

# ReVoKoS – Ein interaktives Vorlesungsskript für die Ausbildung von Ingenieuren in Numerischer Mathematik

Friedemann Kemm, Claus-Dieter Munz, Mirko Pauli, Thomas Westermann

Universität Stuttgart, FH Karlsruhe  
Stuttgart (Germany), Karlsruhe (Germany)

ReVoKoS – Rechnergestützte Vorlesungskonzepte für die numerische Simulation – ist ein Projekt mit dem Ziel, die Möglichkeiten moderner Rechnerumgebungen – Hypertextelemente, Computeralgebrasysteme, moderne Programmiersprachen, Visualisierung – als Lehrwerkzeuge bei der Ausbildung von Ingenieursstudenten in numerischer Mathematik auszunutzen. Mit diesem Instrumentarium lassen sich Konzepte verwirklichen, welche sowohl interaktive Vorführungen von numerischen Rechnungen in der Vorlesung erlauben als auch den Studierenden Anreize geben, eigene Erfahrungen in der numerischen Simulation zu sammeln. Kernstück des in dieser Arbeit beschriebenen Konzepts ist ein interaktives Vorlesungsskript, das in seiner elektronischen Form zur praktischen, interaktiven Vorlesungsnachbereitung anregt. Die technische Realisierung und Bemerkungen zum Einsatz in einer Vorlesung „Numerische Mathematik für Ingenieure“ wird in dieser Arbeit beschreiben.

## Einleitung

Die numerische Simulation ingenieurwissenschaftlicher Probleme wird in der Industrie ein immer wichtigeres Werkzeug für Forschung und Entwicklung. Der Absolvent eines Ingenieurstudiums muss heute in diesem Bereich ein fundiertes Grundlagenwissen besitzen. Dies ist insofern schwierig, da neben dem theoretischen Wissen hierzu auch die eigenen Erfahrungen mit den numerischen Methoden, ihre Umsetzung in Rechenprogramme und die Beurteilung und Interpretation der Ergebnisse gehört. Bislang wurden diese Kenntnisse nur von Studenten erworben, welche im Bereich der numerischen Simulation eine Diplomarbeit anfertigten. Ohne Erfahrung in der Programmierung, Fehlersuche, und insbesondere in der praktischen Anwendung numerischer Verfahren war die Einarbeitung in ein Diplomarbeitsthema in diesem Bereich im allgemeinen aufwendiger als bei anderen Arbeiten, was zum einen die Nachfrage nach solchen Themen einschränkte als auch für die Betreuer einen erhöhten Betreuungsaufwand bedeutete. Die Überlegung, wie die Erfahrung mit numerischen Methoden, aber auch die Motivation der Studierenden für die numerische Mathematik schon möglichst früh verbessert werden kann, führte zu den in diesem Projekt umgesetzten Ansätzen.

Die eigene Erfahrung der Studierenden in der numerischen Simulation lässt sich gewissermaßen nur durch eigenes Rechnen vermitteln. Ein Rechnerpraktikum zu der Grundvorlesung in numerischer Mathematik ist allerdings sehr aufwendig zum einen für den Dozenten, da die Betreuungszeit sehr hoch ist, zum anderen sind die Voraussetzungen für die Studierenden sehr hoch. Sie müssen mit dem Betriebssystem, dem Compiler, der Visualisierungswerkzeugen meist auf einem CIP-Pool an der Universität umgehen können. In neuen Studienplänen im Ingenieurbereich ist die Datenverarbeitung mit einem Programmierpraktikum schon in den ersten Semestern ein Pflicht-

fach. Trotzdem ist man bei der Grundvorlesung „Numerischer Mathematik für Ingenieure“ in dem Rechnerpraktikum am Anfang vor allem damit beschäftigt, die Grundvoraussetzungen in der Umsetzung von numerischen Algorithmen in Rechenprogramme, deren Ausführung, deren Validierung und der Darstellung und Interpretation der Ergebnisse mit den Studierenden zu erarbeiten. Es ist nicht möglich, das Rechnerpraktikum parallel zur Vorlesung zu halten und die erzielten Ergebnisse begleitend mit in der Vorlesung zu diskutieren.

In diesem Artikel wird ein Konzept vorgestellt, wie schon in einer Grundlagenvorlesung „Numerische Methoden für Ingenieure“ durch die direkte Vorführung numerischer Simulationen in der Vorlesung, aber auch durch die Anregung zu Learning by Doing die beiden gleichermaßen wichtigen Aspekte, Theorie und Erfahrung gemeinsam vermittelt werden können. Die Vermittlung theoretischer Kenntnisse mit praktischen Erfahrungen wurde in diesem Projekt durch die Entwicklung eines elektronischen Vorlesungsskripts ermöglicht. Dabei kann der „Leser“ dieses Skripts auf einem Rechner Beispiele aus der Vorlesung durch einen einfachen Aufruf im Skript interaktiv nachrechnen. Er kann dann auch Parameter oder selbst das Beispiel verändern und die Ergebnisse interaktiv visualisieren. Es stellt damit weit weniger Voraussetzungen an den Benutzer als die Entwicklung von eigenen Rechenprogrammen im Rahmen eines Programmierpraktikums.

Computer werden heute sehr oft als Werkzeug zur Bereitstellung und Verarbeitung von Information genutzt. Zunächst wurden sie allerdings als Rechenmaschinen zur Lösung von Ingenieurproblemen entwickelt. Methoden der Informatik und der numerische Mathematik sind dabei für die Bereitstellung von Algorithmen und Programmen nötig. Dieses Skript bringt diese beiden Aspekte wieder zusammen: Das Vorlesungsskriptum bleibt kein reiner Text zum Lesen oder Ausdrucken, die numerischen Algorithmen können für Beispiele direkt ausgeführt

und ausprobiert, der Computer wird entsprechend seiner ursprünglichen Funktion auch gleich als Rechenmaschine mit einbezogen.

## Das interaktive Vorlesungsskriptum

Die numerische Simulation besteht darin, mathematische Modelle für Ingenieurprobleme auf dem Computer zu lösen, um Informationen über das technische Problem zu erhalten. Die Komplexität der mathematischen Modelle erlaubt im allgemeinen keine exakte Lösung sondern nur die näherungsweise Lösung mit Hilfe numerischer Verfahren. Die Entwicklung geeigneter Verfahren zur Berechnung von Näherungslösungen ist Aufgabe der numerischen Mathematik. Den gesamten Prozess von der ingenieurwissenschaftlichen Modellierung des Ausgangsproblems bis hin zum Computerprogramm und dessen Anwendung umfasst die numerische Simulation. Die kritische Auswertung und Interpretation der Simulationsergebnisse in Bezug auf die Näherungsverfahren, aber auch in Bezug auf die ursprüngliche technische Problemstellung erfordert Erfahrung, die den Studierenden ohne den direkten Einsatz von Computern in der Lehrveranstaltung nur unzureichend vermittelt werden kann.

Schon in den Grundvorlesungen sollten die Studierenden möglichst schnell eigene Erfahrungen mit numerischen Simulationen machen - möglichst anhand konkreter Probleme, welche mit dem Thema des Studiums eng zusammenhängen. Damit werden zum einen Fertigkeiten in der numerischen Simulation erworben, zum anderen auch die Motivation an der numerischen Simulation erhöht, da die Studierenden die Verbindung des Ingenieurfachs zu den wichtigen aber auch schwierigen mathematischen und numerischen Methoden besser sehen.

Mit diesem Ziel wurde ein Konzept entwickelt, dessen wesentlicher Baustein ein elektronisches Vorlesungsskript ist, aus dem heraus Beispiele interaktiv ausgeführt werden können. Die Ideen und die Konstruktion der numerischen Methoden werden in der Vorlesung vorgestellt, wobei die entsprechenden Beispiele in dem Skriptum zur Motivation und Demonstration benutzt werden. Die Ergebnisse werden für alle sichtbar über einen Beamer dargestellt und direkt in die Vorlesung einbezogen. Die Diskussion der Ergebnisse kann im Gespräch mit den Zuhörern erfolgen. Zur Nachbearbeitung der Vorlesung können die Studierenden diese Beispiele dann aber auch durch das elektronische Skriptum nachrechnen und haben die Möglichkeit, einzelne Parameter oder ganze Beispiele selbst zu ändern und damit zu „spielen“.

Man braucht also ein druckbares Skriptum mit interaktiven Beispielen, die am Rechner genutzt werden können. Dabei ist es sinnvoll, für den Textteil ein Dateiformat zu verwenden, das einerseits systemübergreifend nutzbar ist und lediglich kostenlose oder kostengünstige Software voraussetzt, andererseits aber die Möglichkeit bietet, die interaktiven Beispiele direkt aus dem Text heraus zu starten. Hierfür bieten sich drei Formate an:

- **HTML:** Dies ist das Standardformat im World Wide Web und daher überall verfügbar. Die Möglichkeiten, externe Dateien und Anwendungen zu verlinken, gehören zu ureigensten Wesen des Dateiformats. Allerdings ist die Darstellung je nach verwendetem Programm und Einstellungen unterschiedlich. Auch der Ausdruck differiert entsprechend. Es wird sich also kaum eine vernünftige Übereinstimmung von Bildschirmdarstellung und Papierversion herstellen lassen.

Überdies muss man, um mathematische Formeln und Zeichen darzustellen, auf Grafiken zurückgreifen. Dies lässt HTML insgesamt als für ein Skriptum nicht geeignet erscheinen.

- **MathML:** Hierbei handelt es sich um eine Erweiterung von HTML, um Mathematik mit Formeln und Sonderzeichen, ebenfalls direkt darstellen zu können. Damit ist ein Nachteil von HTML aufgehoben. Die anderen bleiben allerdings bestehen. Da die Unterstützung von MathML durch die angebotenen Webbrowser nur langsam voranschreitet, kommt der Nachteil der schlechten Verfügbarkeit hinzu.
- **PDF:** Hierbei handelt es sich eigentlich um ein proprietäres Dateiformat der Firma Adobe. Da aber das Betrachterprogramm Acrobat Reader kostenlos und ohne Registrierung abgegeben wird und überdies für alle gebräuchlichen Computerplattformen verfügbar ist, stellt dies keine Einschränkung dar. Auch in diesem Format kann wie bei HTML mit sogenannten Hyperlinks gearbeitet werden. Dabei besteht auch die Möglichkeit, externe Anwendungen aus dem Text heraus aufzurufen. Das Schriftbild ist mit dem Erzeugen der Datei fixiert. Es besteht also die bestmögliche Übereinstimmung zwischen Bildschirmdarstellung und Druckbild. PDF-Dateien lassen sich mittels pdfTeX direkt aus TeX heraus erzeugen. Damit ist ein professionelles Schriftbild sowie eine hervorragende Darstellung mathematischer Formeln und Zeichen gewährleistet.

Nun bleibt noch die Frage offen, wie die interaktiven Beispiele realisiert werden sollen. Hierfür bieten sich wiederum zwei grundsätzliche Wege an:

- Man kann die Beispiele samt graphischer Darstellung in einer höheren Programmiersprache programmieren und als in sich geschlossenes Programm aufrufen. Möchte man dies in System unabhängiger Weise realisieren, so bleibt eigentlich nur noch Java als Programmiersprache übrig. Der Vorteil ist, dass man außer der Existenz von Java auf dem Rechner der Nutzer, was meist gegeben ist, keine weiteren Softwarevoraussetzungen hat. Der sehr erhebliche Nachteil ist der extrem hohe Aufwand beim Erstellen der Programme. Außerdem muss man den Spielraum der Anwender beim Modifizieren der Beispiele von vornherein festlegen. Dabei bedeutet jede weitere Freiheit einen erheblichen zusätzlichen Programmieraufwand.
- Man kann aber auch ein existierendes Computeralgebra- (Maple, Mathematika, Mupad...) oder Numeriksystem (Matlab, Octave, Scilab...) verwenden. Dabei bieten die Computeralgebrasysteme (CA-Systeme) die höhere Flexibilität und Interaktivität, während die Numeriksysteme eine kürzere Ausführungszeit als Vorteil für sich verbuchen können. Da bei der Rechenleistung der heute üblichen PC's und der ohnehin geringen Rechenzeit für die Beispiele diese nicht ins Gewicht fällt, haben wir uns für ein CA-System, in unserem Falle Maple, entschieden. Hier kommt es auch sehr zupass, dass die Firma Waterloo für Windowsysteme eine kostenlose Testversion ihres Programms anbietet. Die dabei weggefallenen Funktionalitäten beeinträchtigen jedoch nicht die Nutzbarkeit der interaktiven Beispiele.

Das Ergebnis ist also eine PDF-Datei, aus der heraus sogenannte Maple-Worksheets bzw. Maple-Hilfeseiten verlinkt werden. Die Papierausgabe stimmt in Seitenzahlen, Gliederun-

gen etc. d.h. dem gesamten Schriftbild mit der Bildschirmgröße überein. Querverweise stellen gleichzeitig Links auf die Verweisstelle dar. Darüber hinaus gibt es spezielle Links – diese sind in einer anderen Farbe dargestellt –, die auf die externen Dateien bzw. Anwendungen zeigen. Klickt man diese an, so wird Maple mit dem entsprechenden Worksheet bzw. der entsprechenden Hilfedatei gestartet.

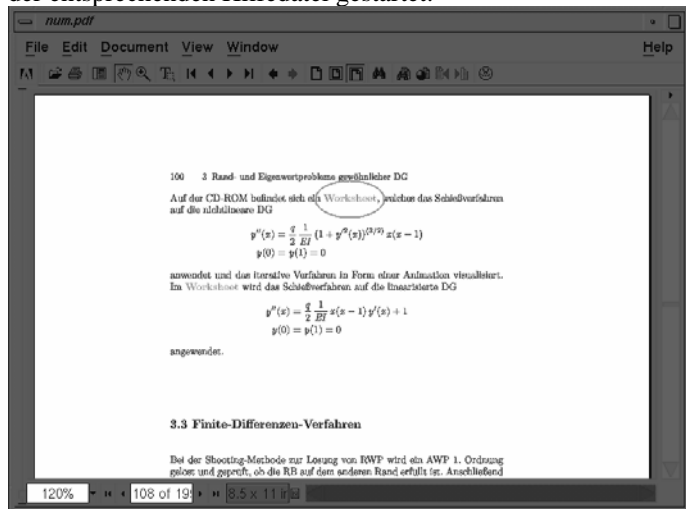


Abbildung 1: Das Skript im Acrobat Reader, eingekreist ein Link auf ein Worksheet

## Die technische Umsetzung

Zunächst braucht man nun eine Anwendung, die es erlaubt, die PDF-Datei bequem zu erstellen und dabei auch gleich die Hyperlinks einzufügen. Hier bietet sich pdfTeX bzw. pdfLaTeX an. LaTeX stellt mit seinem hervorragenden Formelsatz mittlerweile den Quasistandard beim Erstellen mathematischer Texte dar. Da das Ergebnis mit jeder TeX-Installation auf jedem Computersystem gleich aussieht, bietet es sich auch vorzüglich für Projekte an, an denen mehrere Beteiligte arbeiten. Die Hypertextfähigkeit wird über das LaTeX-Paket hyperref erreicht. Benutzt man pdfLaTeX, so wird direkt, will heißen ohne den Umweg über Postscript, eine PDF-Datei erzeugt.

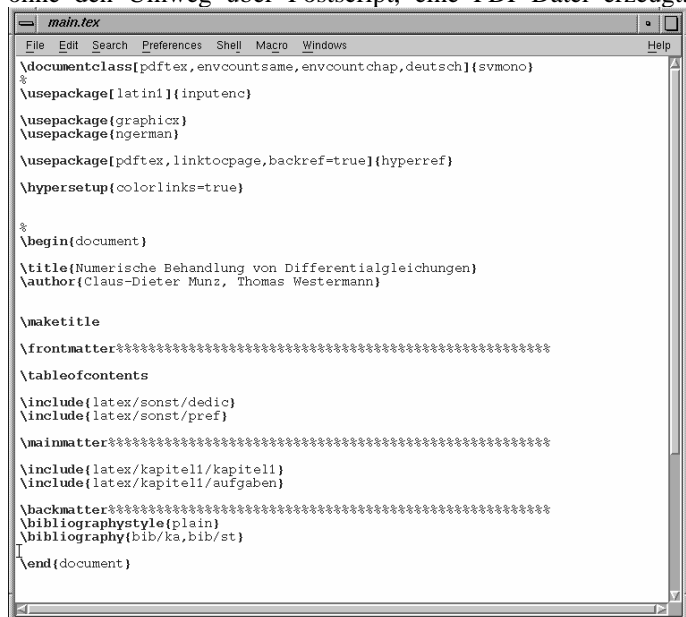


Abbildung 2: Die Hauptdatei im Editor

Dass dieses im Vergleich zu einer normalen LaTeX-Datei nur wenig Mehraufwand macht, lässt sich in Abbildung 2 sehr leicht nachvollziehen. Es sind lediglich der vierte `\usepackage-`

Befehl sowie die Zeile mit dem Befehl `\hypersetup` hinzugekommen. Das optionale Argument „pdfTeX“ teilt den verwendeten Paketen mit, dass pdfLaTeX verwendet wird. Allein die Verwendung von `hyperref` sorgt bereits dafür, dass alle Querverweise im Dokument auch als Link an die entsprechende Stelle ausgeführt werden. Außerdem stellt das Paket den Befehl `\href` bereit, mit dem die Links auf externe Dateien und Anwendungen realisiert werden können. Der in Abbildung 1 eingekreiste Link etwa wird durch den in Abbildung 3 eingekreisten Text erzeugt.

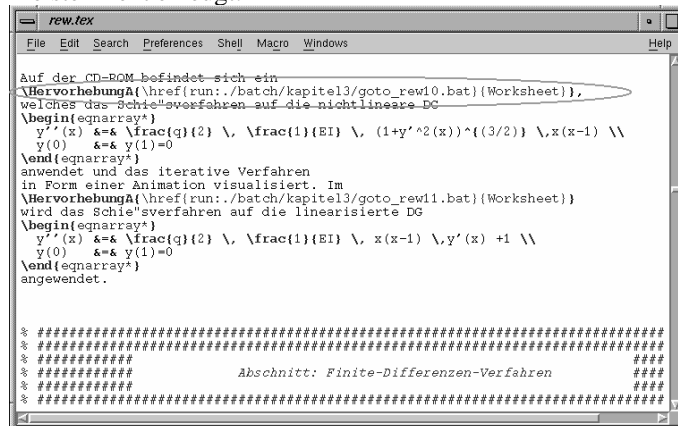


Abbildung 3: Quelltext für den Link auf ein Worksheet

Über die Gestaltung von Maple-Worksheets, um interaktives Arbeiten zu motivieren, die Möglichkeit, eigene Pakete mit eigenen Maple-Routinen zu erstellen, einzubinden und mit entsprechenden Hilfedateien zu versehen, soll an dieser Stelle nicht geschrieben werden. Es seien ein paar Dinge erwähnt, die sich in unserem Fall als sehr nützlich erwiesen haben:

- Die Rechengenauigkeit in Nachkommastellen kann explizit angegeben werden. Dadurch kann man die Effekte von Rundung und Rundungsfehlern sehr schön vermitteln.
- Gerade im Bereich der Differentialgleichungen ist Maple sehr stark. Daher hat man in vielen Beispielen auch die exakte Lösung zur Hand und kann mit ihr vergleichen.
- Auch Namen von aufzurufenden Funktionen können in Variablen des Typs „symbol“ gespeichert werden. Auf diese Weise lassen sich sehr einfach Schnittstellen zur die Auswahl verschiedener Verfahren für ein Problem erzeugen.
- Die Möglichkeit, eigene Routinen ins System einzubauen, erlaubt es, die allfälligen unübersichtlichen Tabellen mit Koeffizienten Diskretisierungsformeln etc. durch kleine Worksheets mit einer Schnittstelle zu geeigneten Routinen zu ersetzen, welche die gewünschten Koeffizienten, Formeln etc. ausgeben. Dabei ist die mögliche Bandbreite ungleich größer als bei jedem Tabellenwerk. Die Ergebnisse ihrerseits können von den Benutzern direkt in eine höhere Programmiersprache (Fortran bzw. C) ausgegeben werden. Was ihren Nutzen gegenüber einer Tabelle wesentlich vergrößert.

## Das neue Konzept

Mit dem oben beschriebenen Skript als Basismaterial lässt sich nun ein Konzept realisieren, das mit traditionellen Mitteln nicht möglich ist. Das Lernen geschieht in drei Stufen, in denen alle Lernkanäle genutzt werden. Dabei lassen sich die

„stärkeren“ Lernkanäle, Tun und Erleben, besser nutzen als dies in einem konventionellen Kurs möglich wäre.

1. **Hören und passives Erleben von numerischer Mathematik:** Dieses geschieht in der Vorlesung. Gerade bei einem großen Publikum – die betreffende Veranstaltung in Stuttgart hat mehr als 200 Hörer – lässt sich hier Interaktion nur eingeschränkt verwirklichen. Das Erleben aber ist bereits möglich, indem die Beispiele aus dem Skriptum im Vortrag verwendet werden. Numerische Methoden können direkt in der Vorlesung ausgeführt werden. Dabei lassen sich auch kleine Lernspiele mit dem Publikum einbauen.
2. **Lesen und eigenes Experimentieren:** Bei der Nachbereitung haben die Teilnehmer ein multimediales Produkt auf dem eigenen PC, das sie einlädt, das Gelesene auch gleich in der Praxis zu erproben. Die vorgearbeiteten Beispiele sind direkt greifbar und die Schnittstellen für eigene Modifikationen einfach. Das Erlernen der Programmiersprache in Maple geschieht automatisch.
3. **Umsetzen der Algorithmen in Struktogramme und eine höhere Programmiersprache:** Solchermaßen vorbereitet ist es für die Studierenden ein relativ kleiner Schritt dahin, konkrete Aufgabenstellungen zu bearbeiten. Dies geschieht in einem durch Tutorien begleiteten Rechnerpraktikum, unterstützt von einer Hörsaalübung, in der allgemeine Hilfestellungen zu Maple, der Programmiersprache (Fortran95), und Struktogrammen gegeben werden.

Im dritten Schritt gehen wir zu einer höheren Programmiersprache wie Fortran über. Alle Programmieraufgaben in dieser Grundvorlesung in Numerik lassen sich auch in der Form von Maple-Worksheets ausführen. Erst für komplexere Probleme sind die Möglichkeiten des CA-Systems erschöpft. Die Rechenzeit als auch der Umfang der Daten wird zu groß. Insofern ist ein Ziel in unserer Ausbildung, auch die Kenntnisse in der Programmierung in diesem Bereich zu vermitteln. Ist dies für den betreffenden Studiengang nicht so sehr wichtig oder steht zu wenig Zeit zur Verfügung, können auch die betreffenden Programmieraufgaben im Rahmen von Maple erledigt werden.

## Der Mehrwert

Worin liegt nun der Mehrwert eines solchen Konzepts? Lohnt es sich, den erforderlichen Aufwand auf sich zu nehmen? Unsere Erfahrung sagt ja.

### Mehr Spaß für Studenten und Dozenten

Die interaktiven Teile der Vorlesung und des Skriptums fördern die Mitarbeit und Motivation der Studierenden sehr stark. Der Dozent hat neue Möglichkeiten, seine Zuhörer in der Vorlesung zu aktivieren. Verschiedene Themen lassen sich hier sehr gut benutzen, gemeinsam Strategien und Verfahren zu erschließen. So ist etwa die Vorführung der Genauigkeitsordnung für ein praktisches Beispiel im Vergleich verschiedener Verfahren eine spannende Sache in der Vorlesung. Genauigkeit und Rechenaufwand, Einfluss der Rundungsfehler, alles dies lässt sich an aktuellen Rechnungen belegen und die Ergebnisse mit den Zuhörern zusammen diskutieren und erarbeiten. Wir glauben auch, dass auch die herabgesetzte Schwelle zum Ausprobieren, Ändern bis hin zum eigenen Programmieren die Studierenden mehr zur Eigeninitiative verleitet. Auch der Dozent hat mehr Spaß an der Vorlesung, wenn er mithilfe

der Maple-Worksheets oft spielerisch numerische Ergebnisse präsentieren kann. Der gesamte Vorgang: Änderung des Programms, Übersetzung in eine ausführbares Programm, Ausführung desselben und Visualisierung der Ergebnisse lässt sich in einem Computeralgebrasystem weitaus einfacher bewerkstelligen und damit erst in eine Vorlesung integrieren.

### Praktische Erfahrungen mit numerischen Verfahren

Ein besonderer Mehrwert ist, dass die Studierenden bereits Erfahrungen mit den numerischen Verfahren sammeln, bevor sie diese zum ersten Mal selber programmieren müssen. Zudem ist die Variabilität der in Maple realisierten Beispiele viel größer als dies bei einer von den Studenten umzusetzenden Programmieraufgabe erreicht werden kann. Durch die herabgesetzte Hemmschwelle zum eigenen Programmieren ist die Akzeptanz und die Motivation bei den Studierenden besser.

### Heranführung an Computeralgebrasysteme

Computeralgebrasysteme gewinnen immer mehr an Bedeutung in der Mathematik und im Ingenieurwesen. Damit ist es sinnvoll und für das spätere Berufsleben nützlich, wenn die Studierenden der Ingenieurwissenschaften schon frühzeitig an solche Systeme herangeführt werden. Im beschriebenen Vorlesungskonzept geschieht dies gewissermaßen nebenher.

### Erste Programmiererfahrung

Den Studierenden wird es leichter gemacht, die ersten Erfahrungen im eigenen Programmieren zu machen. Es tauchen Fragen auf, die nachher in der Praxis für die Beurteilung der Rechnungen eine Rolle spielen, etwa der Umgang mit der Rechengenauigkeit oder die Effizienz der Programmierung.

### Die Langzeitwirkung

Wir hoffen, dass das elektronische Skriptum auch als Nachschlagewerk und Anleitung nach dem Studium noch zum Einsatz kommt. Insbesondere die Möglichkeit, Tabellen von Koeffizienten etc. durch wesentlich leistungsfähigere Worksheets zu ersetzen, macht es hierfür besonders wertvoll. Als Grundlage für weiterführende Lehrveranstaltungen im Bereich der numerischen Simulation wird es augenblicklich schon eingesetzt.

### Das Rechnerpraktikum

Für viele Ingenieurstudiengängen ist es wichtig, in der numerischen Simulation eine solide Ausbildung zu vermitteln. Die Absolventen sollten in der Lage sein, auch selbst numerische Algorithmen zu programmieren. Die Programmieraufgaben in einer Grundvorlesung in Numerik lassen sich in Maple-Worksheets umsetzen. Damit lernt der Studierende die Entwicklung einer Programmstruktur und die Umsetzung in ein Programm. Für komplexere Probleme stehen später kommerzielle Softwaretools zur Verfügung, wobei Teile dann aber oft geändert oder hinzugefügt werden müssen. Insofern ist ein Ziel in unserer Ausbildung, auch dort schon die Kenntnisse in der Programmierung in einer höheren Programmiersprache zu fördern.

Die Vorlesung „Numerische Methoden“ für die Luft- und Raumfahrttechnik in Stuttgart wird jeweils im vierten Fachsemester gelesen. Die Studierenden haben also bereits Fachvorlesungen hinter sich mit Problemen, die samt der Lösungen bekannt sind. In Stuttgart etwa lösen die angehenden Luft- und Raumfahrttechniker im zweiten Semester eine Konstruktionsaufgabe aus gegebenen Stützwerten. Teile daraus ergeben eine

schöne Beispielaufgabe für Interpolationsverfahren. Die Dozentinnen und Dozenten der ingenieurwissenschaftlichen Fächer hatten hier eine ganze Reihe von Vorschlägen. Für die Betreuung des Rechnerpraktikums wurden interessierte Studenten aus den höheren Semester eingesetzt. Auch diese waren gerne bereit, über die Betreuung der Tutorien hinaus, auch bei der Gestaltung der Aufgaben mitzuwirken.

### Das Webforum

Bei den Programmieraufgaben hatten wir zunächst Bedenken, ob die Rechnerräume überhaupt ausreichen. Es hat sich jedoch gezeigt, dass bei Verbreitung von PC's dies kein Problem darstellt. Viele Studierende bearbeiten die Aufgaben am heimischen Rechner. Um dieses Arbeiten zu unterstützen, wurde von uns ein Webforum installiert. In einem solchen Forum gibt es innerhalb vorbereiteter Bereiche die Möglichkeit, Fragen zu stellen bzw. Fragen der Kommilitonen zu beantworten. Es wurde regelmäßig von Tutoren und Dozenten gelesen. Die Antworten sind auch für jeden sichtbar, sodass zum einen immer klar ist, ob eine Frage schon beantwortet wurde, zum anderen aber auch die Möglichkeit besteht, Antworten zu ergänzen oder zu korrigieren. Mit den Antworten erreicht man damit nicht nur den Fragesteller allein sondern alle Teilnehmer. Das Forum ist jederzeit öffentlich zugänglich, natürlich bis auf die Ausfallzeiten des Webservers.

### Die Hörsaalübung

Im Rahmen eines solchen Konzepts muss die Hörsaalübung die praktischen Elemente der Lehrveranstaltung unterstützen. In unserem Fall ergaben sich die folgenden Punkte:

- Da das Skriptum mit Maple-Worksheets arbeitet, die Studierenden im allgemeinen wenig Erfahrungen mit Computeralgebra mitbringen, muss die Übung in dieses System einführen. Dies geschieht zunächst am Anfang des Semesters. Während der Vorlesungszeit werden dann Ergänzungen angeboten, die das Arbeiten mit den Worksheets zum aktuellen Thema erleichtern.
- Hilfen zum Arbeiten mit der Programmiersprache, den Programmierumgebungen, Visualisierungswerkzeugen etc. werden in den Übungen angeboten. Auch hier liegt der Schwerpunkt am Anfang des Kurses.
- Zur Darstellung der Programme ist der Text in der Programmiersprache selbst nicht gerade die beste Wahl. Daher werden immer auch Struktogramme nach DIN66261 verlangt. Auch dieses Werkzeug wird durch die Übung bereitgestellt.
- Auf dem Webforum tauchen Fragen auftauchen, die für die Behandlung in der Übungsstunde günstig sind. Diese Fragen betreffen neben der Theorie und den Verfahren selbst oft auch die oben erwähnten eingesetzten Werkzeuge.

### Die schriftliche Prüfung

Es ist natürlich schwierig, die praktischen Fähigkeiten der Studierenden im Bereich der numerischen Simulation in einer schriftlichen Prüfung abzuprüfen. Es wird hier so vorgegangen, dass das erfolgreiche Bearbeiten der Programmieraufgaben und das zweimalige Vorstellen der Ergebnisse in den Tutorien eine Prüfungsvorleistung darstellt. Die schriftliche Prüfung besteht dann aus

- einem Fragenteil mit Verständnisfragen,

- und einem Aufgabenteil.

Der Aufgabenteil besteht zum überwiegenden Teil aus Programmieraufgaben, bei denen Struktogramme entworfen werden müssen. Der Schritt zum Rechenprogramm wird nicht in der Prüfung behandelt. Gegenüber der üblichen theoretischen Aufgabe bedeutet dies aber oft einen erhöhten Korrekturaufwand. Die bisherigen Erfahrungen zeigen, dass die Studierenden bewusst darauf hin arbeiten, selbst mit numerischen Algorithmen umgehen zu können.

### Technische Probleme

Bei soviel Einsatz von Computertechnik – interaktives Skriptum, Programmierumgebungen, Webforum,- besteht immer die Befürchtung, dass es auch technische Probleme gibt. Diese sollen natürlich nicht verschwiegen werden.

#### Unterschiedliche Mapleaufrufe

Es ist eine seltsame Eigenheit des Softwareherstellers Waterloo, dass er mit jeder neuen Version von Maple für Windows den Programmaufruf ändert. Mal heißt er `maplew`, mal `wmaple` oder er enthält noch die Versionsnummer. Unter Unix/Linux wiederum heißt er prinzipiell `xmaple`. Dies erleichtert es nicht gerade, ein Makro zu stricken, das auf allen Plattformen gilt und die richtige Maple-Version aufruft. Daher wurde ein Windowsprogramm namens `maplestarter.exe` geschrieben, das in der Registry nachsucht, welches denn die aktuellste Maple-Version ist, und diese aufruft. Für Unix/Linux gibt es ein kleines Shellscript namens `maplestarter`, das einfach `xmaple` aufruft.

#### Der Umgang des Acrobat Reader mit Dateipfaden

Der eigentlich von Adobe vorgesehene Weg, Dateien für externe Anwendungen zu verlinken ist der, einen Link direkt auf die Datei zu setzen. Auf dem System ist dann festgelegt, welche Anwendung dafür zuständig ist. Diese wird dann mit der Datei gestartet. So steht es jedenfalls in den Hilfeseiten zum Acrobat Reader. Das Problem ist nur, dass es auf den wenigsten Kombinationen Betriebssystem/Acrobat-Version funktioniert. Was der Acrobat Reader stattdessen tut, ist einen Webbrowser dazwischen zu schalten. Der kann die Dateieindung dann zuordnen und die Anwendung starten. Leider funktioniert hierbei die Übergabe der Dateipfade nicht korrekt. Lediglich mit absoluten Pfaden, lässt sich diese Konstruktion zum Laufen bringen. Nun kann man aber noch nicht einmal bei der Festlegung auf ein Betriebssystem sicherstellen, dass überall die Laufwerksnamen passen. Diese Version der technischen Umsetzung fällt also weg.

Man kann aber auch Anwendungen direkt von PDF aus verlinken. Dazu wird in pdfLaTeX mit Hilfe des Befehls `\href{run:programmaufruf}` ein Link erstellt. Nun würde man gerne Maple mit dem entsprechenden Worksheet aufrufen. Hier gibt es aber eine neue Schwierigkeit: Man kann nur bei einigen Versionen des Acrobat Readers für Windows überhaupt Parameter mit angeben. Diese müssen dann mit # statt einem Leerzeichen vom Programmnamen getrennt sein. Unter Unix/Linux funktioniert dieses grundsätzlich nicht. Dort gibt es aber immerhin das Programm `Xpdf`, das man als Ersatz nehmen könnte. Doch dieses akzeptiert die seltsame Syntax mit # nicht. Es verlangt den ganz normalen Aufruf, wie man ihn auf der Kommandozeile verwenden würde. Damit scheidet auch diese Möglichkeit aus.

Was letztlich gemacht wurde ist, den Aufruf in einer Batchdatei zu kapseln. Diese enthält also den Aufruf von Maple mit dem entsprechenden Worksheet. Dabei ist es dann sogar mög-

lich, eigene Maple-Pakete mitzuladen oder auch direkt auf Hilfedateien zu springen. Beide Möglichkeiten wurden beim ReVoKoS-Skript genutzt. Indem nur der Aufruf in der Datei steht, kann sie von Unix/Linux auch wie ein Shellsript ausgeführt werden und ist somit systemunabhängig. Dieses Vorgehen funktioniert mindestens für die Acrobat-Versionen 3,4 und 5. Der gerade erschienene Acrobat Reader in der Version 6 scheint wieder neue Probleme zu bereiten, sodass das hier beschriebene Vorgehen möglicherweise nochmals geändert werden muss.

### **Die Verfügbarkeit des Webservers**

Das elektronische Skript kann von dem Webserver heruntergeladen werden. Auch die Übungsblätter werden auf die Webseite gelegt. Auch das Forum läuft natürlich über diesen Webserver. Die Abhängigkeit der Vorlesung von dem Webserver ist somit sehr groß. Es kommt durchaus vor, dass der Server eine Zeit lang komplett ausfiel. Wir hatten hierfür allerdings schon ein Notsystem eingerichtet. Es gab eine Liste der Mailadressen aller Beteiligten, so dass die Kommunikation nicht völlig zusammenbrach.