R(3) )
```

```
            LET B(I, 2) = T * ( Q(1)*S(1) + Q(2)*S(2) + Q(3)*S(3) )
```

```
            IF T > 0 THEN
```

```
                LET L(I) = 1
```

```
            ELSE
```

```
                LET L(I) = -1
```

```
            END IF
```

```
        END IF
```

```
    NEXT I
```

```
END SUB
```

```
REM **** Unterprogramm Bildanzeige ****
```

```
SUB Bildanzeige
```

```
    CLEAR
```

```
    PLOT LINES: -D, 0; D, 0
```

```
    PLOT LINES: 0, -D/2; 0, D/2
```

```
    PLOT AREA: -D, -D; D, -D; D, -D/2; -D, -D/2
```

```
    FOR J = 1 TO 17
```

```
        LET I1 = K(J, 1)
```

```
        LET I2 = K(J, 2)
```

```
        IF L(I1) = 1 AND L(I2) = 1 THEN
```

```
            PLOT LINES: B(I1, 1), B(I1, 2); B(I2, 1), B(I2, 2)
```

```
        END IF
```

```
        IF L(I1) = 1 AND L(I2) = -1 THEN
```

```
            LET U = B(I1, 1) + 10000 * ( B(I1, 1) - B(I2, 1) )
```

```
            LET V = B(I1, 2) + 10000 * ( B(I1, 2) - B(I2, 2) )
```

```
            PLOT LINES: B(I1, 1), B(I1, 2); U, V
```

```
        END IF
```

```
        IF L(I1) = -1 AND L(I2) = 1 THEN
```

```
            LET U = B(I2, 1) + 10000 * ( B(I2, 1) - B(I1, 1) )
```

```
            LET V = B(I2, 2) + 10000 * ( B(I2, 2) - B(I1, 2) )
```

```
            PLOT LINES: B(I2, 1), B(I2, 2); U, V
```

```
        END IF
```

```
    NEXT J
```

```
END SUB
```

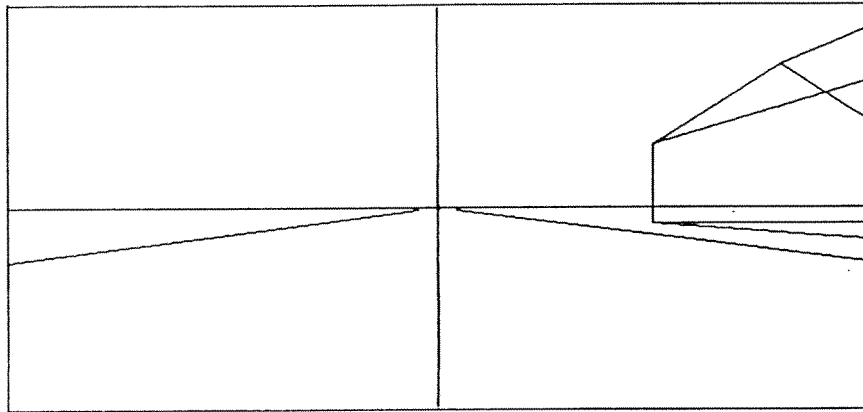


Abb. 14 Korrekte Wiedergabe der Cockpit-Sicht von Abb. 12

Das Unterprogramm Steuerung

Die Steuerung im KS Flight Simulator erfolgt interaktiv über das Zahlenfeld der Tastatur (vgl. Kap. 1).

- Mit der Taste 6, bzw. 4 wird das Flugzeug um $+2^\circ$, bzw. -2° um die Längsachse gedreht.
- Mit der Taste 2, bzw. 8 wird das Flugzeug um $+2^\circ$, bzw. -2° um die Querachse gedreht.
- Mit der Taste 0, bzw. ENTER wird das Flugzeug um $+2^\circ$, bzw. -2° um die Hochachse gedreht.
- Mit der Taste 9, bzw. 3 wird das Flugzeug um $+10$ m, bzw. -10 m entlang der Längsachse verschoben.

Den Tasten 0 bis 9 entsprechen im ASCII-Code die Nummern 48 bis 57. Die ENTER-Taste hat die ASCII-Nummer 3.

```
REM **** Unterprogramm Steuerung ****
```

```
SUB Steuerung
```

```
    REM Tastaturabfrage
```

```
    LET Alpha = 0
```

```
    LET Beta = 0
```

```
    LET Gamma = 0
```

```
    LET Taste = 0
```

```
    IF KEY INPUT THEN
```

```
        GET KEY Taste
```

```
    END IF
```

```
    IF Taste = 54 THEN
```

```
        LET Alpha = 2
```

```
    END IF
```

```
    IF Taste = 52 THEN
```

```
        LET Alpha = -2
```

```
    END IF
```

```
    IF Taste = 50 THEN
```

```
        LET Beta = 2
```

```
    END IF
```

```
IF Taste = 56 THEN
    LET Beta = -2
END IF
```

```
IF Taste = 48 THEN
    LET Gamma = 2
END IF
IF Taste = 3 THEN
    LET Gamma = -2
END IF
```

```
IF Taste = 57 THEN
    LET Vor = 10
END IF
IF Taste = 51 THEN
    LET Vor = -10
END IF
```

REM Berechnung neue Lage

```
DIM NNeu(1 TO 3)
DIM RNeu(1 TO 3)
DIM SNeu(1 TO 3)
```

```
FOR I = 1 TO 3
    LET RNeu(I) = COS(Alpha)*R(I) - SIN(Alpha)*S(I)
    LET SNeu(I) = SIN(Alpha)*R(I) + COS(Alpha)*S(I)
    LET R(I) = RNeu(I)
    LET S(I) = SNeu(I)
NEXT I
```

```
FOR I = 1 TO 3
    LET NNeu(I) = COS(Beta)*N(I) + SIN(Beta)*S(I)
    LET SNeu(I) = -SIN(Beta)*N(I) + COS(Beta)*S(I)
    LET N(I) = NNeu(I)
    LET S(I) = SNeu(I)
NEXT I
```

```
FOR I = 1 TO 3
    LET NNeu(I) = COS(Gamma)*N(I) - SIN(Gamma)*R(I)
    LET RNeu(I) = SIN(Gamma)*N(I) + COS(Gamma)*R(I)
    LET N(I) = NNeu(I)
    LET R(I) = RNeu(I)
NEXT I
```

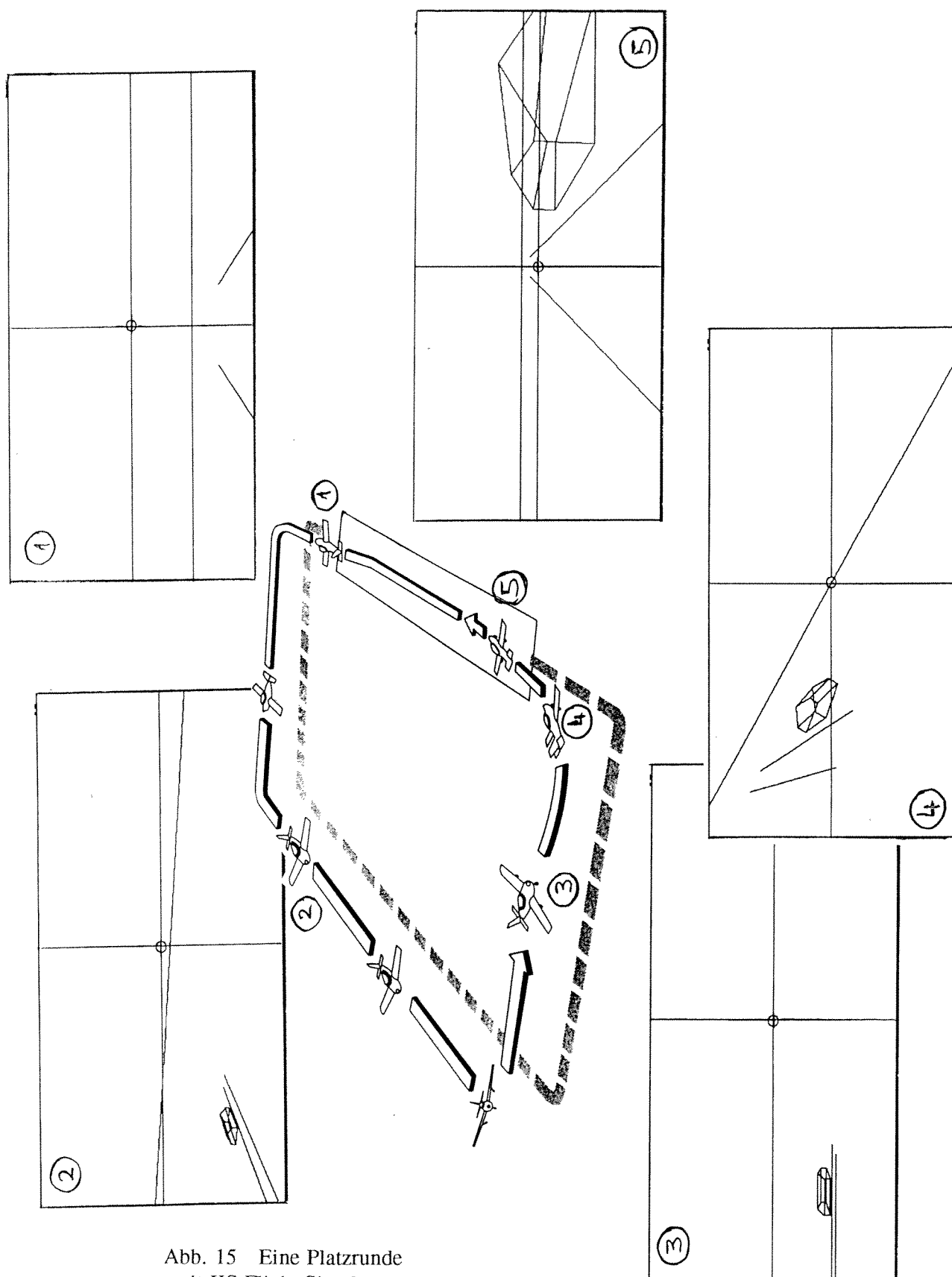
REM Berechnung neue Position

```
FOR I = 1 TO 3
    LET M(I) = M(I) + Vor*N(I)
NEXT I
```

END SUB

Eine Platzrunde

Der erste Flug, egal ob real oder simuliert, ist stets eine sog. Platzrunde (Abb. 15).



Kapitel 5

Ablauf des Projekts

Das Umfeld

Der Autor diese Berichts führte das Projekt KS Flight Simulator im Schuljahr 1996/97 an der Kantonschule Baden in einer 3. Klasse¹ des Typus C durch.

Die 17 teilnehmenden Schüler² brachten aus dem Mathematikunterricht umfassende Kenntnisse aus der Vektorgeometrie mit, aus dem Unterricht in Angewandter Mathematik Fertigkeiten in der zeichnerischen Darstellung räumlicher Objekte und im Umgang mit der Programmiersprache TrueBASIC.

Das ganze Projekt erstreckte sich über 9 Wochen. Pro Woche wurden dafür die zwei Lektionen in Angewandter Mathematik und eine Lektion in Mathematik eingesetzt, insgesamt also 27 Lektionen.

Für die Programmierarbeit standen 12 Macintosh-Computer zur Verfügung.

1. Woche: Projektstart, Einführung ins reale Fliegen, Einführung ins simulierte Fliegen

Der Lehrer informiert die Schüler über das bevorstehende Projekt. Er erläutert die Zielsetzung, beschreibt die Arbeitsform und verteilt den Zeitplan.

Ein Schüler der Klasse hat die fliegerische Vorschulung für Militärpilotenanwärter absolviert. Bestandteil dieser Vorschulung sind rund ein Dutzend Übungsflüge mit einem Fluglehrer. Dieser Schüler vermittelt seinen Kollegen die Grundlagen des Fliegens, in etwa so wie diese in Kap. 1 zusammengestellt sind.

Der Lehrer stellt das kommerzielle Flugsimulationsprogramm «MS Flight Simulator» kurz vor. Die Schüler erhalten eine ganze Lektion Zeit, selbst mit diesem Programm zu spielen.

2. Woche: Konzeption des KS Flight Simulator, Grundlagen der Zentralprojektion

Die Schüler diskutieren in Gruppen, aus welchen Teilen ein Flugsimulationsprogramm besteht und wie diese zusammenwirken. Sie halten ihre Überlegungen in Ablaufdiagrammen fest. Es entsteht die Grobstruktur des Programms (vgl. Kap. 4).

Der Lehrer konkretisiert für die weitere Arbeit die wichtigsten Komponenten des KS Flight Simulator. Die Vorgaben an die Schüler decken sich mit den Ausführungen in Kap. 2.

Die Schüler reaktivieren ihre Kenntnisse über Zentralprojektionen. Sie lösen dazu zwei Aufgaben. Sie sind als Aufgaben 1 und 2 im Anhang dieses Berichts abgedruckt. (Perspektive Darstellungen sind in dieser Klasse im Mathematikunterricht nicht speziell thematisiert worden. Das Thema wurde aber da und dort – zum Beispiel in der Vektorgeometrie – angeschnitten. Zudem sind einige Elemente – zum Beispiel Fluchtpunkte – vom Zeichenunterricht her bekannt.)

¹ 12. Schuljahr, eineinhalb Jahre vor der Matur.

² Schülerinnen gab es in der Klasse keine.

3. und 4. Woche: Formeln zur Berechnung der Cockpit-Sicht, Drehungen im Raum

Der Lehrer erstellt zusammen mit der Klasse eine Liste mit Eigenschaften der Zentralprojektion und mit Tips für das perspektive Zeichnen.

Die Schüler stossen anhand weiterer Aufgaben bis zu den allgemeinen Formeln für die Berechnung der Cockpit-Sicht vor (vgl. Kap. 3). Zwei dieser Aufgaben sind als Aufgaben 3 und 4 im Anhang dieses Berichts abgedruckt.

Die Schüler leiten selbständig die Zusammenhänge zwischen alter und neuer Lage des Flugzeugs nach einer Drehung um eine der drei Hauptachsen her (vgl. Kap. 3).

5. Woche: Einstieg in die Programmierarbeit

Der Lehrer verteilt und kommentiert das von ihm erstellte Rahmenprogramm (vgl. Kap. 4). Er teilt die Schüler in Viererteams auf. Die bekanntermassen guten Programmierer der Klasse werden auf die Teams verteilt. Die Teams erhalten den Auftrag, die Unterprogramme Bildberechnung und Bildanzeige zu implementieren und auszutesten. Allfällig auftretende Probleme sollen möglichst genau lokalisiert werden. Es entsteht Version 1 des KS Flight Simulator (vgl. Kap. 4).

6. Woche: Zwischenhalt

Viel zu reden und viel zu erklären geben die in Version 1 auf dem Monitor ab und zu inkorrekt dargestellten Kanten der Landschaft (vgl. Kap. 4). Erst die genaue Analyse einer problematischen Situation schafft Klarheit. Die Teams verarbeiten die neuen Erkenntnisse zu Version 2 des KS Flight Simulator (vgl. Kap. 4).

Nun ist es Zeit, etwas mit dem KS Flight Simulator zu spielen. Die einen fliegen eine Platzrunde (vgl. Kap. 4), die anderen einen Looping. Einige Teams machen sich bereits daran, eine einfache Steuerung zu implementieren. Es entsteht Version 3 des KS Flight Simulator (vgl. Kap. 4).

7. Woche: Prüfung

Die Prüfung ist im Anhang abgedruckt. Für die 4 Aufgaben standen 3 Lektionen Zeit zur Verfügung. Die Aufgaben 1 und 2 decken den Mathematikteil, die Aufgabe 3 den Informatikteil der Wochen 1 – 6 ab. Die Aufgabe 4 verlangt Physikkenntnisse, welche im Rahmen des Projekts KS Flight Simulator nicht thematisiert werden, den Schülern aber aus dem Physikunterricht bekannt sind.

Als bestes Resultat erzielen drei Schüler die Note 6, als schlechtestes Resultat zwei Schüler die Note 2.5. Das Arithmetische Mittel der Noten liegt bei 4.4.

8. Woche: Abschluss der Programmierarbeit

Inzwischen hat das Projekt KS Flight Simulator eine Eigendynamik entwickelt. Einige Schüler haben das Programm ausserhalb der Schulzeit stark ausgebaut. Hinzu kommen unter anderem neue Landschaftselemente, eine Steuerung über die Maus, eine Instrumentenanzeige und Motorengeräusch.

9. Woche: Projektabschluss

Als Projektabschluss schaut sich die Klasse gemeinsam den Film «Top Gun» an.

Einige Schülerkommentare¹

«Es war interessant, die Mathematik einmal in einem etwas anderen Rahmen zu erleben. Fast jeder kennt einen Flugsimulator und hat vielleicht schon damit gespielt. Bestimmt hat auch jeder, der schon einmal virtuell geflogen ist, seine Mängel am Programm kundgetan, sei es bezüglich der Graphik oder der Bedienung. Mir wurde erst während diesem Projekt klar, was für ein Aufwand betrieben wird, um einen professionellen Flight Simulator zu programmieren, dass also solche Mängel mit grossen mathematischen oder informatischen Problemen verbunden sind.

Die Einführung ins Projekt habe ich leider verpasst, aber ich denke, das gesamte Projekt war gut durchdacht und geplant.

Die ganze Arbeit war eine gute Übung, die Vektorgeometrie noch einmal anzuwenden und die Sprache TrueBASIC etwas besser kennenzulernen, wobei gerade im Informatik-Teil die grosse Diskrepanz bezüglich der Kompetenz der verschiedenen Schüler stark zur Geltung kam.

..... Das Problem mit dem falschen und richtigen Zeichnen von Kanten haben nicht alle ganz begriffen.

Die Prüfung fand ich in Ordnung.»

«Ich denke, dass ein solches Projekt von der Schülerschaft gerne aufgenommen wird und zwar wegen seiner Vielseitigkeit. Ich finde, dass der Mathematik- und Informatik-Teil gut proportioniert waren. Was ich jedoch noch schön finden würde, wäre wenn man das Physikalische noch etwas aufrollen würde. So hätte man sich noch mit Auftrieb und Luftwiderstand beschäftigen können.

Auch gut fand ich die Einführung von Kornel ins reale Fliegen.

Alles in allem würde ich das Projekt einfach noch etwas aufblasen und mehr Zeit dafür aufwenden.»

«Das Projekt wurde sehr gut vorbereitet und strukturiert. Zuerst wurde man über das Fliegen generell informiert, was sehr interessant war. Dann wurde man mit einfachen Beispielen in das Thema eingeführt. Die Arbeit am Computer war nicht immer ganz einfach. Es war sehr nützlich, einmal hinter die Kulissen eines Computer-Flugsimulators zu schauen. Zeitlich gesehen konnten wir, so glaube ich, etwas gutes auf die Beine stellen.»

«Ein gutes Projekt, um die Nützlichkeit der Vektorrechnung zu demonstrieren. Es gab sicherlich auch manchen einen besseren Einblick in die Aufgabenstellungen der Informatik. Vielleicht hätten Sie uns selber den eigentlichen Programmablauf planen lassen sollen. Das wäre vielleicht lehrreicher gewesen, als nur die Unterprogramme zu schreiben.»

«Persönlich fand ich das Thema Flugsimulation sehr interessant. Das Projekt war gut koordiniert, der Zusammenhang zwischen der theoretischen Berechnung und der Realität/dem Flugsimulator war deutlich. Was mich ein wenig irritierte war die Arbeit am Computer. Die Gruppen waren zu gross. Ich hatte auch ein wenig Probleme mit dem Programmieren, was vom Fehlen in der 2. Klasse (wegen dem Austauschjahr) herrührte. Umso grösser war danach das Erfolgserlebnis, als unser Simulator tatsächlich einigermassen wahrheitsgetreu flog und der Hangar über den Bildschirm flitzte.»

«Die Idee mit uns einen Flugsimulator (mindestens ansatzweise) zu programmieren finde ich genial. Der FS ist ein Computerprogramm, das jeder kennt und somit wird die AM ein weiteres Mal zur wirklichen *angewandten* Mathematik.

Ich finde die Grundstruktur des Aufbaus gut.

Die einzige Kritik, die ich anbringen möchte, ist das Tempo. Für mich sind die mathematischen Teile zu langwierig behandelt worden. Wir wären weitergekommen, wenn wir mehr Zeit zum Programmieren gehabt hätten.»

¹ Einige der abgedruckten Kommentare wurden vom Autor dieses Berichts stilistisch leicht redigiert.

Sechs gute Gründe

Warum sind Flugsimulationen ein interessantes Thema für den Mathematikunterricht an Gymnasien? Im folgenden sind dafür sechs gute Gründe angeführt und erläutert.

1. Grund: Interesse

Fliegen fasziniert fast alle Leute, Junge und Alte, Männer und Frauen gleichermassen. Flugsimulatoren faszinieren – zugegebenermassen – vor allem junge Männer. Dennoch: Wann bietet sich je im Mathematikunterricht eine solch gute Startposition? Nutzen wir die Chance!

«MS Flight Simulator» ist **das** Spielprogramm schlechthin. Es wurde seit seiner Lancierung vor 15 Jahren weltweit über 10 Millionen mal verkauft. Viele Schüler und Schülerinnen haben schon selbst damit gespielt.

2. Grund: Modellbildung

Flugsimulatoren sind komplex. So komplex, dass sie in einem ersten Zugang stark vereinfacht werden müssen. Die wesentlichen Komponenten müssen herausgearbeitet und in die Sprache der Mathematik und Informatik übersetzt werden. Dabei gibt es nicht nur **den** einen Weg. Es sind eine Vielzahl von Ansätze möglich. Sie müssen diskutiert und bewertet werden. Die Modellbildungsphase ist von grundlegender Bedeutung. Nur ein gut durchdachtes, klar strukturiertes Modell hat eine Chance, auch tatsächlich realisiert zu werden.

3. Grund: Interdisziplinarität

Ein sachgerechter Zugang zum Thema Flugsimulatoren verlangt neben fliegerischem Spezialwissen gute Kenntnisse in den drei gymnasialen Fächern Physik, Mathematik und Informatik. Damit eignet sich dieses Thema ganz besonders für das Schwerpunktfach Physik und Angewandte Mathematik.

4. Grund: Räumliches Anschauungsvermögen

Die Auseinandersetzung mit Flugsimulatoren fördert das Orientierungsvermögen im Raum ganz enorm. Aus der Cockpit-Sicht des Piloten sieht die Welt ganz anders aus. Selbst Lehrer und Lehrerinnen, die jahrelang Darstellende Geometrie unterrichtet haben, dürften zunächst Mühe haben, nicht die Orientierung zu verlieren. Was ist rechts und links? Was oben und unten? Während eines Fluges purzelt so manches durcheinander. Um auch bei komplizierteren Manövern klaren Kopf zu bewahren, müssen die entsprechenden Raumbewegungen aus verschiedenen Perspektiven betrachtet und immer wieder mental durchlaufen werden.

5. Grund: Vektorrechnung

Die Vektorrechnung nimmt im Mathematikunterricht derzeit (noch) einen breiten Platz ein. Es gibt kaum eine Serie von Maturitätsaufgaben in Mathematik, welche nicht eine oder gar zwei Aufgaben zur Vektorrechnung enthalten. In diesen Aufgaben werden in der Regel aber nur mehr oder weniger standardisierte Berechnungen rund um Punkte, Geraden, Ebenen und Kugeln verlangt. Eine darüber hinausgehende Bedeutung haben die Objekte nicht.

Flugsimulatoren setzten zwei andere Akzente in der Vektorrechnung. Erstens haben die Punkte, Geraden und Ebenen eine konkrete Bedeutung wie zum Beispiel Augpunkt, Sehstrahl oder Bildebene. Zweitens werden die Rechnungen allgemein formuliert, automatisiert und dem Rechner zur x-fachen Ausführung übergeben. Die Stärke des Kalküls Vektorrechnung wird dadurch erst richtig ausgenützt.

6. Grund: Handlungsspielraum

Flugsimulatoren sind beliebig erweiterbar. Je nach Zeit, Interessenlage und Leistungsvermögen können einzeln, in Gruppen oder in der ganzen Klasse einige Aspekte vertieft oder neue Aspekte aufgegriffen werden. Die folgende Liste gibt einige Anregungen.

Physikalische Erweiterungen:

- Der Kurvenflug wird physikalisch modelliert. Eine Kurve wird mit Hilfe des Querruders geflogen. Die Querneigung und die Geschwindigkeit des Flugzeugs bestimmen den Kurvenradius.

Mathematische/algorithmische Erweiterungen:

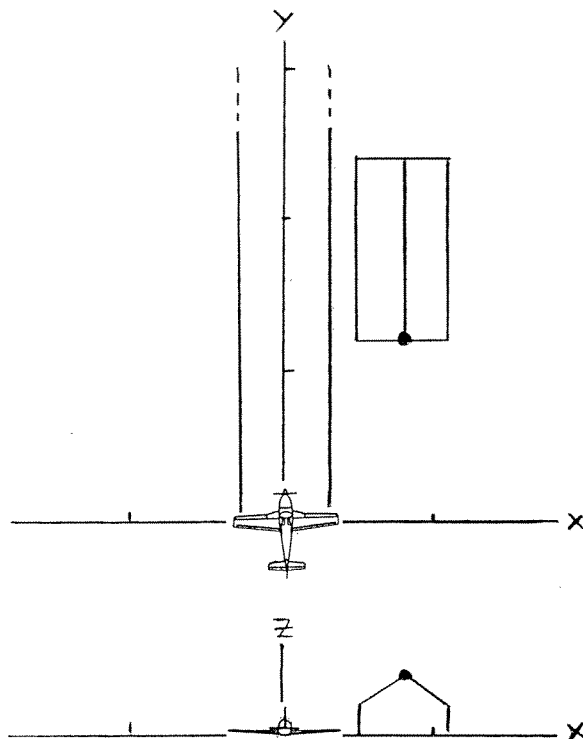
- Die Sichtbarkeit wird berücksichtigt. Verdeckte Kanten werden nicht gezeichnet.
- Die Landschaft wird zu einem Flächenmodell erweitert. Den einzelnen Flächen werden unterschiedliche Grauwerte zugeordnet.
- Die Lichtverhältnisse werden mitberücksichtigt. Flächen werden umso heller dargestellt, je stärker sie dem Licht ausgesetzt sind. Gebäude werfen Schatten.
- Es werden zusätzliche Sichtweisen eingeführt, etwa die Verfolgung eines Flugs vom Tower aus. Der Benutzer kann zwischen der Cockpit-Sicht und der Towersicht hin- und herschalten.

Informatische Erweiterungen:

- Die Instrumente zur Flugüberwachung werden hinzugefügt. Dadurch wird die Orientierung während eines Flugs wesentlich verbessert.
- Die Steuerung wird mit der Maus implementiert. Dadurch kommt man dem Steuerknüppel näher.
- Die Steuerung wird einem Autopiloten übergeben, zumindest in speziellen Flugphasen wie etwa dem Starten und Landen.

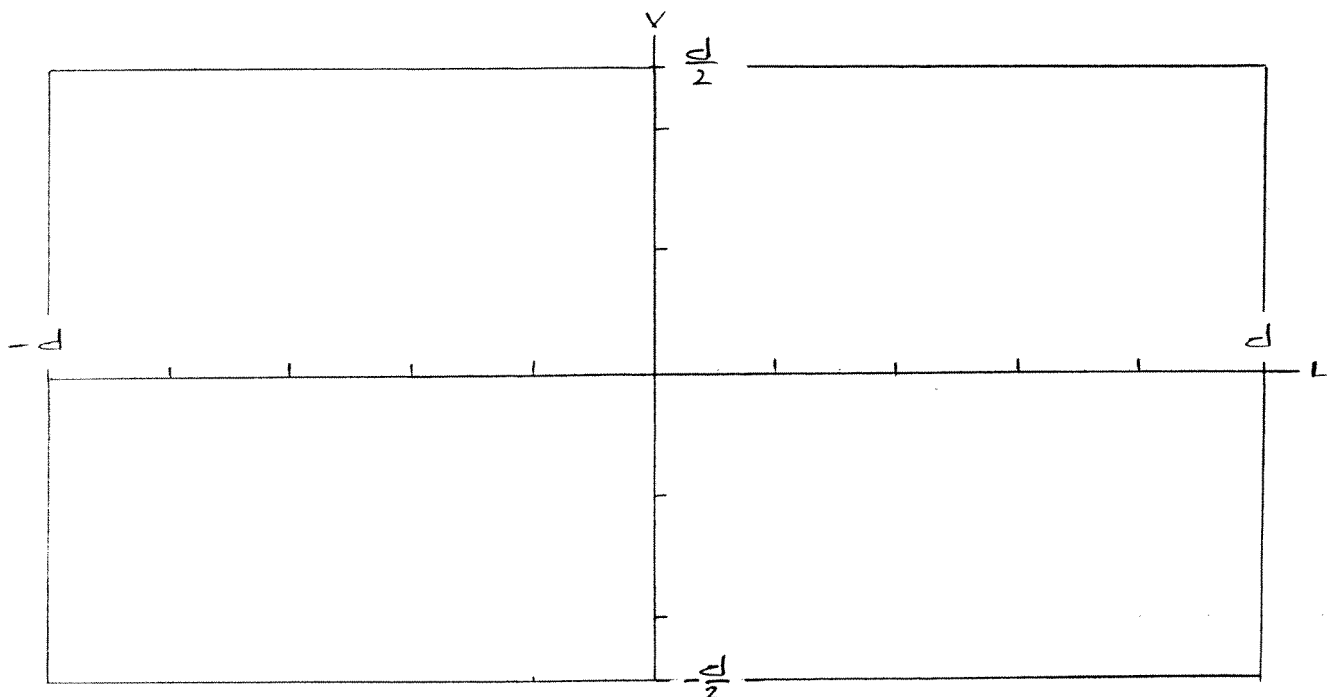
Aufgaben

Aufgabe 1

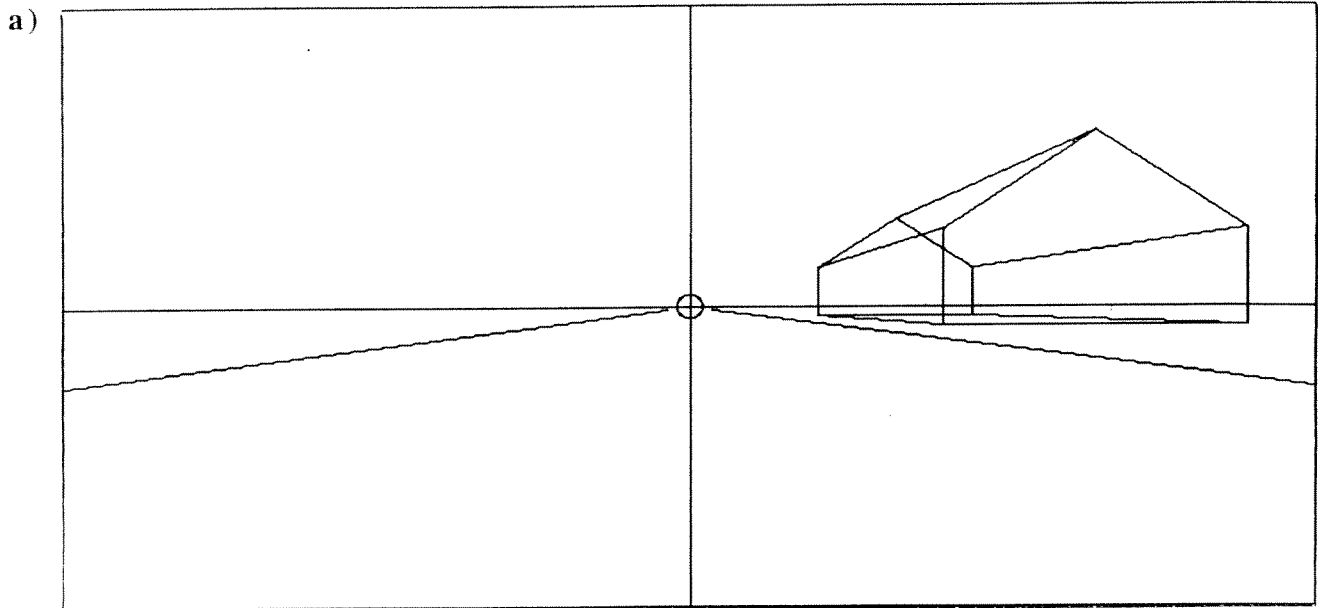


Das Flugzeug hat die Position $M: (0, 0, 2)$ und die oben gezeichnete Lage.

- Zeichnen Sie in die Vorlage die Cockpitsicht des Piloten in etwa korrekt ein.
- Geben Sie die Prinzipien an, nach denen Sie die Zeichnung a) angefertigt haben.
- Geben Sie die Lage des Flugzeugs an
- Berechnen Sie die Cockpit-Sicht des Piloten. Machen Sie das exemplarisch anhand des vorderen Punkts des Dachfirsts des Hangars.



Teillösung zu Aufgabe 1



- d) Position: $M: (0, 0, 2)$
 Lage: $\vec{n} = (0, 1, 0), \vec{r} = (1, 0, 0), \vec{s} = (0, 0, 1)$
 Bildebene: $\pi: (x, y, z) = (0, 0, 2) + u \cdot (1, 0, 0) + v \cdot (0, 0, 1)$
 Distanz: $d: 0.5$
 Augpunkt: $A: (0, -0.5, 2)$
 Objektpunkt: $P: (40, 60, 20)$
 Sehgerade: $g: (x, y, z) = (0, -0.5, 2) + t \cdot (40, 60.5, 18)$
 Bildpunkt: Schnittpunkte von g mit π

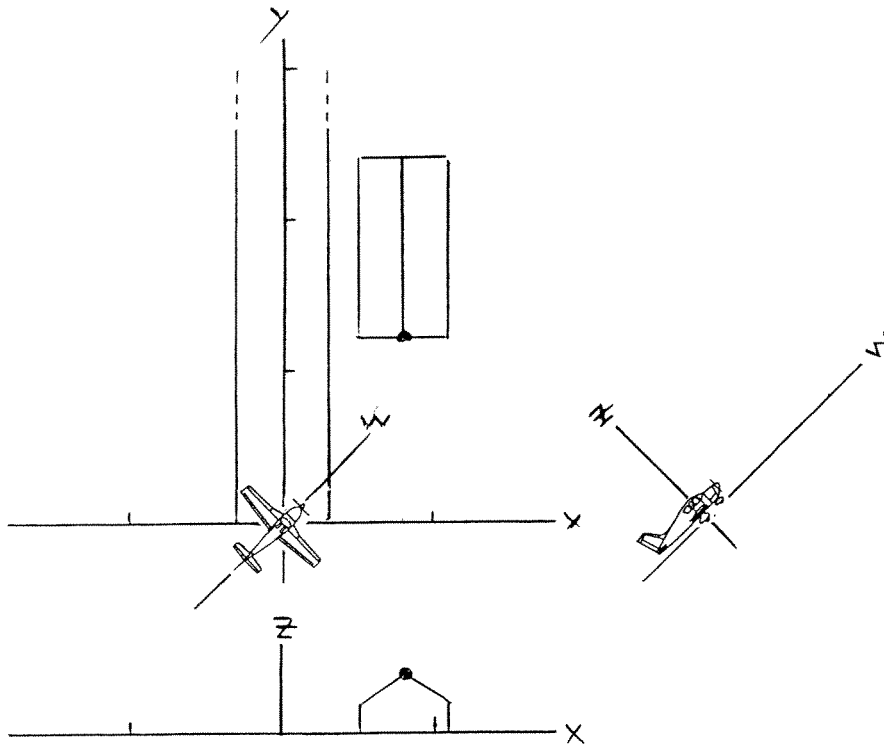
$$(0, -0.5, 2) + t \cdot (40, 60.5, 18) = (0, 0, 2) + u \cdot (1, 0, 0) + v \cdot (0, 0, 1)$$

$$\begin{aligned} 40.0 \cdot t - u &= 0.0 \\ 60.5 \cdot t &= 0.5 \\ 18.0 \cdot t - v &= 0.0 \end{aligned}$$

$$u = 0.3306, v = 0.1488$$

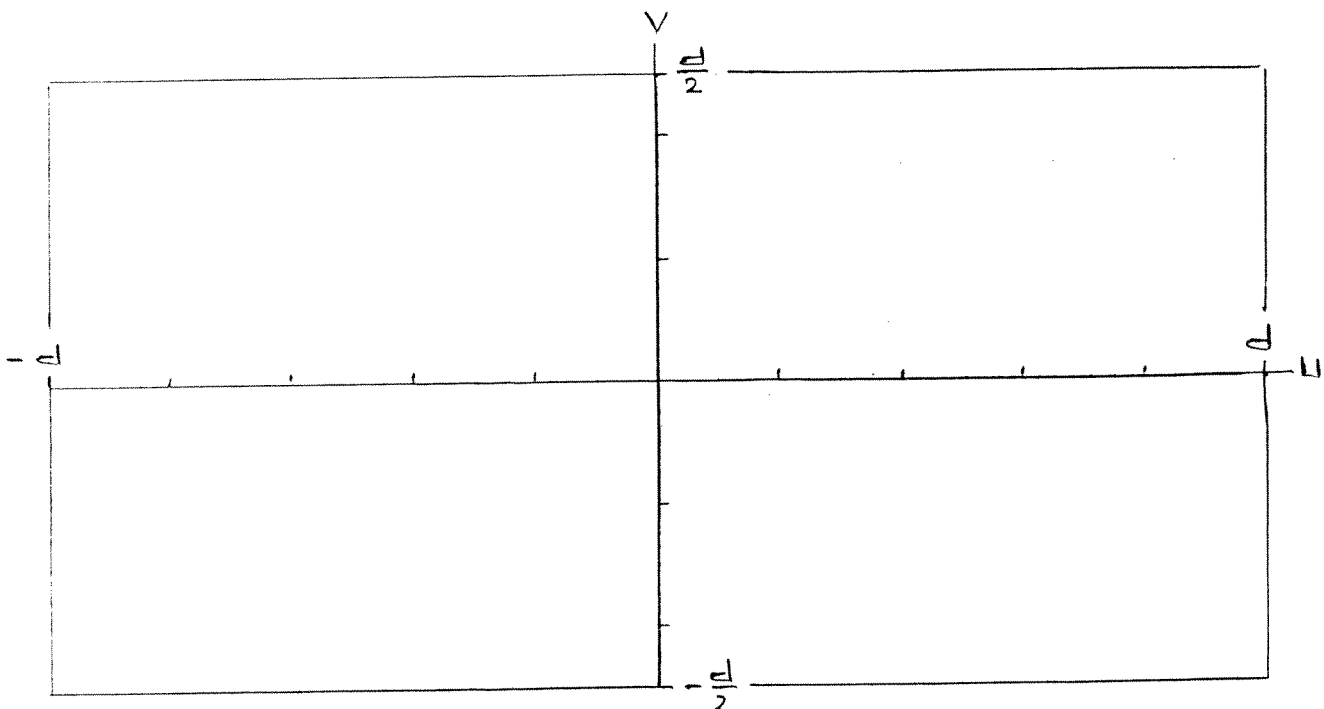
=====

Aufgabe 2

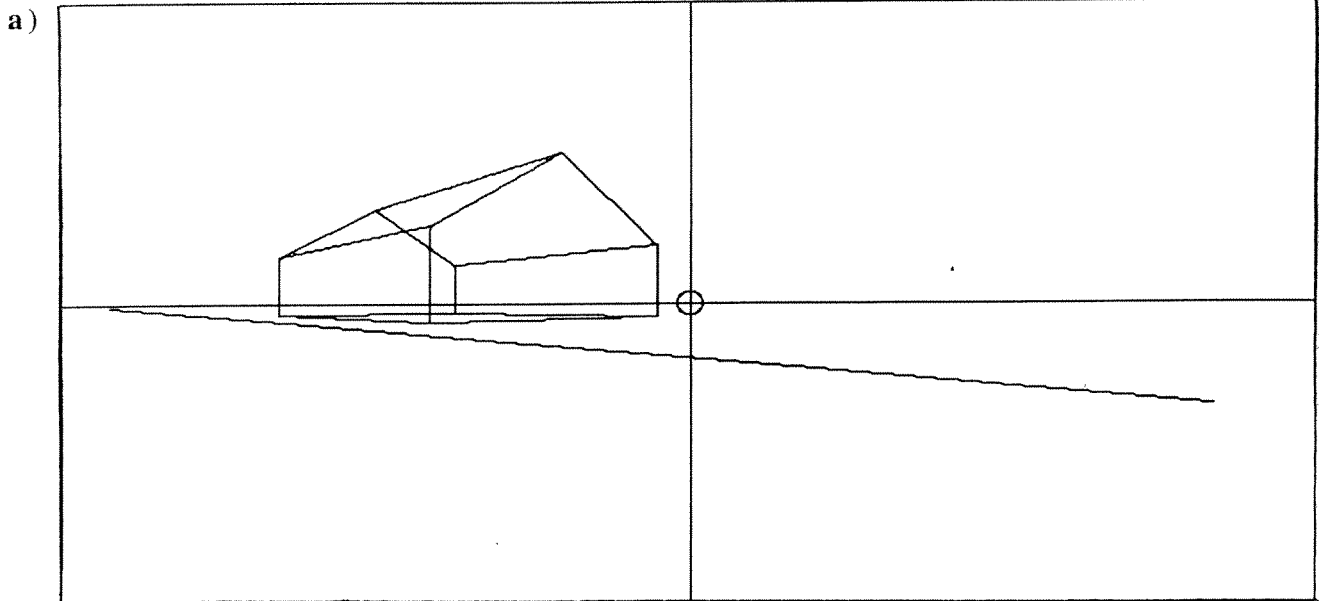


Das Flugzeug hat die Position $M: (0, 0, 2)$ und die oben gezeichnete Lage.

- Zeichnen Sie in die Vorlage die Cockpitsicht des Piloten in etwa korrekt ein.
- Geben Sie die Prinzipien an, nach denen Sie die Zeichnung a) angefertigt haben.
- Geben Sie die Lage des Flugzeugs an
- Berechnen Sie die Cockpit-Sicht des Piloten. Machen Sie das exemplarisch anhand des vorderen Punkts des Dachfirsts des Hangars.

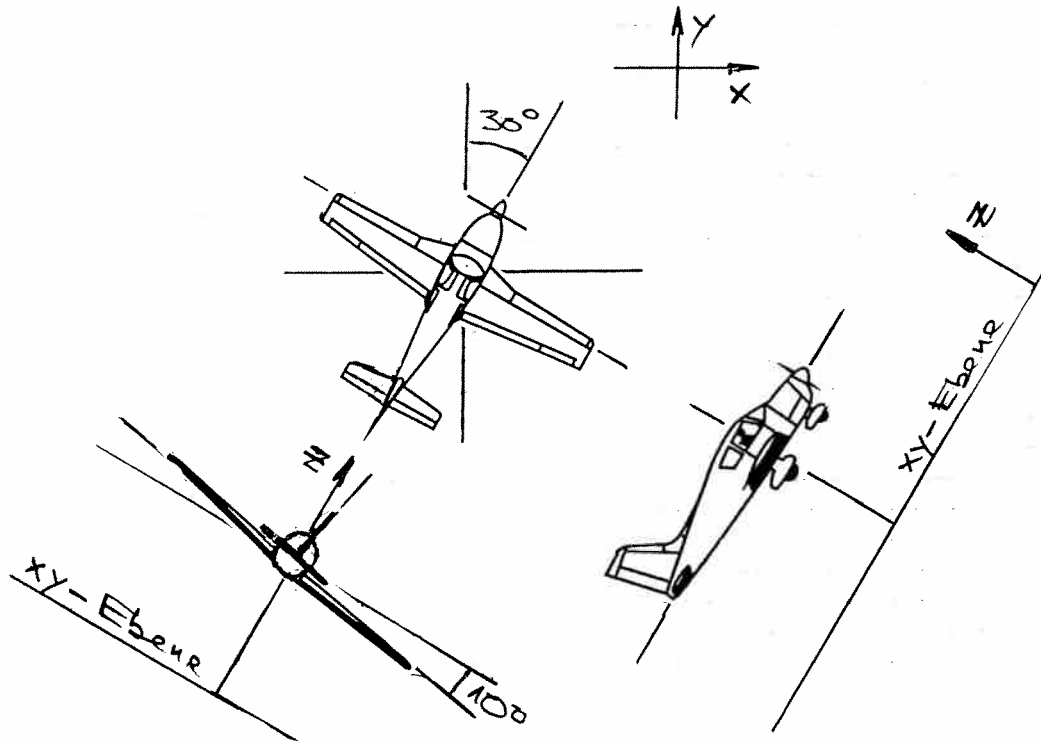


Teillösung zu Aufgabe 2



- d) Position: M: (0, 0, 2)
- Lage: $\vec{n} = (0.7071, 0.7071, 0), \vec{r} = (0.7071, -0.7071, 0), \vec{s} = (0, 0, 1)$
- Bildebene: $\pi: (x, y, z) = (0, 0, 2) + u \cdot (0.7071, -0.7071, 0) + v \cdot (0, 0, 1)$
- Distanz: d: 0.5
- Augpunkt: A: (-0.3536, -0.3536, 2)
- Objektpunkt: P: (40, 60, 20)
- Sehgerade: $g: (x, y, z) = (-0.3536, -0.3536, 2) + t \cdot (40.3536, 60.3536, 18)$
- Bildpunkt: Schnittpunkt von g mit π
- $$(-0.3536, -0.3536, 2) + t \cdot (40.3536, 60.3536, 18)$$
- $$= (0, 0, 2) + u \cdot (0.7071, -0.7071, 0) + v \cdot (0, 0, 1)$$
- $$40.3536 \cdot t - 0.7071 \cdot u = 0.3536$$
- $$60.3536 \cdot t + 0.7071 \cdot u = 0.3536$$
- $$18.0000 \cdot t + v = 0.0000$$
- $$u = -0.0982, v = -0.1262$$
- =====

Aufgabe 3

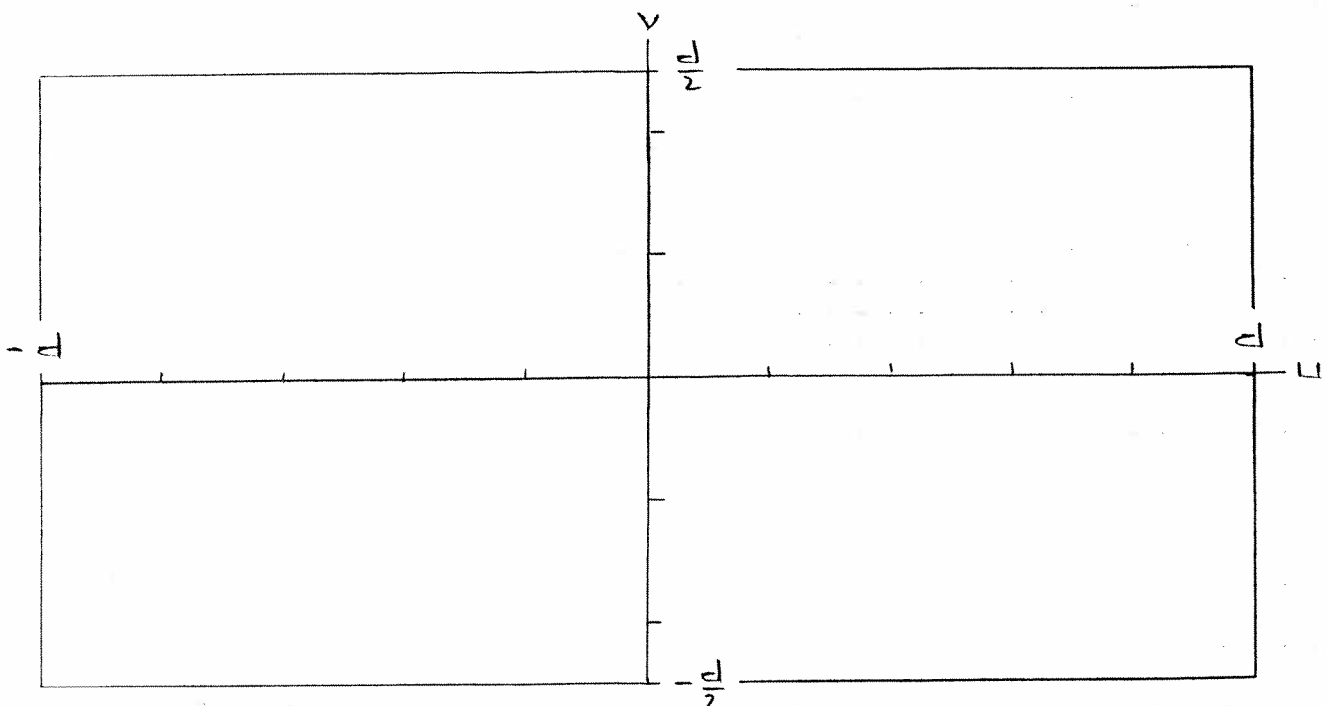


Das Flugzeug hat die Position $M: (-100, -100, 52)$ und die oben gezeichnete Lage.

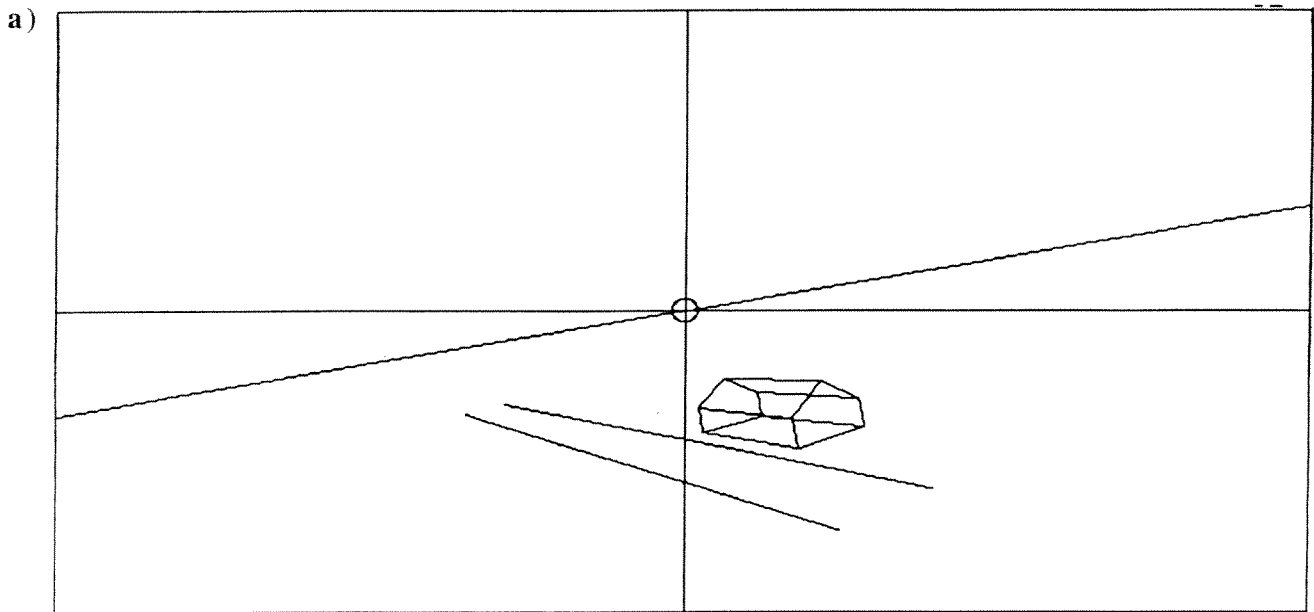
- Zeichnen Sie in die Vorlage den Geländehorizont, die Piste und den Hangar aus Sicht des Piloten.
- Zeigen Sie, dass das Flugzeug die folgende Lage hat:

$$\vec{n} = (0.5, 0.8660, 0), \vec{r} = (0.8529, -0.4924, -0.1736), \vec{s} = (0.1504, -0.0868, 0.9848)$$

- Berechnen Sie die Cockpit-Sicht des Piloten. Machen Sie das exemplarisch anhand des vorderen Punkts des Dachfirsts des Hangars.



Lösung zu Aufgabe 3



b) $\vec{n} = (\sin 30^\circ, \cos 30^\circ, 0) = (0.5, 0.8660, 0),$

$$\vec{r} = (\cos 30^\circ \cdot \cos 10^\circ, -\sin 30^\circ \cdot \cos 10^\circ, -\sin 10^\circ) = (0.8529, -0.4924, -0.1736)$$

$$\vec{s} = \vec{r} \times \vec{n} = (0.1504, -0.0868, 0.9848)$$

c) Position: M: $(-100, -100, 52)$

Lage: $\vec{n} = (0.5, 0.8660, 0), \vec{r} = (0.8529, -0.4924, -0.1736), \vec{s} = (0.1504, -0.0868, 0.9848)$

Bildebene: $\pi: (x, y, z) = \vec{OM} + u \cdot \vec{r} + v \cdot \vec{s}$
 $= (-100, -100, 52) + u \cdot (0.8529, -0.4924, -0.1736) + v \cdot (0.1504, -0.0868, 0.9848)$

Distanz: $d = 0.5$

Augpunkt: $\vec{OA} = \vec{OM} - d \cdot \vec{n} = (-100.25, -100.4330, 52)$

Objektpunkt: P: $(40, 60, 20)$

Sehgerade: $g: (x, y, z) = \vec{OA} + t \cdot \vec{AP} = (100.25, -100.4330, 52) + t \cdot (140.25, 160.4330, -32)$

Bildpunkt: Schnittpunkt von g mit π

$$\vec{OA} + t \cdot \vec{AP} = \vec{OM} + u \cdot \vec{r} + v \cdot \vec{s}$$

$$140.2500 \cdot t - 0.8529 \cdot u - 0.1504 \cdot v = 0.2500$$

$$160.4330 \cdot t + 0.4924 \cdot u + 0.0868 \cdot v = 0.4330$$

$$-32.0000 \cdot t + 0.1736 \cdot u - 0.9848 \cdot v = 0.0000$$

$$u = 0.1104, v = -0.0582$$

=====

Aufgabe 4

a) Kommentieren Sie die folgende Rechnung Zeile für Zeile. Das Kreislein $\circ$ steht für das Skalarprodukt von Vektoren.

$$\begin{aligned}
 (1) \quad & \overrightarrow{OA} + t \cdot \overrightarrow{AP} = \overrightarrow{OM} + u \cdot \overrightarrow{r} + v \cdot \overrightarrow{s} \\
 (2) \quad & t \cdot (d \cdot \overrightarrow{n} + \overrightarrow{MP}) - u \cdot \overrightarrow{r} - v \cdot \overrightarrow{s} = d \cdot \overrightarrow{n} \\
 (3) \quad & (t \cdot (d \cdot \overrightarrow{n} + \overrightarrow{MP}) - u \cdot \overrightarrow{r} - v \cdot \overrightarrow{s}) \circ \overrightarrow{n} = (d \cdot \overrightarrow{n}) \circ \overrightarrow{n} \\
 (4) \quad & t \cdot (d \cdot (\overrightarrow{n} \circ \overrightarrow{n}) + (\overrightarrow{MP} \circ \overrightarrow{n})) - u \cdot (\overrightarrow{r} \circ \overrightarrow{n}) - v \cdot (\overrightarrow{s} \circ \overrightarrow{n}) = d \cdot (\overrightarrow{n} \circ \overrightarrow{n}) \\
 (5) \quad & t \cdot (d + \overrightarrow{MP} \circ \overrightarrow{n}) = d \\
 (6) \quad & t = \frac{d}{d + \overrightarrow{MP} \circ \overrightarrow{n}} \\
 & \text{=====}
 \end{aligned}$$

b) Berechnen Sie u und v analog zu t in Teilaufgabe a).

c) Überprüfen Sie die Formeln für t, u und v anhand von Aufgabe 3.

d) Reflektieren Sie die Rechnungen in Teilaufgaben a) und b). Warum verlaufen sie so erfolgreich?

Prüfung über das Projekt KS Flight Simulator

Kantonsschule Baden / Klasse 3bR / 2. Semester Schuljahr 1996/97 / ADM

Zeit:	3 Lektionen.
Hilfsmittel:	Zeichengeräte, Taschenrechner, handgeschriebene Zusammenfassung im Umfang von zwei A4-Seiten, Formelsammlung DMK/DPK, TrueBASIC-Benutzerhandbuch.
Bewertung:	Total möglich sind 38 Punkte. 30 Punkte geben die Note 6, 18 Punkte die Note 4.
Hinweise:	Alle Lösungen sind so zu kommentieren, dass die beim Lösen gemachten Überlegungen klar erkennbar sind. Rechnungen sind in Einzelschritten darzulegen.

Aufgabe 1: Sicht des Fluglotsen (12 Punkte)

Abb. 1.1 zeigt im Grund- und Aufriss die Piste und den Hangar «unseres» Flughafens. Im Punkt A: $(-30, -10, 20)$ überwacht ein Fluglotse den Flugverkehr. Er blickt in Richtung Nordost. Sein Kopf ist um 15° nach unten und weder nach links noch nach rechts geneigt.

- a) Zeichnen Sie in die **Abb. 1.2** die Sicht des Fluglotsen in etwa korrekt ein.
- b) Berechnen Sie zum Punkt P: $(25, 120, 10)$ den zugehörigen Bildpunkt. Wählen Sie $d = 0.5$ als Abstand zwischen Augpunkt A und Bildebene.

Aufgabe 2: Schatten der Lampe (Teil 1) (6 Punkte)

Abb. 2 zeigt im Schrägriss die Piste und den Hangar «unseres» Flughafens. Im Punkt L: $(-20, 30, 40)$ befindet sich eine Lampe, welche Licht in alle Richtungen abstrahlt.

- a) Zeichnen Sie in **Abb. 2** den Schatten, welcher der Hangar auf den Boden (auf die xy-Ebene) wirft, so korrekt wie möglich ein.
- b) Berechnen Sie zum Punkt Q: $(40, 60, 20)$ den zugehörigen Schattenpunkt mit Vektorrechnung.

Aufgabe 3: Schatten der Lampe (Teil 2) (10 Punkte)

Abb. 3 zeigt ein in TrueBASIC geschriebenes, teilweise noch unvollständiges Programm, welches zu einer vorgegebenen Lichtquelle den Schatten aller Objekte einer Landschaft erzeugt.

- a) Erläutern Sie den Aufbau des Programms.
- b) Vervollständigen Sie das Unterprogramm Berechnung.
- c) Diskutieren Sie, inwiefern das Unterprogramm Anzeige unzureichend ist.

Aufgabe 4: Horizontaler Kurvenflug (10 Punkte)

Abb. 4.1 zeigt ein Flugzeug vor, während und nach einem horizontalen Kurvenflug. Die Geschwindigkeit beträgt während des ganzen Manövers $v = 65 \text{ m/s}$.

- a) Der Pilot leitet die Kurve ein, indem er das Flugzeug nach rechts neigt. Begründen Sie physikalisch, warum das Flugzeug eine Querneigung haben muss, um eine Kurve fliegen zu können.
- b) Zeigen Sie, dass bei einer Querneigung von $\alpha = 30^\circ$ der Kurvenradius $R \approx 750 \text{ m}$ beträgt.
- c) Nachdem der Pilot das Flugzeug geneigt hat, belässt er es während $t = 15 \text{ s}$ in Querlage. Dabei dreht sich das Flugzeug um einen Winkel φ (vgl. Abb. 4.1). Zeigen Sie, dass $\varphi \approx 75^\circ$ beträgt.
- d) Das Flugzeug hat unmittelbar vor dem Kurvenflug die Position $P_1: (300, 200, 100)$ und die Lage $\vec{n}_1 = (-0.906, 0.423, 0)$, $\vec{r}_1 = (0.423, 0.906, 0)$, $\vec{s}_1 = (0, 0, 1)$ (vgl. Abb. 4.1). Berechnen Sie die Position P_2 und die Lage $\vec{n}_2$, $\vec{r}_2$, $\vec{s}_2$ unmittelbar nach dem Kurvenflug. Orientieren Sie sich dabei an **Abb. 4.2**.

Abb. 1.1

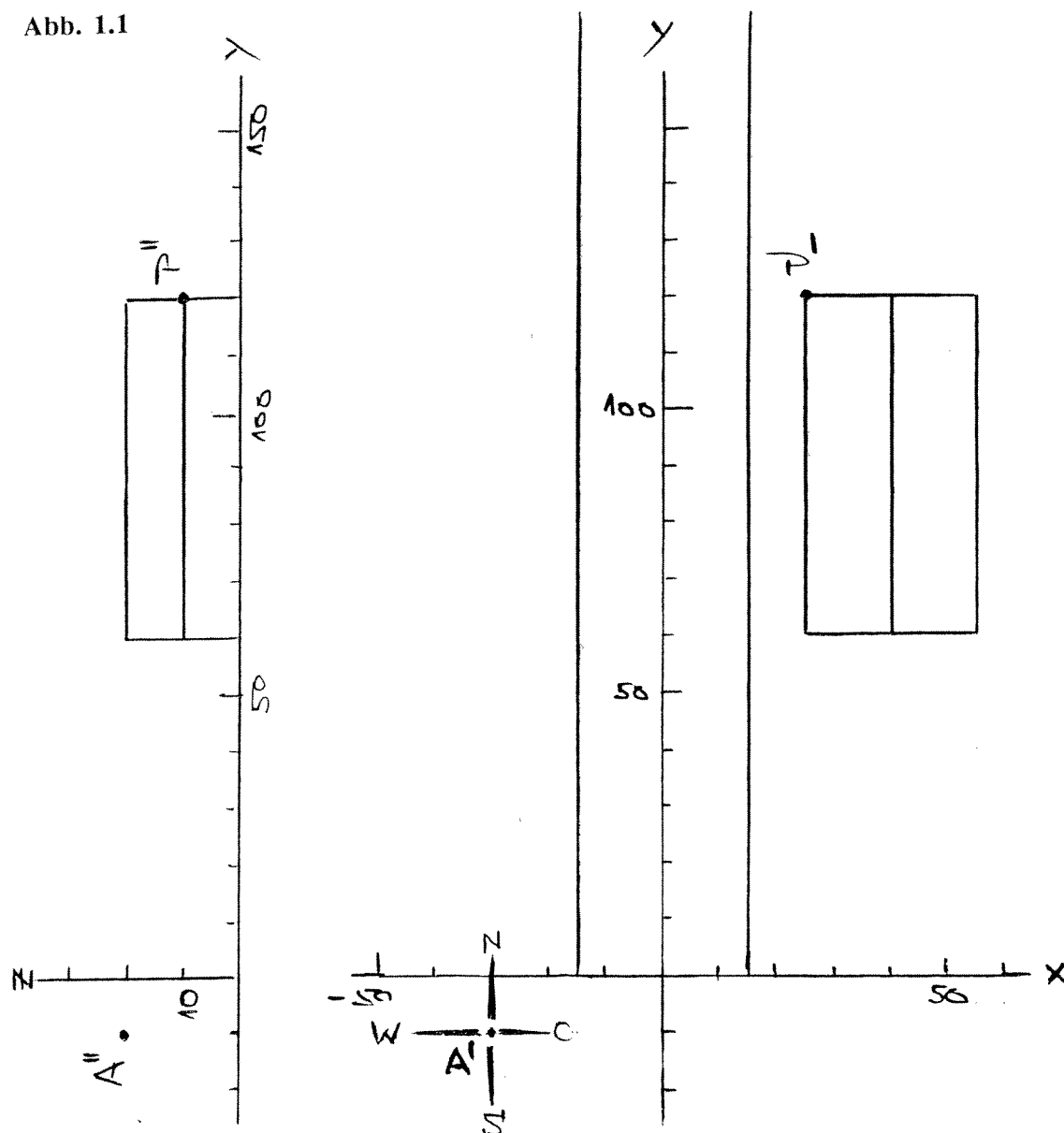


Abb. 1.2

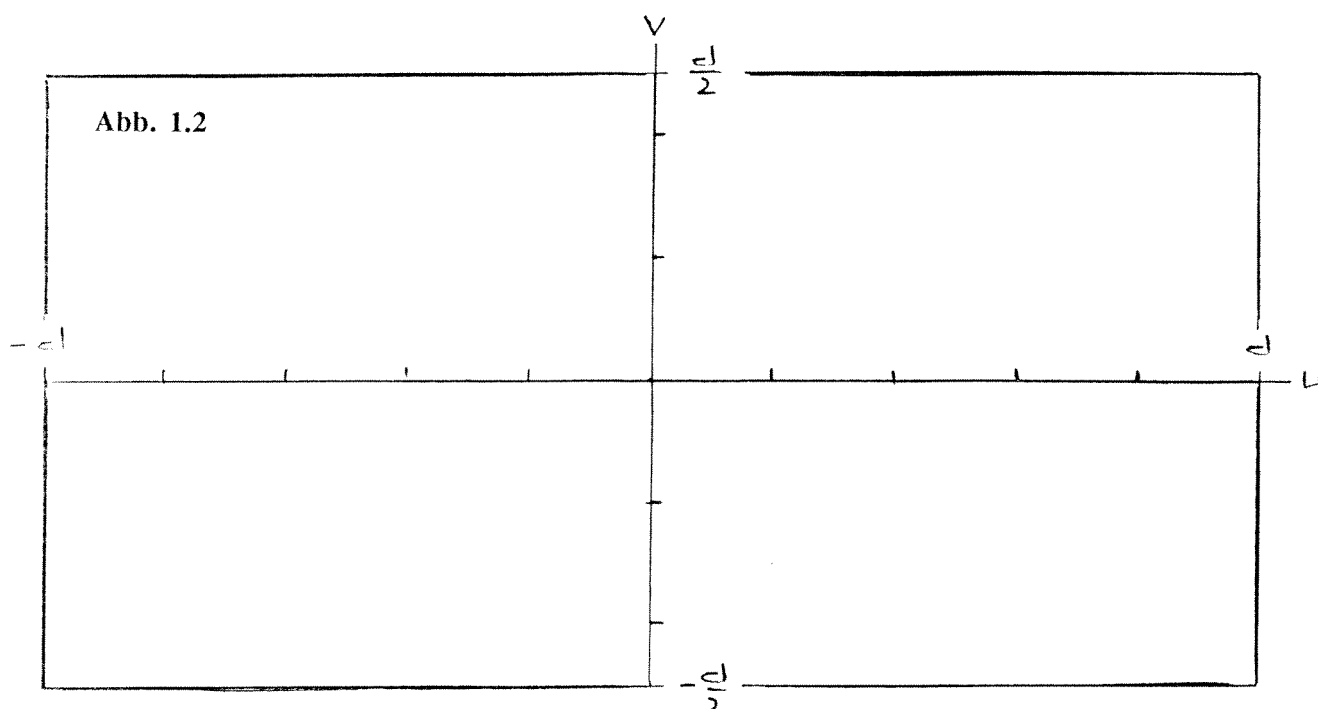


Abb. 2

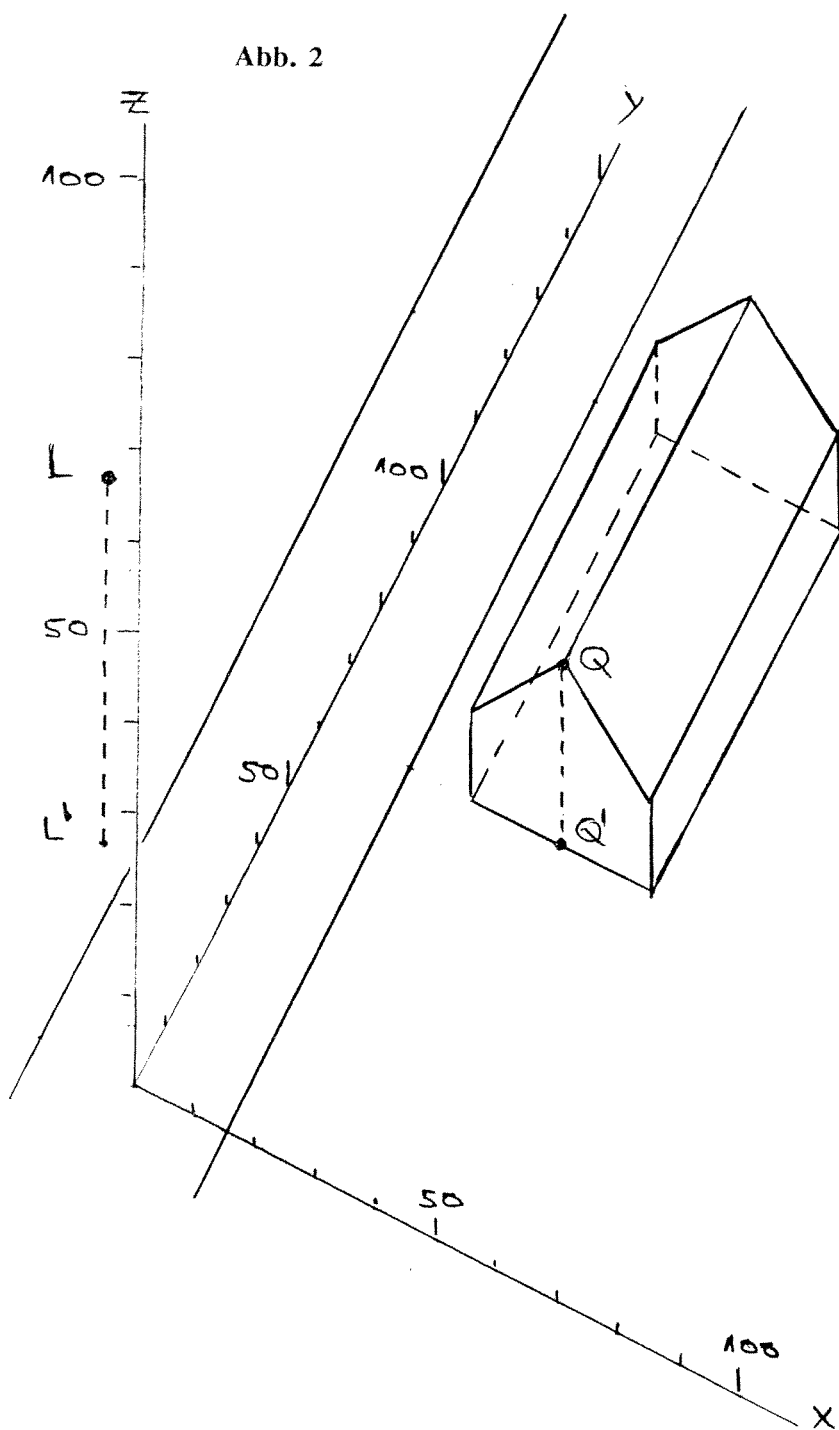


Abb. 3

```
DIM L(1 TO 3)
DIM P(1 TO 100, 1 TO 3)
DIM K(1 TO 100, 1 TO 2)
DIM S(1 TO 100, 1 TO 2)

SUB Daten
  LET L(1) = -20
  LET L(2) = 30
  LET L(3) = 40

  LET P(1, 1) = -15
  LET P(1, 2) = 0
  LET P(1, 3) = 0

  LET P(2, 1) = -15
  LET P(2, 2) = 400
  LET P(2, 3) = 0

  .....

  LET P(14, 1) = 40
  LET P(14, 2) = 120
  LET P(14, 3) = 20

  LET K(1, 1) = 1
  LET K(1, 2) = 2

  LET K(2, 1) = 3
  LET K(2, 2) = 4

  .....

  LET K(17, 1) = 13
  LET K(17, 2) = 14
END SUB

SUB Berechnung

  .....

END SUB

SUB Anzeige
  CLEAR
  FOR J = 1 TO 17
    PLOT LINES: S(K(J, 1), 1), S(K(J, 1), 2); S(K(J, 2), 1), S(K(J, 2), 2)
  NEXT J
END SUB

SET WINDOW -50, 350, -50, 250
CALL Daten
CALL Berechnung
CALL Anzeige

END
```

Abb. 4.1

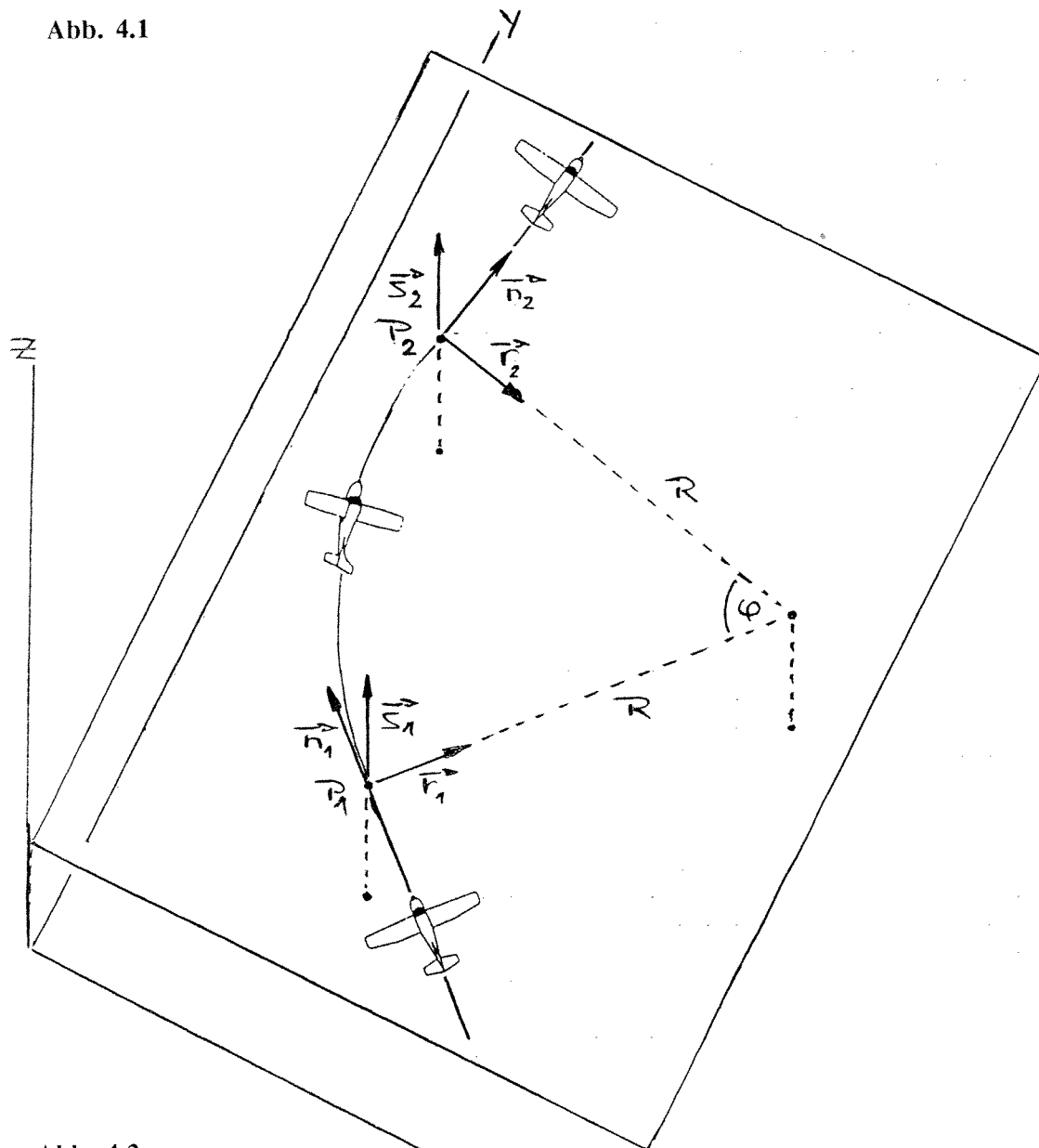
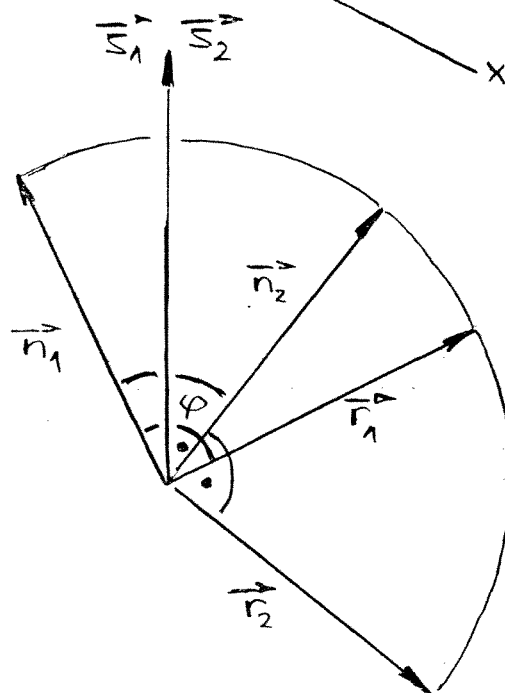


Abb. 4.2



Quellenverzeichnis

Bücher

- [1] Jürgen Mies: Flugtechnik. Motorbuch Verlag, Stuttgart, 1. Aufl. 1996. ISBN 3-613-01725-3.
- [2] Microsoft Flight Simulator, User's Guide for Apple Macintosh. Microsoft Corporation, 1991.
- [3] Werner Leinhos: Flight Simulator for Windows 95 – PC Praxis Buch. Data Becker, Düsseldorf, 1. Aufl. 1996. ISBN 3-8158-1265-8.
- [4] Michael Ellis: Flight Simulator for Windows 95 – Das Bordbuch. Econ Taschenbuch Verlag, Düsseldorf, 1. Aufl. 1997. ISBN 3-612-29310-9.
- [5] Karl Frey: Die Projektmethode. Beltz Verlag, Weinheim und Basel, 6. Aufl. 1995. ISBN 3-407-25123-8.

Artikel

- [6] A. K. Dewdney: Ein beschwingter Fantasieflug über eine menschenleere Datenbank. Spektrum der Wissenschaft, Computer-Kurzweil, September 1986, Seiten 8 – 13.

Programme

- [7] Microsoft Flight Simulator, Version 4 for Apple Macintosh. Microsoft Corporation, USA, 1993.
- [8] TrueBASIC, Version 4 for Apple Macintosh. TrueBASIC Incorporated, USA, 1993.

Filme

- [9] Top Gun. Regie: Tony Scott. Hauptdarsteller: Tom Cruise. Produktion: Paramount Pictures, USA, 1986.

Texte

Die Einstimmung ist weitgehendst aus [6] übernommen.

Abbildungen

Die Illustrationen auf dem Titelblatt, die Abbildung zur Einstimmung, Abb. 1 – 5 in Kap. 1 sind aus [2] entnommen.

89-01	H. Walser	Fraktale
89-02	H.R. Schneebeli	Zwei Fallstudien zur Geometrie
89-03	W. Büchi	Astronomie im Mathematikunterricht
89-04	M. Adelmeyer	Theorem von Sarkovskii
90-01	U. Kirchgraber	Von Mathematik und Mathematikunterricht
90-02	A. Kirsch	Das Paradoxon von Hausdorff, Banach und Tarski: Kann man es "verstehen"?
91-01	A. Barth	Formalisierung und künstliche Intelligenz – eine mögliche Behandlung in der Schule
91-02	U. Kirchgraber	Smale's Beweis des Fundamentalsatzes
91-03	M. Federer	Preistheorie
91-04	M. Gauglhofer	Zur Theorie der sozialen Entscheidungen: Das Arrow-Paradoxon bei Abstimmungen über mehrere Alternativen
92-01	U. Kirchgraber	Chaotisches Verhalten in einfachen Systemen
93-01	M. Huber, U. Manz, H. Walser	Annäherung an den Goldenen Schnitt
93-01(I)	M. Huber, U. Manz, H. Walser	Approccio alla Sezione Aurea
93-02	P. Gallin, H. Keller, H. Stocker	Perspektive und Axonometrie
93-02(I)	P. Gallin, H. Keller, H. Stocker	Prospettiva e Assonometria
93-03	H.R. Schneebeli, N. Sigris, F. Spirig	Verzweigungsphänomene
93-03(I)	H.R. Schneebeli, N. Sigris, F. Spirig	Fenomeni di Biforcazione
93-04	H. Biner, H.P. Dreyer, W. Hartmann, A. Moretti	Der Fallschirmspringer
93-04(I)	H. Biner, H.P. Dreyer, W. Hartmann, A. Moretti	Il Paracadutista
93-05	H.R. Schneebeli	Alles fliesst – Mit dem Graphikrechner zu den Quellen der Analysis
93-06	H. Biner	Kongruenzabbildungen und Symmetrien im Euklidischen Raum
93-07	U. Kirchgraber	Hundert Jahre Störungstheorie – Differentialgleichungen von Poincaré bis Nekhoroshev
94-01	U. Maurer	Kryptologie: Mathematik zwischen Anwendung und Ästhetik
94-02	H. Klemenz	Computergestützte Raumgeometrie
94-03	F. Barth	Erstens kommt es anders und zweitens als man denkt - Paradoxien im Umfeld der bedingten Wahrscheinlichkeit
94-04	W. Henn	Auto und Verkehr – Beispiele aus der Analysis zum realitätsnahen Mathematikunterricht
95-01	N. Sigris	Auf der Kippe

95-02	U.Kirchgraber U. Kirchgraber, N. Sigris	Als Poincaré, Hadamard und Perron die Invarianten Mannigfaltigkeiten entdeckten. Feigenbaum-Universalität: Beschreibung und Beweisskizze
95-03	A. Gächter	Infinitesimalgeometrie - am Beispiel der Kreisevolvente
95-04	P. Gallin	Grund- und Aufrissmethode in der Wahrscheinlichkeitsrechnung
95-05	P. Bolli	The unreasonable effectiveness of mathematics
95-06	G. Schierscher	Verfolgungsprobleme
96-01	W. Burgherr	Schwimmende Prismen mit Schlagseite
96-02	M. Struwe	Sattelpunkte oder Variationsprinzipien in Geometrie und Mechanik
96-03	M. Huber	Warum denn ist $\exp(x^2)$ nicht elementar integrierbar?
97-01	B. Eicke, E. Holzherr	Analysis – mit dem Computer-Algebra-System des TI-92 (Preis: Fr. 8.- incl. MWSt)
97-02	C. Blatter	Notizen zu einer Mathematik fürs Leben
97-03	F. Spirig	Elemente zur Kugelgeometrie
98-01	U. Kortenkamp, J. Richter-Gebert	Geometry and Education in the Internet Age
98-02	M. Adelmeyer	KS Flight Simulator