



**BERUFSAKADEMIE MANNHEIM
STAATLICHE STUDIENAKADEMIE**

Fachrichtung Informationstechnik

Praktikumsbericht
3. Semester

***Implementierung eines Verfahrens zur Bestimmung von
Flugzeiten in der Anflugphase***

Von:

*Jeanne Kirgizov
Matrikel-Nr. 170269
TIT 04 A GR*



Deutsches Zentrum
für Luft- und Raumfahrt e.V. in
der Helmholtz-Gemeinschaft

Lilienthalplatz 7
38108 Braunschweig

Institut für Flugführung
Abteilung Luftverkehrssysteme

Betreuer:
Dipl.-Ing. F. Knabe

Dokument Information

Betreuer:	Dipl.-Ing. F. Knabe
Zuständiger Autor:	Jeanne Kirgizov
Weitere Autor(en):	
Projekt / Zielfeld:	
Zugänglichkeitsstufe:	C/II (im Institut für Flugführung unbegrenzt zugänglich)
Datei:	Dokument1
Version:	0.02
Speicherdatum:	2006-03-21
Gesamtseitenzahl:	22

Freigabe:

Die Freigabe erfolgt lt. gesondertem Freigabebformblatt:

© 2006, DLR, Institut für Flugführung:

Dieses Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwendung innerhalb und außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des DLR, Institut für Flugführung, unzulässig und wird zivil- und strafrechtlich verfolgt. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Änderungsverfolgung

Version	Datum	Geänderte Seiten / Kapitel	Bemerkungen
0.01	10.03.2006	Kapiteln: 1, 2, 4, 5, 7	
0.02	20.03.2006		

Inhaltsverzeichnis

1	Einleitung	5
2	Aufgabenstellung	6
3	Betrachtungsbereich ASMA	7
4	Eingabedaten	8
5	Verwendete Softwarekomponenten	9
5.1	Datenbankverbindung mit ODBC	9
5.2	Installation und Konfiguration von ODBC-Treibern	10
5.3	Zugriff auf SQL	10
5.4	MFC als Unterstützung für ODBC	11
5.4.1	CDatabase-Klasse	11
5.4.2	CRecordset-Klasse	11
6	Implementierung des Programms	12
6.1	Entwicklungsumgebung und Bibliotheken	12
6.1.1	Vektoren in C++	12
6.1.2	Maps in C++	12
6.2	Aufbau des Programms	12
6.2.1	CSpreadSheet-Klasse	13
6.2.2	Klasse CGauss_Krueger	14
6.2.3	Klasse CSchnitt_Gerade_Kreis	14
6.2.4	Klasse Winkelberechnung	16
7	Fazit	18
8	Literatur	19
9	Internetquellen	19
10	Bildverzeichnis	19
11	Anhang A	20
11.1	Transversale Mercator Projektion	20
11.2	Gauß-Krüger-Koordinaten	20
12	Anhang B	22

1 Einleitung

„Air transport punctuality in Europe is one of the major concerns for the industry and a constant source of complaints from the passengers. Not only are unpunctual flights a major inconvenience for the passengers, especially when connections are missed, but they also include large “tactical” and “strategic” costs for airlines and the airline community as a whole. Hence, reducing air transport delays to the minimum is of major importance for passengers, airlines and airports [6].“

Es ist das Ziel der Luftfahrtindustrie in Europa, die Pünktlichkeit¹ des Flugverkehrs zu steigern.

Der achte Performance Review Report [6] befasst sich mit den elf größten europäischen Flughäfen (z. B. Amsterdam, Barcelona, Paris Charles de Gaulle, Rome Fiumicino, Frankfurt, London, Madrid, Zürich usw.). Besondere Bedeutung kommt der effizienteren Nutzung der bestehenden Zeitressourcen zu. Durch die Planung und Optimierung der Zeitverteilung soll ein Effizienzgewinn erzielt werden. Eines der wichtigsten Untersuchungsgebiete ist dabei die Pünktlichkeit der Flugzeuge.

Auf diesem Gebiet sollen die „taktischen“ und „strategischen“ Kosten gesenkt werden. Unter „taktischen“ Kosten versteht man die zusätzlichen Kosten, die durch eine Verzögerung im Flugverkehr verursacht werden, wie z.B. die zusätzlichen Treibstoffverbrauchskosten. „Strategische“ Kosten sind auf die zusätzlichen Zeitpuffer bezogen, die zur Erhöhung der Pünktlichkeit im Luftverkehr beitragen.

Um die Verzögerungszeiten während des Fluges zu kontrollieren, werden die so genannten Pufferzeiten eingeführt.

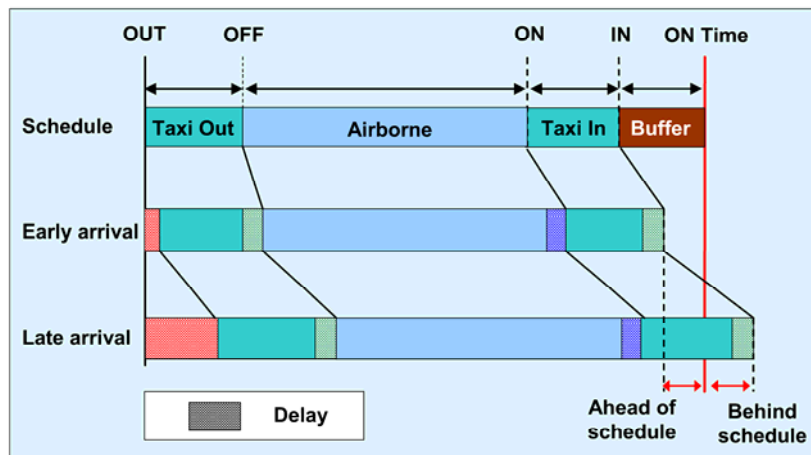


Abbildung 1: Aufteilung von Flugzeiten [6]

Diese Zeitpuffer tragen zur Steigerung der Pünktlichkeit im Flugverkehr bei, da der Zeitvorrat in der Flugplanung ausgedehnt wird. Als Folge dessen steigen die Flugzeiten, was zu höheren Kosten führt. Laut dem achten Performance Review Report würde die Verkürzung der Zeitpuffer bei der Hälfte aller europäischen Flüge von nur fünf Minuten zu einem jährlichen Gewinn für die Luftfahrtindustrie von ca. einer Milliarde Euro führen.

Im diesem Zusammenhang wird die Kürzung der Flugzeiten durch die Reduzierung der Variabilität der Flugzeit als Ansatzpunkt angesehen.

¹ Der Anteil von Flügen, der weniger als 15 Minuten verspätet ist verglichen mit den veröffentlichten An- und Abflugzeiten

2 Aufgabenstellung

Variabilitäten, die zu längeren Zeitpuffern beitragen entstehen in allen Flugphasen.

Die EUROCONTROL-Studie konstatiert einen Mangel an Daten zu den Flügen in der flughafennahen Phase (En-Terminal-Phase). In Rahmen dieser Arbeit soll diese letzte Phase (Lande-anflugphase) des Fluges näher betrachtet werden. Dazu soll ein Programm entwickelt werden, das die Landeanflugzeit und andere Anflugparameter eines Fluges aus einem gegebenen Datensatz berechnet. Dabei sind die Anflugzeit des Fluges vom Eintrittspunkt in den flughafennahen Bereich bis zur Landung, sowie die Anflugrichtung und die Position des Flugzeugs von Bedeutung. Die Auswertung dieser Daten soll anschließend zu einer genaueren Festsetzung der Pufferzeiten beitragen.

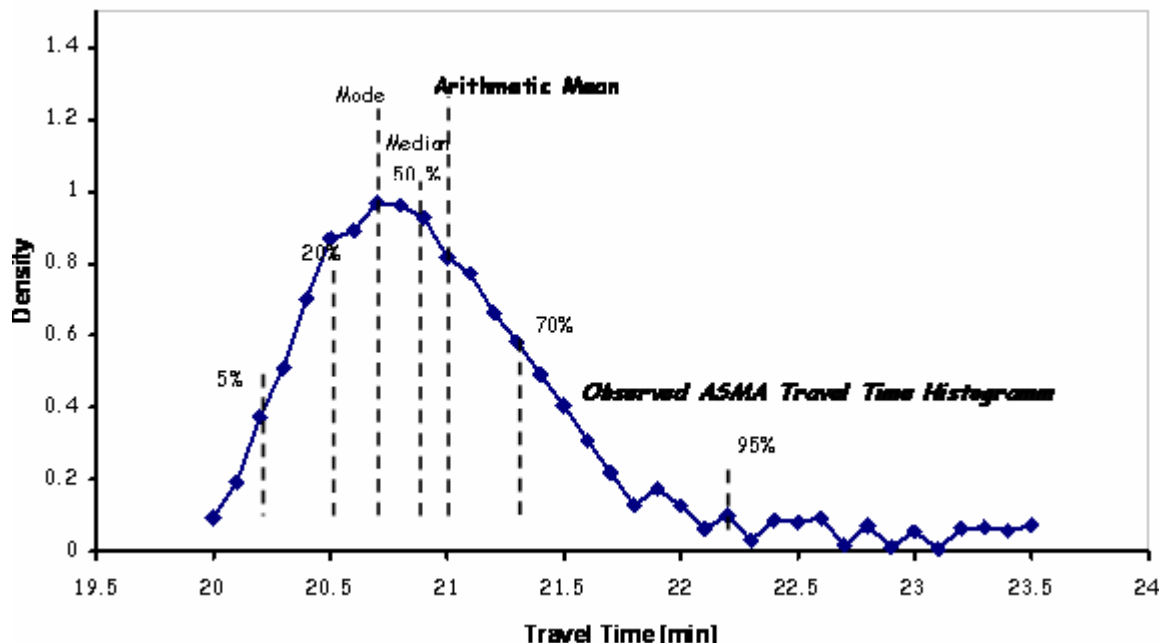


Abbildung 2: Histogramm² der Flugzeitverteilung in der letzten Flugphase (Beispiel) [7]

Die in Abbildung 2 dargestellte Kurve zeigt ein Beispiel einer statistischen Verteilung der benötigten Flugzeiten in der ASMA-Phase (Arrival Sequencing and Metering Area).

Hier soll eine Einengung der Verteilungskurve erfolgen, was zum nachfolgenden Abbau der Pufferzeiten führen könnte.

Das Ergebnis dieser Studienarbeit soll eine genauere Analyse der Flugzeit im Anflugbereich ermöglichen und zur Erhöhung der Pünktlichkeit im Luftverkehr, zur Flugzeiterparnis und auch zu einem umweltschonenden Verhalten einen Beitrag leisten.

² Graphische Darstellung der Häufigkeitsverteilung von Messwerten(WWW03)

3 Betrachtungsbereich ASMA

In dieser Studienarbeit wird die Betrachtungsfläche auf den flughafennahen Bereich (en-terminal) begrenzt. Der Kern dieses Gebietes wird als Terminal Maneuvering Area (TMA³) bezeichnet. In diesem Bereich kommt es zu großen Unterschieden in der Anflugzeit, die durch Warteschleifen verursacht werden.

Alle Messungen und Berechnungen in diesem Bericht beziehen sich auf die Fläche um die TMA. Dabei handelt es sich um einen als ASMA bezeichneten erweiterten Betrachtungsbereich, der einen Kreis mit einem Radius von 100 NM⁴ um den Flughafen darstellt.

Die Betrachtung beginnt im äußeren Bereich an dem Punkt, an dem der im Anflug befindliche Flieger den 100 NM Radius zum Endflughafen überfliegt und endet mit der Landung [5].

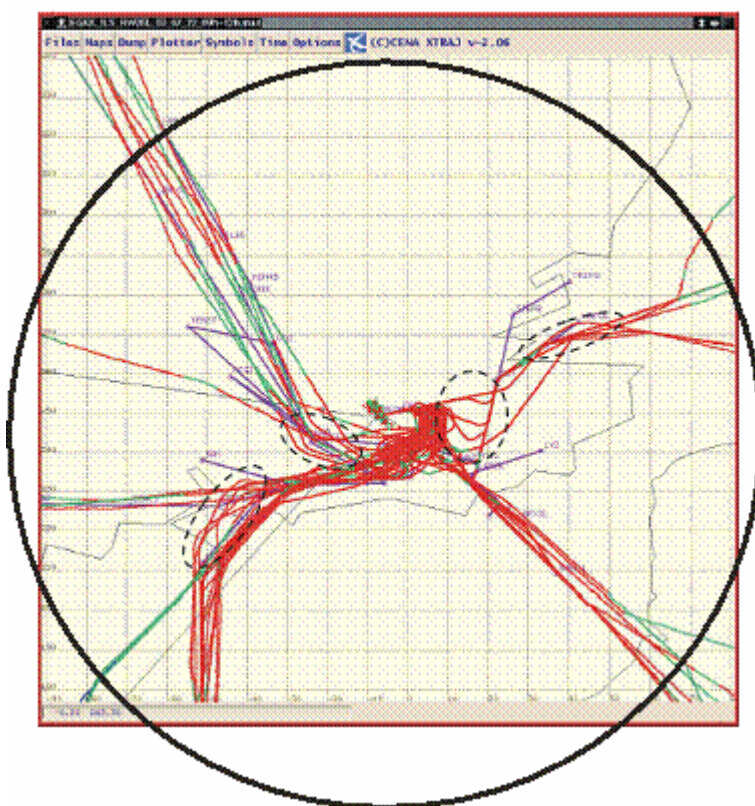


Abbildung 3: *Arrival Sequencing and Metering Area (ASMA)* um den Flughafen London-Gatwick [7]

Dieses Kreiskonzept ist universell für alle europäischen Flughäfen einsetzbar und stellt eine allgemeingültige Leistungsvergleichsbasis zur Verfügung.

Die Kreisfläche wird in acht Sektoren unterteilt, um die jeweilige Anflugrichtung genauer zu bestimmen.

³ Definierter Luftraum um einen Flughafen, in dem hauptsächlich An- und Abflüge über festgelegte Routen geleitet werden (WWW02).

⁴ Die nautische Meile (Seemeile) ist ein in der Luftfahrt gebräuchliches Längenmaß (1 NM = 1852 m).

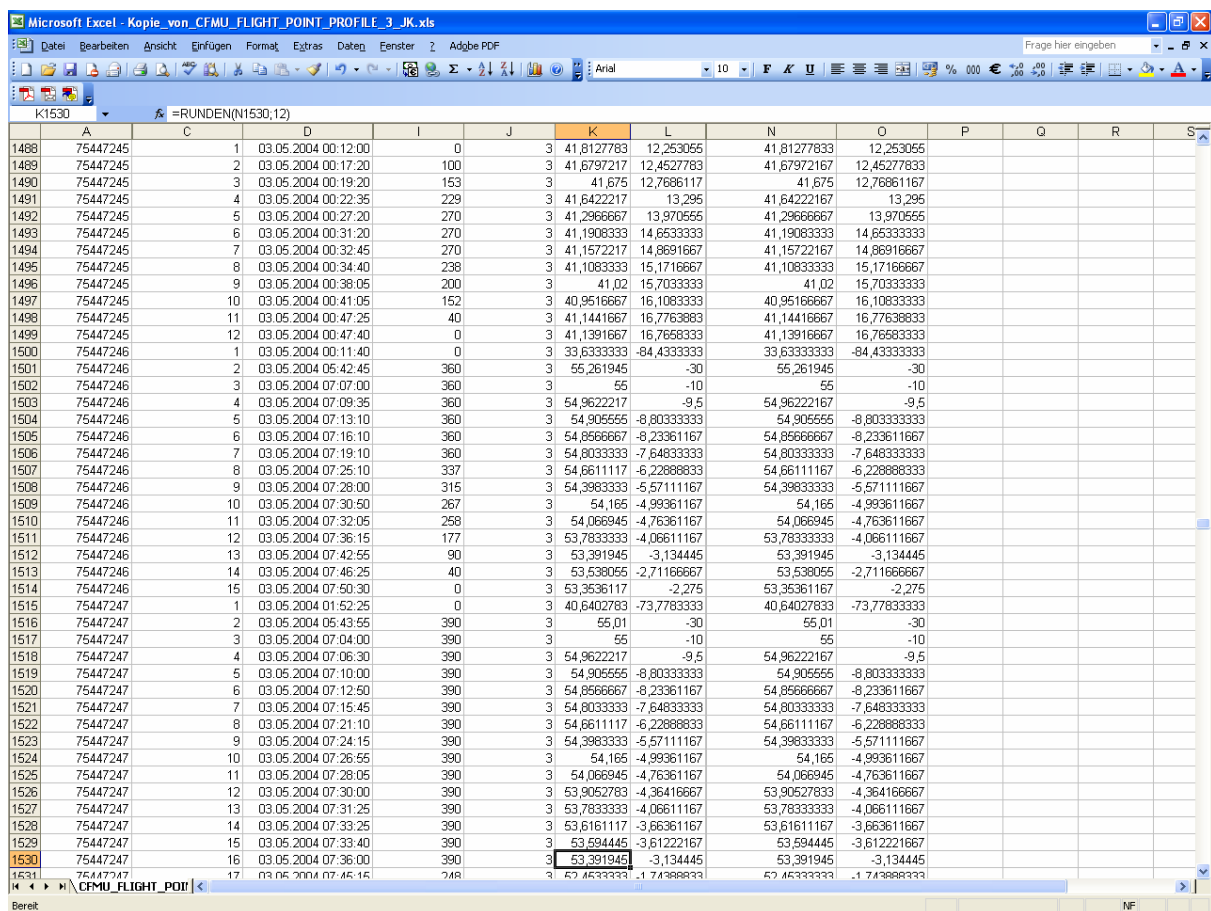
4 Eingabedaten

Die Flugzeuge bewegen sich auf unterschiedlichen Bahnen. Die jeweiligen Datensätze zu den Flugspurpunkten der einzelnen Flugzeuge werden in entsprechenden Exceltabellen geliefert. Diese Tabellen enthalten die planmäßigen und die tatsächlichen Daten zu den zu untersuchenden Fliegern. Jedes Flugzeug landet auf einem bestimmten Flughafen, um welchen ein imaginärer Kreis mit dem Radius von 100 NM gelegt wurde.

Excel ist ein Tabellenkalkulationsprogramm von Microsoft. Es erlaubt Formeln, Funktionen und umfangreiche Berechnungen durchzuführen. Im diesem Projekt wird eine Exceltabelle allerdings als eine kleine Datenbank eingesetzt, in der die Daten für jedes einzelne Flugzeug gespeichert werden. Die Exceldatei umfasst jeweils eine Tabelle, die aus zwölf Spalten und über zweitausend Zeilen besteht.

Die Flüge werden nach Flugzeug-ID sortiert dargestellt und umfassen jeweils eine Reihe von Flugspurpunkten (geographische Länge und Breite), Flugfläche⁵, Datum, Zeitpunkt usw. verfügen. Betrachtet man die Flugfläche, kann man feststellen, wo der Bestimmungsflughafen liegt, da in der Liste mit den verwendeten Daten die Flughöhe Null erreicht wird.

Die Tabelleneinträge kann man in zwei Teile unterteilen: Im ersten Teil (Zeile 2 bis 1487) werden die geplanten Flugdaten gespeichert. Ein kennzeichnendes Merkmal dafür ist die Zahl „1“ in der *Flug-Model*-Spalte. Im zweiten Teil (Zeile 1488 bis 2830) werden die tatsächlich geflogenen Flugdaten abgelegt. Ein kennzeichnendes Merkmal für diese Daten ist die Zahl „3“ in der *Flug-Model*-Spalte.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1488	75447245	1	03.05.2004 00:12:00		0	3	41,8127783	12,253055		41,81277833	12,253055		41,81277833	12,253055				
1489	75447245	2	03.05.2004 00:17:20		100	3	41,6797217	12,4527783		41,67972167	12,45277833		41,67972167	12,45277833				
1490	75447245	3	03.05.2004 00:19:20		153	3	41,675	12,7686117		41,675	12,76861167		41,675	12,76861167				
1491	75447245	4	03.05.2004 00:22:35		229	3	41,6422217	13,295		41,64222167	13,295		41,64222167	13,295				
1492	75447245	5	03.05.2004 00:27:20		270	3	41,2966667	13,970555		41,29666667	13,970555		41,29666667	13,970555				
1493	75447245	6	03.05.2004 00:31:20		270	3	41,1908333	14,6533333		41,19083333	14,65333333		41,19083333	14,65333333				
1494	75447245	7	03.05.2004 00:32:45		270	3	41,1572217	14,8691667		41,15722167	14,86916667		41,15722167	14,86916667				
1495	75447245	8	03.05.2004 00:34:40		238	3	41,1083333	15,1716667		41,10833333	15,17166667		41,10833333	15,17166667				
1496	75447245	9	03.05.2004 00:38:05		200	3	41,02	15,7033333		41,02	15,70333333		41,02	15,70333333				
1497	75447245	10	03.05.2004 00:41:05		152	3	40,9516667	16,1083333		40,95166667	16,10833333		40,95166667	16,10833333				
1498	75447245	11	03.05.2004 00:47:25		40	3	41,1441667	16,7763883		41,14416667	16,77638833		41,14416667	16,77638833				
1499	75447245	12	03.05.2004 00:47:40		0	3	41,1391667	16,7658333		41,13916667	16,76583333		41,13916667	16,76583333				
1500	75447246	1	03.05.2004 00:11:40		0	3	33,6333333	-84,4333333		33,63333333	-84,43333333		33,63333333	-84,43333333				
1501	75447246	2	03.05.2004 05:42:45		360	3	55,261945	-30		55,261945	-30		55,261945	-30				
1502	75447246	3	03.05.2004 07:07:00		360	3	55	-10		55	-10		55	-10				
1503	75447246	4	03.05.2004 07:09:35		360	3	54,9622217	-9,5		54,96222167	-9,5		54,96222167	-9,5				
1504	75447246	5	03.05.2004 07:13:10		360	3	54,905555	-8,8033333		54,905555	-8,80333333		54,905555	-8,80333333				
1505	75447246	6	03.05.2004 07:16:10		360	3	54,8566667	-8,2336117		54,85666667	-8,23361167		54,85666667	-8,23361167				
1506	75447246	7	03.05.2004 07:19:10		360	3	54,8033333	-7,6483333		54,80333333	-7,64833333		54,80333333	-7,64833333				
1507	75447246	8	03.05.2004 07:25:10		337	3	54,6611117	-6,2288833		54,66111167	-6,22888333		54,66111167	-6,22888333				
1508	75447246	9	03.05.2004 07:28:00		315	3	54,3983333	-5,5711117		54,39833333	-5,57111167		54,39833333	-5,57111167				
1509	75447246	10	03.05.2004 07:30:50		267	3	54,165	-4,9936117		54,165	-4,99361167		54,165	-4,99361167				
1510	75447246	11	03.05.2004 07:32:05		258	3	54,066945	-4,7636117		54,066945	-4,76361167		54,066945	-4,76361167				
1511	75447246	12	03.05.2004 07:36:15		177	3	53,7833333	-4,0661117		53,78333333	-4,06611167		53,78333333	-4,06611167				
1512	75447246	13	03.05.2004 07:42:55		90	3	53,391945	-3,134445		53,391945	-3,134445		53,391945	-3,134445				
1513	75447246	14	03.05.2004 07:46:25		40	3	53,538055	-2,7116667		53,538055	-2,71166667		53,538055	-2,71166667				
1514	75447246	15	03.05.2004 07:50:30		0	3	53,3536117	-2,275		53,35361167	-2,275		53,35361167	-2,275				
1515	75447247	1	03.05.2004 01:52:25		0	3	40,6402783	-73,7783333		40,64027833	-73,77833333		40,64027833	-73,77833333				
1516	75447247	2	03.05.2004 05:43:55		390	3	55,01	-30		55,01	-30		55,01	-30				
1517	75447247	3	03.05.2004 07:04:00		390	3	55	-10		55	-10		55	-10				
1518	75447247	4	03.05.2004 07:06:30		390	3	54,9622217	-9,5		54,96222167	-9,5		54,96222167	-9,5				
1519	75447247	5	03.05.2004 07:10:00		390	3	54,905555	-8,8033333		54,905555	-8,80333333		54,905555	-8,80333333				
1520	75447247	6	03.05.2004 07:12:50		390	3	54,8566667	-8,2336117		54,85666667	-8,23361167		54,85666667	-8,23361167				
1521	75447247	7	03.05.2004 07:15:45		390	3	54,8033333	-7,6483333		54,80333333	-7,64833333		54,80333333	-7,64833333				
1522	75447247	8	03.05.2004 07:21:10		390	3	54,6611117	-6,2288833		54,66111167	-6,22888333		54,66111167	-6,22888333				
1523	75447247	9	03.05.2004 07:24:15		390	3	54,3983333	-5,5711117		54,39833333	-5,57111167		54,39833333	-5,57111167				
1524	75447247	10	03.05.2004 07:26:55		390	3	54,165	-4,9936117		54,165	-4,99361167		54,165	-4,99361167				
1525	75447247	11	03.05.2004 07:28:05		390	3	54,066945	-4,7636117		54,066945	-4,76361167		54,066945	-4,76361167				
1526	75447247	12	03.05.2004 07:30:00		390	3	53,9052783	-4,3641667		53,90527833	-4,36416667		53,90527833	-4,36416667				
1527	75447247	13	03.05.2004 07:31:25		390	3	53,7833333	-4,0661117		53,78333333	-4,06611167		53,78333333	-4,06611167				
1528	75447247	14	03.05.2004 07:33:25		390	3	53,6161117	-3,6636117		53,61611167	-3,66361167		53,61611167	-3,66361167				
1529	75447247	15	03.05.2004 07:33:40		390	3	53,594445	-3,6122217		53,594445	-3,61222167		53,594445	-3,61222167				
1530	75447247	16	03.05.2004 07:36:00		390	3	53,391945	-3,134445		53,391945	-3,134445		53,391945	-3,134445				
1531	75447247	17	03.05.2004 07:45:15		248	3	53,4533333	-1,7438883		53,45333333	-1,74388833		53,45333333	-1,74388833				

Abbildung 4: Ausschnitt der Exceltabelle mit den Eingabedaten

Der in der Abbildung 4 wiedergegebene Screenshot einer solchen Tabelle beinhaltet die tatsächlichen Daten zu den einzelnen Flügen. Die erste Spalte gibt die einzelnen Nummer der Flüge, die zweite Spalte die Folgenummer der einzelnen Flugspurpunkte und die dritte das Datum und den Zeitpunkt,

⁵ Flight level (FL), in der Luftfahrt verwendete Druckhöhe.

die den einzelnen Positionen eines Fliegers entsprechen, wieder. Die nächste Spalte beinhaltet die Flugfläche zu den Flugspurpunkten und die fünfte die Zahl, die einen tatsächlichen Vorgang andeutet. In den letzten Spalten stehen die geographischen Koordinaten jeder einzelnen Position eines Fliegers.

Die einzelnen Flüge unterscheiden sich in der Anzahl ihrer jeweiligen Flugspurpunkte. Diese Zahl variiert zwischen drei und achtzehn Einträgen.

5 Verwendete Softwarekomponenten

Das Programm arbeitet mit bestimmten Softwarekomponenten, die für die Anbindung an die Datenbank benötigt werden. Als Datenquelle wird in diesem Projekt eine Exceltabelle angesehen. Zu den verwendeten Softwarekomponenten zählt die ODBC-Schnittstelle, die mit Hilfe spezieller Treiber auf eine Datenbank zugreifen kann. In diesem Kapitel wird der Zugriff auf eine Datenquelle näher betrachtet.

5.1 Datenbankverbindung mit ODBC

Die ODBC (Open Database Connectivity) ist eine Entwicklung von Microsoft. Sie ermöglicht einen Zugriff auf heterogene ⁶SQL-Datenbanken. Über diese Datenbankverbindung wird das Einlesen der Daten aus einer *.xls-Datei in das Programm realisiert.

ODBC basiert auf SQL (Structured Query Language), einem Standard für Datenzugriff. Um über diese Schnittstelle mit einer anderen Datenbank kommunizieren zu können, müssen zusätzliche Treiberprogramme unter Windows installiert werden.

Die ODBC-Schnittstelle bietet eine Möglichkeit, Daten direkt aus einer Datenbank einzulesen. ODBC wird mit aktuellen Microsoft Office Versionen automatisch installiert, ist betriebssystemspezifisch und vom Datenbankprogramm abhängig. Durch die ODBC-Verbindung können alle Arten von Datenbanken angesprochen werden.

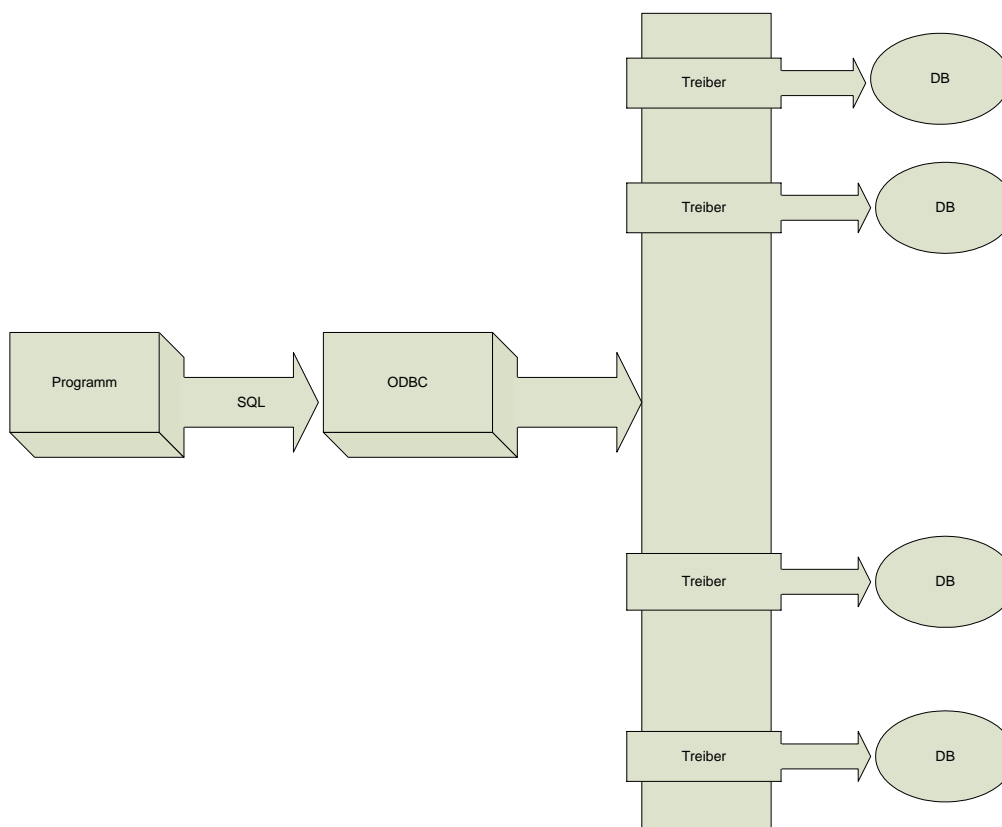


Abbildung 5: Zugriff auf eine Datenbank über ODBC-Schnittstelle

⁶ Inhomogen, uneinheitlich, verschieden, ungleichartig

5.2 Installation und Konfiguration von ODBC-Treibern

Damit eine Verbindung mit Excel funktioniert und die Daten aus einer Exceltabelle eingelesen werden können, soll der ODBC-Treiber auf dem Rechner installiert und für die bestimmte Anwendung (in diesem Fall Excel) konfiguriert werden.

Falls auf dem betreffenden Rechner kein ODBC-Protokoll installiert worden ist, muss das Office-Setup neu gestartet und anschließend der Modus „Features hinzufügen/entfernen“ gewählt werden.

Für den Zugriff auf den SQL-Server ist eine ODBC-Datenquelle Voraussetzung. Um diese einzurichten, geht man wie folgt vor:

Über den in Abbildung 6 dargestellten ODBC-Datenquellen-Administrator (Start→Systemsteuerung→Verwaltung→Datenquellen) kann man den installierten ODBC-Treiber neu konfigurieren und dem verwendeten Datenbankformat anpassen.

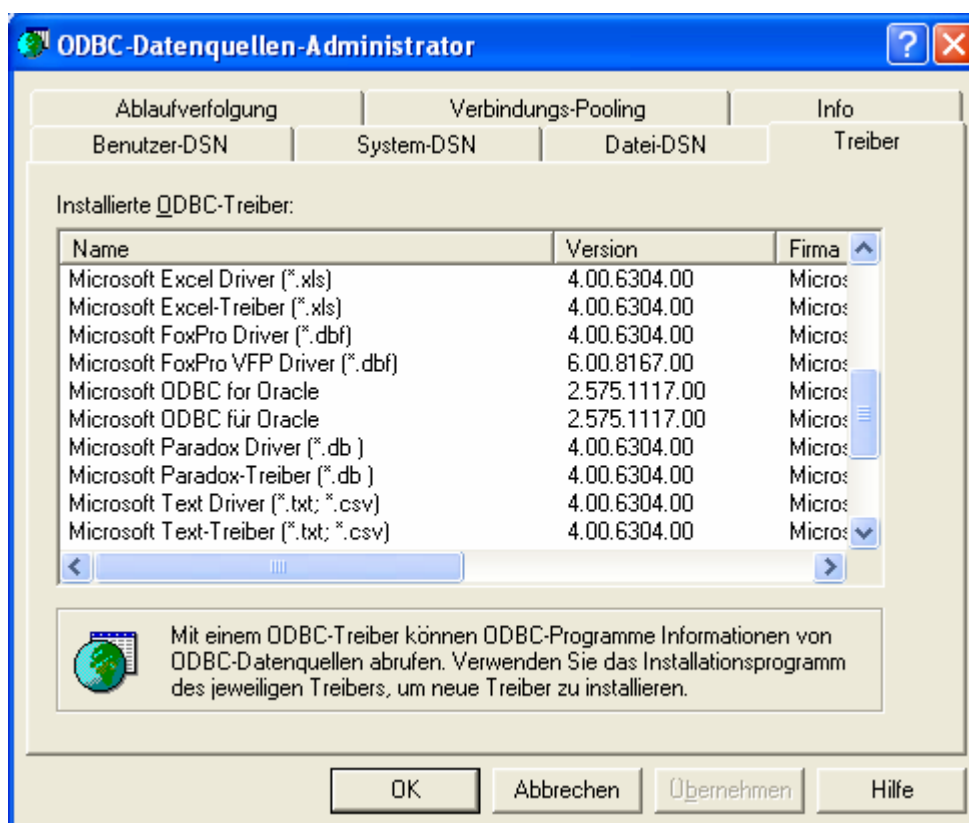


Abbildung 6: ODBC-Datenquellen-Administrator

5.3 Zugriff auf SQL

Über die ODBC-Schnittstelle kann man aus Excel-Anwendung heraus auf SQL zugreifen.

SQL ist eine Abkürzung für Structured Query Language, übersetzt „Strukturierte Abfragesprache“. Sie dient in Datenbanksystemen zur Definition von Daten und zur Informationsgewinnung. SQL definiert die Sprache, die verwendet wird, um mit Datenbankinhalten zu arbeiten. Die Hauptaufgabe von SQL-Ausdrücken ist das Lesen oder Verändern von vorhandenen Daten der Datenbank oder das Hinzufügen von neuen Datenbankeinträgen.

Über SQL-Befehle lassen sich die Daten aus den verknüpften Datenbanken benutzerdefiniert verwalten.

SQL verfügt allgemein über Kommandos zur Datenbearbeitung (Data Manipulation Language (DML)), und Kommandos, mit denen das Datenbank Design geändert werden kann (Data Definition Language (DDL)).

Im Programm werden die Befehle aus DML-Kategorie angewendet:

- SELECT-Anweisung zur Abfrage von bestimmten Daten der Datenbank. Mit diesen Ausdrücken können einzelne oder mehrere Datensätze oder Felder von Datensätzen ausgegeben werden. Dabei kann auf eine oder mehrere Tabellen zugegriffen werden.
- INSERT-Ausdrücke werden zur Speicherung von neuen Datensätzen in die Datenbank eingesetzt.
- DELETE-Befehle werden zum Löschen bestehender Daten verwendet.
- DDL-Anweisungen werden ebenso im Programm gebraucht:
 - CREATE-Ausdrücke erzeugen und definieren Datenbanken und Tabellen.
 - DROP-Befehle dienen zum Löschen von Datenbanken und Tabellen.

Diese standardisierten SQL-Befehle werden in der Klasse zum Einlesen einer Exceltabelle angewandt, um die Kommunikation mit einer ODBC-Schnittstelle zu ermöglichen und über einen entsprechenden Treiber auf eine Excel-Datenbank (*.xls) zuzugreifen.

5.4 MFC als Unterstützung für ODBC

Die MFC-Library (Microsoft Foundation Class) stellt eine breite Unterstützung für ODBC zur Verfügung. Es gibt eine Reihe von Klassen in dieser Bibliothek, die die ODBC-„Funktionalitäten“ „kapseln“ (umschließen). Zu diesen Klassen gehören `CDatabase` und `CRecordset`.

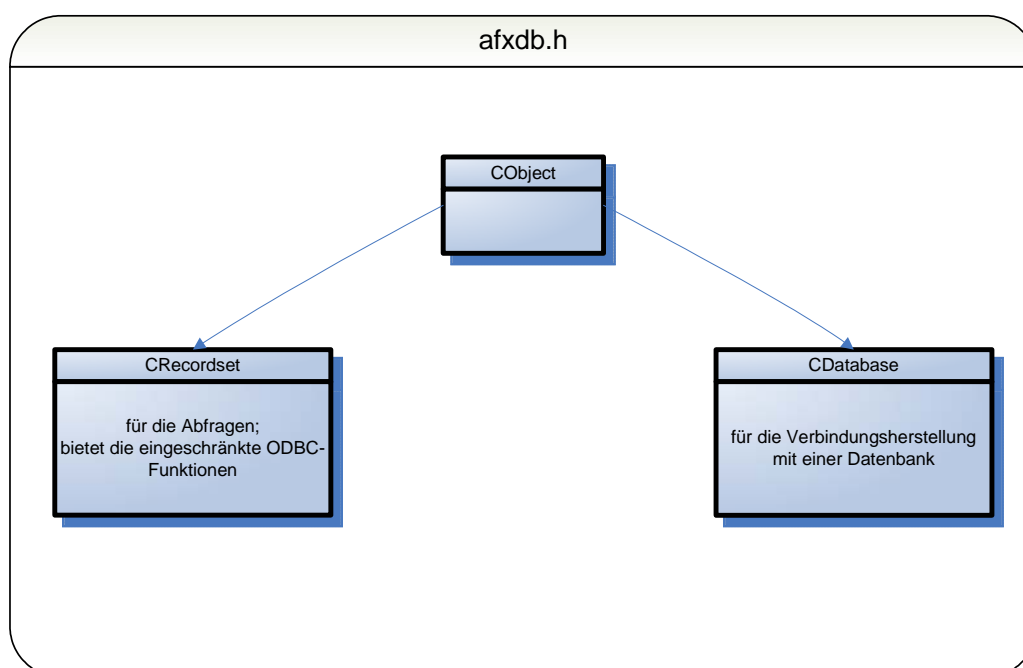


Abbildung 7: MFC-Klassen

5.4.1 CDatabase-Klasse

Nachdem eine Exceltabelle mit dem Konstruktor erzeugt wurde, wird die Verbindung zur ODBC-Schnittstelle mit einer Funktion `Open()` hergestellt um eine Verbindung zur Datenbank zu erreichen.

Ein `Close`-Aufruf löst einen `RollBack()` für alle laufenden Aktionen und Transaktionen aus, d.h. dass mit der `RollBack()` Funktion alle Änderungen, die während der Transaktion durchgeführt worden waren, zurückgesetzt werden.

5.4.2 CRecordset-Klasse

Die `CRecordset`-Klasse stellt eine Anfrage an das DBMS⁷ und verwaltet die Antwort. Mit einem `CRecordset`-Objekt kann man in der Antwortmenge navigieren und einzelne Elemente modifizieren.

⁷ Datenbank Management System

Der Konstruktor benötigt in dieser Klasse die Adresse der `CSpreadSheet`-Variablen.

```
CRecordset::CRecprdset(CSpreadSheet *pDatabase = NULL);
```

Durch `Open()` wird die `CRecordset`-Klasse mit einem SQL-Befehl verbunden.

6 Implementierung des Programms

6.1 Entwicklungsumgebung und Bibliotheken

Die Wahl der Programmiersprache war zur freien Wahl gestellt, da noch keine schon vorprogrammierten Schnittstellen vorhanden waren. Das Programm wird objektorientiert in der Sprache C++ unter der Entwicklungsumgebung Visual Studio C++ 6.0 geschrieben. Bei der Wahl der Sprache zur Programmimplementierung, fiel die Entscheidung zu Gunsten von C++ Sprache, da sie eine der mächtigsten und ausdrucksstärksten Programmiersprachen ist. Aufbauend auf C, einer konventionellen Programmiersprache, bietet C++ zusätzlich objektorientierte Sprachmerkmale und ermöglicht den Zugriff auf die Bibliotheken, die für die Lauffähigkeit des Programms von Bedeutung sind.

Im Programm werden Bibliotheken wie `odbcinst.lib`, `odbc32.lib` und `odbccp32.lib` eingebunden, um die Verbindung zu einer Datenbank herzustellen und bestimmte Operationen auf Datensätzen durchzuführen. Den Zugriff auf eine Datenbank ermöglicht das Einbinden einer Header-Datei `afxdb.h`.

Die Funktionen von C-Standard-Bibliothek (STL⁸) werden ebenso benutzt. Dabei werden Bibliotheksmodule wie `<sstream>`, `<vector>`, `<map>`, `<string>` usw. verwendet.

6.1.1 Vektoren in C++

Im Programm wird mit einem Modul `<vector>` gearbeitet. Dabei wird jeder einzelne Flieger mit allen seinen Flugspurpunkten und dazu gehörigen Daten in ein dynamisches Container-Template abgespeichert.

`<Vector>` ist ein eigener Datentyp in C++, der für die Verwaltung von Vektoren verantwortlich ist. Er bietet einige Vorteile gegenüber normalen Feldern in C: Bei diesem Bibliotheksmodul kann die Größe eines Feldes dynamisch während Laufzeit festgelegt und angepasst werden. Dabei lässt sich die Anzahl der wirklich benutzten Elemente des Feldes während der Laufzeit bestimmen.

Die dynamische Verwaltung des Speichers wird dazu benutzt, um den Arbeitsspeicher sparsam zu verwenden.

6.1.2 Maps in C++

Die Menge aller Flieger wird in einem `<map>` verwaltet. Ein Map ist ein sortierter assoziativer Container in STL. Die in einem Map enthaltenen Elemente sind Wertepaare, die jeweils aus Key und Value bestehen. Dabei ist aber zu beachten, dass jeder Key in einer Map nur einmal enthalten sein darf.

```
typedef map<int, acftVec> aircraftMap; //ein Map
//die Keys sind vom Typ "int",
//die values vom Typ „acftVec“
```

6.2 Aufbau des Programms

Das Programm besteht aus den drei Klassen `CSpreadSheet`, `CGauss_Krueger` und `CSchnitt_Gerade_Kreis`.

In der `CSpreadSheet`-Klasse sind die Methoden zur Manipulation einer Exceltabelle mit C++ enthalten. Die Klasse `CGauss_Krueger` liefert die Funktionen zur Umrechnung von geographischen in die Gauß-Krüger-Koordinaten. Die `CSchnitt_Gerade_Kreis`-Klasse umfasst die unterschiedlichen Algorithmen für die im Programm wichtigen Berechnungen.

⁸ Standard Template Library

Die korrekte Arbeitsweise des Programms und der durchgeführten Berechnungen wurde durch Ergebnisvergleiche mit der Exceltabelle und eines im Internet gefundenen Tools zum Gauß-Krüger-Verfahren (Siehe Abbildung 8) bestätigt (WWW01).

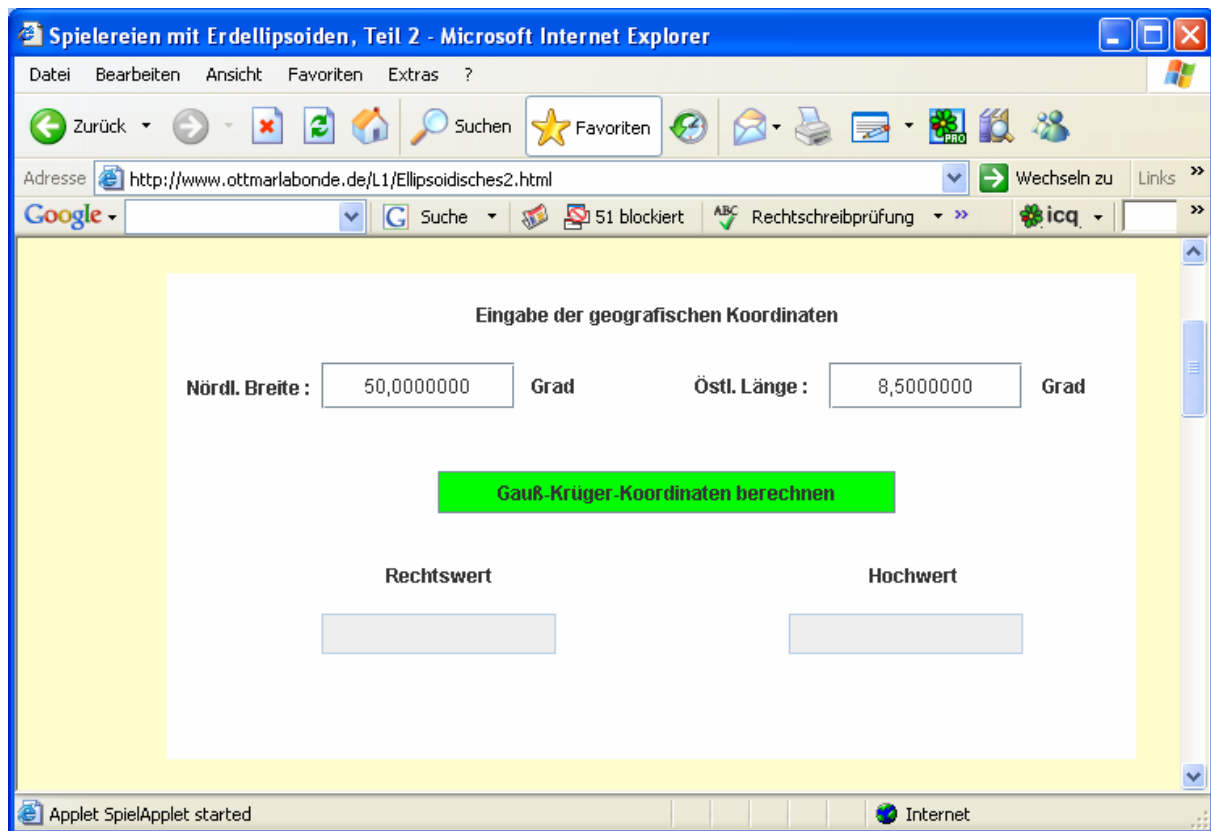


Abbildung 8: Oberfläche zur Berechnung der Gauß-Krüger-Koordinaten (WWW01)

Nach dem Starten des Programms werden die Daten zuerst aus einer Exceltabelle ausgelesen. Als nächstes werden die Berechnungen mittels dieser Daten durchgeführt und als Ergebnis in einer *.txt-Datei abgelegt. Die Berechnungsvorgänge werden in den folgenden Kapiteln erläutert.

6.2.1 CSpreadSheet-Klasse

Die Hauptaufgabe der `CSpreadSheet`-Klasse ist das zellen-, reihen- und spaltenweise Auslesen aus einer Exceltabelle.

Jede Zeile und Spalte in der Exceltabelle darf nur einmal vorkommen, da die entsprechenden Daten sonst überschrieben werden.

Die Klasse benutzt eine ODBC-Schnittstelle und erfordert die Verwendung der MFC (Microsoft Foundation Class).

Mit dem Konstruktor `CSpreadSheet()` wird ein Excel-File (*.xls) oder eine *.txt (*.csv)-Datei zum Auslesen oder in die Datei schreiben geöffnet. Falls es sich um eine Exceldatei handelt, wird die interne Flag `m_bExcel` auf `true` gesetzt. In der Variable `SheetOrSeparator` wird der Name vom Excel-Sheet oder, falls die Datei ein Textfile ist, der Typ des Separators, gespeichert. Der Name der Datei wird dann in der Variable `m_sFile` abgelegt.

Die Methode `ReadRow()` liest eine Reihe aus einer geöffneten Exceldatei ein. Falls eine Reihe leer ist, dann wird zur nächsten Reihe übergegangen und diese eingelesen. Durch diese Methode wird jede Reihe zunächst in einzelne Spalten aufgeteilt.

Mit der Methode `ReadColumn()` können Spalten aus Exceldateien eingelesen werden. Dafür stehen zwei verschiedene Möglichkeiten zur Verfügung:

- `bool ReadColumn(CStringArray &ColumnValues, short column)`
liest die gewünschte und durch die Variable `column` bezeichnete Spalte aus der geöffneten Exceldatei ein.
- `bool ReadColumn(CStringArray &ColumnValues, CString column, bool Auto=true)`
hat fast die gleiche Funktionsweise, allerdings gilt die Variable `column` für jeden Excelspaltennamen (z.B. A, B, usw.) oder den Feldnamen der Header-Reihe. Die Variable `Auto` wird auf `false` gesetzt, wenn es nicht erwünscht ist, dass die Methode automatisch ermittelt, ob unter der Variable `column` ein Name der Excelspalte oder der Name der Header-Reihe der Exceltabelle zu verstehen ist (für den Fall, dass der Feldname der Header-Reihe aus zwei Buchstaben besteht). Die `return`-Anweisung zeigt an, ob das Einlesen aus einer Spalte erfolgreich gewesen ist.

Die Methode `ReadCell()` liest eine einzelne Zelle aus einer geöffneten Exceltabelle ein. Falls die Zelle leer ist, dann wird zur Zelle der nächsten Reihe übergegangen.

Diese Funktion `BeginTransaction()` ist einer entsprechenden SQL-Funktion ähnlich. Sie wird benutzt, um den Start eines Datentransfers einzuleiten. Dabei wird die interne Flag zur Bestimmung des Transaktionsstatus auf `true` gesetzt. Aus der Definition von Transaktionen ergibt sich ihr grundsätzlicher Aufbau. Transaktionen müssen einen Anfang haben, über ein oder mehrere (SQL-) Befehle verfügen und einen definierten Endpunkt haben. Man beschreibt eine Transaktion durch den Begriff ACID: Es ist die Abkürzung der Begriffe Atomicity, Consistency, Isolation und Durability. Dabei werden folgende Eigenschaften von Transaktionen gemeint:

- Atomicity/Atomarität: Bezeichnet die Eigenschaft von Transaktionen, d. h. ob die Transaktionen komplett sind oder gar nicht ausgeführt werden können.
- Consistency/Konsistenz: Eine Transaktion überführt eine Datenbank von einem konsistenten Zustand in einen anderen konsistenten Zustand. Jeder Befehl wird vollständig ausgeführt.
- Isolation: Eine Transaktion wird in ihrer Ausführung nicht durch parallel ausgeführte Befehle beeinträchtigt. Das Ergebnis einer Transaktion ist immer konstant.
- Durability/Dauerhaftigkeit: Nach einer erfolgreichen Ausführung einer Transaktion wird das Ergebnis dauerhaft in der Datenbank gespeichert.

6.2.2 Klasse `CGauss_Krueger`

Die Berechnungsfunktionen der Klasse `CGauss_Krueger` werden nach einem Gauß-Krüger-Algorithmus aufgebaut.

Wie oben beschrieben, werden die Eingangskordinaten in geographischer Form geliefert, d.h. dass jeder Flugspurpunkt von jedem Flieger eine Position in Länge (LON) und Breite (LAT) zugewiesen bekommt. Um aber mit diesen Werten mathematische Operationen durchzuführen, muss man die Koordinatenebene festlegen und die geographischen Koordinaten in zweidimensionale Daten umrechnen können.

Die theoretischen Grundlagen des Gauß-Krüger-Verfahrens sind im Anhang A dargestellt.

Ein Flieger (Flugzeug-ID), mit allen seinen Flugspurpunkten, soll als Objekt jeweils eines Koordinatensystems betrachtet werden. Dabei ist zu beachten, dass die Umrechnung in Gauß-Krüger-Koordinaten für jeden einzelnen Flug getrennt erfolgt, da der Bezugsmeridian gleich bleiben muss, um ein einziges Koordinatensystem beizubehalten. Nachdem der Bezugsmeridian für den Endflughafen errechnet wird, wird dieser Wert für die restlichen Flugspurpunkte als Bezugsniveau genommen.

6.2.3 Klasse `CSchnitt_Gerade_Kreis`

Diese Klasse `CSchnitt_Gerade_Kreis` ist eine Basis für die Berechnungen, die mit dem Schnittpunkt von einer Geraden und dem Kreis zusammenhängen. Die Gerade wird dabei durch die beiden Flugspurpunkte gelegt, die den geringsten Abstand zum Kreis aufweisen, wobei ein Punkt innerhalb

und der andere außerhalb des Kreises liegt. Diese beiden Punkte werden durch Vergleich des Abstandes aller zu einem Flieger zugehörigen Flugspurpunkte mit dem Radius des Kreises gewonnen.

Der Schnittpunkt Gerade-Kreis wird folgendermaßen berechnet: Aus einer Geradengleichung folgt:

$$x = x_1 + s \cdot (x_2 - x_1)$$

$$y = y_1 + s \cdot (y_2 - y_1)$$

Die Kreisgleichung lautet:

$$(x - x_0)^2 + (y - y_0)^2 = R^2, \text{ wobei } x_0/y_0 - \text{Flughafenkoordinaten}$$

$$x_1/y_1 - \text{Äußerer Punkt}$$

$$x_2/y_2 - \text{Innerer Punkt}$$

Die Werte aus der Geradengleichung werden in die Kreisgleichung eingesetzt und der Parameter s wird berechnet. Anschließend werden die Koordinaten des Schnittpunktes Gerade-Kreis berechnet.

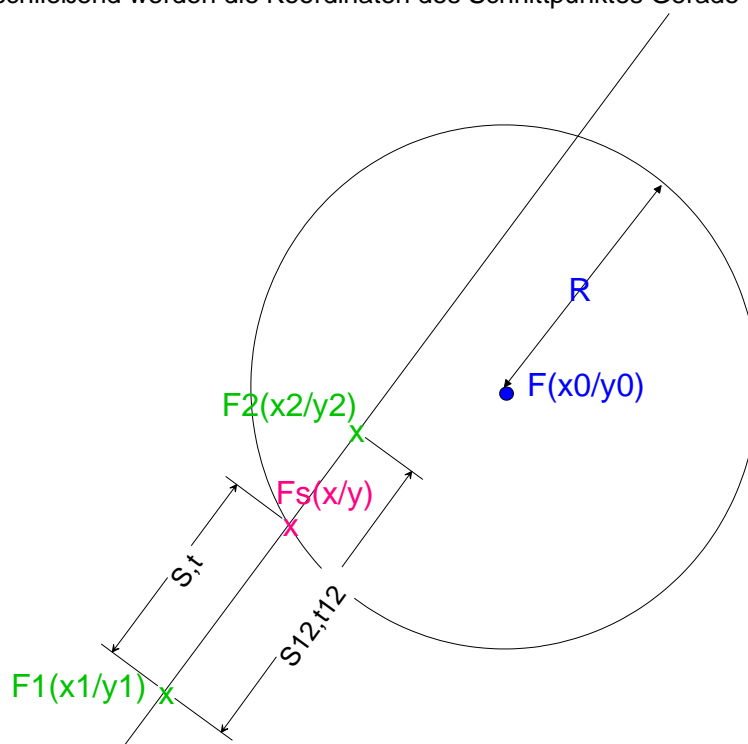


Abbildung 9: Berechnungsskizze

Nachdem die Koordinaten des Schnittpunktes berechnet wurden, wird die Proportion $\frac{\text{Strecke}}{\text{Zeit}}$ ermittelt und die Ankunftszeit am Schnittpunkt bestimmt: Das Verhältnis des Weges S_{12}

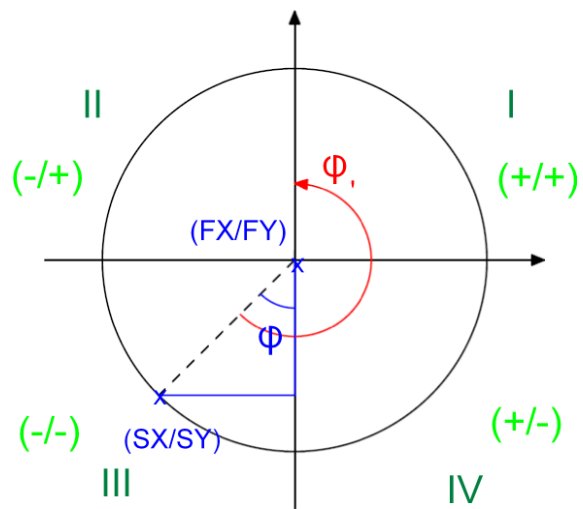
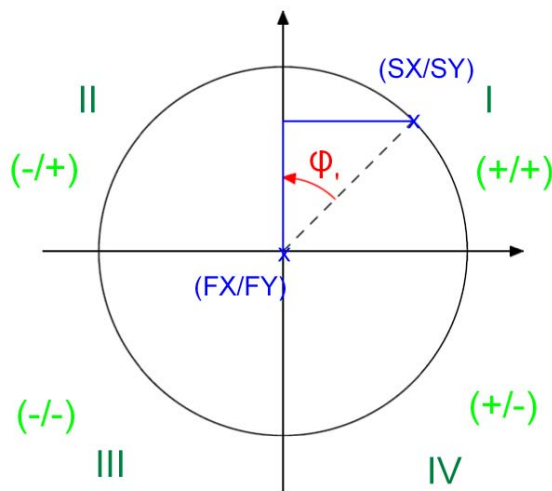
($S_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$) (vom Flugspurpunkt F1 zum Flugspurpunkt F2) zum Weg S ist gleich dem Verhältnis der beiden Zeiten, die das Flugzeug für diese Strecken benötigt, d. h.

$$\frac{S_{12}}{S} = \frac{t_{12}}{t}. \text{ Aufgelöst nach } t \text{ ergibt sich die gesuchte Zeit: } t = \frac{S \cdot t_{12}}{S_{12}}$$

6.2.4 Klasse Winkelberechnung

Flieger kommen aus unterschiedlichen Richtungen in das betrachtete Gebiet. Dabei ist der Winkel zwischen der Strecke Einflugpunkt-Mittelpunkt und der positiven y -Achse für die Verkürzung der Pufferzeit von Bedeutung. Um diesen Winkel zu bestimmen, wird nach dem folgenden Verfahren vorgegangen: Der Kreis wird in vier Quadranten unterteilt, wobei die x - und y -Koordinaten der Flieger im ersten Quadrant beide positiv sind, im zweiten Quadrant ist die x -Koordinate negativ und y -Koordinate positiv, im dritten Quadrant sind die beiden Koordinaten negativ, und im vierten bekommen die x -Werte positive und die y -Werte negative Vorzeichen. Anhand dieser Vorzeichenregelung wird bestimmt, in welchem Quadrant sich der Flieger gerade befindet.

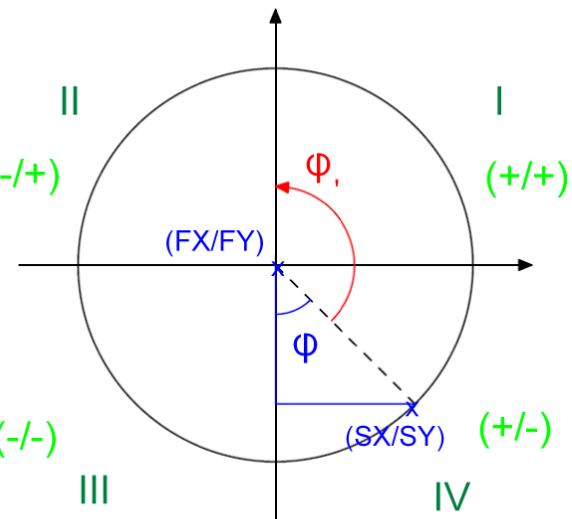
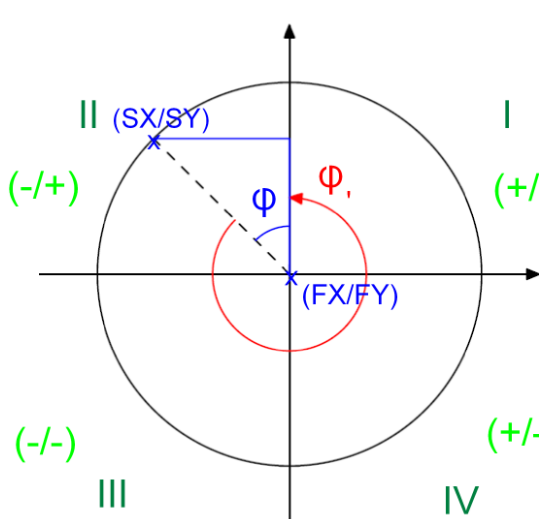
Die Position des Flugzeugs wird gemäß der Berechnungen im Kapitel 6.2.3 genau bestimmt. Die Koordinaten des Endflughafens sind ebenso bekannt. Anhand dieser Daten berechnet man den Winkel φ im rechtwinkligen Dreieck. Um den gesuchten Winkel φ zu bestimmen, wird folgende Methode angewandt:



$$\varphi_1 = \arctan \frac{SX - FX}{SY - FY}$$

$$\varphi = \arctan \frac{SX - FX}{SY - FY}$$

$$\varphi_1 = \varphi + 180^\circ$$



$$\varphi = \arctan \frac{SX - FX}{SY - FY}$$

$$\varphi_1 = 360^\circ - \varphi$$

$$\varphi = \arctan \frac{SX - FX}{SY - FY}$$

$$\varphi_1 = 180^\circ - \varphi$$

7 Fazit

Im Laufe der Praxisarbeit wurde ein Programm entwickelt, mit dem es möglich ist, die für die Zeitvariabilitätenverminderung wichtigen Daten zu gewinnen. Diese Daten wurden mit den Prognosedaten verglichen und haben sich in der Testphase als zutreffend erwiesen.

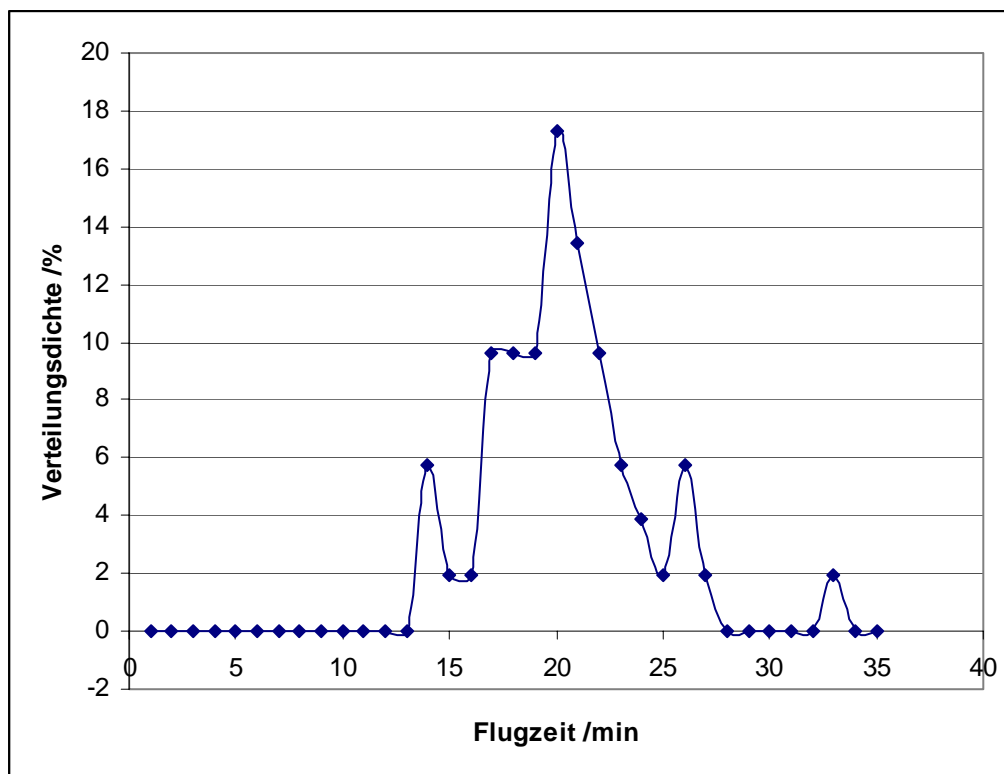


Abbildung 10: Histogramm der Flugzeitverteilung

In Abbildung 10 wird die tatsächliche Verteilung der Flugzeiten, die man mit den aus dem Programm gewonnenen Daten bekommt, dargestellt. Der Verlauf entspricht tendenziell dem des statistischen Histogramms in Abbildung 2. Die einzelnen Unregelmäßigkeiten entstehen durch eine relativ geringe Anzahl von Flügen im Beispieldatensatz.

Die Länge der erforderlichen Zeitpuffer soll anhand der gewonnenen Daten vermindert werden, da sie von der Variabilität der Flugzeiten abhängig ist.

Für jeden Flug lässt sich der Zeitintervall zwischen dem Einflug in die ASMA-Zone und der Landung bestimmen. Zudem lassen sich die für die Position des Fliegers wichtige Richtung (Sektor: Nord, Nord-Ost, Ost, Süd-Ost, Süd, Süd-West, West oder Nord-West) sowie der Einflugwinkel bestimmen.

Im weiteren Verlauf des Projektes sollen die mit dem Programm gewonnenen Daten zur Verminderung der Pufferzeiten beitragen, wobei die kontinuierlicheren und planmäßigeren Flüge ein Potential zur erhöhten Effizienz und Umweltfreundlichkeit im flughafennahen Bereich aufweisen. Dies bedeutet gleichzeitig, dass die Flughafenkapazitäten besser und effektiver genutzt werden können.

8 Literatur

- [1] Kahmen, Heribert: Vermessungskunde II; Verlag: Walter de Gruyter; 1. Auflage (1986)
- [2] Balzert, Helmut: Lehrbuch der Software-Technik; Akademischer Verlag: Spektrum (1998)
- [3] Kernighan, Brian W.: Programmieren in C; Verlag: Hanser; 1990
- [4] Bronstein, I. N., Semendjajew, K. A., Musiol, G.; Taschenbuch der Mathematik; Verlag: Harry
- [5] Solf, Martin: Einflüsse auf die Variabilität von Anflugzeiten bei Verkehrsflugzeugen; Diplomarbeit; Dezember 2005
- [6] EUROCONTROL: Report on Punctuality Drivers at Major European Airports; Brüssel: 2005
- [7] Knabe, F; Schmitz, R; Temme, A: Variability of Arrival Flight Times (2006), DLR IB in Vorbereitung

9 Internetquellen

- (WWW01) Spielereien mit Erdellipsoiden: <http://www.ottmarlabonde.de/L1/Ellipsoidisches2.html>
- (WWW02) Skyguide Glossary, TMA: <http://www.skyguide.ch/de/Tools/Glossary/glossaryT.xml>
- (WWW03) Wikipedia-Histogramm: <http://de.wikipedia.org/wiki/Histogramm>
- (WWW04) Mercator Projektion: <http://www.explorermagazin.de/gps/grids.htm>
- (WWW05) Kartenprojektion: <http://www.kowoma.de/gps/geo/Projektionen.htm>
- (WWW06) Meridianstreifensystem: <http://www.geog.fu-berlin.de/~jkrywkw/eva/koord1.html#Merkator%20Projektion>

10 Bildverzeichnis

Abbildung 1: Aufteilung von Flugzeiten [6].....	5
Abbildung 2: Histogramm der Flugzeitverteilung in der letzten Flugphase (Beispiel) [7].....	6
Abbildung 3: <i>Arrival Sequencing and Metering Area (ASMA)</i> um den Flughafen London-Gatwick [7].....	7
Abbildung 4: Ausschnitt der Exceltabelle mit den Eingabedaten.....	8
Abbildung 5: Zugriff auf eine Datenbank über ODBC-Schnittstelle	9
Abbildung 6: ODBC-Datenquellen-Administrator	10
Abbildung 7: MFC-Klassen.....	11
Abbildung 8: Oberfläche zur Berechnung der Gauß-Krüger-Koordinaten (WWW01)	13
Abbildung 9: Berechnungsskizze	15
Abbildung 10: Histogramm der Flugzeitverteilung	18
Abbildung 11: Transversale Mercator Projektion (Zylinderprojektion) (WWW04).....	20
Abbildung 12: Meridianstreifensystem in Deutschland [1]	21

11 Anhang A

11.1 Transversale Mercator Projektion

Als Grundlage für Navigationskarten dient noch heute das 1569 von Gerhard Mercator entwickelte und nach ihm benannte Projektionsverfahren.

Bei einer Zylinderprojektion sind die Verzerrungen dort am geringsten, wo der Zylinder die Erdoberfläche berührt.

Daher wird bei topografischen Karten der Projektionszylinder um 90° gekippt. Er berührt den Erdkörper an einem Längengrad, dem sog. Bezugsmeridian. Eine definierte Fläche zu beiden Seiten des Bezugsmeridians, der sog. Meridianstreifen wird dann auf dem Zylinder abgebildet. Je schmaler der abzubildende Meridianstreifen gewählt wird, desto geringer fallen nachher die Verzerrungen aus.

Mit dieser Methode lässt sich die dreidimensionale Oberfläche des Erdkörpers weitgehend flächentreu in der Ebene abbilden. Die geografischen Meridiane erscheinen später auf der Karte als gerade Strahlenbüschel mit dem Pol als Mittelpunkt. Die Breitenkreise werden als konzentrische und parallele Teilkreise mit dem Pol als Mittelpunkt abgebildet. Die Verzerrung wächst jedoch mit zunehmender nördlicher/südlicher Breite. Die transversale⁹ Mercator-Projektion kommt daher für eine Abbildung der Polregionen nicht in Betracht

Die Transversale Mercator Projektion wird auch als querachsige Zylinderprojektion bezeichnet (WWW04), (WWW05).

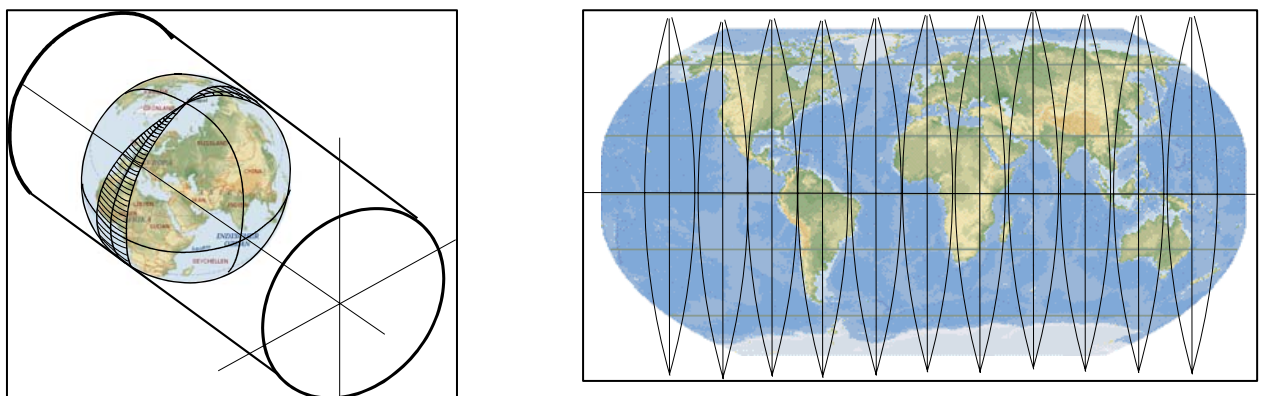


Abbildung 11: Transversale Mercator Projektion (Zylinderprojektion) (WWW04)

11.2 Gauß-Krüger-Koordinaten

Das Gauß-Krüger-System gehört zu den bedeutendsten Bezugssystemen in Deutschland und in Europa.

Die transversale Mercator Projektion diente als Grundlage der von Carl Friedrich Gauß in den Jahren 1827 bis 1860 durchgeführten topografischen Landesaufnahme des Landes Hannover. Johann Heinrich Louis Krüger (1857-1923) hat die Idee der Gaußschen Abbildung auf die Oberfläche des Rotations-Ellipsoids übertragen und dies in einer Arbeit mit dem Titel "Konforme Abbildung des Erd-Ellipsoids in der Ebene" im Jahre 1912 vorgestellt. Das Projektionsverfahren ist die Grundlage vieler moderner topografischer Kartenwerke und wurde 1927 als amtliche Vermessungsmethode in Deutschland eingeführt.

⁹ Quer zur Körperachse verlaufend

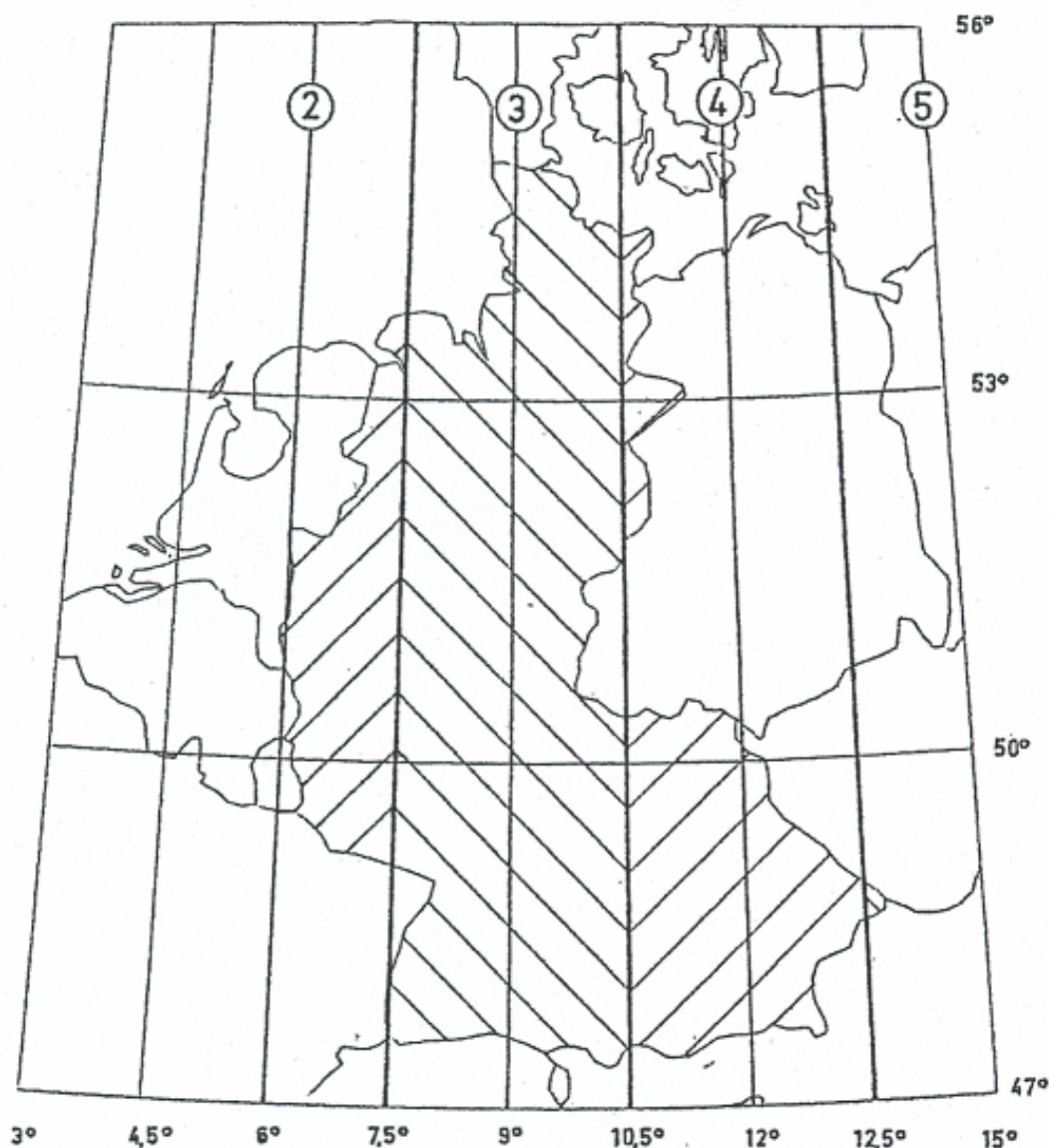


Abbildung 12: Meridianstreifensystem in Deutschland [1]

Die Gauß-Krüger-Projektion ist eine querachsige Zylinderprojektion mit einem abgebildeten Meridian. Für das amtliche Kartenwerk ist eine Streifenbreite von 3 Grad mit den Hauptmeridianen der Längengrade 6°, 9° und 12° festgelegt (siehe Abbildung 12). Alle Berechnungen werden auf einen Bessel-Ellipsoid bezogen (WWW06).

Bei der Abbildung eines solchen Meridianstreifens wird der Hauptmeridian auf die x- Achse eines ebenen Koordinatensystems abgebildet, der Abstand eines Punktes vom Hauptmeridian wird auf der Ordinate abgebildet. Um negative Ordinaten zu vermeiden, werden zur Ordinate 500 km addiert und als so genannter Rechtswert angegeben, der den Wert

$$\text{Rechtswert (Ordinate)} = R = N \cdot 10^6 + 500000 + Y \text{ [m]}$$

annimmt. Dabei ist N die Kennziffer der Streifennummer, die durch Division der Gradzahl des Hauptmeridians durch die Meridianstreifenbreite (hier 3°) entsteht. Die Abszissen x werden vom Äquator aus auf dem Hauptmeridian gemessen und heißen Hochwerte.

$$\text{Hochwert (Abszisse)} = X$$

12 Anhang B

