Studienarbeit

angefertigt am Fachgebiet Strömungsmaschinen der Universität Gesamthochschule Kassel

Entwurf von Schaufeln im Netzwerk

Betr.: Prof. Dr.-Ing. M. Lawerenz
Dipl.-Inform. M. Uelschen

Verfasser: Michael Pfütz
Matr.-Nr.: 72 31 70

Inhaltsverzeichnis

1 Einleitung **1**

2 Verteilte Anwendungen im World Wide Web **2**

2.1 Grundlagen des Internets 2

2.1.1 Die Entstehung des Internets 2

2.1.2 Kommunikation der Rechner im Internet 4

2.1.3 World Wide Web 4

2.2 Dynamisch verteilte Anwendungen 7

2.3 Auswahl der Programmiersprache 8

2.4 Java 9

2.4.1 Die Java Virtuelle Maschine 9

2.4.2 Die Zugriffsrechte 11

2.4.3 Java Programm-Beispiel 11

2.5 Beispiele aus dem Internet 17

3 Entwurf von Schaufeln **25**

3.1 Zweidimensionaler Entwurf 25

3.1.1 Mathematische Grundlagen des B–Splines 25

3.1.2 Schwerpunktberechnung des B–Splines 26

3.2 Dreidimensionaler Entwurf 29

4 Zusammenfassung **30**

4.1 Schwierigkeiten 30

4.2 Ausblicke 31

A Bedienungsanleitung **32**

A.1 Hardware-Voraussetzungen 32

A.2 Software-Voraussetzungen 32

A.3 Bedienung des Turbo-Designers 33

A.3.1 Die Arbeitsfläche und deren Elemente 34

A.3.2 Die Befehle im *File*-Menü 36

A.3.3 Die Befehle im *View*-Menü 39

A.3.4 Die Befehle im *Option*-Menü 41

A.3.5 Die Befehle im *Help*-Menü 41

A.3.6 Übersicht Tastenkürzel 43

A.3.7 Die Bedienung des Fensters *Preferences* 44

A.3.8 Die Informationen im Fenster *Properties* 44

A.3.9 Konventionen der Turbo-Designer-Dateien 46

A.4 Die Bedienung des Turbo-Viewers 47

A.4.1 Der *Netloader* 47

A.4.2 Der 3D-Viewer 48

A.5 Einstellungen der Zugriffsrechte im Sun-HotJava-Browser . . . 50

Abbildungsverzeichnis

2.1 Darstellung der Beispiel HTML-Datei 7
2.2 Zusammenspiel des Servers mit dem Client 10
2.3 Das Java-Applet `Hallo.class` 11
2.4 Die Baumstruktur der Pakete 14
2.5 Die Ausgabe des Java-Applets `Hallo.class` 17
2.6 Profil mit Skelettlinie 18
2.7 Das Java-Applet zur Bestimmung von NACA Profilen 19
2.8 Das Java-Applet zur Simulation eines zweidimensionalen Windkanals (Konturansicht) 20
2.9 Das Java-Applet zur Simulation eines zweidimensionalen Windkanals (Linendarstellung) 20
2.10 AnswerWeb Home page for Airfoil 22
2.11 Darstellung eines Mediums um ein Profil 23
2.12 Auswahl der verschiedenen Darstellungen 23
2.13 Eingabeformular zur Berechnung eines NACA Profils und deren cp-Werte 24
2.14 Darstellung eines Profils der NACA-Serie 24

3.1 Flächen der Dreiecke 29

A.1 Der Turbo-Designer 33
A.2 Arbeitsfläche und deren Elemente 34
A.3 Der im Turbo-Designer repräsentierte Querschnitt durch das Profil einer Axialmaschine 35
A.4 Das File-Menü 36

A.5 Die Datei-Auswahbox (Netloader) 37
A.6 Das Linux-übliche Menü zum Datei laden 38
A.7 Das Windows95-übliche Menü zum Datei speichern 38
A.8 Das *View*-Menü 39
A.9 Das *Options*-Menü 41
A.10 *Help*-Menü 41
A.11 Fenster *Information about* 42
A.12 Fenster *Basics* 43
A.13 Das Fenster Preferences 45
A.14 Das Fenster Properties 45
A.15 Netloader 48
A.16 3D-Viewer 49
A.17 Menü für die Einstellungen der Zugriffsrechte 51
A.18 Button *Advanced* >> 51
A.19 Menü zum Hinzufügen des Domainnamens 52
A.20 *Advanced Security Settings*: System Zugriff 52
A.21 *Advanced Security Settings*: Datei Zugriff 53
A.22 Warnmeldung 54

Kapitel 1

Einleitung

Zur weiteren Verbesserung von Turbomaschinen bedarf es speziell angepaßter Beschaufelungen. Zum geometrischen Entwurf wurden in der Vergangenheit C-, NACA-, Kreisbogen-, Doppelkreisbogen (DCA)- und Mehrfachkreisbogen (MCA)-Profile eingesetzt. Moderne Ansätze verwenden Spline-Funktionen. Mit Hilfe dieser Spline-Funktionen können Beschaufelungen individuell für eine bestimmte Anwendung am Computer entworfen werden.

Die Idee dieser Studienarbeit besteht darin, ein interaktives Entwicklungswerkzeug für Schaufelprofile in der plattformunabhängigen Sprache Java zu entwickeln. Diese bietet den Vorteil gegenüber den meisten anderen Programmiersprachen, daß sie nur einmal kompiliert werden muß. Sie kann auf den unterschiedlichsten Rechnersystemen eingesetzt werden.

In Rahmen dieser Studienarbeit sind zwei Module (sogenannte Java Applets) entworfen und implementiert worden. Beim ersten Modul (Turbo-Designer) handelt es sich um einen graphischen Editor zum interaktiven Entwurf von zweidimensionalen Profilschnitten mit periodischen B–Splines. Dieser kann charakteristische geometrische Größen der Beschaufelung berechnen und anzeigen. Beim zweiten Modul (Turbo-Viewer) handelt es sich um eine Anwendung zur dreidimensionalen Darstellung von Schaufeln mit Tensorprodukt-B–Spline-Flächen.

Kapitel 2

Verteilte Anwendungen im World Wide Web

2.1 Grundlagen des Internets

2.1.1 Die Entstehung des Internets

Das Internet (Abk. für International Network) ist aus einer vom US-Verteidigungsministerium im Jahre 1969 in Auftrag gegebenen Entwicklung, dem ARPANET (Advance Research Projects Agency Network), hervorgegangen. ARPANET war als dezentrales militärisches Netz und Forschungsnetz konzipiert, das im Kriegsfall auch bei Ausfällen von Verbindungen und Knoten (einzelner Computer, der an das Netz angeschlossen ist) weiter funktionieren sollte. Es sollte auch die Zusammenarbeit verschiedener Forschungsstellen erleichtern und verband zunächst nur die Universitäten von Kalifornien in Los Angeles, das Stanford Research Institute, die Universität von Kalifornien in Santa Barbara und die Universität in Utah, wobei es nur Forschern zur Verfügung stand, um sich auf anderen Computern einzuloggen. Bald schon wurden Methoden zur Dateiübertragung und zur elektronischen Nachrichtenübermittlung (Email) in das System eingebaut. Damit das Netz auch funktionsfähig bleibt, wenn eine oder mehrere Leitungen zerstört werden, muß das System automatisch auf andere umschalten und auf diese Weise selbständig eine Verbindung zwischen zwei Orten herstellen können. Dieser Vorgang wird als Dynamic Rerouting bezeichnet.

Im Zuge des Aufkommens anderer Netzwerke entstand der Bedarf, eine Möglichkeit zu finden, unterschiedliche Systeme so zu verbinden, daß diese un-

eingeschränkt miteinander kommunizieren können. Unter dem Namen Internetting Project gab die DARPA (der neue Name der ARPA; das D steht für Defense) in den 70er Jahren die Entwicklung einer entsprechenden Methode in Auftrag.

Die Lösung dieses Problems bestand in den systemunabhängigen Übertragungsprotokollen TCP/IP (Abk. für Transmission Control Protocol/Internet Protocol), die auf alle am Netz angeschlossenen Rechnern installiert werden. Ein Protokoll besteht aus einer Reihe von Regeln, die festlegen, wie Daten zwischen zwei Rechnern ausgetauscht und eventuell bei der Übertragung aufgetretene Fehler beseitigt werden können. TCP ist für den Auf- und Abbau der Verbindung zwischen den einzelnen Arbeitsstationen im Netzwerk zuständig. Es steuert den Datenfluß und stellt die Vollständigkeit der Übertragung sicher. IP ist für die Organisation und Adressierung der Daten zuständig. Dazu werden die Daten für die Übertragung in Datenpakete aufgeteilt und beim Empfänger wieder zu den Gesamtdaten zusammengefügt.

Im Laufe der Zeit wurden immer mehr Netze (LANs[1]) mit dem Internet verbunden. Hierauf ist die heutige Struktur zurückzuführen. Das Internet setzt sich aus vielen, voneinander unabhängigen Netzen zusammen. Eine zentrale Verwaltung oder Koordination gibt es nicht. Jedes Teilnetz ist technisch und organisatorisch sowie finanziell unabhängig (wenn es nicht gerade von einer übergeordneten Stelle finanziert wird). Das Internet verbindet nicht einzelne Rechner direkt miteinander, sondern schafft Kommunikationswege für die nächstgrößere Einheit, das sind in sich geschlossene Gruppen von Computern. Über eine solche Gruppe - ein LAN - können alle dort angeschlossenen Computer Teil des Internets werden. Jeder Computer im Internet hat eine eindeutige IP-Adresse, unter der dieser alleine zu erreichen ist.

Das Internet ist die Menge aller Netzwerke, die unter TCP/IP arbeiten und miteinander in Verbindung stehen. Allerdings gibt es auch im Internet Netzwerke, die nicht mit TCP/IP arbeiten, deren Daten werden von einem Gateway in das jeweils benötigte Format übersetzt.

Für die verschiedenen Aufgaben, die über das Internet ausgeführt werden können, stehen eine Reihe von sogenannten Diensten (Services) zur Verfügung. Meist ist der Name des jeweiligen Dienstes mit dem Namen des zugehörigen Programms identisch, z. B. Email (Nachrichten versenden), Telnet (interaktiver Zugriff auf einen entfernten Rechner) und FTP (Dateien zwischen zwei Computern übertragen) (vergl. Grieser Irlbeck in [4], April in [5]).

Nachdem die Benutzeroberfläche jahrelang eher spartanisch war, zum Aufruf

[1] Local Area Network: Räumlich begrenzte Netzwerke

von Programmen und Funktionen war die Eingabe von Unix (Betriebssystem) -Befehlen erforderlich, sind seit Anfang der 90er Jahre Programme verfügbar, die über Menüs gesteuert werden. Inzwischen gibt es die wichtigsten Dienstprogramme für die verschiedensten grafischen Betriebssystem-Oberflächen.

2.1.2 Kommunikation der Rechner im Internet

Das Terminal-Modell Beim Terminal-Modell wird nur die Rechenleistung eines zentralen Computers, häufig ein Großrechner, verwendet. Die Terminals dienen nur der Datenein- und -ausgabe. Es sind an ihm bis zu mehrere hundert Terminals angeschlossen. Mit Hilfe von Terminalemulationen können auch andere Rechner sich wie ein Terminal verhalten.

Das Client/Server-Modell Durch das Client/Server-Modell werden dynamisch verteilte Anwendungen ermöglicht. Es werden für die Befehlseingabe und den Zugriff auf Systemressourcen zwei miteinander kommunizierende Programme verwendet. Bei dieser Aufgabenstellung bezeichnet man das Programm, das die Befehle des Benutzers entgegennimmt, als Client. Dieses befindet sich auf dem Rechner des Benutzers und nimmt auch dessen Rechenleistung in Anspruch.

Das entsprechende Serverprogramm befindet sich auf dem Computer, auf dem alle zur Verfügung stehenden Informationen gespeichert sind. Es nimmt den Auftrag seines Clients entgegen, bearbeitet ihn und schickt dann die geforderten Informationen wieder mit Hilfe eines Übertragungsprotokolls zurück. Meistens handelt es sich hierbei um einfache Dateizugriffe. Aber auch Mischformen sind möglich, bei denen die Rechenleistung beider Systeme gefragt sind. Die Bezeichnung Server wird sowohl für den File-Server, also einen Rechner, als auch für Software verwendet.

2.1.3 World Wide Web

Ein wesentlicher Bestandteil des Internets ist das World Wide Web (Abk: WWW; wörtlich: weltweites Netz), ein gigantisches Hypermedia-System (in Anlehnung an den Begriff Multimedia). Über dieses läßt sich auf Dokumente zugreifen, die auf einem beliebigen Host oder Server im WWW abgelegt sind. Das WWW wurde 1993 am europäischen Forschungszentrum für Kernphysik CERN in Lausanne (Schweiz) entwickelt.

Das WWW stellt textuelle, graphische und auditive Informationen sowie Interaktionsmöglichkeiten auf WWW-Seiten zur Verfügung, auf die über graphische Benutzeroberflächen, sogenannte WWW-Browser, zugegriffen wird. Diese Seiten werden durch die Auszeichnungssprache HTML (Hypertext Markup Language) beschrieben. So wird mit ihr die Struktur und darin enthaltene Verweise (Links) auf andere Dokumente im WWW des darzustellenden Dokuments beschrieben. Diese Struktureigenschaften betreffen z. B. Titel, Absätze, Tabellen, Bilder, Eingabe- und Auswahlfelder, Schaltflächen, Tondateien und sonstige Multimediadateien. Es werden jedoch keine exakten Positionen, Einrückungen, Schriftarten und Schriftgrößen vorgeschrieben. Sie bieten die Möglichkeit, diese nicht nur linear von vorne nach hinten zu betrachten, sondern über Querverweise von einer Dokumentstelle zu einer anderen springend. Das Layout, das der Browser darstellt, ist von seinen graphischen und multimedialen Darstellungsmöglichkeiten abhängig. So kann die gleiche HTML-Datei auf unterschiedlichen Browsern auch unterschiedlich dargestellt werden. Es können sogar einige Teile fehlen.

Einige Browser (z. B. Netscape Communicator, Microsoft Explorer und Operer) stellen auch einige nicht dem HTML-Standard entsprechende Erweiterungen zur Verfügung. Solche Erweiterungen werden normalerweise von anderen Browsern ignoriert. Um z. B. Ton- und andere Multimediadateien abzuspielen, sind oft noch sogenannte Plug-ins nötig. Das sind Zusatzprogramme für den Browser, an die solche Dateien weitergereicht werden, um sie entsprechend behandeln zu können. Hierbei handelt es sich um einen plattformabhängigen Maschinencode. Sie müssen für das jeweilige Betriebssystem geschrieben sein und haben nach einer Installation uneingeschränkten Systemzugriff (Schwickert in [2], Reibold Schmitz Seeger in [9]).

Es gibt auch noch weitere Sprachen, die so ähnlich wie HTML funktionieren, so z. B. die Virtual Reality Markup Language (VRML). Sie dient einer grafischen Simulation dreidimensionaler Räume und Objekten mit eingeschränkten Interaktionsmöglichkeiten. Die Idee von VRML besteht darin, solche Räume über das WWW zu verbinden und mehreren Benutzern gleichzeitig zu erlauben, in diesen Räumen zu agieren. Durch Anklicken von Objekten kann man in andere Welten gelangen oder Dokumente, wie z. B. HTML-Seiten, in den WWW-Browser laden. Auch kann ein VRML-Raum auf andere Ereignisse reagieren. So kann z. B. eine Mausbewegungen in eine Rotation eines bestimmten grafischen Objekts, umgewandelt werden. VRML-Welten können mit anderen WWW-Techniken, wie z. B. Java-Applets, kombiniert werden (Diehl in [3]).

Beispiel einer HTML-Datei

```
<html>
  <head>
    <title>Beispiel</title>
  </head>
  <body>

    <h1>Schriftgröße 1</h1>
    <center>
      <h2>Schriftgröße 2, zentriert</h2></center>
    <div align="right">
      <h3><i>Schriftgröße 3, rechts, kursiv</i></h3></div>
    <h4>Schriftgröße 4</h4>
    <h5>Schriftgröße 5</h5>
    <h6>Schriftgröße 6</h6>
    <h4>Nummern Liste:</h4>

    <ol>
      <li><b>Einleitung</b></li>
      <li>Verteilte Anwendungen im World Wide Web</li>
      <li>Entwurf von Schaufeln</li>
    </ol>
  </body>
</html>
```

Die Anweisungen an den Browser, für die Darstellung der HTML-Seite, sind sogenannte *Tags*. Als *Tag* (engl. Schildchen, Edikett) werden die Anweisungen in spitzen Klammern (`< ... >`) bezeichnet. Zum Beenden einer Anweisung wird der gleiche *Tag* mit vorgestelltem Schrägstrich (`</ ... >`) verwendet. Der *Tag* `<html>` gibt dem Browser an, daß er die folgenden Daten als HTML-Anweisungen interpretieren soll. Die HTML-Datei teilt sich in die Bereiche `head` und `body` auf. Im Bereich `head` stehen Informationen, die nicht direkt dargestellt werden, wie z. B. der Titel der Seite, der Zeichensatz, der verwendet werden soll und das Programm, mit dem die Seite erstellt wurde. Unter `body` steht der Inhalt der HTML-Seite, der im Browser-Fenster dargestellt werden soll.

Das HTML-Format stellt sechs verschiedene Einstellungen für Überschriften zur Verfügung. Der dafür verwendete *„Headline"-Tag* (`<h1>` bis `<h6>`) kann die Werte 1 bis 6 annehmen, wobei der Wert 1 die größte Schrift und der Wert 6 die kleinste Schrift festlegt. Zur Formatierung der Wörter und Absätze auf

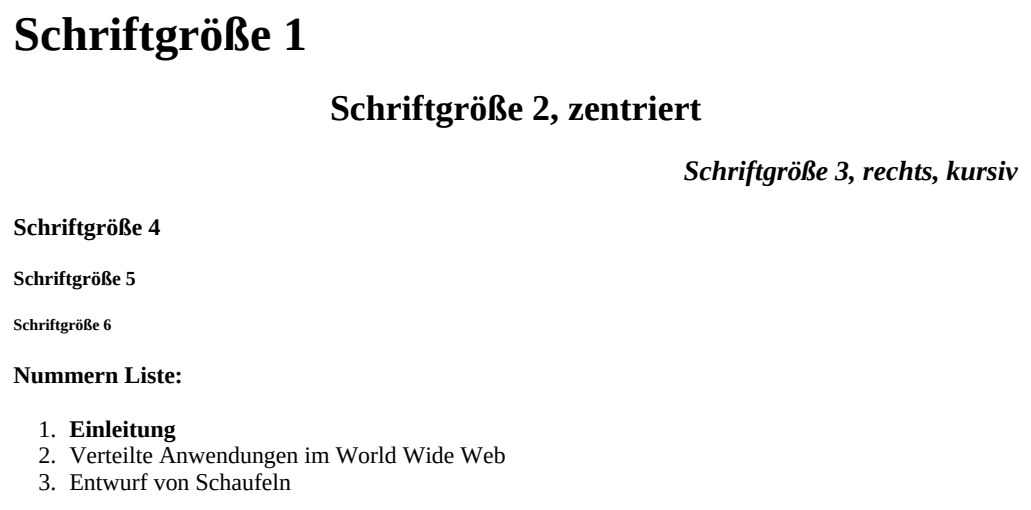


Abbildung 2.1: Darstellung der Beispiel HTML-Datei

der HTML-Seite stehen noch zahlreiche weitere *Tags* zur Verfügung, z. B.: `<center>`: Zentrierter Text, `<div align=" ... ">` (center, left, right): Ausrichtung, `<i>`: Kursivschrift (italic), `<b>`: Fettdruck (bold), `<li>` (list): Liste und `<ol>`: Numerieren einer Liste.

2.2 Dynamisch verteilte Anwendungen

Programme auf WWW-Seiten Von Netscape und Microsoft wurden noch weitere Möglichkeiten zur WWW-Seiten-Gestaltung geschaffen. Bei Microsoft ist dies VBScript (Visual-Basic Script), bei Netscape JavaScript (siehe auch Kapitel 2.4), beides sind Skriptsprachen, die in eine HTML-Datei eingebunden werden. Der jeweilige Browser liest diese und führt sie mit einem integrierten VB/JavaScript-Interpreter aus. Mit diesen beiden Sprachen lassen sich schon einfache interaktive Programme schreiben.

Wesentlich weiter gehen Java und Microsoft ActiveX. Java ist von Sun entwickelt worden, es gibt jedoch auch Umsetzungen anderer Hersteller. Mit diesen Programmiersprachen lassen sich umfangreiche Programme in WWW-Seiten integrieren, die dynamisch verteilt im Netz laufen, d. h. es werden jeweils nur die Programmteile vom Server geladen, die der Client zur Ausführung braucht. Ein Programm auf dem Client kann mit einem Programm auf dem Server zusammenarbeiten.

Bei diesen Programmiersprachen besteht jedoch ein gewisses Sicherheits-

Risiko, falls Programme Befehle enthalten, die auf Systemressourcen des Client zugreifen können und diese nicht von dem jeweiligen Browser abgefangen werden. Unseriöse Programmierer können mit dessen Hilfe über WWW-Seiten auf den Client zugreifen und dort unerwünschte Aktionen auslösen (z. B. Dateien einsehen, löschen, schreiben usw.).

ActiveX Die Programmiersprache ActiveX ist zum Programmieren umfangreicher Anwendungen geeignet. Mit ihr gibt es nur wenige Einschränkungen beim Zugriff auf die Systemressourcen. Bei ActiveX-Programmen handelt es sich um Maschinencode. Dieser muß für das jeweilige Betriebssystem kompiliert sein. AvtiveX integriert sich fast nahtlos in Microsoft Windows. Es basiert auf dessen OLE (Object Linking and Embedding; Verknüpfen und Einbinden von Objekten) und auf DCOM (Distributed Component Object Model; Möglichkeit, Softwarekomponenten im Netz verteilt zu halten). Hierdurch kann ein ActiveX-Programm mit Windows-Programmen kommunizieren. Dies birgt aber nicht unerhebliche Risiken. Die Gefahr, daß ein Programm unerlaubt auf den Client-Rechner zugreift, ist dadurch gegeben. Auf anderen Betriebssystemen müssen OLE und DCOM nachgebildet werden. ActiveX ist bis jetzt nur im Microsoft-Explorer für Microsoft-Windows vorhanden (Reibold Schmitz Seeger in [9]).

Java Die Programmiersprache Java und die dazugehörigen Klassenbibliotheken bieten eine portable, objektorientierte Programmierumgebung. Systemspezifische Besonderheiten können nicht voll ausgenutzt werden, da die Kompatibilität der Java-Programme zu den unterschiedlichsten Systemen gewährleistet werden soll. Java-Programme müssen nicht für die unterschiedlichsten Systeme extra kompiliert werden, sondern nur einmal (siehe Kapitel 2.4.1).

Bei Java-Anwendungen (Applets), diese können auf WWW-Seiten integriert werden, ist der Zugriff auf die Systemressourcen stark eingeschränkt. Der Anwender muß den Zugriff explizit freigeben. Mit Java sind auch Programme möglich, die, wie z. B. C- oder Fortran-Programme, lokal auf einem Rechner laufen oder beide Möglichkeiten bieten.

2.3 Auswahl der Programmiersprache

Um möglichst vielen Benutzern die Möglichkeit zur Anwendung von verteilten Applikationen geben zu können, sollten diese auf möglichst vielen Rech-

nern mit unterschiedlichen Betriebssystemen laufen. ActiveX ist zur Zeit nur unter Microsoft-Explorer lauffähig und muß für andere Betriebssysteme extra kompiliert werden. Der Explorer ist noch nicht in jeder Version für die einzelnen Betriebssysteme ActiveX-fähig. Java ist dagegen hardware- und betriebssystemunabhängig und hat das bessere Sicherheitskonzept (siehe Kapitel 2.4.2). Es gibt mehrere Browser für die einzelnen Betriebssysteme, die Java unterstützen.

2.4 Java

Oftmals werden unter dem Begriff Java die Bezeichnungen Java-Applets, Java-Applikationen und JavaScript verwechselt. Java-Applets und Java-Applikationen sind Programme, die in der Programmiersprache Java, die von Sun entwickelt wurde, geschrieben sind.

Java-Applets werden durch ein Applet-*Tag* (siehe Kapitel 2.4.3) aus einer HTML-Datei in einem für Java-Applets geeigneten Browser gestartet. Die Applets können, wie auch die HTML-Datei, über das Internet geladen werden. Jedes Applet besteht mindestens aus einer eigenständigen Datei.

Java-Applications sind lokale Programme, die in der Sprache Java geschrieben wurden.

JavaScript ist eine eigenständige Scriptsprache, die von Netscape entwickelt wurde. Der Programmcode wird direkt in die HTML-Datei eingefügt. Sie ist nicht so leistungsfähig wie Java.

2.4.1 Die Java Virtuelle Maschine

Java-Programme werden zuerst durch einen Compiler in einen Bytecode übersetzt. Dieser ist jedoch nicht direkt auf einer konkreten Maschine lauffähig. Für die konkrete Maschine muß es einen Bytecode-Interpretierer geben, die Java Virtuelle Maschine (VM). Die Software-Emulation der VM verlängert die Laufzeit der Programme. Durch die Verwendung der VM ist Java maschinenunabhängig. Es muß jedoch eine VM für die konkrete Maschine installiert sein. Es gibt jedoch auch schon Ansätze, wo die konkrete Maschine Teile oder den gesamten Bytecode selber verarbeiten kann. Dadurch entfällt die Software-Emulation der VM ganz bzw. zum Teil.

Bei einem Applet wird zuerst durch einen Browser, der auf dem Rechner des Anwenders (Client) läuft, eine HTML-Seite vom Rechner des Anbieters

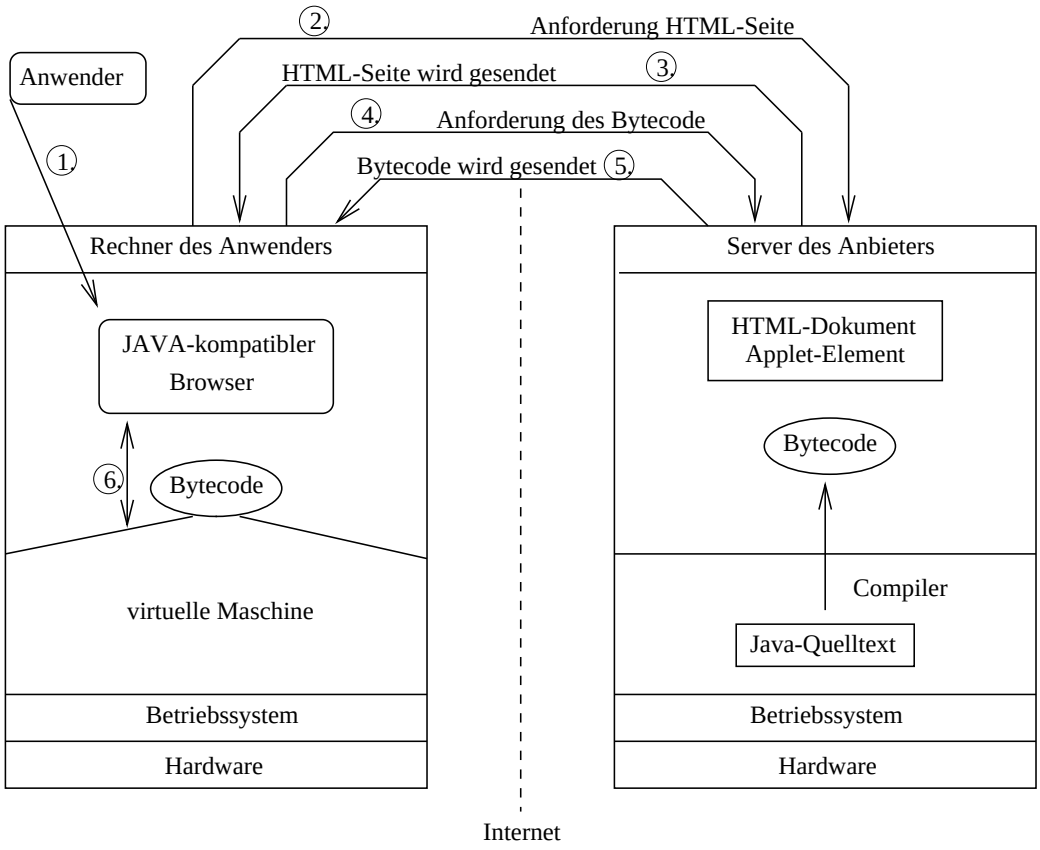


Abbildung 2.2: Zusammenspiel des Servers mit dem Client

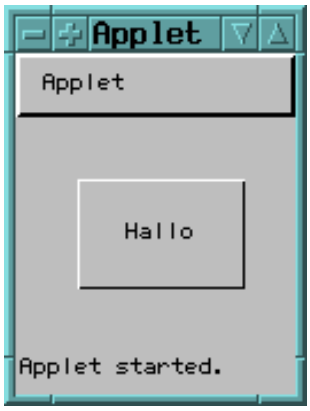


Abbildung 2.3: Das Java-Applet `Hallo.class`

(Server) angefordert ①② (in der Abbildung 2.2). Dieser sendet die entsprechende Seite ②. In dieser HTML-Datei ist ein Applet-*Tag* (siehe Kapitel 2.4.3), in dem steht, welche Java-Bytecodes (Java-Classen) vom Server angefordert werden müssen. Diese werden angefordert ④, vom Server gesendet ⑤ und der virtuellen Java-Maschine auf dem Client zur Bearbeitung übergeben. Diese arbeitet zusammen mit dem Browser und führt den Code aus ⑥.

2.4.2 Die Zugriffsrechte

Bei Programmen, die über das Internet gestartet werden, ist es besonders wichtig, daß nicht jedes Programm auf lokale Ressourcen zugreifen kann. Der Zugriff auf lokale Ressourcen (Speicher, Dateisystem, Drucker, etc.) wird durch die virtuelle Maschine kontrolliert. Keine Anwendung kann unter Java direkt auf diese zugreifen. Der Anwender kann bestimmte Ressourcen für ein Programm freigeben. Einem Applet ist es z. B. bei Standardeinstellungen in den meisten Browsern nicht erlaubt, auf das lokale Dateisystem zuzugreifen.

2.4.3 Java Programm-Beispiel

Anhand des folgenden Programms (Abbildung 2.3) soll gezeigt werden, wie Java auf ein Ereignis, wie z. B. das Anklicken eines Knopfes mit der Maus, reagiert. Unter Java gibt es einige Hilfsmittel, um eine Benutzerschnittstelle (engl. user interface; kurz UI) erstellen zu können. UI ist ein weiter Begriff, der jegliche Art der Kommunikation zwischen einem Programm und

seinen Anwendern einbezieht. Häufig stellen Applets und Applications mit Hilfe einer GUI (graphic user interface) Informationen dar, die der Anwender interaktiv beeinflussen kann. Der abstrakte Fensterwerkzeugsatz (engl. Abstract Window Toolkit; kurz AWT), Teil der Java-Umgebung, enthält einen vollständigen Satz an Klassen, die zur Entwicklung von GUI-Programmen notwendig sind.

AWT bietet viele Standard-GUI-Komponenten, wie Schaltflächen, Listen, Menüs und Textflächen, an. Es enthält ebenfalls Container (z. B. Anwendungsfenster oder Menüleisten) und aufwendigere Komponenten, wie z. B. ein Dialogfenster zum Öffnen und Sichern von Dateien (Reibold Schmitz Seeger in [8]).

Das Programm-Beispiel, es ist ein Applet, wird mit einem Java-kompatiblen Browser gestartet.
Hierzu lädt der Browser die folgende HTML-Datei (`Hallo.html`):

```
<HTML>
<HEAD>
<TITLE>Applet Demo</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Hallo.class" WIDTH=100 HEIGHT=100></APPLET>
</BODY>
</HTML>
```

Als Applet-*Tag* wird die folgende Anweisung bezeichnet:

```
<APPLET CODE="Hallo.class" WIDTH=426 HEIGHT=266></APPLET>
```

Aus diesem Teil weiß der Browser, daß dieser ein Applet `Hallo.class` mit einer Darstellungsgröße von 426 x 266 Bildpunkten zu starten und diesem den nötigen Platz für die graphische Anzeige zu reservieren hat.

Nachfolgend der Quelltext des `Hallo.class`-Applets:

```
/*
  HalloApplet
  (C) 1998 Michael Pfuetz, GhK

  Applet zur Demonstration eines Event
  benoetigt AWT 1.1.X
```

```
*/

import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class Hallo extends Applet
  implements ActionListener
{
  Button HalloButton;

  public void init()
    {
      setSize(100,100);
      HalloButton = new Button();
      HalloButton.setLabel("Hallo");
      HalloButton.setBounds(25,25,75,50);
      HalloButton.setBackground(new Color(12632256));
      add(HalloButton);
      HalloButton.addActionListener(this);
    }

  public void actionPerformed(ActionEvent event)
    {
      Object object = event.getSource();
      if (object == HalloButton)
System.out.println("Hallo");
    }
}
```

Der Quelltext beginnt mit drei `import`-Anweisungen. Durch das Importieren von Klassen oder Paketen kann sich eine Klasse einfach auf Klassen in anderen Paketen beziehen. In der Programmiersprache Java werden zum Zusammenfassen von Klassen *Pakete* verwendet. Diese sind in einer Baumstruktur organisiert. Einzelne Äste werden durch Punkte („.“) getrennt (siehe Abbildung 2.4).

Jedes Applet muß eine Unterklasse der Klasse `Applet` definieren. Im Hallo-Applet wird diese Unterklasse `Hallo` genannt.

Applets erben aus der Klasse einen Großteil ihrer Funktionalität, angefangen bei der Kommunikation mit Browsern bis hin zu der Fähigkeit, auf Benutzer-

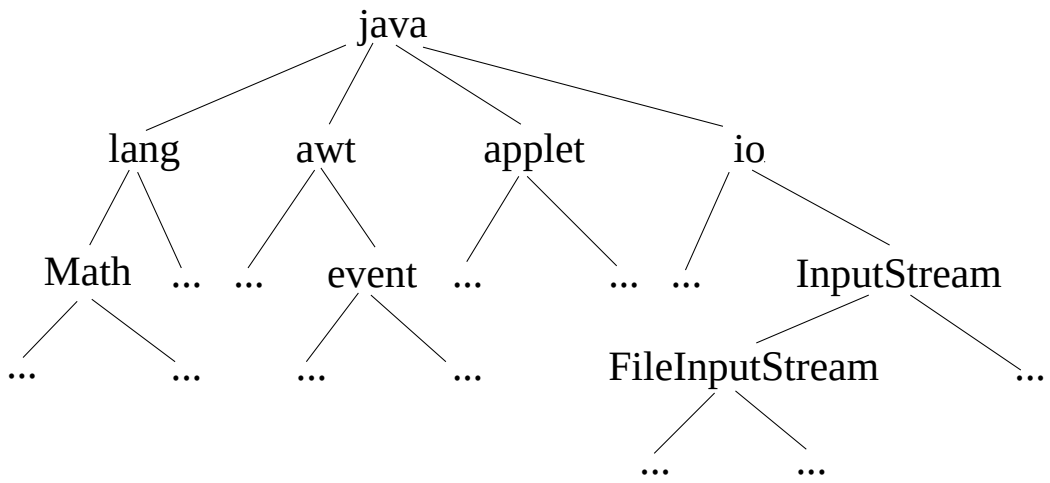


Abbildung 2.4: Die Baumstruktur der Pakete

ereignisse zu reagieren.

Das Hallo-Applet implementiert genau zwei Methoden, nämlich die Methoden `init` und `actionPerformed`. Jedes Applet muß mindestens eine der folgenden Methoden implementieren: `init`, `start` oder `paint`. Im Gegensatz zu Java-Applikationen brauchen Java-Applets *keine* Methode `main` zu implementieren. Durch das Importieren der Pakete kann man sich im Programm auf die Klassen `Applet`, `ActionListener`, `ActionEvent`, `Button` und `Color` beziehen.

Wird auf das Explizite-importieren verzichtet, muß auf nicht lokal definierte Bezeichner mit dem gesamten Namen zugegriffen werden, wie das folgende Beispiel zeigt:

```
public class Hallo2 extends java.applet.Applet
  implements java.awt.event.ActionListener
{
  java.awt.Button HalloButton;

  public void init()
    {
      setLayout(null);
      setSize(100,100);
      HalloButton = new java.awt.Button();
      HalloButton.setLabel("Hallo");
      HalloButton.setBounds(25,25,75,50);
```

```
      HalloButton.setBackground(new java.awt.Color(12632256));
      add(HalloButton);
      HalloButton.addActionListener(this);
    }

  public void actionPerformed(java.awt.event.ActionEvent event)
    {
      Object object = event.getSource();
      if (object == HalloButton)
System.out.println("Hallo");
    }
}
```

Das Paket `Java.awt` enthält die am häufigsten benutzten Klassen im AWT. Im Paket `Java.awt.event` sind Klassen definiert, die zur Ereignisbehandlung nötig sind. Das Paket `Java.lang` wird immer automatisch importiert, so daß die Klasse `System` ohne `import`-Anweisung benutzt werden kann.

Am Anfang des Blocks, der die Klasse `Hallo` definiert, steht:

```
public class Hallo extends Applet
  implements ActionListener
{
```

Das Schlüsselwort `extends` zeigt an, daß `Hallo` eine Unterklasse der genannten Klasse `Applet` ist. In der Klasse `Applet` ist z. B. die Fähigkeit implementiert, auf Browseranfragen zu reagieren. Mit dem Schlüsselwort `implements` werden eine oder mehrere Schnittstellen implementiert. Im Beispiel handelt es sich um eine Schnittstelle namens `ActionListener`. Vor dem Schlüsselwort `class` kann einer von drei Modifikatoren stehen, um die Klasse öffentlich (`public`), abstrakt (`abstract`) oder endgültig (`final`) zu deklarieren.

Der im Beispiel verwendete Modifikator `public` deklariert, daß die Klasse von Objekten außerhalb des aktuellen Paketes verwendet werden kann. Per Voreinstellung kann eine Klasse nur von anderen Klassen im selben Paket, in dem sie deklariert worden ist, benutzt werden. Der Modifikator `abstract` deklariert die Klasse als abstrakte Klasse. Eine abstrakte Klasse kann abstrakte Methoden enthalten (Methoden ohne Implementierung). Mit dem Modifikator `final` wird eine Klasse als endgültig definiert, d. h., sie kann keine Unterklassen bilden (Campione in [8]).

Die nächste Zeile im Beispiel:

```
  Button HalloButton;
```

definiert eine Referenz mit dem Namen `HalloButton` vom Objekt-Typ `Button` aus dem Paket `Java.awt` ohne Speicherplatz anzufordern. Diese Klasse enthält alle Methoden, um einen Knopf (Button) zu beschreiben.

Wenn das Applet vom Browser dazu aufgefordert wird, sich zu initialisieren, wird folgender Block ausgeführt:

```
public void init()
  {
    setSize(100,100);
    HalloButton = new Button();
    HalloButton.setLabel("Hallo");
    HalloButton.setBounds(25,25,75,50);
    HalloButton.setBackground(new Color(12632256));
    add(HalloButton);
    HalloButton.addActionListener(this);
  }
```

Mit `setSize(100,100)` wird dem Applet eine Größe von 100 x 100 Bildpunkten zugeordnet. Mit der `new`-Anweisung wird ein Button mit dem Namen HalloButton initalisiert, d. h., der nötige Speicher wird bereitgestellt. Diesem Button werden danach einige Einstellungen übergeben, zuerst seine Beschriftung (Hallo), danach seine Lage/Größe und seine Farbe. Mit `add(HalloButton)` wird er zum Applet hinzugefügt. Der `HalloButton` bekommt eine Schnittstelle zum `ActionListener` und diesem wird mit `this` das Applet übergeben.

Aktiviert der Anwender eine Komponente - im Beispiel durch Anklicken des `HalloButtons` - bemerkt der plattformspezifische Teil des AWT das Ereignis (engl. event) und erzeugt ein `Event`-Objekt. Das Ereignisbehandlungssystem des AWT reicht das `Event`-Objekt die Komponentenhierarchie hoch und gibt damit jeder Komponente die Möglichkeit, auf das Ereignis zu reagieren. Die Komponenten können unterschiedlich auf das Ereignis reagieren. Sie können es ignorieren, abändern, abfangen oder auf etliche andere Arten reagieren. Der Button im Beispiel reicht das Ereignis weiter, so daß der nächste Block des Beispiels auf das Ereignis reagieren kann:

```
public void actionPerformed(ActionEvent event)
  {
    Object object = event.getSource();
    if (object == HalloButton)
      System.out.println("Hallo");
  }
```

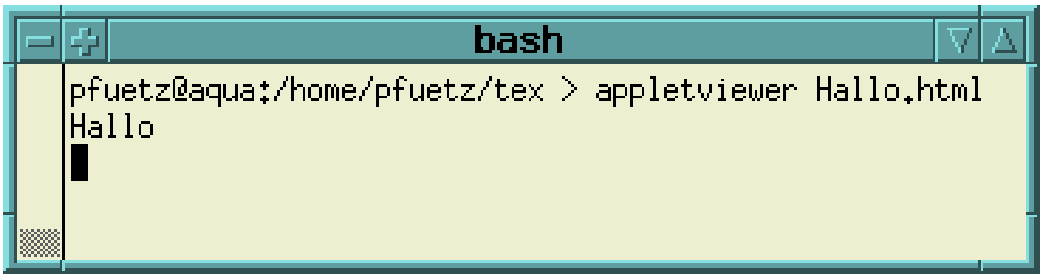


Abbildung 2.5: Die Ausgabe des Java-Applets `Hallo.class`

Der Methode `actionPerformed` wird das Ereignis `ActionEvent event` übergeben. Danach wird die Art des Objekts, die das Ereignis ausgelöst hat, in der neu deklarierten und initialisierten Variablen `object` gespeichert. In der Zeile:

```
if (object == HalloButton)
```

wird festgestellt, ob es sich bei dem Objekt um den `HalloButton` handelt; ist dieses der Fall, wird der folgende Befehl ausgeführt:

```
System.out.println("Hallo");
```

Daraufhin wird im Systemfenster der Text „Hallo“ ausgegeben (siehe Abbildung 2.5).

2.5 Beispiele aus dem Internet

DLR Unter der Adresse: http://www.sm.go.dlr.de/~jens/NACA.html der DLR (siehe Abbildung 2.7) werden *NACA 4-digits airfoil series* und *NACA 5-digits airfoil series* Profile mit Hilfe eines Java-Applet dargestellt. Es gibt mehrere Schiebebalken, mit denen die Form und Lage des Profils in dem Bereich der NACA Spezifikationen verändert werden kann. Dieses veränderte Profil wird der entsprechenden NACA-Nummer zugeordnet.

Um die Profil-Geometrie zu verändern, lassen sich einige Dinge einstellen: *Maximum Camber (Cmax)* (Maximale Wölbung der Skelettlinie (camber line)), *Max. Camber Position (XCmax)* (X-Lage der maximalen Wölbung der Skelettlinie) und *Thickness (t)* (Maximale Dicke des Profils).

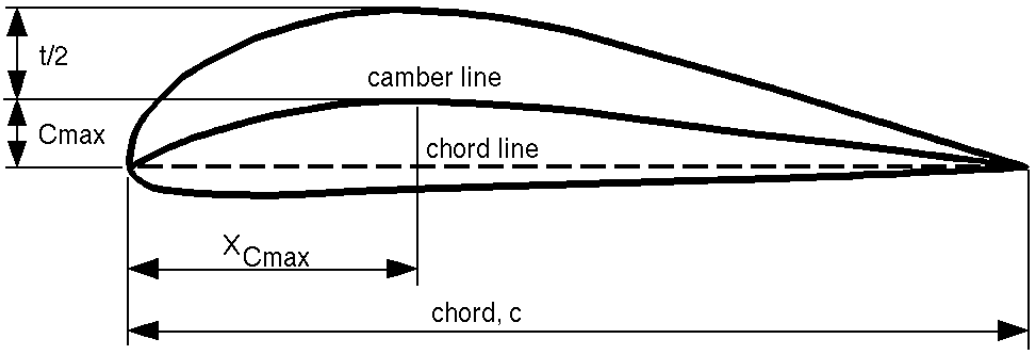


Abbildung 2.6: Profil mit Skelettlinie

Zur Manipulation der Darstellung lassen sich *Number of Points* (Anzahl der berechneten Punkte zur Profildarstellung), *Point Distribution* (Verteilung der Punkte), *Point Size* (Größe der Punkte, mit denen das Profil dargestellt wird), *Zoom* (Vergrößerung des Profils), *X-Offset* (Verschiebung des Profils in X-Richtung) und *Y-Offset* (Verschiebung des Profils in Y-Richtung) einstellen. Siehe hierzu auch Abbildung 2.6.

Computational Aerospace Laboratory, MIT Unter der Adresse http://raphael.mit.edu/Java/ des MIT (siehe Abbildung 2.8 und 2.9) wird ein zweidimensionaler Windkanal mit Hilfe eines Java-Applets simuliert. Es lassen sich verschiedene Dinge anzeigen: *Density* (Verteilung der Dichte um das Profil), *Pressure* (Verteilung des Druckes um das Profil), *Temperature* (Verteilung der Temperatur um das Profil), *Mach Number* (Verteilung der Machzahl um das Profil), *Axial Velocity* (Verteilung der axialen Geschwindigkeit um das Profil), *Normal Velocity* (Verteilung der normalen Geschwindigkeit um das Profil), *Total Pressure* (Verteilung des totalen Drucks um das Profil).

Einstellen lassen sich die *Inlet Mach Number* (Einlaß-Mach-Zahl), *4th Order Damping* und *2nd Order Damping* (Dämpfung der Koeffizienten vierter und zweiter Ordnung) -ein bestimmter Betrag an Dämpfung ist notwendig, um numerische Instabilitäten aus der Lösung zu entfernen-.

Außer der Konturansicht (eine Farbdarstellung des Fluss-Feldes um das Profil) läßt sich auch eine Linienansicht (Schaubild einer Quantität gegenüber linkem und rechtem Rand des Profils in axialer Richtung) anzeigen (siehe Abbildung 2.9).

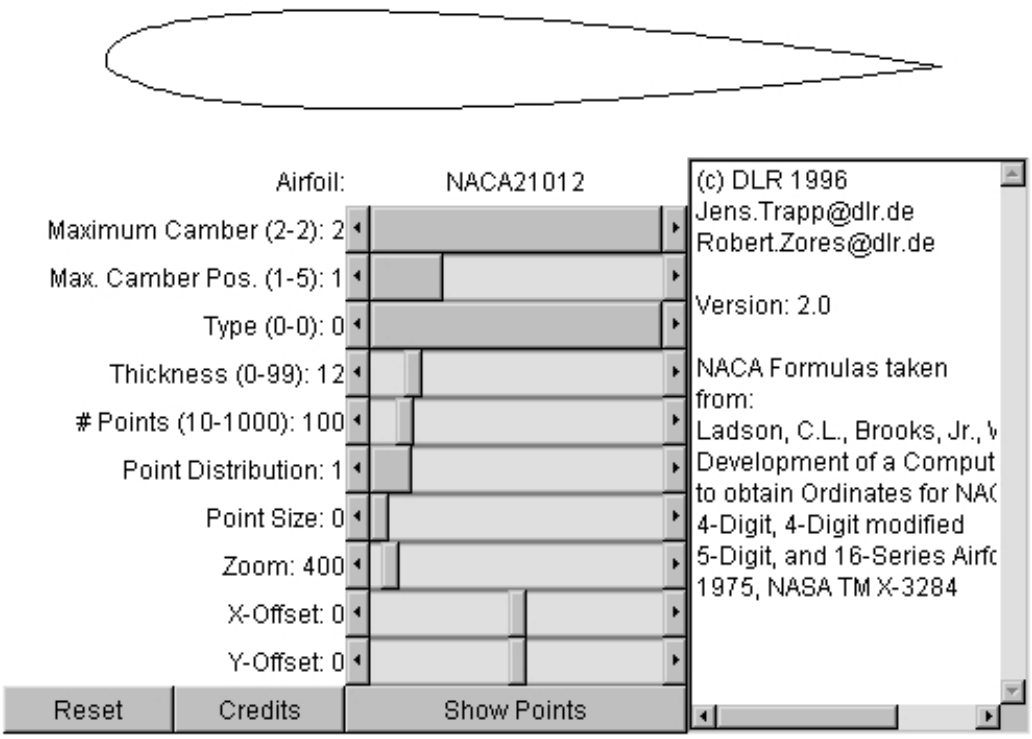


Abbildung 2.7: Das Java-Applet zur Bestimmung von NACA Profilen

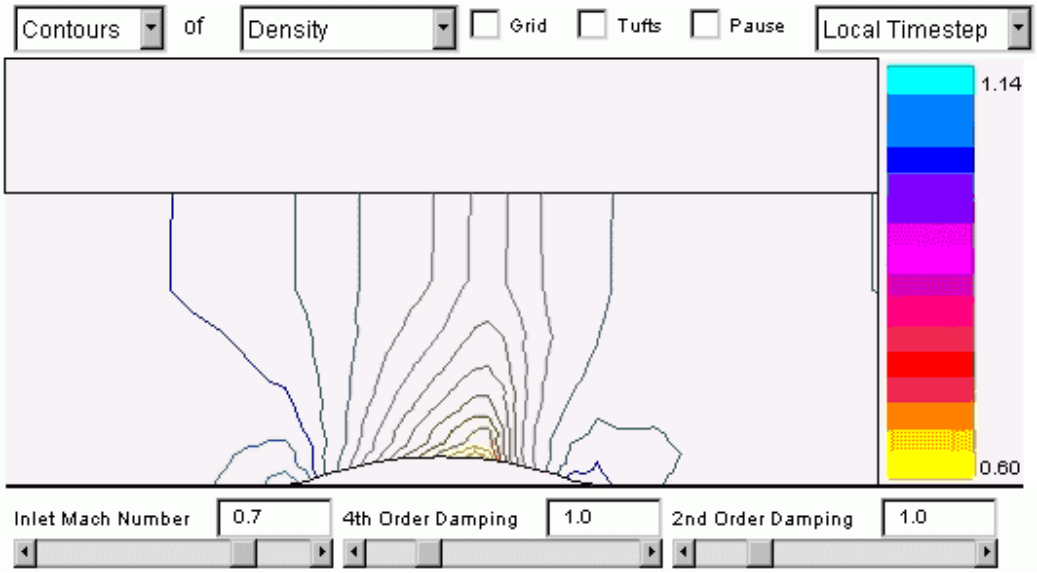


Abbildung 2.8: Das Java-Applet zur Simulation eines zweidimensionalen Windkanals (Konturansicht)

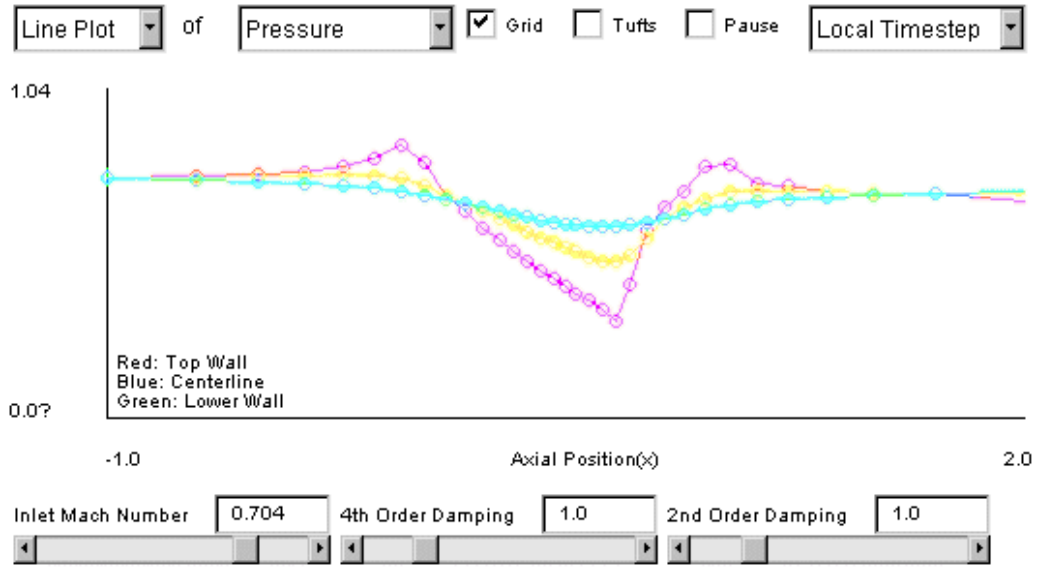


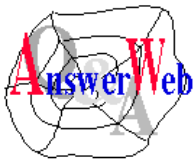
Abbildung 2.9: Das Java-Applet zur Simulation eines zweidimensionalen Windkanals (Linendarstellung)

AnswerWeb Home page for Airfoil Unter der Adresse: http://www-interact.eng.cam.ac.uk/CAL95/answer.html (siehe Abbildung 2.10) werden Fragen zu umströmten Profilen und deren Antworten gesammelt. Diese können auf verschiedene Art abgefragt werden. Als: Baumstruktur, geordnet nach verschiedenen Themen; eine Liste mit allen Fragen, die gestellt wurden; ein Resultat einer Stichwortsuche; Datenbank mit häufig gestellten Fragen. Es können auch Fragen per Email gestellt werden, die an einen Experten weitergereicht werden. Die Seiten dieser Adresse sind im HTML-Format, ohne besondere Extras, gehalten.

UNIVERSITÁ DEGLI STUDI DI GEOVA Unter der Adresse: http://www.idra.unige.it/~irro/profilo_e.html (siehe Abbildung 2.11 und 2.12) werden einige animierte Grafiken im Gif-Format angeboten. Sie stellen, in Form eines Films, ohne eigentliche Interaktion, die Umströmung, Geschwindigkeitsverteilung, Druckverteilung eines Mediums um ein Profil, das seinen Winkel zur Durchströmungsrichtung ändert, dar. Außerdem werden die Kräfte an einem umströmten Profil, das seinen Winkel zur Durchströmungsrichtung ändert, dargestellt.

NAVEL POSTGRADUATE SCHOOL Unter der Adresse: http://atemi.aa.nps.navy.mil/panel/panel.html (siehe Abbildung 2.13 und 2.14) werden *NACA 4-digits airfoil series* und *NACA 5-digits airfoil series* Profile und cp-Werte mit einem Fortran-Programm auf dem Server berechnet. Hierzu muß ein Formular, das in JavaScript geschrieben ist, ausgefüllt werden, was an den Server geschickt wird. Die Ergebnisse werden mit Hilfe eines CGI-Programms (externes Programm auf dem Server) als Bild oder als Texttabelle an den Browser übergeben. Einstellen lassen sich in dem Formular folgende Dinge: Vier- oder fünfstellige NACA-Nummer, der Zuströmwinkel, Anzahl der berechneten Ebenen und ob die Ausgabe als Bild oder Texttabelle erfolgen soll.

Das Fortranprogramm steht auch als Download bereit.



The AnswerWeb Home page for Airfoil

Look for Questions

The *AnswerWeb* for **Airfoil** can be viewed in a number of different ways,

- As a tree structure with branches for each subject category *Tree Structure*
- As a list of all the questions that have been asked. *A ll Questions*
- Results from a keyword search *Keyword Search*
- View database as a list of Frequently Asked Questions (FAQ). You can print this out as one document. *View as FAQ*

Ask a Question

The AnswerWeb allows you to ask questions if you can not find the answer to your query. To ask a question simply click on

Ask a question !

Your question will be emailed to an expert. Once the question is answered it will be put into the database.

Toolbar

The toolbar below is available from all pages. It gives you access to all the systems main functions.

Home | Ask | back | Search | List | Help

Help

Abbildung 2.10: AnswerWeb Home page for Airfoil

FLOW AROUND AN AIRFOIL

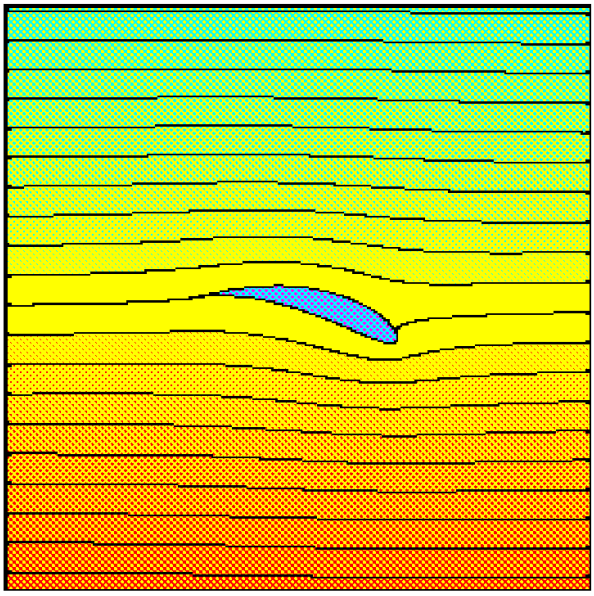


Abbildung 2.11: Darstellung eines Mediums um ein Profil

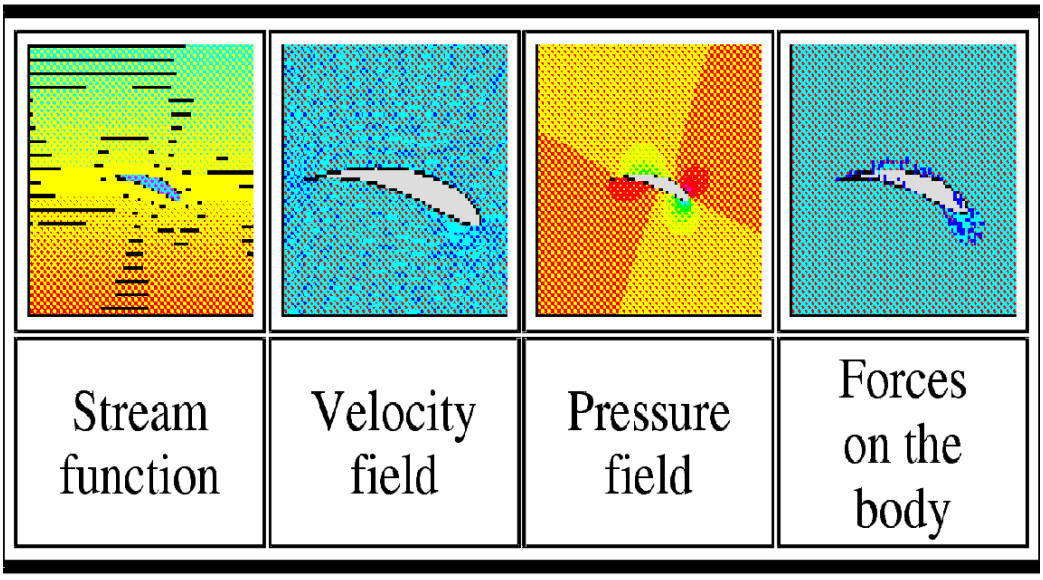


Abbildung 2.12: Auswahl der verschiedenen Darstellungen

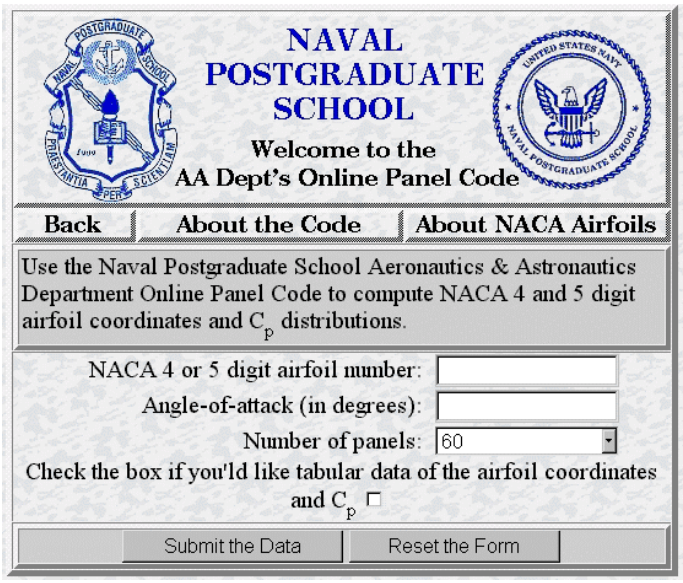


Abbildung 2.13: Eingabeformular zur Berechnung eines NACA Profils und deren cp-Werte

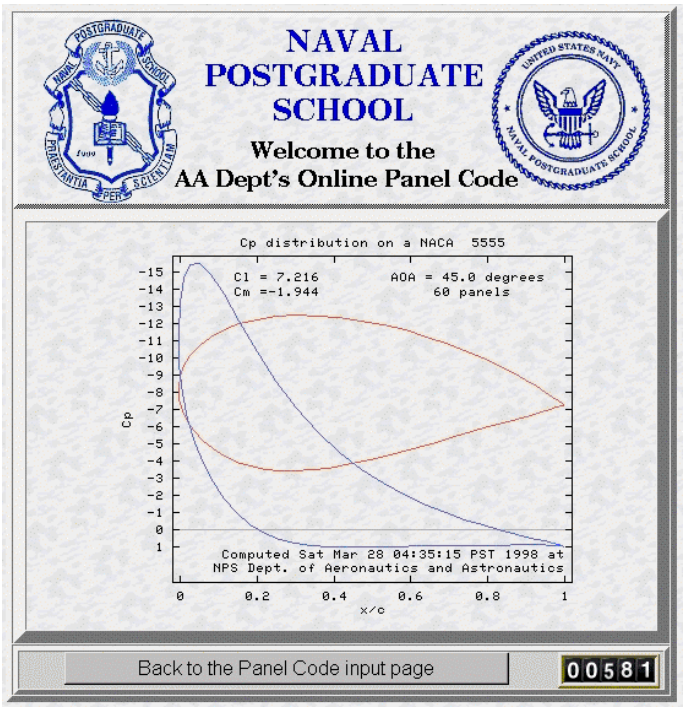


Abbildung 2.14: Darstellung eines Profils der NACA-Serie

Kapitel 3

Entwurf von Schaufeln

3.1 Zweidimensionaler Entwurf

3.1.1 Mathematische Grundlagen des B–Splines

Die in dieser Studienarbeit verwendeten Basis–Splines (B–Splines) beruhen auf der folgenden Basisfunktion (Lawerenz Uelschen Jensen in [10]).

Definition 1 (Basisfunktion) Es sei $t = \{t_0, t_1, \ldots, t_m\}$ eine Folge von nicht fallenden reellen Zahlen, d. h. $t_i \leq t_{i+1}$ für $i = 0, \ldots, m-1$. Die i-te Basisfunktion vom Grad p (Ordnung $p+1$) ergibt sich zu

$$N_{i,0}(t) = \begin{cases} 1 & \text{falls } t_i \leq t < t_{i+1} \\ 0 & \text{sonst} \end{cases}$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t)$$

Die einzelnen t_i werden als Träger, Knotenpunkt oder kurz als Knoten bezeichnet. Die folgenden Eigenschaften lassen sich aus der o. g. Definition der Basisfunktionen ableiten.

1. Für ein $t \notin [t_i, t_{i+p+1}]$ gilt $N_{i,p}(t) = 0$. Dieses wird als lokale Stütz- oder Trägereigenschaft bezeichnet.

2. In jedem Knotenintervall $[t_j, t_{j+1})$ sind maximal $p+1$ der Basisfunktionen $N_{i,p}(t)$ nicht null.

3. Die Eigenschaft $\forall i, p, u : N_{i,p}(u) \geq 0$ wird als Nichtnegativität bezeichnet.

4. Alle Ableitungen von $N_{i,p}(t)$ existieren innerhalb eines Knotenintervalls. An den Knotenpunkten ist $N_{i,p}(t)p-k$ mal stetig differenzierbar, wobei k die Multiplizität des Knotenpunktes ist.

5. Für $p > 0$ hat $N_{i,p}(t)$ genau ein Maximum.

Mit Hilfe der soeben eingeführten Basisfunktionen folgt die Definition einer B–Spline Kurve (Lawerenz Uelschen Jensen in [10]).

Definition 2 (B–Spline Kurve) Es seien $\{N_{i,p}(t)\}$ nach Definition 1 gebildete Basisfunktionen und $\{P_i\}$ Punkte im $\mathbb{R}^2$ (bzw. $\mathbb{R}^3$) mit $i = 0, \ldots, n$. Eine parametrische B–Spline Kurve $B(t)$ vom Grad p (Ordnung $p+1$) wird dann durch

$$B(t) = \sum_{i=o}^{n} N_{i,p}(t) P_i$$

definiert.

Bei einer Periodischen Kurve ergibt sich der Knotenvektor zu

$$t = \{\underbrace{t_0, t_1, \ldots, t_{p-1}}_{pKnoten}, t_p, \ldots, t_{m-p}, \underbrace{t_{m-p+1}, \ldots, t_m}_{pKnoten}\},$$

wobei die ersten bzw. letzten p Knoten eine periodische Fortsetzung vorangehender Knoten sind, d. h. $t_{p-i-1} = t_{p-i} - (t_{n-i+1} - t_{n-i})$ und $t_{n+i+2} = t_{n+i+1} + (t_{p+i+1} - t_{p+i})$ für $i = 0, \ldots, p-1$. Für die Kontrollpunkte gilt dementsprechend die Beziehung $P_{n+1-p+i} = P_i$ für $i = 0, \ldots, p-1$. Die Periodenlänge ergibt sich zu $n + 1 - p$.

Die Kurve ist im periodischen Fall auf $[t_p, t_{m-p})$ definiert. Oftmals werden die Kontrollpunkte auch als *de Boor-Punkte* bezeichnet. Die lineare Verbindung der Kontrollpunkte $\{P_i\}$ wird als Kontrollpolygon bezeichnet (Lawerenz Uelschen Jensen in [10]).

3.1.2 Schwerpunktberechnung des B–Splines

Eine Möglichkeit, um den Schwerpunkt des B–Splines zu berechnen, ist, die Approximation durch ein Polygon mit n-Punkten. Dieses wird durch $n-2$

Dreiecke zerlegt (siehe Abbildung 3.1). Die Schwerpunktkoordinate eines einzelnen Dreiecks ist wie folgt zu berechnen:

Die Schwerpunktskoordinaten eines Dreiecks mit den Eckpunkten

$$P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$$

errechnet sich in x-Richtung[6]:

$$\xi = \frac{x_1 + x_2 + x_3}{3}$$

und in y-Richtung[6]:

$$\eta = \frac{y_1 + y_2 + y_3}{3}$$

Die Schwerpunktkoordinate zweier Flächen berechnet sich aus dem Teilschwerpunktsatz:

Teilschwerpunktsatz[7]
Der Schwerpunkt S der Fläche A liegt auf der Verbindungslinie der beiden Teilflächenschwerpunkte S_1 und S_2:

$$A \cdot x_s = A_1 \cdot x_{S_1} + A_2 \cdot x_{S_2}$$

$$A \cdot y_s = A_1 \cdot y_{S_1} + A_2 \cdot y_{S_2}$$

A:	Fläche
A_1,A_2:	Teilflächen von $A(A = A_1 + A_2)$
$S = (x_S; y_S)$:	Schwerpunkt der Fläche A
$S_1 = (x_{S_1}; y_{S_1})$:	Schwerpunkt der Teilfläche A_1
$S_2 = (x_{S_2}; y_{S_2})$:	Schwerpunkt der Teilfläche A_2

Die Flächen der Dreiecke, die zur Berechnung nötig sind, berechnen sich:

Der Flächeninhalt A eines Dreiecks mit den Eckpunkten[6]:

$$P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$$

$$A = \frac{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)}{2}$$

Daraus ergibt sich: Der erste Punkt jedes Dreiecks (siehe Abbildung 3.1)

ist der Punkt Nr. 1. Also, das erste Dreieck besteht aus den Punkten 1, 2 und 3, das zweite aus den Punkten 1, 3 und 4 usw. Je zwei von ihnen haben eine Diagonale gemeinsam und zusammen überdecken sie - ganz gleich, welche Ecke als Ausgangspunkt der Diagonalen (Punkt Nr. 1) gewählt wurde - die ganze Fläche des Polygons lückenlos. Umläuft man jedes dieser Dreiecke in dem als positiv festgesetzten Sinne, so wird auch das Polygon in diesem Sinne umlaufen. Jede Diagonale wird dabei einmal in der einen und im Nachbardreieck in der entgegengesetzten Richtung durchlaufen. *Der Flächeninhalt des Polygons ist die Summe der Flächeninhalte der Dreiecke.* Sollte das Polygon nicht konvex sein, ergeben sich Dreiecke, die über das Polygon hinaus gehen und Dreiecke, die gar nicht in das Polygon hineinreichen. Rechnet man nach der über den Flächeninhalt der Dreiecke getroffenen Festsetzung im entgegengesetzten Sinne umlaufende Dreiecke als negativ, so ergibt sich wieder der Flächeninhalt des Polygons als algebraische Summe der Dreiecksflächen, falls das Polygon nicht überschlagen ist, d. h., falls seine Seiten sich nicht schneiden (Gootwald Küstner Hellwich Kästner in [6]).

Die Schwerpunktskoordinaten eines Dreiecks i des Polygons, mit den Eckpunkten

$$P_1(x_1, y_1), P_k(x_k, y_k), P_{k+1}(x_{k+1}, y_{k+1})$$

errechnet sich in x-Richtung:

$$x_i = \frac{x_1 + x_k + x_{k+1}}{3}$$

und in y-Richtung:

$$y_i = \frac{y_1 + y_k + y_{k+1}}{3}$$

Die Schwerpunktkoordinate des B–Spline in x-Richtung ergibt sich:

$$x_{gesamt} = \frac{\sum_{i=1}^{n-2} x_i \cdot A_i}{\sum_{i=1}^{n-2} A_i}$$

und in y-Richtung:

$$y_{gesamt} = \frac{\sum_{i=1}^{n-2} y_i \cdot A_i}{\sum_{i=1}^{n-2} A_i}$$

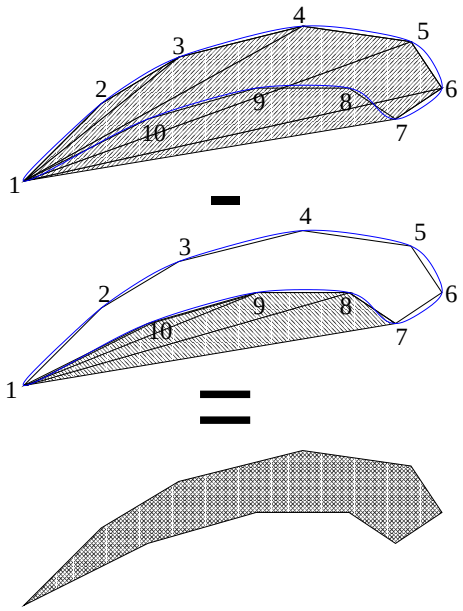


Abbildung 3.1: Flächen der Dreiecke

Der Flächeninhalt eines Dreiecks i des Polygons, mit den Eckpunkten

$$P_1(x_1, y_1), P_i(x_k, y_k), P_{k+1}(x_{k+1}, y_{k+1})$$

errechnet sich:

$$A_i = \frac{x_1(y_k - y_{k+1}) + x_k(y_{k+1} - y_1) + x_{k+1}(y_1 - y_k)}{2}$$

Der Flächeninhalt des gesamten Polygons:

$$A_{gesamt} = \sum_{k=2}^{n-2} \frac{x_1(y_k - y_{k+1}) + x_k(y_{k+1} - y_1) + x_{k+1}(y_1 - y_k)}{2}$$

3.2 Dreidimensionaler Entwurf

Der 3D-Teil (Turbo-Viewer) der Studienarbeit ist eine Erweiterung des Demonstrations-Programms `ThreeD.class`, der Distribution des JDK1.1.x von Sun, um eine Anpassung und dem Netloader.

Kapitel 4

Zusammenfassung

4.1 Schwierigkeiten

Dadurch, daß die Programmiersprache Java noch recht neu ist, gibt es noch wenig Bücher über diese. Während der Studienarbeit erschienen auch neue Versionen des Java-AWT. Zu Beginn wurde Java 1.0.x benutzt. Später wurden die Programme auf Java 1.1.x umgestellt, was zur Zeit die aktuelle Version ist. Viele Bücher sind bzw. waren aber noch auf die ältere Version ausgelegt. Zwischen der UNIX- und der Windows-Java-Version gibt es einige Ungereimtheiten, die jedoch umgangen werden konnten. Zum Beispiel sind Ergebnisse einer Mausabfrage nicht dieselben. Die Internet-Browser sind in den aktuellsten Versionen, bis auf den HotJava-Browser, auch nicht 100% Java 1.1.x-fähig, teilweise bzw. mit Hilfe eines Patches aber für den Turbo-Designer schon brauchbar. Die Geschwindigkeit der einzelnen Java-Versionen läßt auch noch zu wünschen übrig, wobei diese sich bei den unterschiedlichen Betriebssystem-Versionen an verschiedenen Stellen besonders bemerkbar macht. Bei der Windows95-Version ruckelt im Vergleich zur LINUX-Version auf demselben Rechner die Turbo-Viewer-Anzeige beim Rotieren eines Models. Bei eingeschaltetem *properties*-Fenster im Turbo-Designer ruckelt dagegen die LINUX-Version.

Beim Einlesen von Dateien mit Hilfe des *StreamTokenizer*[1], die Zahlen enthalten, kommt es zu Problemen, wenn diese Zahlen im Fließkomma-Format vorkommen (z. B. 1e-6).

[1] ein Java-Hilfsprogramm zum Einlesen von Zeichen aus einer Datei

4.2 Ausblicke

Der Turbo-Designer dient dem Entwurf von Profilen über das Internet, die an einen *Server* gesendet werden können, um dort zu einer Schaufel berechnet zu werden. Die so berechnete Schaufel läßt sich mit dem Turbo-Viewer betrachten. Auch ist die Möglichkeit gegeben, daß Studenten im Fachbereich Maschinenbau, die Zugang zum Internet haben, eigene Freiformprofile entwerfen oder betrachten können, ohne sich selber spezielle Software installieren zu müssen.

In einer weiteren Arbeit kann das hier entwickelte Entwurfsprogramm an eine Strömungsberechnungen gekoppelt werden.

In einer späteren Version des Turbo-Designers und des Turbo-Viewers könnte eine Doppelpufferung des Bildschirminhaltes eine ruckelfreie Anzeige gewährleisten. Auch eine Zertifizierung des Turbo-Designers durch eine Zertifizierungsstelle könnte hilfreich sein, da hierdurch eine eindeutige Identifizierung des Ursprungs eines Programms möglich ist. Hierdurch heben einige Internet-Browser die Sicherheitsbarrieren auf, so daß der Turbo-Designer auf das Datei-System des Clients, ohne besondere Einstellungen, zugreifen kann.

Die Funktionalität des Turbo-Designers kann noch um geometrische Operationen (Rotieren, Skalieren, Verschieben, Spiegeln ...) und Ableitungen von geometrischen Eigenschaften (Winkel, Skelettlinie, Dickenverteilung ...) erweitert werden.

Eine spätere Umstellung des Turbo-Designers und des Turbo-Viewers auf die Java AWT-Version 1.2.x könnte eine Beschleunigung der Bildschirmausgabe ergeben, da diese spezielle Befehle für die 2D- und 3D-Animation enthalten soll. Dies ist jedoch erst sinnvoll, wenn diese Version auch in den Internet-Browsern unterstützt wird.

Anhang A

Bedienungsanleitung

A.1 Hardware-Voraussetzungen

Folgende Rechner sind für Java geeignet:

- SPARC-Architekturen unter Solaris 2.3, 2.4 oder 2.5
- Intel x86-Architektur (oder kompatible) unter Microsoft Windows NT bzw. 95
- Intel x86-Architektur (oder kompatible) unter Linux
- Macintosh mit PowerPC- oder Motorola-Prozessor (68030 oder besser) und System 7.5

Java selbst ist hardwareunabhängig. Jedoch gibt es nicht immer für die unterschiedlichsten Systeme die neuesten Java-Versionen, da sich Java zur Zeit noch in der Weiterentwicklung befindet. Grundsätzlich läßt sich sagen, daß die Entwicklung von Java-Applets/-Applikationen und deren Ausführung auf jedem modernen Computer möglich ist. Gerade die Lauffähigkeit des gleichen Codes auf verschiedenen Betriebssystemen und auf verschiedenen Prozessortypen ist schließlich ein Hauptgrund für den Erfolg von Java.

A.2 Software-Voraussetzungen

Bevor Applikationen ausgeführt werden können, muß es einen Java-Bytecode-Interpreter, wie z. B. im Sun JDK (Java Development Kit), im Microsoft

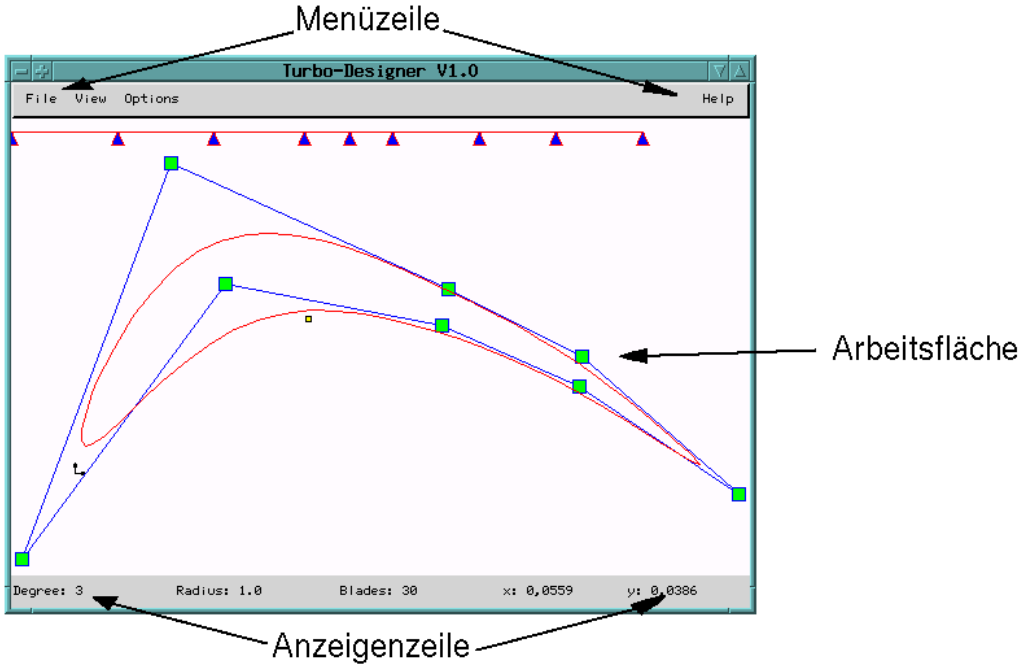


Abbildung A.1: Der Turbo-Designer

JDK oder im Symatec JDK, für das eingesetzte Betriebssystem geben. Um Applets auszuführen, muß es einen Browser mit Java-Bytecode-Interpreter, wie z. B. in folgenden Browsern geben: Sun HotJava, Sun JDK-Appletviewer, Netscape Communicator, Microsoft-Explorer. Leider sind diese aber nicht immer vollständig implementiert und nicht immer in der richtigen Version vorhanden. Für die in dieser Studienarbeit entwickelten Programme ist ein JDK1.1.x notwendig. Zum Entwickeln eigener Applets/Applikationen wird ein Java-Compiler benötigt, wie z. B. der im Sun JDK enthaltene.

A.3 Bedienung des Turbo-Designers

Der Turbo-Designer besteht aus der Arbeitsfläche, der Anzeigenzeile und der Menüzeile (siehe Abbildung A.1)

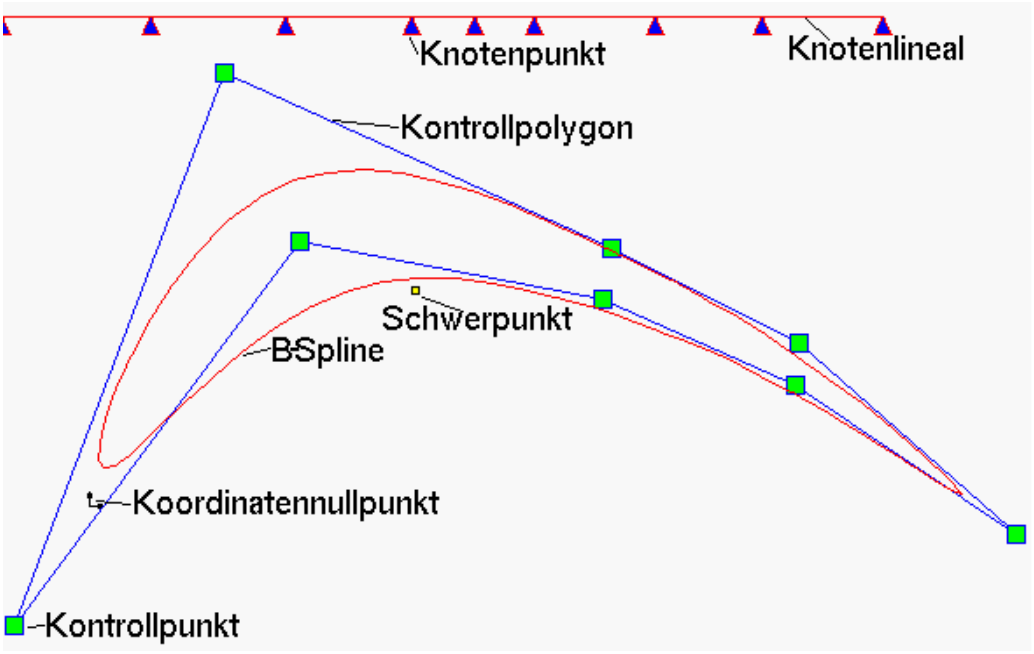


Abbildung A.2: Arbeitsfläche und deren Elemente

A.3.1 Die Arbeitsfläche und deren Elemente

Auf der Arbeitsfläche (siehe Abbildung A.2) finden sich Kontrollpunkte P_i mit dem zugehörigen Kontrollpolygon, der B–Spline und das Knotenpunktlineal mit dessen Knotenpunkten t_i. Jedoch werden nicht die periodischen Ergänzungen der Knoten- und Kontrollpunkte angezeigt, da diese sich aus den angezeigten Punkten ergeben.

Das B–Spline im Turbo-Designer repräsentiert einen Querschnitt durch ein Schaufelprofil. In der Abbildung A.3 wird der repräsentierte Querschnitt für eine Axialmaschine dargestellt. Die Maschinenachse wird mit z bezeichnet, die Lage des Querschnittes wird durch eine Abstandskoordinate r und einer Winkelkoordinate φ, die sog. Polarkoordinaten, beschrieben. Die Länge l bleibt in allen Darstellungen gleich. Die Breite b ist in der Darstellung des Turbo Designer im Vergleich zur Draufsicht breiter, da der Turbodesigner eine Abwicklung des Querschnittes darstellt.

Die Knotenpunkte, Kontrollpunkte und das Kontrollpolygon: Alle Knotenpunkte sind auf dem Knotenlineal aufgereiht. Die Verbindung der Kontrollpunkte wird als Kontrollpolygon bezeichnet. Mit Hilfe der Kontroll-

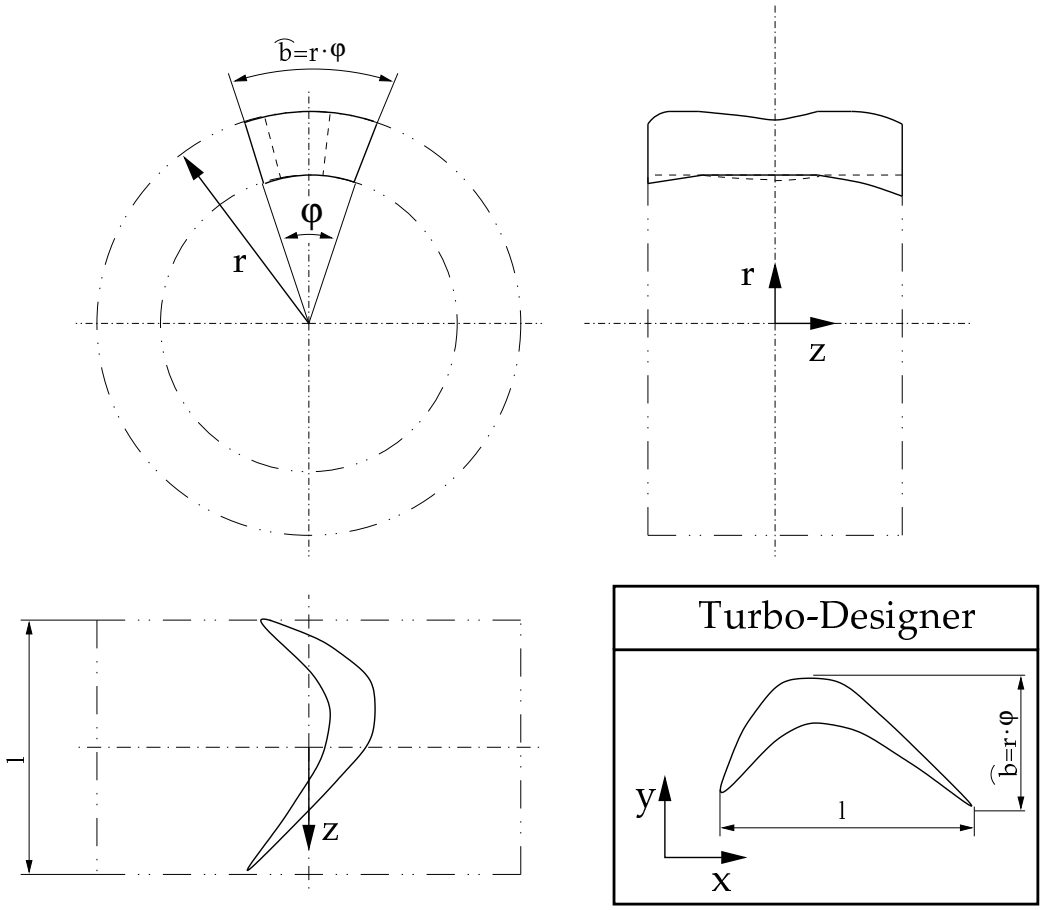


Abbildung A.3: Der im Turbo-Designer repräsentierte Querschnitt durch das Profil einer Axialmaschine

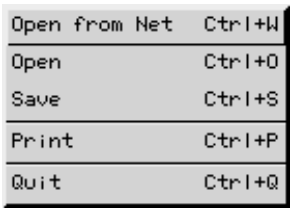


Abbildung A.4: Das File-Menü

punkte und der Knotenpunkte kann die äußere Form des B–Splines manipuliert werden. Dazu werden diese verschoben bzw. eingefügt und gelöscht. Um ein zügiges Manipulieren auf langsamen Rechnern zu gewährleisten, sollte im Fenster *Preferences* die Option *Centre of gravity* abgeschaltet werden. Zum Einfügen eines Kontrollpunktes gibt es zwei Methoden:

Entweder es wird mit der Maus auf einen vorhandenen Kontrollpunkt mit der rechten Maustaste gedrückt (dieses ändert seine Farbe von grün nach rot) und in dem erscheinenden Menü *insert* gewählt, so wird ein neuer Punkt gleich neben dem gewählten Punkt eingefügt,

oder es wird mit der Maus auf dem Knotenlineal zwischen zwei Knotenpunkten ein Einfügepunkt gewählt, auf dem Kontrollpolygon wird ein kleines blaues Viereck dargestellt, die rechte Maustaste gedrückt und in dem erscheinenden Menü *insert* gewählt. An der Position des kleinen blauen Vierecks wird ein neuer Kontrollpunkt eingefügt.

Gleichzeitig wird ein neuer Knotenpunkt mit eingefügt, damit die Definition des B–Splines erfüllt wird!

Um einen Kontrollpunkt zu löschen, wird mit der Maus ein vorhandener Kontrollpunkt oder der zugehörige Knotenpunkt mit der rechten Maustaste gewählt (der Kontrollpunkt ändert seine Farbe von grün nach rot bzw. von grün nach gelb) und in dem erscheinenden Menü *erase* gewählt. Hierdurch wird, falls die Definition des B–Splines dieses zuläßt, der entsprechende Kontrollpunkt und ein Knotenpunkt gelöscht.

A.3.2 Die Befehle im *File*-Menü

Im *File*-Menü kann man Dateien laden bzw. speichern, den B–Spline drucken und den Turbo-Designer beenden (siehe Abbildung A.4). Das Drucken der Darstellung des B–Splines und seiner Kontrollelemente, das Laden und Speichern einer Datei im lokalen Dateisystem ist nur möglich, wenn dies dem

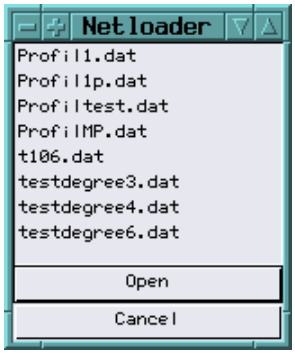


Abbildung A.5: Die Datei-Auswahbox (Netloader)

Turbo Designer ausdrücklich erlaubt wurde. Hierzu sind die Zugriffsrechte des Applets entsprechend zu ändern (siehe Abschnitt A.5).

Laden einer Datei: Um eine Datei zu laden, gibt es zwei Methoden. Die erste besteht darin, eine Datei über das Internet vom Server zu laden. Hierbei kann nur auf die vom Programm vorgesehenen Dateien zugegriffen werden. Dies geht im *File*-Menü mit dem Menüpunkt *Open from Net* bzw. mit der Tastenkombination (Strg)+(W). Hierbei erscheint eine Auswahlbox, in der die entsprechenden Dateien angezeigt werden (siehe Abbildung A.5).

Die zweite besteht darin, eine Datei vom Client zu laden. Dazu dient der Menüpunkt *Open* im *File*-Menü bzw. mit der Tastenkombination (Strg)+(O). Hierbei erscheint das im jeweiligen Betriebssystem übliche Menü (siehe Abbildung A.6). Falls versucht wird, eine Datei zu laden, die nicht das richtige Format hat, werden Fehlermeldungen ausgegeben.

Speichern einer Datei im lokalen Dateisystem Das Speichern einer Datei erfolgt im *File*-Menü mit dem Menüpunkt *Save* bzw. mit der Tastenkombination (Strg)+(S). Hierbei erscheint das im jeweiligen Betriebssystem übliche Menü (siehe Abbildung A.7).

Drucken des B–Splines Um die Darstellung des B–Splines und seiner Kontrollelemente auszudrucken, muß im *File*-Menü der Menüpunkt *Plot* bzw. die Tastenkombination (Strg)+(P) gewählt werden. Hierbei erscheint das im jeweiligen Betriebssystem übliche Menü.

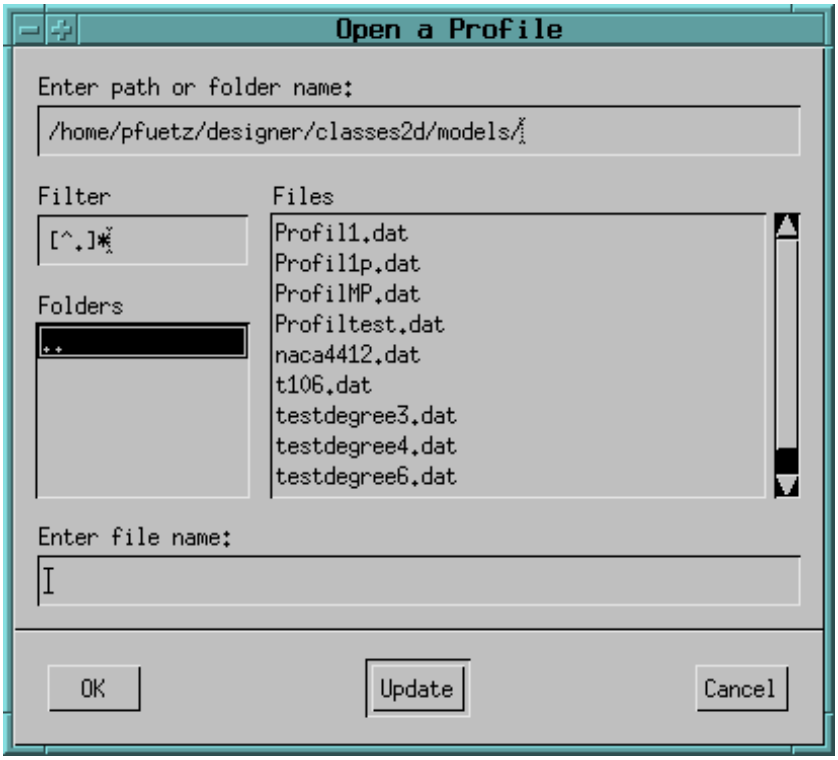


Abbildung A.6: Das Linux-übliche Menü zum Datei laden

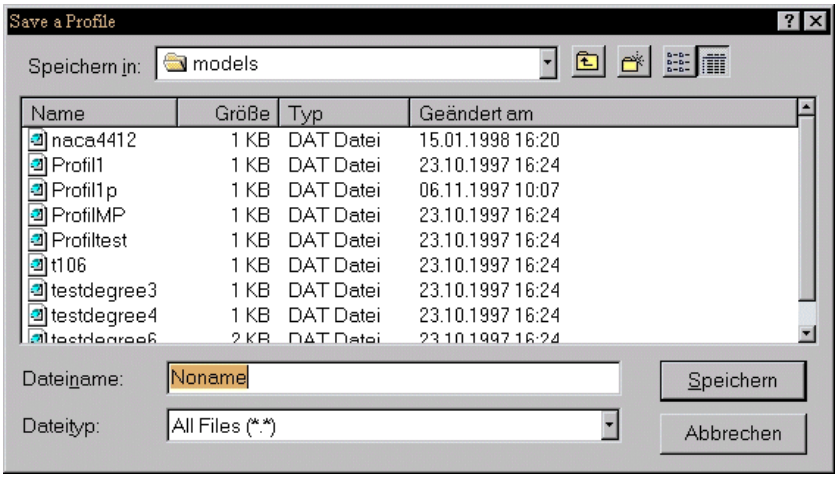


Abbildung A.7: Das Windows95-übliche Menü zum Datei speichern

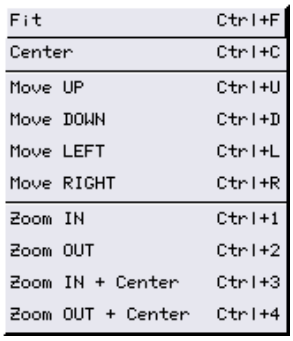


Abbildung A.8: Das *View*-Menü

Beenden des Turbo-Designers Um den Turbo-Designer zu beenden, wird der Menüpunkt *Quit* bzw. die Tastenkombination (Strg)+(Q) gewählt.

A.3.3 Die Befehle im *View*-Menü

Zur Manipulation der Darstellung des B–Splines mit dem Kontrollpolygon gibt es einige Befehle im *View*-Menü (siehe Abbildung A.8).

B–Spline in Arbeitsbereich einpassen Um die Darstellung im Arbeitsbereich des Turbo-Designers einzupassen, gibt es im *View*-Menü den Menüpunkt *Fit*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(F) gewählt werden.

B–Spline im Arbeitsbereich zentrieren Um die Darstellung im Arbeitsbereich des Turbo-Designers zu zentrieren, gibt es im *View*-Menü den Menüpunkt *Center*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(C) gewählt werden.

B–Spline nach oben bewegen Um die Darstellung des B–Splines nach oben zu verschieben, gibt es im *View*-Menü den Menüpunkt *Move UP*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(U) gewählt werden.

B–Spline nach unten bewegen Um die Darstellung des B–Splines nach unten zu verschieben, gibt es im *View*-Menü den Menüpunkt *Move DOWN*.

Diese Funktion kann auch mit der Tastenkombination (Strg)+(D) gewählt werden.

B–Spline nach links bewegen Um die Darstellung des B–Splines nach links zu verschieben, gibt es im *View*-Menü den Menüpunkt *Move LEFT*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(L) gewählt werden.

B–Spline nach rechts bewegen Um die Darstellung des B–Splines nach rechts zu verschieben, gibt es im *View*-Menü den Menüpunkt *Move RIGHT*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(R) gewählt werden.

B–Spline heranzoomen Um die Darstellung zu vergrößern, gibt es im *View*-Menü den Menüpunkt *Zoom IN*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(1) gewählt werden.

B–Spline wegzoomen Um die Darstellung zu verkleinern, gibt es im *View*-Menü den Menüpunkt *Zoom OUT*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(2) gewählt werden.

B–Spline heranzoomen und zentrieren Um die Darstellung zu vergrößern und gleichzeitig zu zentrieren, gibt es im *View*-Menü den Menüpunkt *Zoom IN + Center*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(3) gewählt werden.

B–Spline wegzoomen und zentrieren Um die Darstellung zu verkleinern und gleichzeitig zu zentrieren, gibt es im *View*-Menü den Menüpunkt *Zoom OUT + Center*. Diese Funktion kann auch mit der Tastenkombination (Strg)+(4) gewählt werden.

Bei der Befehlswahl über eine Tastenkombination im *View*-Menü wird eine geringere Veränderung vorgenommen als bei der Wahl über die Menüpunkte. Durch längeres Halten einer Tastenkombination wird der zugehörige Befehl kontinuierlich ausgeführt.

Abbildung A.9: Das *Options*-Menü

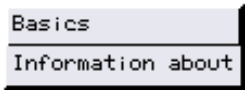


Abbildung A.10: *Help*-Menü

A.3.4 Die Befehle im *Option*-Menü

Grad des B–Splines verändern Um den Grad des B–Splines zu verändern, gibt es im *Option*-Menü (siehe Abbildung A.9) den Menüpunkt *Degree +1* bzw. *Degree -1*. Dies geht allerdings nur, falls die Voraussetzungen hierfür erfüllt sind.

Das Fenster *Preferences* bzw. *Properties* öffnen Um das Fenster *Preferences* bzw. *Properties* zu öffnen, muß im *Option*-Menü (siehe Abbildung A.9) der Menüpunkt *Preferences* bzw. *Properties* gewählt werden. Im Fenster *Preferences* sind einige Einstellungen möglich, die im Kapitel A.3.7 erklärt werden. Im Fenster *Properties* werden einige Informationen angezeigt, die im Kapitel A.3.8 erklärt werden.

A.3.5 Die Befehle im *Help*-Menü

Das Fenster *information about* öffnen Um das Fenster *Information about* zu öffnen, muß im *Help*-Menü (siehe Abbildung A.10) der Menüpunkt *Information about* gewählt werden. Daraufhin erscheint ein Fenster (siehe Abbildung A.11) mit Informationen über den Turbo-Designer und das lokale System.

Das Fenster *Basics* öffnen Um das Fenster *Basics* zu öffnen, muß im *Help*-Menü (siehe Abbildung A.10) der Menüpunkt *Basics* gewählt werden.

Abbildung A.11: Fenster *Information about*

Daraufhin erscheint ein Fenster (siehe Abbildung A.12) mit einem Hilfstext zum Turbo-Designer.

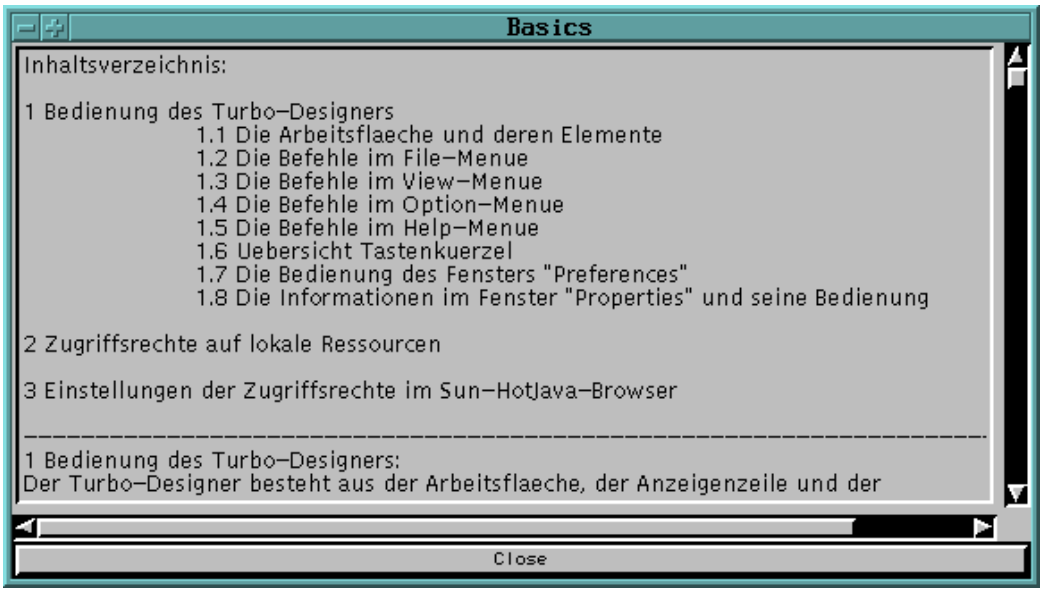


Abbildung A.12: Fenster *Basics*

A.3.6 Übersicht Tastenkürzel

Strg+W	Open from Net	Datei vom Server laden
Strg+O	Open	Datei vom Client laden
Strg+S	Save	Datei auf den Client speichern
Strg+P	Print	B–Spline drucken
Strg+Q	Quit	Turbo-Designer beenden
Strg+F	Fit	B–Spline auf Fenstergröße einpassen
Strg+C	Center	B–Spline zentrieren
Strg+U	Move UP	B–Spline nach oben bewegen
Strg+D	Move DOWN	B–Spline nach unten bewegen
Strg+L	Move LEFT	B–Spline nach links bewegen
Strg+R	Move RIGHT	B–Spline nach rechts bewegen
Strg+1	Zoom IN	B–Spline heranzoomen
Strg+2	Zoom OUT	B–Spline wegzoomen
Strg+3	Zoom IN + Center	B–Spline heranzoomen und zentrieren
Strg+4	Zoom OUT + Center	B–Spline wegzoomen und zentrieren

A.3.7 Die Bedienung des Fensters *Preferences*

Im Fenster *Preferences* (siehe Abbildung A.13) können einige Dinge eingestellt werden. Hierzu kann in das Eingabefeld unter der jeweiligen Funktionsbeschreibung ein Wert eingegeben werden.

Unter den Feldern *Blades* und *Radius* kann die Anzahl der Schaufeln auf dem Umfang bzw. der Radius, auf dem der Profilschnitt (B–Spline) liegt, eingegeben werden. Diese Einstellungen sind für den Turbo-Designer nicht nötig, sie werden für andere Programme beim Speichern mit in die Ausgabedatei (siehe Kapitel A.3.9), geschrieben.

Bei *Number of Plotpoints* kann die Anzahl der berechneten Punkte, durch die das B–Spline dargestellt wird, eingestellt werden. Ein höherer Wert bedeutet höhere Genauigkeit der Darstellung des B–Splines, aber auch eine längere Berechnungszeit, ein niedrigerer Wert entsprechend umgekehrt. Der Standardwert ist 60 Punkte.

Für Testzwecke gibt es einen Schalter *Debug*. Er ist bei den Standardeinstellungen nicht aktiviert. Im Normalfall ist dies auch nicht nötig. Sollten Unregelmäßigkeiten bei der Darstellung des B–Splines auftreten, sollte dieser aktiviert werden. Diese Option kostet jedoch zusätzliche Rechenzeit.

Um eine graphische Anzeige des B–Spline-Schwerpunktes zu erreichen, kann mit dem Schalter *Centre of gravity* diese aktiviert werden. Er ist bei den Standardeinstellungen deaktiviert. Er verbraucht zusätzliche Rechenzeit und sollte auf langsamen Rechnern während des Schaufelentwurfs deaktiviert werden, um einen schnelleren Bildschirmaufbau zu erreichen.

A.3.8 Die Informationen im Fenster *Properties*

Im Fenster *Preferences* (siehe Abbildung A.14) werden verschiedene Informationen über den aktuellen B–Splines angezeigt. Es benötigt jedoch zusätzliche Rechenzeit. Um genauere Ergebnisse zu erhalten, kann mit der Taste *precision high* eine genauere Berechnung erreicht werden. Diese wird aber nur einmal pro Tastendruck durchgeführt. Um die Ergebnisse zu vergleichen, kann durch Drücken der Taste *precision low* auf die sonst angezeigten Werte umgeschaltet werden.

Folgende Dinge werden angezeigt:

Centre of gravity x: Abstand des B–Spline-Schwerpunktes in X-Richtung zum Koordinatennullpunkt, *Centre of gravity y:* Abstand des B–Spline-Schwerpunktes in Y-Richtung zum Koordinatennullpunkt, *Area A:* Fläche

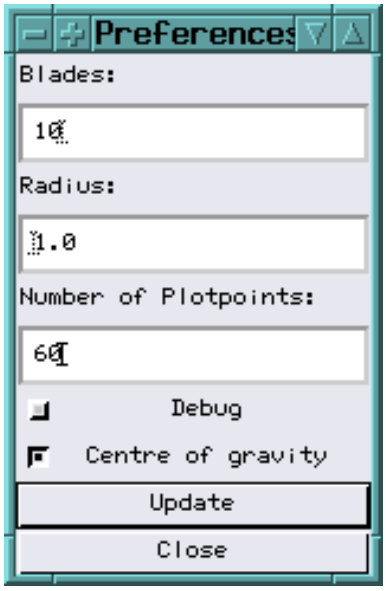


Abbildung A.13: Das Fenster Preferences

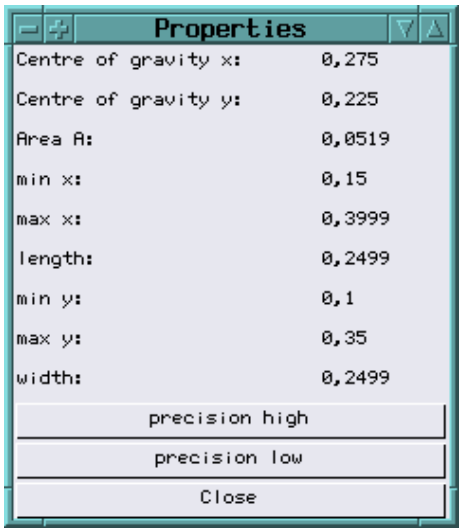


Abbildung A.14: Das Fenster Properties

des B–Splines, *min x:* minimale X-Position des B–Splines, *max x:* maximale X-Position des B–Splines, *length:* Länge des B–Splines ($x_{max} - x_{min}$), *min y:* minimale Y-Position des B–Splines, *max y:* maximale Y-Position des B–Splines, *width:* Breite des B–Splines ($y_{max} - y_{min}$).

A.3.9 Konventionen der Turbo-Designer-Dateien

Eine Turbo-Designer-Datei besteht aus mehreren Zahlenblöcken, die durch jeweils eine leere Zeile getrennt werden.

Die im zuerst gefundenen Block enthaltenen Zahlen werden als Kontrollpunkte mit periodischer Ergänzung interpretiert,

der nächste als Knotenpunkte mit periodischer Ergänzung.

Danach folgen drei Blöcke mit nur einer Zahl, der Grad des B–Splines, die Anzahl der Schaufeln auf dem Umfang und der Radius, auf dem der Profilschnitt (B–Spline) liegt.

Alle danach folgenden Zeilen werden ignoriert. Zeilen, die mit dem Zeichen `#` beginnen, werden ebenfalls ignoriert.

Beispiel einer Turbo-Designer-Datei:

```
#
# Turbo Designer V1.0, (c) 97, 98 Uni Kassel
# Institut für Thermische Energietechnik
# Fachgebiet Strömungsmaschinen
# http://www.ite.maschinenbau.uni-kassel.de/turbo
#
# API Version: 45.3
# Java Vendor: Sun Microsystems Inc., ported by Randy Chapman and Steve Byrne
# Architectur: x86
#
# Autor: Michael Pfütz
# Datum: 31.03.1998
#

# Control points
          0.25           0.25
          0.25           0.75
          0.75           0.75
          0.75           0.25
```

```
          0.25            0.25
          0.25            0.75

# Knot sequence
          -1
           0
           1
           2
           3
           4
           5
           6
           7

# Degree
             2

# Blades
            30

# Radius
             1
```

A.4 Die Bedienung des Turbo-Viewers

Der Turbo-Viewer dient der Darstellung eines 3D-Drahtmodels der kompletten Schaufel. Diese kann mit der Maus am Bildschirm beliebig rotiert werden. Der Turbo-Viewer wird in einem Java-fähigen Browser mit der HTML-Datei `netloader.html` geladen.

A.4.1 Der *Netloader*

Der *Netloader* (siehe Abbildung A.15) dient dazu, zwischen verschiedenen 3D-Modellen zu wählen. Er wird immer vor dem eigentlichen 3D-Viewer ausgeführt. Um ein Model zu laden, muß dieses im *Netloader* markiert werden, danach muß der Knopf *Open* gedrückt werden. Danach wird das 3D-Model im Viewer dargestellt. War kein Model ausgewählt, erscheint im Viewer `NO MODEL`. Um ein neues Model zu betrachten, muß der Viewer mit dem *Netloader* neu aufgerufen werden.

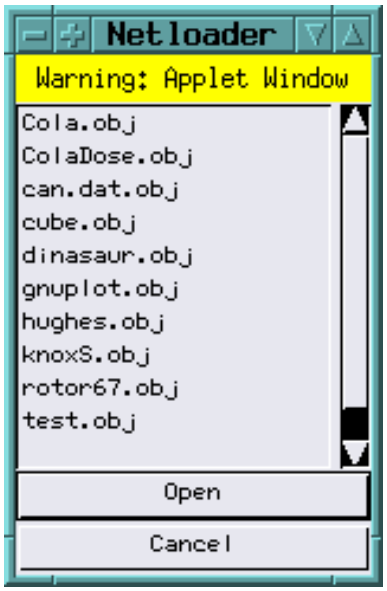


Abbildung A.15: Netloader

A.4.2 Der 3D-Viewer

Die Bedienung des Turbo-Viewers (siehe Abbildung A.16) beschränkt sich auf das Rotieren des 3D-Models. Zum Rotieren des Models wird mit der Maus auf den Hintergrund des Viewers geklickt und der Mausknopf gehalten, durch Bewegen der Maus auf diesem dreht sich das Model entsprechend. Das Model wird entweder in einer HTML-Datei festgelegt oder vom Netloader ausgewählt.

Konventionen der 3D-Eingabedatei

Der Turbo-Viewer benötigt eine Eingabedatei mit folgendem Format:
Als erstes ein Block mit Knotenpositionen. Dieser besteht aus mehreren Zeilen, die jeweils mit einem `v` beginnen, nach dem `v` kommen die X-, Y- und Z-Koordinaten des jeweiligen Knotens. Alle Angaben werden durch ein Leer- oder Tab-Zeichen getrennt.

Danach folgen Zeilen mit Anweisungen, welche Knoten miteinander zu verbinden sind. Sie sollten, falls mehrere Knoten in einem Ring verbunden werden sollen, mit einem `f` beginnen und danach die Nummer der jeweiligen zu verbindenden Knoten. Falls nur zwei Knoten miteinander verbunden werden sollen, muß die Zeile mit einem `l` beginnen, darauf folgen in diesen Zeilen nur

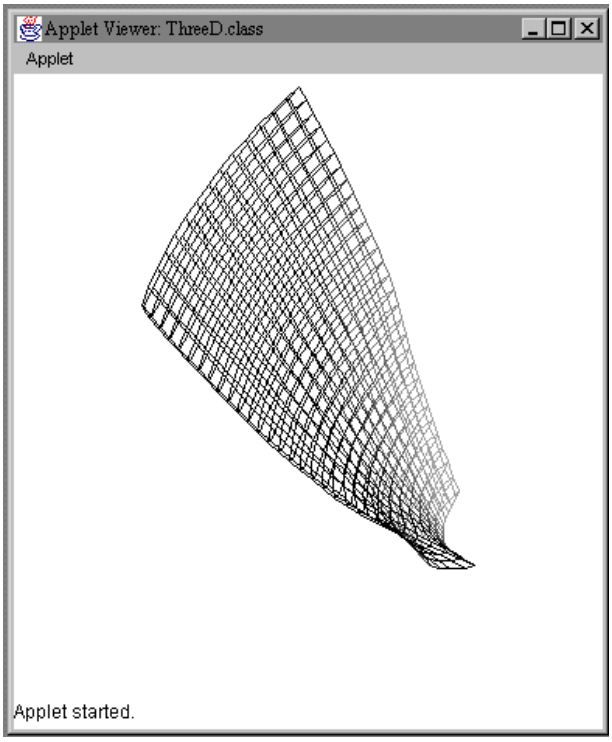


Abbildung A.16: 3D-Viewer

die Nummern der beiden Knoten. Auch in diesen Zeilen werden alle Angaben durch Leer- oder Tab-Zeichen getrennt.

Nach diesen Blöcken können weitere Blöcke mit Knotenpositionen bzw. Verbindungen folgen.
Beispiel einer Eingabedatei, die einen Würfel darstellt:

```
v 0 0 0
v 1 0 0
v 1 1 0
v 0 1 0
v 0 0 1
v 1 0 1
v 1 1 1
v 0 1 1
f 1 2 3 4
f 5 6 7 8
l 1 5
l 2 6
l 3 7
l 4 8
```

Konvertierung einer Gnu-Plot-Datei in eine Turbo-Viewer-Datei

Um Dateien im GNU-Plot-Format zu verarbeiten, gibt es einen Konverter namens `dat2obj.class`, er kann auch gleich eine HTML-Datei für den Viewer erzeugen.

A.5 Einstellungen der Zugriffsrechte im Sun-HotJava-Browser

Um auf das lokale Dateisystem des Clients zugreifen zu können, muß im HotJava-Browser einiges eingestellt werden.

Dies geschieht unter dem Menüpunkt ***Applet Security*** ... (siehe Abbildung A.17). Hier muß auf den Button ***Advanced*** >> geklickt werden, der sich rechts unten auf der Seite *Applet Security* befindet(siehe Abbildung A.18). Dort muß der Button ***Add Site*** angeklickt werden. In dem erscheinenden Menü (siehe Abbildung A.19) muß der Domainname
photon.ite.maschinenbau.uni-kassel.de

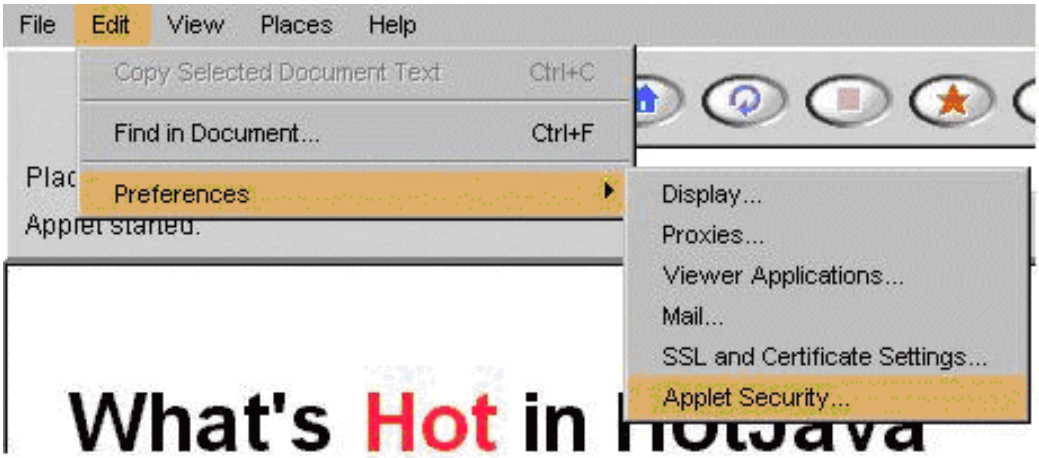


Abbildung A.17: Menü für die Einstellungen der Zugriffsrechte

Abbildung A.18: Button *Advanced* >>

eingetragen werden und mit ***Add*** bestätigt werden. Im unteren Bereich der Seite müssen noch einige Einstellungen (siehe Abbildung A.20) gemacht werden:

```
[ ] Use default permissions for this site or certificate.
[x] Applet can open windows without warning banner
[x] Applet may access all properties
[ ] Applet may access clipboard
[x] Applet may access print jobs
[ ] Applet may launch local applications
```

Unter dem Punkt ***Access to Files*** können noch Verzeichnisse und Dateien des Clients angegeben werden, auf die ohne Nachfrage zugegriffen werden kann. Die Einstellungen der Check-Boxen sollten sicherheitshalber, wie hier geschehen (siehe Abbildung A.21), vorgenommen werden:

```
[x] Warn before granting access to other files
[x] Warn before granting write acess to other files
[x] Warn when applets tries to delete a file
[ ] Don't warn when deleting files in above list
```

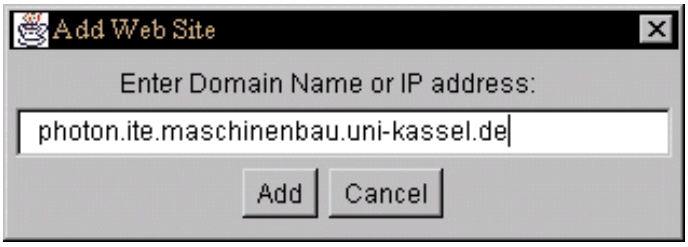


Abbildung A.19: Menü zum Hinzufügen des Domainnamens

Abbildung A.20: *Advanced Security Settings*: System Zugriff

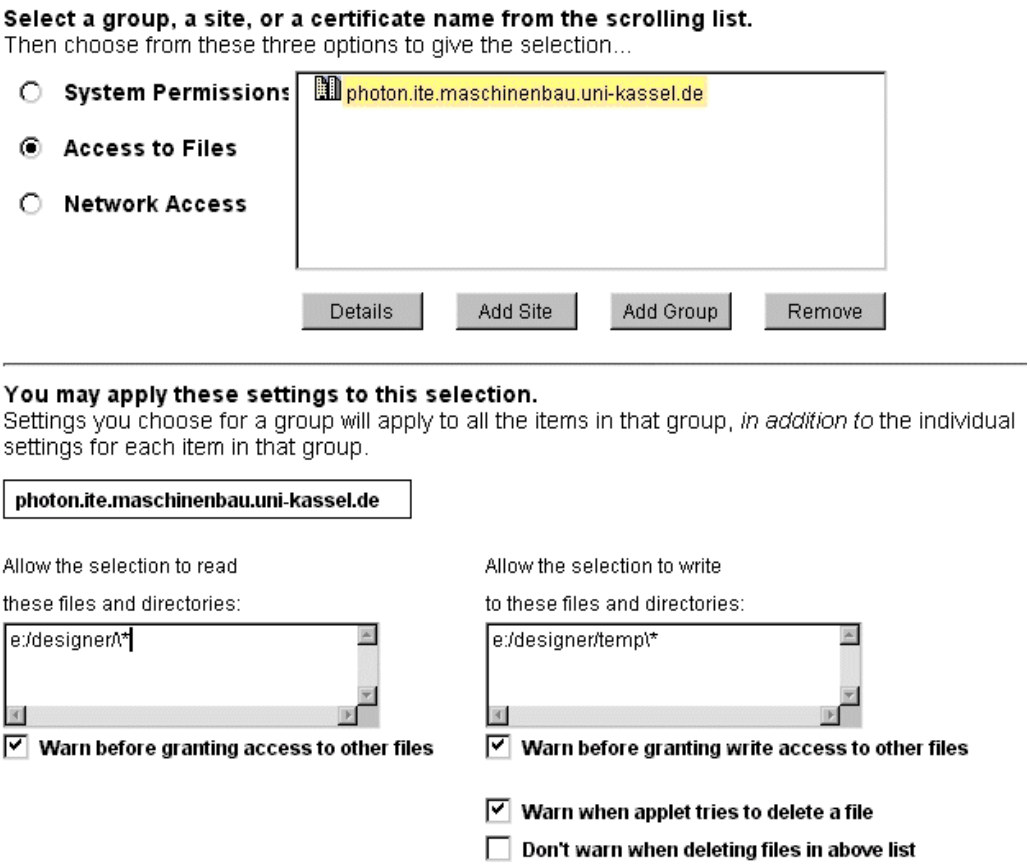


Abbildung A.21: *Advanced Security Settings*: Datei Zugriff

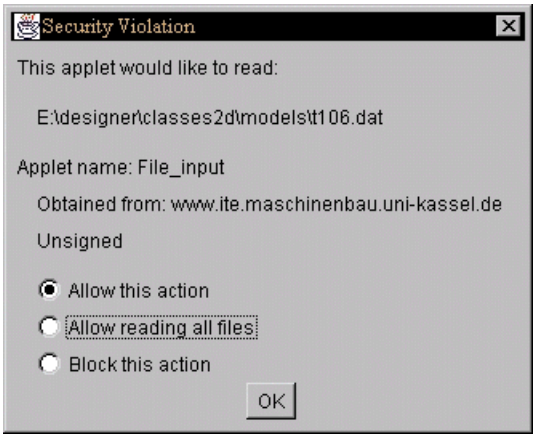


Abbildung A.22: Warnmeldung

Hierdurch wird bei einem Zugriff auf andere Dateien bzw. Verzeichnisse die Warnmeldung in Abbildung A.22 ausgegeben. Hier kann der Zugriff entsprechend bewilligt oder verweigert werden.

Literaturverzeichnis

[1] Martin Lawerenz, Skript zur Vorlesung „Fluiddynamik der Strömungsmaschinen 1 WS 97/98 Teil 1“. Universität Gesamthochschule Kassel, Institut für Thermische Energietechnik, Fachbereich Strömungsmaschinen, 1997.

[2] Axel C. Schwickert, Das aktuelle Schlagwort: HTML-Hypertext Markup Language. Informatik Spektrum Band 20, Seiten 168-169, Springer-Verlag, 1997.

[3] Stephan Diehl, Das aktuelle Schlagwort: VRML. Informatik Spektrum Band 20, Seiten 294-295, Springer-Verlag, 1997.

[4] Franz Grieser und Thomas Irlbeck,Computer-Lexikon: Das Nachschlagewerk zum Thema EDV. C.H.Beck, München, 1995.

[5] Konstantin April, Einstieg Internet: Infos und Tools fürs Internet mit Windows 3.x und Windows 95. Naumann und Göbel Verlagsgesellschaft mbH.

[6] S. Gottwald, H. Küstner, M. Hellwich und H. Kästner. Handbuch der Mathematik: Ein Ratgeber für Schule und Praxis, zum Selbststudium besonders geeignet. VEB Bibliographisches Institut Leipzig, 1986

[7] Lothar Papula, Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler, Vieweg, 1994.

[8] Mary Campione und Kathy Walrath, Das Java Tutorial: Objektorientierte Programmierung für das Internet, Addison Wesley Longman Verlag GmbH, 1997

[9] Helmut F. Reibold, Ulrich Schmitz, Jürgen Seeger. Die Rivalen: Was ActiveX von Java, JavaScript und Plug-ins unterscheidet. iX, Februar 1997

[10] Martin Lawerenz, Michael Uelschen, Sandra Jensen. Dreidimensionaler Entwurf axialer Beschaufelungen und Geometrieoptimierung. Interner Bericht 1/98. Universität Gesamthochschule Kassel, Institut für Thermische Energietechnik, Fachbereich Strömungsmaschinen, April 1998

Index

3D-Eingabedatei, 48

Abwicklung, 34
ActiveX, 7, 8
Anwendungen, 8
Applet, 9, 12
Applet-Tag, 12
Applets, 8
Area A, 44
AWT, 11
Axialmaschine, 34

B–Spline, 1
    Aus B–Spline zoomen, 40
    Aus B–Spline zoomen und zentrieren, 40
    Grad des B–Splines erhöhen, 41
    Grad des B–Splines verringern, 41
    im Arbeitsbereich zentrieren, 39
    in Arbeitsbereich einpassen, 39
    In B–Spline zoomen, 40
    In B–Spline zoomen und zentrieren, 40
    nach links bewegen, 40
    nach oben bewegen, 39
    nach rechts bewegen, 40
    nach unten bewegen, 39
B–Splines, 25
Basics, 41
Basis–Splines, 25
Beenden des Turbo-Designers, 39
Benutzerschnittstelle, 11
Blades, 44
Breite des B–Splines, 46
Bytecode-Interpretierer, 9

Center, 39
Centre of gravity, 44
Centre of gravity x, 44
Centre of gravity y, 44
Client, 4
Communicator, 5

Debug, 44
Degree
    erhöhen, 41
    verringern, 41
Drucken des B–Splines, 37

Explorer, 5, 8

Fenster
    Basics, 41
    Information about, 41
    Preferences, 41
    Properties, 41
Fit, 39
Fläche des Profils, 44
Fließkomma-Format, 30

Gateway, 3
Geschwindigkeitsoptimierung, 44
GUI, 11

Help-Menue, 41–42
heranzoomen, 40
heranzoomen und zentrieren, 40
Hilfstext zum Turbo Designer, 41

HTML-Datei, 5, 12
HTML-Hypertext Markup Language, 4
Hypermedia, 4

Information about, 41
Informationen über den Turbo Designer, 41
IP-Adresse, 3

Java, 7
  Anwendungen, 8
  Applets, 8, 9
  Applications, 9
  Script, 7, 9
  Virtuelle Maschine, 9
Java-AWT
  Version 1.0.x, 29
  Version 1.1.x, 29
  Version 1.2.x, 31

Knotenpunkte, 34
Knotenpunktlineal, 34
Kontrollpolygon, 34
Kontrollpunkte, 34

Länge des B–Splines, 46
Laden
  einer Datei, 36
  einer Datei vom Client, 37
  einer Datei vom Server, 37
LAN, 3
length, 46
Local Area Network, 3

Maschinenaxe, 34
max x, 46
max y, 46
maximale X-Position, 46
maximale Y-Position, 46
Microsoft
  ActiveX, 7
  Explorer, 5, 8
min x, 46
min y, 46
minimale X-Position, 46
minimale Y-Position, 46
Move
  DOWN, 39
  LEFT, 40
  RIGHT, 40
  UP, 39

Netscape
  Communicator, 5
Number of Plotpoints
  Eingabefeld, 44

Option-Menue, 41
Options, 41

periodische Ergänzung, 34
Plug-ins, 5
Polarkoordinaten, 34
Preferences
  öffnen, 41
Programm Hallo, 11
Properties, 41

Querschnitt, 34

Radius, 44

Schaufel
  Anzahl, 44
Schaufelprofil
  Querschnitt, 34
Schaufelprofile, 1
Schwerpunkt
  Abstand in X-Richtung, 44
  Abstand in Y-Richtung, 44
Schwerpunktkoordinaten, 26
Server, 9
  Fileserver, 4
Speichern

Speichern einer Datei, 37
Strömungsberechnung, 31
Sun
Java, 7

Tag, 5, 9, 12
Tastenkürzel
Übersicht, 43
TCP/IP, 3
Terminal-Modell, 4

Uebertragungsprotokoll, 3
UI, 11

VBScript, 7
View-Menue, 39–40
Virtual Reality Markup Language, 5
VM, 9
VRML, 5

wegzoomen, 40
wegzoomen und zentrieren, 40
width, 46
Winkelkoordinate, 34
WWW, 4
WWW-Seiten
HTML, 4
Programme, 7

Zoom
heranzoomen, 40
heranzoomen und zentrieren, 40
wegzoomen, 40
wegzoomen und zentrieren, 40
Zugriffsrechte, 11, 31
HotJava Browser, 50

Quelltexte

Turbo-Designer V1.0

Ich versichere hiermit, daß ich diese Studienarbeit selbst verfaßt habe.
Kassel, 15. Mai 1998

Michael Pfütz