

---

Konrad-Zuse-Zentrum für Informationstechnik Berlin  
Heilbronner Str. 10, D-1000 Berlin 31



J. Lügger

W. Dalitz

**Verteilung mathematischer Software mittels elektronischer Netze  
– die elektronische Softwarebibliothek eLib –**

---

Herausgegeben vom  
Konrad-Zuse-Zentrum für Informationstechnik Berlin  
Heilbronner Str. 10  
1000 Berlin 31  
Verantwortlich: Dr. Klaus André  
Umschlagsatz und Druck: Rabe KG Buch-und Offsetdruck Berlin

ISSN 0933-789X

# Verteilung mathematischer Software mittels elektronischer Netze - die elektronische Softwarebibliothek eLib -

J. Lügger, W. Dalitz  
Konrad-Zuse-Zentrum für Informationstechnik Berlin - ZIB  
Heilbronner Str. 10, D 1000 Berlin 31

## Statt eines Abstracts

Library - Definitionen aus "Oxford Advanced Learners Dictionary of Current English":

1. room or building for a collection of books kept there for reading.
2. the public library: maintained by a town or city council, etc.
3. writing and reading room in a private house.
4. series of books issued by a publisher in uniform binding and connected in some way.

## 1 Aspekte der fortschreitenden Dezentralisierung

Die Bedeutung von Softwarebibliotheken für den Bereich des Wissenschaftlichen Rechnens ist allgemein anerkannt. Weniger offensichtlich ist, daß mit der fortschreitenden Verlagerung rechenintensiver Aufgaben von den großen Universalrechnern zu leistungsfähigen Arbeitsplatzsystemen und der zunehmenden Vernetzung strukturell auch hier ein Wandel eingetreten ist.

Der einzelne Wissenschaftler wird zunehmend von zentralen Rechenkapazitäten unabhängig und verlagert z.B. seine Modellierungsaufgaben auf Arbeitsplatzrechner seines Instituts. Dort kann er auf der Basis eines lokalen Netzes mit seinen Fachkollegen Daten und Programme austauschen und er tritt auf dem Wege über elektronische Netze mit externen Fachkollegen in Verbindung. Im Verlauf seiner beruflichen Laufbahn wird er sehr wahrscheinlich mit einer Reihe von unterschiedlichen Arbeitsplatzsystemen arbeiten müssen, sei es, daß sein Institut einen Systemwechsel oder er selbst einen Institutswechsel vornimmt.

Was solch ein Wissenschaftler - abgesehen von seinem Fachwissen - über einen längeren Zeitraum oder auch über einen Ortswechsel hinaus für sich erhalten kann, wird seine von ihm und seinen Kollegen entwickelte Software sein. Er wird also darauf achten, daß seine Software eine Reihe von dafür notwendigen technischen Voraussetzungen erfüllt, wie z.B. Portabilität, Wartbarkeit etc. sowie - und darauf kommt es uns hier besonders an - Unabhängigkeit von speziellen Schnittstellen.

Natürlich ist heute - im Wissenschaftsbereich - kaum eine Anwendung "stand-alone" programmiert. Im Gegenteil: es werden Basisalgorithmen des jeweiligen Computerherstellers oder Bausteine aus einer kommerziellen Softwarebibliothek eingesetzt. Bei einem Systemwechsel entsteht dann der Wunsch, dieselbe Softwarebasis möglichst identisch wieder verfügbar zu haben. Oft hat man verloren oder muß teuer bezahlen, wenn nun die Softwarequellen der benutzten Bausteine nicht zur Verfügung stehen. Wer selbst für eigene Zwecke oder für Nutzung durch Dritte Software entwickelt, kennt dieses Problem.

Kurz gesagt: Mit der zunehmenden Verbreitung von Arbeitsplatzsystemen steigt der Bedarf an Quellen allgemein verfügbarer wissenschaftlich-technischer Grundsoftware. Insbesondere besteht ein zunehmendes Interesse an Quellen qualitativ hochwertiger Basisalgorithmen und Forschungscode aus allen Bereichen des Wissenschaftlichen Rechnens.

Glücklicherweise steht diesem Bedarf heute ein breites Spektrum an öffentlich verfügbarer mathematischer Software gegenüber /Boisvert 85/, /Dongarra 87/. Bei den Entwicklern besteht ein starkes Interesse daran, ihre hochspezialisierten Verfahren unter realistischen Bedingungen zum Einsatz zu bringen, damit sich Entwicklung und Anwendung wechselseitig befruchten können.

Beide Seiten sind dabei auf die gemeinsame Bewertung der Verfahren angewiesen. Der Entwickler: indem er seine Verfahren möglichst weit verbreitet und durch den breit gestreuten Einsatz möglichst umfangreiche Erfahrungen gewinnt, die er z.B. in Robustheit und Sicherheit seiner Verfahren ummünzen kann. Der wissenschaftliche Anwender: indem er qualitative Verbesserungen der Grundbausteine in eine Leistungssteigerung seiner eigenen Entwicklungen umsetzen kann. Beide Gruppen entwickeln ihre Ideen im Grunde wechselseitig und kooperativ. Und zum Medium, in dem sie sich ausdrücken und ihre Erfahrungen formulieren, gehört nicht nur der wissenschaftlich-technische Bericht, sondern auch die entwickelte Software.

Für jede dieser Formen der Erfahrung ist ein geeignetes Forum sinnvoll. Wir schlagen vor, daß auch für Software aus den Bereichen des Wissenschaftlichen Rechnens entsprechende Plattformen eingerichtet werden sollen, die etwa in der Art von wissenschaftlichen Bibliotheken zu betreiben sind: Öffentlich zugänglich, natürlich kostenfrei zu nutzen, ausgestattet mit Algorithmen, Testprogrammen bzw. Testdaten sowie allgemeinen Software-sammlungen, getragen von wissenschaftlichen Institutionen oder deren Trägern.

Mit den Fortschritten in der internationalen Vernetzung ergeben sich besondere Chancen für die Realisierung solcher Softwarebibliotheken. Sie können auf dem Wege über elektronische Netze angeboten und von fast jedem Arbeitsplatzsystem aus genutzt werden. Besonders interessant ist, daß mit dem Abschluß der Normung des Dialogs und des elektronischen Nachrichtendienstes als Kommunikationsprotokolle im Bereich der ISO und der fortschreitenden Verwendung dieser Protokolle jetzt eine natürliche Zugriffsmöglichkeit auf solche Bibliotheken besteht. Die Auswahl einer Informationseinheit kann online erfolgen, während die Übermittlung der Information, z.B. eines Programms, eines Testsatzes, eines Handbuchs oder auch eines Index-Dokuments der Softwarebibliothek, off-line durchgeführt werden kann.

Dabei sind die Verhältnisse im Bereich des "Deutschen Forschungsnetzes - DFN" für die Einrichtung einer elektronischen Bibliothek besonders günstig. Hier kann mit dem "Deutschen Wissenschaftsnetz - WIN" ein X.25-orientiertes Netz pauschal und damit quasi kostenneutral genutzt werden. Auf dieser Basis steht auch der X.400-orientierte elektronische Nachrichtendienst des DFN mit seinen Gateways für elektronische Nachrichten in andere Netze, wie z.B. in das Internet oder in das BITNET oder in das uucp, ebenfalls kostenneutral zur Verfügung.

Im folgenden sollen die Erfahrungen beim Aufbau einer speziellen elektronischen Softwarebibliothek, der am ZIB entwickelten eLib, diskutiert werden. Die eLib war ursprünglich für die Weitergabe von am ZIB entwickelter Software für das Wissenschaftli-

che Rechnen gedacht. Über die eLib steht derzeit aber auch eine umfangreiche Sammlung mathematischer Algorithmen und Softwaresysteme aus dem amerikanischen Raum zur Verfügung.

Die eLib befindet sich noch in der Entwicklung - wesentliche Teile fehlen noch. Im Verlauf der Diskussion wird sich zeigen, daß der technische Apparat der kleinere Teil des Problems ist und daß, wenn man auf "Accessibility" Wert legt, noch nicht alle modernen Tools und Techniken selbstverständlich sind, die wir als Benutzer heutiger Arbeitsplatzrechner so angenehm und nützlich finden. Zuvor soll jedoch das Modell der Softwarebibliothek - wie wir es sehen - dargelegt werden.

## 2 Elemente der elektronischen Softwarebibliothek eLib

Wir beschreiben die elektronische Softwarebibliothek auf verschiedene Arten. Einerseits kann die eLib als ein eigenständiges Programm gesehen werden, das man über ein Netz anwählt, um Software aus ihrem Programm Magazin zu erhalten. Andererseits steht die eLib im Verbund mit zwei weiteren Softwarebibliotheken, der NetLib von J.J. Dongarra und E. Grosse und der Reduce-NetLib von A.C. Hearn, die am ZIB gemeinsam mit der eLib auf demselben "Server" eingesetzt werden. Hier ist die elektronische Bibliothek eher als Rechner mit Plattenspeicher zu sehen, der über ein Netz erreichbar ist. Denkt man etwas weiter über die Bausteine einer elektronischen Bibliothek nach, kommt man noch zu anderen Interpretationen. Man kann die Netzverbindungen und sogar einzelne Gateways als zur elektronischen Bibliothek gehörig betrachten. Wir werden die eLib im folgenden im engeren Sinne als Programm sehen und nur in manchen Fällen von dieser Perspektive abweichen.

An erster Stelle steht für uns die breite Erreichbarkeit der eLib. Der Zugang erfolgt im Normalfall im Dialog<sup>1</sup> über das Datex-P-Netz der internationalen Postverwaltungen (PAD, X.25, X.29) oder über das im Wissenschaftsbereich weit verbreitete Internet (telnet). Auch über den elektronischen Nachrichtendienst kann auf die eLib zugegriffen werden. Dabei werden die Protokolle der CCITT bzw. der ISO (X.400) und des Internet (TCP/IP) eingesetzt.

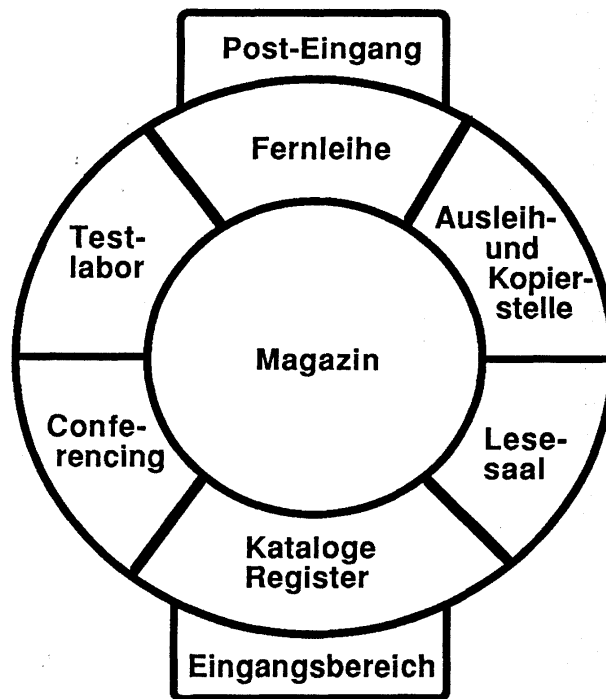
Die elektronische Bibliothek eLib besitzt, ganz wie eine Bibliothek im normalen Sprachgebrauch, einen "Eingangsbereich" mit allgemeinen Informationen und Hinweisen für Benutzer. Im Gegensatz zu einer normalen Bibliothek ist sie allerdings Tag und Nacht "geöffnet" und kann von vielen Orten her jederzeit "betreten" werden. Man kann sie anonym "besuchen", ohne daß man sich explizit identifizieren muß.

Gleich neben dem "Eingangsbereich" gibt es einen mathematischen Sachkatalog, der nach dem "Guide to Available Mathematical Software" /Boisvert 90/ organisiert ist. Dieser sogenannte GAMS-Index beinhaltet ein problemorientiertes Klassifikationsschema, das dem Benutzer umfangreicher und sehr heterogener Softwaresammlungen aus dem Bereich des Wissenschaftlichen Rechnens die effektive Identifizierung der für ihn geeigneten Software erlauben soll.

<sup>1</sup>Datex-P bzw. WIN: (Germany) 45050331033;

Internet: eLib.zib-berlin.de (130.73.108.11) login: elib

Die Verwendung eines solchen Schemas ist nicht ganz unproblematisch, erscheint hier jedoch angebracht. Im "Magazin" der eLib befinden sich zur Zeit über 1000 Programmquellen und Dokumente. Rechnet man die mit der eLib im Verbund betriebenen Bibliotheken NetLib /*Dongarra 87*/ und Reduce-NetLib mit hinzu, so kommen noch ca. 5000 weitere Software- und Informationsmodule hinzu. Ein Teil der über die NetLib verfügbaren Software wird gegenwärtig noch nach dem GAMS-Index klassifiziert und in die eLib integriert, allerdings nicht die gesamte Software der NetLib.



**Bild 1.** Modellvorstellung einer Softwarebibliothek

Neben dem GAMS-Index gibt es den Library-Index. Die eLib enthält wie die NetLib keine alleinstehenden Softwaremodule, sondern ganze Softwarebibliotheken. Sie ist also eher als eine Sammlung von Softwarebibliotheken zu sehen. Im Library-Index sind die Informationsmodule nach Bibliotheken geordnet "aufgestellt". Wir stimmen mit */Boisvert/* darin überein, daß ein Library-Index von der Mehrzahl der Anwender nicht als ideales Ordnungsschema für die Softwaresuche gesehen und gewünscht wird. Wir unterstützen diesen dennoch in der eLib, weil er aus der Perspektive der Softwareentwickler eine natürliche Unterteilung der gesamten Software darstellt und eine elegante Adressierungsart für die einzelnen Softwaremodule anbietet. Auch NetLib und Reduce-NetLib arbeiten mit einem Library-Index. Ein allgemeiner Library-Index bot sich als eine Integrationsstufe von eLib, NetLib und Reduce-NetLib an.

Ferner gibt es in der eLib einen Modul-Index, in dem alle Softwaremodule alphabetisch "aufgereiht" sind. Jeder Modul ist dabei mit einer Kurzbeschreibung versehen. Damit ist die Möglichkeit gegeben, im Modul-Index nach fachspezifischen Stichwörtern

zu suchen und auf diese Weise geeignete Software zu finden. Wir weisen jedoch darauf hin, daß es sich dabei nicht um einen standardisierten Satz von Schlüsselwörtern im Sinne eines Thesaurus handelt. Die Kurzbeschreibungen sind in der Regel direkt aus den Softwarequellen extrahiert - sie stammen von Softwareentwicklern mit verschiedenem Erfahrungshintergrund und folglich unterschiedlichem Sprachgebrauch.

Die Frage, ob es sinnvoll ist, für die Erschließung von Software zusätzlich zum GAMS-Index noch einen Thesaurus zu entwickeln, müssen wir hier offen lassen. Wir haben uns bei der eLib nicht dazu entschließen können. Potentielle Benutzer müssen die einem Thesaurus zu Grunde liegende Terminologie erst erlernen. Das ist für sie nicht einfach. Obendrein kann die Entwicklung eines Thesaurus nicht isoliert geschehen. Sie verlangt neben der Bestimmung auch die fortlaufende Abstimmung der gemeinsamen Terminologie sowohl mit Softwareentwicklern als auch mit Benutzern. Wir sehen darin eine schwierige und sehr aufwendige Aufgabe.

Die verschiedenen Indizes der eLib sind ebenso wie der "Eingangsbereich" in der Art eines hierarchischen Menübaumes organisiert. Wie man mit Menüs umgeht, ist heute allgemein bekannt. Man gelangt durch fortschreitende Selektion bzw. Auswahl zu weiteren Menüs bzw. zu Programmen und Dokumenten, die man in einem Fenster der eLib inspizieren kann ("Lesesaal"). Die ausgewählten Informationseinheiten werden dabei gewissermaßen automatisch vom "Magazin" in das Fenster gebracht. Dort kann man in ihnen blättern oder Stichworte suchen oder Zeileninhalte auf reguläre Ausdrücke testen etc.

Durch diese Inspektion von Programmquellen kann ein Benutzer in den meisten Fällen unmittelbar eine Entscheidung treffen, ob es sich für ihn lohnt, ein Programm oder Dokument zu seinem lokalen Arbeitsplatzrechner zu transferieren oder nicht. Ein Programm kann er dabei mit allen seinen Unterprogrammen zusammenpacken. Programmquellen, Dokumentation oder auch eLib-Menüs kann er per elektronischer Post an seine lokale Mailbox<sup>2</sup> verschicken ("Leih- und Kopierstelle"). Falls er dabei auf - etwa durch Lizenzbedingungen - geschützte Software zugreift, muß er sich zuvor entsprechend ausgewiesen haben.

An dieser Stelle sollte eine der wichtigsten Entwurfsentscheidungen plausibel sein: Die eLib ist als Kommando-Shell realisiert - sie wird über eine Kommandozeile gesteuert. Die eLib steht in ihrer technischen Anlage einem Programmiersystem näher als etwa einer Datenbank. Sie richtet ihr Dienstleistungsangebot ja auch an den wissenschaftlich-technischen Programmierer und verwendet deshalb eine der Programmierung nahestehende Benutzeroberfläche. Die Objekte der Manipulationen sind die Programme und Dokumente selbst, ganz wie bei Benutzern von normalen Bibliotheken, die die Bücher selbst verlangen.

Diese Vorgehensweise unterscheidet sich von der bei wissenschaftlich-technischen Informationsdiensten üblichen. Dort werden auf elektronischem Wege nur Hinweise oder allenfalls Abstrakts auf Bücher und andere Veröffentlichungen angeboten. Die eigentliche "Beschaffung" arbeitet bei solchen Systemen gewissermaßen nur indirekt, d. h. auf einer Menge von Formularen, die in der Regel in einer Datenbank gespeichert sind. Für die Speicherung von Programmen mit ihren Abhängigkeiten bzw. Volltexten sind Datenbanken weniger geeignet. Die im Rahmen des "Information Retrieval" entwickelten Modelle

<sup>2</sup>Dazu muß sich ein Benutzer registriert haben und die eLib muss seine Adresse verifizieren.

für den Benutzerdialog sind unserer Ansicht nach für eine elektronische Softwarebibliothek weniger hilfreich.

Die geeignete Basis für die Speicherung von Programmquellen ist heute noch immer das hierarchische Dateisystem. In der eLib wird es im "Magazin" benutzt und durch den Library-Index, der die einzelnen Knoten benennt und in knapper Form beschreibt, direkt erschlossen. Der GAMS- und der Modul-Index weisen ebenfalls mehr oder weniger direkt in dieses Programm-Magazin.

Die Benutzeroberfläche der eLib ist im Prinzip fensterorientiert angelegt, wenn auch das eLib-Fenster aus Gründen, die in der Kommunikationstechnik liegen, dem zeilenorientierten Dialog heute noch näher steht als dem im Rahmen einer graphischen Oberfläche geführten Dialog (weiteres dazu siehe unten).

Für die Benutzerführung wird im wesentlichen ein - vom Betreiber der eLib programmierbares - Menüsystem eingesetzt, in das neben den Indizes der eLib auch die wichtigsten Kommandos eingebaut sind. Wichtige Grundfunktionen sind schon jetzt so ausgelegt, daß sie später relativ leicht in eine graphische Oberfläche übertragen werden können.

Die eLib ist für uns in erster Linie ein menügesteuerter "Browser" mit Kommandozeile. Es ist aber auch möglich, sie off-line über ein Interface für elektronische Post anzusprechen. Für diese Form der Anfrage haben wir zusätzlich eine spezielle Kommandoform von der NetLib konzeptionell übernommen und für die eLib eingerichtet. Die eLib kann somit auch von Anwendern eingesetzt werden, die nur über eine elektronische Mailbox verfügen, aber keinen Zugriff auf den Dialogdienst haben.

Im ursprünglichen Konzept der eLib war vorgesehen, daß man über eine "Fernleihe" auf Dokumente anderer Bibliotheken zugreifen kann. Zunächst wurde die Fernleihe bezüglich der NetLib realisiert. Die Fernleihe war hier zunächst nur für off-line-orientierten Anfragen sinnvoll, da die originale NetLib nur off-line bedient werden kann. Mit der späteren Implementierung eines Dialogverbundes zwischen eLib, NetLib und Reduce-NetLib mußten möglichst alle Informationseinheiten der NetLib in ihrer aktuellen Form online auf dem "Server" der eLib vorgehalten werden.

Bei der nächsten Version der eLib ist daran gedacht, auch den Filetransfer zur Informationsübertragung anzubieten. Der Vorteil besteht darin, daß man bei der Übermittlung mittels Filetransfer nicht erst lange auf das Eintreffen einer elektronischen Nachricht warten muß. Der eigentliche Vorgang der Informationsübertragung kann dadurch besser kontrolliert werden - auch umfangreiche Dateien könnten praktisch sofort übermittelt und verfügbar gemacht werden. Nachteilig ist, daß z.B. der international durch die ISO genormte FTAM Dateitransfer zumindest heute noch nicht allgemein verbreitet ist und mit dem in der Internet-Welt populären "ftp" würde nur eine spezielle Benutzergemeinschaft unterstützt.

Eines der Hauptprobleme für den Benutzer einer elektronischen Bibliothek liegt in der qualitativen Bewertung der vorliegenden Software. Deshalb war schon im ersten Entwurf der eLib vorgesehen, einen Konferenzdienst in der Art eines problemorientiert strukturierten Bulletin-Boards - etwa in der Art des CoSy-Systems */Guelph/* - einzurichten. Eingeschriebene Anwender und Autoren sollten ihre Kommentare zu den Algorithmen - und dazu gehören auch Rechenergebnisse und Bewertungen - im Rahmen von öffentlich zugänglichen elektronischen Konferenzen hinterlegen können. Von diesem Konferenzdienst wurde vorerst nur eine erste Stufe entwickelt.



Ferner ist an ein Prüflabor für spezielle, ausgewählte Verfahren gedacht. Bestimmte mathematische Algorithmen, die nicht mit ihrer ganzen Quelle zur Verfügung stehen (z.B. Lizenzprodukte) sollten dennoch allgemein getestet und geprüft werden können. Dazu müßte ein geeigneter Formalismus für die Spezifikation von Testaufgaben entwickelt werden, der z.B. über den off-line Dienst der eLib zum Einsatz gebracht werden könnte. Falls öffentlich zugängliche Software getestet werden soll, wäre auch zu überlegen, ob es nicht besser ist, die generierten Testprogramme zu den Daten zu schicken, anstatt die Daten zu den Testprogrammen. In diesem Falle könnte die für ein Testlabor notwendige Rechenkapazität auf die eigentlichen Nutzer dieser Einrichtung verteilt werden.

Nicht zuletzt - eine elektronische Softwarebibliothek kann nur so nützlich sein, wie die Beiträge, die zu ihr geleistet werden. An dieser Stelle ist die wissenschaftliche Gemeinschaft gefragt und zur Mitarbeit aufgerufen. Das Konrad-Zuse-Zentrum für Informationstechnik Berlin kooperiert hier als Betreiber der eLib mit der gemeinsamen Fachgruppe "Numerische Software" von GI, GAMM und DMV bei der Sammlung von Software insbesondere im deutschen Raum. Die Bereitstellung von Software aus den Staaten verlangt die Kooperation mit anderen Netzbibliotheken und Softwareherstellern. Beides wird durch den Einsatz elektronischer Netze und Dienste entscheidend unterstützt.

### 3 Der on-line Dienst der eLib

Zum heutigen Zeitpunkt ist allein der zeilenorientierte Dialog (PAD, X.25, X.29) international standardisiert und weithin verfügbar. Unter den fensterorientierten Dialogen genießt X11 zwar zunehmend an Popularität, ist jedoch in gewisser Weise hersteller-spezifisch an die Internet-Protokollwelt gebunden. Wir haben uns, um eine möglichst breite Verfügbarkeit zu erreichen, zu folgendem Kompromiß entschlossen, der von minimalen technischen Voraussetzungen ausgeht.

Der eLib-Dialog ist grundsätzlich zeilenorientiert aufgebaut. Aus den einzelnen Zeilen wird ein Fenster aufgebaut, in dem ein Ausschnitt der von der eLib zu zeigenden Information zu sehen ist, also Teile von Programmen, allgemeinen Dokumenten oder auch Menüs. Welchen Teil der Information die eLib zeigt, wird außen am Fenster durch entsprechende Maßzahlen angezeigt. Nach jedem eLib-Kommando wird im Prinzip immer ein ganzes Fenster mit Information ausgegeben. Wo immer möglich wird allerdings eine Reduktion der zu übertragenden Zeichenmenge durch eine besondere Anpassung<sup>3</sup> an spezielle Terminaltypen angeboten.

Wir bewerten mit der Realisierung dieses Konzeptes breite Erreichbarkeit heute noch höher als einen möglichen Zuwachs an Benutzerkomfort, wie er sich etwa bei Verwendung einer X11-orientierten Benutzeroberfläche ergeben könnte. Später, wenn sich ein leistungsfähigeres, fensterorientiertes Dialogprotokoll allgemein etabliert hat, ist ein Migrationsschritt in diese Richtung nicht ausgeschlossen.

Die eLib war von Beginn an menüorientiert angelegt. Sie sollte selbsterklärend und einfach zu bedienen sein. Gerade im Umgang mit hierarchischen Dateisystemen zeigen jedoch rein auf Menüs und Fenster ausgelegte Systeme ihre Schwächen. Wie könnte man

<sup>3</sup>Technisch setzen wir dazu die "termcap/terminfo"-Schnittstelle auf UNIX-Basis ein.

z.B. den Umgang mit regulären Ausdrücken vollständig in eine so geartete Oberfläche einbetten? Die eLib umfaßt auch Kommandos, die sich nicht in eine rein graphische Oberfläche übertragen lassen. Sie wird auch später - wie schon heute - über eine Kommandozeile verfügen.

Die Menüs der eLib sind nicht etwa im eLib-Prozessor fest einprogrammiert, sondern vom Betreiber der eLib frei programmierbar. Das gesamte Konzept ist darauf hin ausgerichtet, daß Menüs sehr viele Elemente enthalten und textuell auch sehr umfangreich sein können. Die Selektion einzelner Menüelemente ist deshalb sowohl numerisch (über Ziffernfolgen) als auch textuell (über alphanumerische Zeichenketten) möglich. Zusätzlich sind spezielle Kommandos zur Selektion vorgesehen, mit denen man einzelne Menüelemente auch dann anwählen kann, wenn man sich "außerhalb" des betreffenden Menüs befindet.

Im Hauptmenü der eLib ist z.B. der GAMS-Index zu finden. Man kann ihn mit der numerischen Selektion 3 oder mit der textuellen Auswahl 'gams index ' oder mit dem Spezialkommando **gams** anwählen. Wir sparen uns die Abbildung des daraus folgenden Menüs mit der obersten Ebene des GAMS-Index und verweisen hierfür auf die Originalarbeit /Boisvert 90/ bzw. auf die eLib selbst. Mit dem Kommando **gams i** steigt man eine Stufe tiefer in die GAMS-Menüs ein und kommt etwa zu folgendem Fensterausschnitt:

[1]	Differential and integral equations (General)	- i0
[2]	Ordinary differential equations (ODE's)	- i1
[3]	Initial value problems	- i1a
[4]	General, nonstiff or mildly stiff	- i1a1
5	One-step methods (e.g., Runge-Kutta)	- gams i1a1a
6	System of first order diff. equations	- gams i1a1a1
7	System of second order diff. equations	- gams i1a1a2
8	Multistep methods	- gams i1a1b
[9]	Extrapolation methods	- i1a1c
10	System of first order diff. equations	- gams i1a1c1
11	System of second order diff. equations	- gams i1a1c2
12	Stiff and mixed algebraic- diff. equations	- gams i1a2
13	System of first order diff. equations	- gams i1a2a
14	Linearly implicit ODE's and diff. algebraic systems of first order up to index 1	- gams i1a2b
15	Special applications	- gams i1a2c
[16]	Multipoint boundary value problems	- i1b

Wer den ursprünglichen GAMS-Index kennt, wird sofort bemerken, daß der Index an einigen, für Zwecke der eLib notwendigen Stellen erweitert, aber nicht strukturell geändert ist. Schon der nächste Schritt - hier etwa durch die Selektion des Menüelements zu **gams i1a2b** - führt zu einem Menü mit Programmen und ihren Kurzbeschreibungen, die unter dem Index i1a2b "Linearly implicit ODE's and diff. algebraic ..." klassifiziert sind.

Aus dieser Strukturierung des Zugriffs auf Software folgen einige subtile Verbesserungen gegenüber dem "GAMS Interactive Consultant", in dem der GAMS-Index erstmalig implementiert wurde /Boisvert 85/. Beim "GAMS Interactive Consultant" benötigt man fünf Auswahlsschritte, um von der obersten Ebene zur Ebene i1a2b zu kommen. Dort muß dann zunächst eine Suche gestartet werden, durch die ermittelt wird, ob dieser Index überhaupt mit Software belegt ist. Bei der eLib genügen zwei Schritte von der obersten Ebene herab und schon nach dem ersten erhält man die Aussage, daß der Index i1a2b auf

Software verweist. Anderenfalls wäre die entsprechende Menünummer eingeklammert und könnte nicht selektiert werden, wie dieses z.B. bei den oben gezeigten Menüelementen 1 bis 4 des Menüs zum GAMS-Index "i" gegeben ist. Diese Reduktion der notwendigen Auswahlsschritte ist insbesondere auch im Zusammenhang mit e-Mail Anfragen an die eLib wichtig (s.u.). Als weiteren Vorteil sehen wir an, daß die unteren Ebenen des GAMS-Index von der eLib im vollständigen Kontext aller GAMS-Indizes mit gleichem Hauptindex gezeigt werden.

Im dritten Interaktionsschritt kann also direkt eine Softwarequelle ausgewählt und im eLib-Fenster betrachtet werden. In der Regel findet sich dabei auch entsprechende Anwenderdokumentation oder eine Parameterbeschreibung als Kommentar in der Quelle. Falls man sich dazu entscheidet, eine selektierte Quelle zur Mailbox auf seinem lokalen System zu transferieren, genügt das Kommando `send` dafür. Natürlich muß der eLib die Adresse der Mailbox zumindest einmal im Dialog bekannt gemacht worden sein. Das geschieht indirekt durch Angabe eines Benutzernamens und des zugehörigen Paßwortes.

Die Verwendung des Paßwortes verdient besondere Beachtung. Man kann die eLib nämlich anonym benutzen. Man braucht sich nicht zu Beginn des Dialogs mit der eLib zu identifizieren. Das ist erst dann nötig, wenn man spezielle Rechte ausnutzen will.

Beim elektronischen Postversand steht allerdings nicht so sehr das Benutzungsrecht im Vordergrund, sondern vielmehr ein betrieblicher Grund. Man kann von Benutzern nicht verlangen, daß sie sich mit den am Einsatzort der eLib lokal gebräuchlichen Adressierungsschemata für elektronische Post auskennen. Am ZIB setzt die eLib z.B. Unix-Mail in einem lokalen Netz unter Verwendung eines X.400-Mail-Servers ein. Daraus resultiert eine nicht allgemein übliche Adressform. Die Adreßangabe kann insbesondere dann Schwierigkeiten bereiten, wenn die adressierte Mailbox nur auf dem Wege über einen oder mehrere Gateways in andere Netze zu erreichen ist.

Bei der eLib muß man deshalb seine Mailbox registrieren lassen, etwa indem man einmal sein Paßwort zusammen mit einem "register"-Befehl von seiner aktuellen Mailbox aus an die eLib schickt<sup>4</sup>. Diese ermittelt aus solch einer Nachricht die Reply-Adresse, notiert sie sich im entsprechenden Benutzerbereich und schickt eine Quittung zurück. Sobald man die Antwort der eLib erhalten hat, weiß man, daß die verwendete Adresse funktioniert. Im anderen Falle erhält die Systemadministration der eLib die nicht zugestellte elektronische Post zurück und kann entsprechende Maßnahmen treffen.

Der Zugriff auf geschützte Quellen ist ebenfalls technisch an das Paßwort gebunden. Die eLib ist dafür eingerichtet, Software aus bestimmten Bibliotheken u. U. nicht vollständig, sondern nur eingeschränkt zur Verfügung zu stellen. Dann zeigt sie bei Selektion ggf. nur die Dokumentation einer solchen Quelle im Fenster und verschickt auch nur diese im Falle eines "send"-Kommandos. Falls man sich aber mit einem Paßwort identifiziert hat, an das die volle Zugriffsberechtigung für diese Bibliothek gebunden ist, stellt die eLib auch die ganze Quelle zur Verfügung.

Der Eintrag von Paßwörtern kann bis zu einem gewissen Grade durch Selbstanmeldung und ohne Eingriffe von Seiten der eLib-Systemverwaltung eingerichtet werden. Die Vergabe von Zugriffsrechten für geschützte Software macht aber immer eine besondere Prüfung und den manuellen Eingriff erforderlich.

<sup>4</sup>Bitte fordern Sie ggf. ein Paßwort bei der Systemadministration an:  
eLib-Adm@eLib.zib-berlin.dbp.de

#### 4 Der off-line Dienst der eLib

Der on-line Zugriff auf die eLib ist effektiv und einfach, jedoch auch recht teuer. Nicht jeder potentielle Benutzer hat die Möglichkeit des Zugriffs mittels Datex-P oder Internet. Im Wissenschaftsbereich verfügt heute jedoch eine große Anzahl von Anwendern über eine elektronische Mailbox. Diesen bietet die eLib mit ihrem off-line Dienst eine preisgünstigere, wenn auch etwas weniger komfortable Zugriffsvariante an. Man schickt eine Anfrage, die wie eine Folge von Dialogbefehlen aufgebaut ist, etwa wie folgt an die eLib<sup>5</sup>. Dabei ist zu beachten, daß die Subject-Zeile signifikant ist und von der eLib ausgewertet wird.

```
To: eLib@sc.zib-berlin.dbp.de
Subject: * testmail: the star designates a comment
        send index
```

Die bereits oben erwähnte Gliederung des GAMS-Index in drei, statt in einer variable Anzahl von Menüebenen, hat im off-line Betrieb einen besonderen Effekt. Es genügt eine einzige Anfrage, um herauszufinden, ob zu einem GAMS-Index Software vorliegt oder nicht. Eine Anfrage der Form

```
Subject: * schicke Liste der Routinen zum Index d2a1
        send index from gams d2a1
```

erfüllt diesen Zweck. Die eLib schickt die Liste aller unter "d2a1" klassifizierten Module an den Absender zurück. Findet die eLib unter einem GAMS-Index keine Software, so schickt sie das Submenü zum Haupt-Kapitel zurück. An den Menünummern kann man erkennen, ob etwa zu einem anderen Index desselben Hauptindex Software vorliegt oder nicht. In unserem Falle ist aber z.B. die Routine DGECO aus der LINPACK-Bibliothek unter "d2a1" klassifiziert. Die Anforderung

```
Subject: * schicke Routine DGECO aus dem Index d2a1
        send dgeco from gams d2a1
```

veranlaßt nun die eLib, die Softwarequelle von DGECO mit allen Unterrouinen, von denen diese abhängt, zusammenzupacken und zurückzuschicken. Das "send"-Kommando der Form

```
send <routine> from <library-or-list>
```

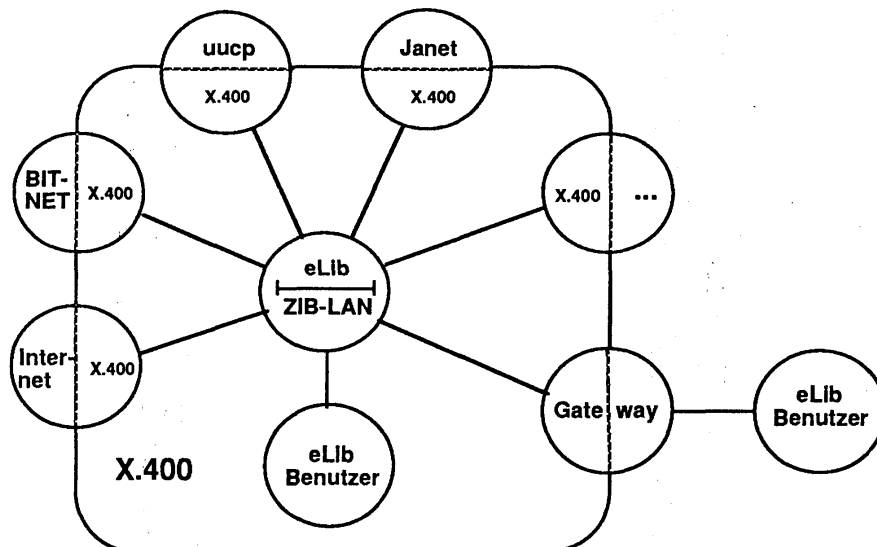
wurde von der NetLib entlehnt und erweitert. Mit dieser Kommandoform wurde die Möglichkeit eröffnet, die eLib in einem homogenen Verbund mit NetLib und Reduce-NetLib einzusetzen (weiteres s. u.).

Die eLib verwendet im off-line Betrieb dieselbe Kommandosprache wie im Dialog. Natürlich kann man mehrere eLib-Anfragen in einem einzigen elektronischen Brief zusammenfassen. Bei dieser Vorgehensweise muß man sich allerdings ein klares Bild darüber machen, wie die eLib reagiert. Im Falle eines Fehlers schickt die eLib für Diagnosezwecke die komplette Folge von Eingabezeilen und resultierenden eLib-Fenstern zurück - ganz so, als ob Sie einen Dialog geführt hätten. Solch ein Ablaufprotokoll kann man auch explizit von der eLib anfordern.

<sup>5</sup>X.400: S=eLib; OU=sc; P=zib-berlin; A=dbp; C=de;  
Internet: eLib@elib.zib-berlin.de

Bezüglich elektronischer Nachrichten wird eine vom DFN vertretene Linie verfolgt. Die eLib verwendet für elektronische Nachrichten, die das lokale Netz des ZIB überschreiten, im wesentlichen X.400, ein inzwischen weit verbreitetes Protokoll, das von der ISO standardisiert ist und den Verkehr zwischen unterschiedlichen e-Mail Systemen regelt. Mit dieser Vorgehensweise können insbesondere vom DFN betriebenen Gateways in Wissenschaftsnetze, die andere Protokolle verwenden, genutzt werden.

Die eLib wird damit aus praktisch allen wichtigen Wissenschaftsnetzen per elektronischer Post erreichbar. Diese Verfahrensweise ist praktikabel, solange die Kostenfrage wie bisher im WIN, dem X.25-Netz des DFN, geregelt ist. Innerhalb des WIN sind die Transferkosten mit einer jährlichen Pauschalen abgedeckt. Bezüglich der eLib sind die Kosten mit dem Beitrag des ZIB abgedeckt. Kosten für elektronische Post, die die Grenzen des WIN überschreitet, übernimmt zur Zeit das DFN. Weder für den Nutzer noch für das ZIB als Betreiber der eLib entstehen bei einer Anfrage und der daraus resultierenden Antwort, die einen zum Teil erheblichen Umfang annehmen kann, zusätzliche Kosten.



**Bild 2.** Kommunikationstechnik bei Anfragen mittels elektronischer Nachrichten

Wegen der bekannten Bedeutung des Internet für den Wissenschaftsbereich ist das ZIB seit einiger Zeit parallel zur in Bild 2 gezeigten Verbindung direkt an das Internet angeschlossen. Deshalb kann die eLib jetzt vom Internet aus besonders effektiv genutzt werden. Da das ZIB lokal ebenfalls Internet-Protokolle einsetzt, sind Schnittstellenumsetzer hier nicht notwendig. Wegen der besonderen Betriebsart des Internet können Anfragen an die eLib effizient und ebenfalls kostenneutral abgewickelt werden.

Wir möchten darauf hinweisen, daß diese Art des Netzeinsatzes bzw. der Kostenverteilung nicht selbstverständlich ist, aber eine essentielle Voraussetzung für den Betrieb von elektronischen Softwarebibliotheken darstellt. In der Tat wissen wir nicht, ob das ZIB die Dienste der eLib auf Dauer kostenfrei anbieten kann.

## 5 Schritte für den Verbund von eLib, NetLib und Reduce-NetLib

Für die Integration von Software in eine elektronische Bibliothek muß ein verhältnismäßig hoher Aufwand getrieben werden. Zu Beginn steht dabei der Kontakt mit den Entwicklern, die ihre Software in der Regel für eine Veröffentlichung in der eLib erst aufbereiten müssen. Am Schluß sind die einzelnen Module nicht nur technisch in eine solche Form zu bringen, daß sie von der eLib aus dem Magazin der Programme abgerufen werden können, sondern sie müssen auch klassifiziert und in die jeweiligen Indizes und Menüs integriert werden. Allein der dafür notwendige Aufwand zwingt zur Konzentration auf einen relativ beschränkten geographischen und fachlichen Raum.

Von Anfang an war deshalb in der eLib eine Zusammenarbeit mit anderen Netzbibliotheken vorgesehen. Für die eLib sollten Forschungscode insbesondere des wissenschaftlichen Rechnens vor allem im deutschsprachigen Raum gesammelt werden. Software aus dem amerikanischen Raum sollte auf dem Wege über die NetLib erschlossen werden. In einem ersten technischen Ansatz wurde dazu eine "Fernleihe" von der eLib am ZIB in Berlin zur NetLib bei AT&T in Murray Hill, New Jersey eingerichtet.

Die Idee bestand darin, in Berlin zunächst einen NetLib-Knoten zu installieren, in dem anfangs nicht viel mehr als die verschiedenen Index-Dateien vorgehalten werden sollten. Bei einer Anforderung an diesen Knoten, eine bestimmte Softwarequelle auszuliefern, sollte diese Anfrage zunächst mit dem in Berlin vorhandenen Softwarebestand befriedigt werden. Falls geforderte Softwaremodule nicht vorrätig waren, sollten diese von der originalen NetLib aus den Staaten geholt werden. Längere Wartezeiten entstehen dabei nur beim jeweils ersten Zugriff auf eine Quelle.

Wir gingen davon aus, daß sich am Knoten des ZIB in relativ kurzer Zeit ein Bestand mit den wichtigsten Softwaremodulen aus der NetLib etablieren würde. Da die Quelle des NetLib-Prozessors öffentlich zur Verfügung stand, konnte ein solcher "NetLib-Cache" in relativ kurzer Zeit eingerichtet werden. Er arbeitet heute noch, auch wenn die NetLib am ZIB inzwischen in einer wesentlich vollständigeren Form angeboten wird. Der "Cache" kann genau wie die originale NetLib per elektronischer Post<sup>6</sup> angeschrieben bzw. abgefragt werden.

Im nächsten Schritt wurde in Kooperation mit E. Grosse der NetLib-Knoten am ZIB inhaltlich möglichst weitgehend mit dem Original bei AT&T synchronisiert. Ursprünglich sollte dieses auf der Basis einer Startversion mit allen im Oktober 89 in der NetLib vorhandenen Quellen - insgesamt ca. 150 MByte an Software und Dokumenten - geschehen. Updates sollten vom ZIB per elektronischer Post aus den Staaten angefordert und explizit in den Berliner Knoten integriert werden. Dieser Versuch schlug fehl - im Grunde wegen des Umfangs der zu transferierenden Quellen und wegen Koordinierungsschwierigkeiten bei der Integration und Synchronisierung.

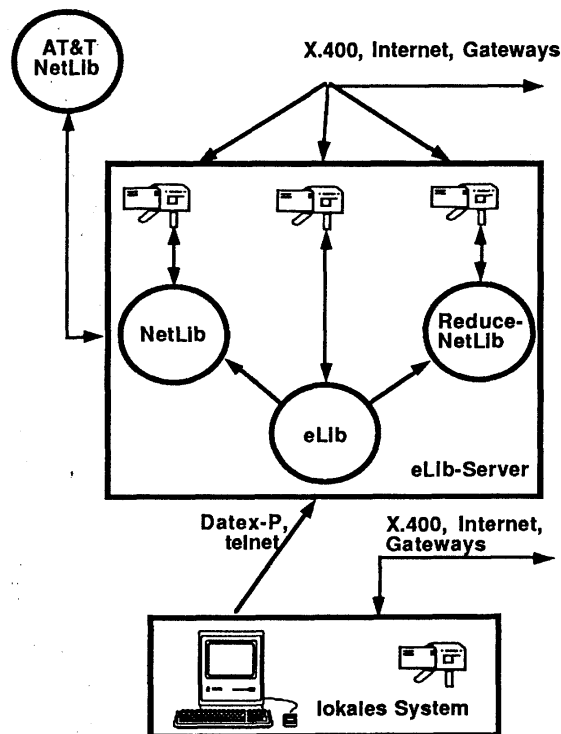
Erst mit dem Anschluß des ZIB an das Internet und dem Einsatz des "ftp"-Dateitransfers konnten diese Probleme adäquat gelöst werden. Heute kann der Berliner NetLib-Knoten innerhalb weniger Stunden nach dem Eintreffen neuer Quellen auf den

<sup>6</sup>netlib@elib.zib-berlin.dbp.de;

Internet: netlib@elib.zib-berlin.de

aktuellen Stand gebracht werden. Damit waren die Voraussetzungen für eine weitergehende Integration von NetLib und eLib gegeben.

Die eLib wurde mit der NetLib technisch derart verbunden, daß aus dem eLib-Dialog eine NetLib-Anforderung unmittelbar abgesetzt werden kann. Die eLib simuliert dabei gleichsam eine Anfrage an die NetLib und stellt das Ergebnis in ihrem Fenster bereit, wo es inspiziert und anschließend auch per elektronischer Post verschickt werden kann. Das entsprechende Kommando ermöglicht also einen interaktiven Zugriff auf Leistungen der NetLib - und es wird in der eLib mit einem speziellen Menü unterstützt.



**Bild 3.** on-line und off-line Verbund von eLib, NetLib und RedLib

Mit derselben Technik wurde die Reduce-NetLib<sup>7</sup>, die auch den NetLib-Prozessor einsetzt, ebenfalls am ZIB für den Einsatz im Dialog bereitgestellt werden. Hier wurde die Integration mit der eLib noch eine Stufe weiter getrieben. Die Index-Dateien der Reduce-NetLib sind so aufgebaut, daß man durch eine relativ einfache textuelle Analyse die Bezeichnungen der von ihnen adressierten Bibliotheken, Programme und Dokumente automatisch feststellen kann. Aus den Index-Dateien läßt sich damit automatisch ein Submenü für die eLib konstruieren, das direkt in die Software der Reduce-NetLib verweist. Dadurch wird es möglich, einen Zugang zur Software der Reduce-NetLib zu strukturieren, der allein mit menüorientierten Techniken der Auswahl bzw. Selektion auskommt - ein spezielles Kommando ist dann nicht mehr notwendig.

<sup>7</sup>Am ZIB auch per e-Mail erreichbar: redlib@elib.zib-berlin.dbp.de;  
Internet: redlib@elib.zib-berlin.de

In einem vorerst letzten Schritt des Verbundes haben wir das von der NetLib entlehnte send-Kommando für die eLib verallgemeinert und in der folgenden Formen bereitgestellt:

- \* schicke generellen Index von eLib, Netlib und Reduce-NetLib  
send index
- \* schicke den vollständigen Index einer Softwarebibliothek oder -Liste  
send index from <library-or-list>
- \* schicke eine spezielle Routine - und alles, wovon sie abhängt  
send <module> from <library-or-list>
- \* schicke genau eine Routine - und keine davon abhängigen Modulen  
send only <module> from <library-or-list>

Der Term <library-or-list> kann einen speziellen GAMS-Index oder eine andere allgemeine Liste der eLib bezeichnen, wie z.B. den Modul-Index oder die Liste der Kurzfassungen der Schriften eines speziellen Instituts. Der Term kann insbesondere eine Softwarebibliothek der eLib oder auch der NetLib oder der Reduce-NetLib kennzeichnen - die eLib wird die Anfrage ggf. an die entsprechende Bibliothek weiterleiten. Schickt der Anwender eine solche Anfrage an die eLib, so kann es ihm gleichgültig sein, ob er mit der eLib nur eine oder mehrere Bibliothekssysteme anspricht.

Die eLib ermöglicht somit den fensterorientierten Dialogzugriff mit Menüunterstützung auch für NetLib und Reduce-NetLib, zwei ursprünglich allein für den Zugriff mittels elektronischer Post eingerichtete Softwarebibliotheken. Das Leistungsspektrum dieser drei Bibliothekssysteme ist am ZIB in seiner ganzen Breite zusätzlich auch allein auf dem Wege über elektronische Post nutzbar.

## 6 Zusammenfassung

Mit zunehmender Dezentralisierung im Bereich des Wissenschaftlichen Rechnens steigt auch der Bedarf an Softwarequellen qualitativ hochwertiger mathematischer Algorithmen. Für den Software- und Informationsaustausch wird die Einrichtung öffentlicher, elektronischer Softwarebibliotheken vorgeschlagen. Am Beispiel der eLib des ZIB werden die notwendigen technischen Konzepte, die u.a. auch in den heutigen Gegebenheiten der technischen Kommunikation begründet sind, diskutiert. Die eLib ist fenster- und menüorientiert aufgebaut und folglich im Dialog sehr einfach zu benutzen. Sie kann aber auch mit ihrer Kommandosprache allein auf dem Wege über elektronische Post abgefragt werden. Über die eLib sind, nach dem Verbund mit den Netzbibliotheken NetLib und Reduce-NetLib jetzt etwa 6000 mathematische Programme und allgemeine Dokumente verfügbar. Das Dienstleistungsangebot der eLib setzt Beiträge aus einer aktiven Benutzergemeinschaft voraus. Es besteht die Hoffnung, daß durch den öffentlichen Softwareaustausch sich Entwickler und Anwender wechselseitig in ihrer Arbeit befruchten.



## Dank

Wir danken Dr. J. J. Dongarra und Dr. E. Grosse für die freundliche Unterstützung bei der Bereitstellung der NetLib und Dr. A. C. Hearn für die Unterstützung bei der Bereitstellung der Reduce-NetLib am ZIB. Wir sind Dr. R. F. Boisvert für die hilfreiche Bereitstellung des GAMS-Systems mit einer Fülle von Softwareklassifikationen in besonderer Weise verpflichtet. Wir danken Herrn S. Madlung und Frau B. Kirkorowicz für die Mitarbeit bei der Programmierung der eLib.

## 7 Literaturverzeichnis

- /Boisvert 90/* R.F. Boisvert, S.E. Howe, D.K. Kahaner  
The Guide to Available Mathematical Software Problem Classification System  
National Institute of Standards and Technology, Rep. NISTIR 4475,  
Nov. 1990
- /Boisvert 85/* R.F. Boisvert, S.E. Howe, D.K. Kahaner  
GAMS: A Framework for the Management of Scientific Software  
ACM TOMS, Vol. 11, No. 4, Dec. 1985, pp 313-355
- /DFN 90/* Am Deutschen Forschungsnetz angeschlossene Institutionen  
DFN Mitteilungen - jeweils im neuesten Heft
- /Dongarra 87/* J.J. Dongarra, E. Grosse  
Distribution of Mathematical Software via Electronic Mail  
Comm. ACM Vol 30, No 5, pp 403-407
- /Guelph 87/* CoSy Computer Conferencing System  
University of Guelph, Ontario, Dec. 1987

SC 86-1. P. Deuflhard; U. Nowak. *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems*. (vergriffen) In: Ber. Bunsenges. Phys. Chem., vol. 90, 1986, 940-946

SC 86-2. H. Melenk; W. Neun. *Portable Standard LISP for CRAY X-MP Computers*.

SC 87-1. J. Anderson; W. Galway; R. Kessler; H. Melenk; W. Neun. *The Implementation and Optimization of Portable Standard LISP for the CRAY*.

SC 87-2. R. E. Bank; T. F. Dupont; H. Yserentant. *The Hierarchical Basis Multigrid Method*. (vergriffen) In: Numerische Mathematik, 52, 1988, 427-458.

SC 87-3. P. Deuflhard. *Uniqueness Theorems for Stiff ODE Initial Value Problems*.

SC 87-4. R. Buhtz. *CGM-Concepts and their Realizations*.

SC 87-5. P. Deuflhard. *A Note on Extrapolation Methods for Second Order ODE Systems*.

SC 87-6. H. Yserentant. *Preconditioning Indefinite Discretization Matrices*.

SC 88-1. W. Neun; H. Melenk. *Implementation of the LISP-Arbitrary Precision Arithmetic for a Vector Processor*.

SC 88-2. H. Melenk; H. M. Möller; W. Neun. *On Gröbner Bases Computation on a Supercomputer Using REDUCE*. (vergriffen)

SC 88-3. J. C. Alexander; B. Fiedler. *Global Decoupling of Coupled Symmetric Oscillators*.

SC 88-4. H. Melenk; W. Neun. *Parallel Polynomial Operations in the Buchberger Algorithm*.

SC 88-5. P. Deuflhard; P. Leinen; H. Yserentant. *Concepts of an Adaptive Hierarchical Finite Element Code*.

SC 88-6. P. Deuflhard; M. Wulkow. *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable*. (vergriffen) In: IMPACT, 1, 1989, 269-301.

SC 88-7. H. Melenk; H. M. Möller; W. Neun. *Symbolic Solution of Large Stationary Chemical Kinetics Problems*.

SC 88-8. R. H. W. Hoppe; R. Kornhuber. *Multi-Grid Solution of Two Coupled Stefan Equations Arising in Induction Heating of Large Steel Slabs*.

SC 88-9. R. Kornhuber; R. Roitzsch. *Adaptive Finite-Element-Methoden für konvektionsdominierte Randwertprobleme bei partiellen Differentialgleichungen*.

SC 88-10. S.-N. Chow; B. Deng; B. Fiedler. *Homoclinic Bifurcation at Resonant Eigenvalues*.

SC 89-1. H. Zha. *A Numerical Algorithm for Computing the Restricted Singular Value Decomposition of Matrix Triplets*.

SC 89-2. H. Zha. *Restricted Singular Value Decomposition of Matrix Triplets*.

SC 89-3. W. Huamo. *On the Possible Accuracy of TVD Schemes*.

SC 89-4. H. M. Möller. *Multivariate Rational Interpolation: Reconstruction of Rational Functions*.

SC 89-5. R. Kornhuber; R. Roitzsch. *On Adaptive Grid Refinement in the Presence of Internal or Boundary Layers*.

SC 89-6. W. Huamo; Y. Shuli. *MmB-A New Class of Accurate High Resolution Schemes for Conservation Laws in Two Dimensions*.

SC 89-7. U. Budde; M. Wulkow. *Computation of Molecular Weight Distributions for Free Radical Polymerization Systems*.

SC 89-8. G. Maierhöfer. *Ein paralleler adaptiver Algorithmus für die numerische Integration*.

SC 89-9. H. Yserentant. *Two Preconditioners Based on the Multi-Level Splitting of Finite Element Spaces*.

SC 89-10. R. H. W. Hoppe. *Numerical Solution of Multicomponent Alloy Solidification by Multi-Grid Techniques*.

- SC 90-1. M. Wulkow; P. Deuffhard. *Towards an Efficient Computational Treatment of Heterogeneous Polymer Reactions.*
- SC 90-2. P. Deuffhard. *Global Inexact Newton Methods for Very Large Scale Nonlinear Problems.*
- SC 90-3. K. Gatermann. *Symbolic solution of polynomial equation systems with symmetry.*
- SC 90-4. F. A. Bornemann. *An Adaptive Multilevel Approach to Parabolic Equations I. General Theory & 1D-Implementation.*
- SC 90-5. P. Deuffhard; R. Freund; A. Walter. *Fast Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems.*
- SC 90-6. D. Wang. *On Symplectic Difference Schemes for Hamiltonian Systems.*
- SC 90-7. P. Deuffhard; U. Nowak; M. Wulkow. *Recent Developments in Chemical Computing.*
- SC 90-8. C. Chevalier; H. Melenk; J. Warnatz. *Automatic Generation of Reaction Mechanisms for Description of Oxidation of Higher Hydrocarbons.*
- SC 90-9. P. Deuffhard; F. A. Potra. *Asymptotic Mesh Independence of Newton-Galerkin Methods via a Refined Mysovskii Theorem.*
- SC 90-10. R. Kornhuber; R. Roitzsch. *Self Adaptive FEM Simulation of Reverse Biased p-n Junctions.*
- SC 90-11. K. Gatermann; A. Hohmann. *Symbolic Exploitation of Symmetry in Numerical Path-following.*
- SC 90-12. A. Walter. *Improvement of Incomplete Factorizations by a Sparse Secant Method.*
- SC 90-13. F. A. Bornemann. *An Adaptive Multilevel Approach to Parabolic Equations II.*
- SC 90-14. J. Ackermann; M. Wulkow. *MACRON - A Program Package for Macromolecular Reaction Kinetics.*
- SC 90-15. M. Wulkow; J. Ackermann. *Numerical Treatment of Polyreactions - Recent Developments.*
- SC 90-16. Hege; Hinnerk; Stuben. *Vectorization and Parallelization of Irregular Problems via Graph coloring.*
- SC 90-17. D. Wang. *Symplectic Difference Schemes for Perturbed Hamiltonian Systems.*
- SC 90-18. H. Caprasse; J. Demaret; K. Gatermann; H. Melenk. *Power-Law Type Solutions of Fourth-Order Gravity for Multidimensional Bianchi I Universes.*
- SC 90-19. U. Nowak. *A Note on Dynamic Sparsing in Stiff Extrapolation Methods.*
- SC 90-20. A. Walter. *Sparse Secant Methods for the Iterative Solution of Large Nonsymmetric Linear Systems.*
- SC 90-21. A. Walter. *Convergence Behavior of Sparse Secant Methods for Incomplete Factorization*

**Veröffentlichungen des Konrad-Zuse-Zentrum für Informationstechnik Berlin**  
**Technical Reports**

**Februar 1991**

TR 86-1. H. J. Schuster. *Tätigkeitsbericht (vergriffen)*

TR 87-1. H. Busch; U. Pöhle; W. Stech. *CRAY-Handbuch. - Einführung in die Benutzung der CRAY.*

TR 87-2. H. Melenk; W. Neun. *Portable Standard LISP Implementation for CRAY X-MP Computers. Release of PSL 3.4 for COS.*

TR 87-3. H. Melenk; W. Neun. *Portable Common LISP Subset Implementation for CRAY X-MP Computers.*

TR 87-4. H. Melenk; W. Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running COS Version 3.3*

TR 87-5. H. Melenk; W. Neun. *REDUCE Users Guide for the CRAY 1 / X-MP Series Running COS. Version 3.3*

TR 87-6. R. Buhtz; J. Langendorf; O. Paetsch; D. A. Buhtz. *ZUGRIFF - Eine vereinheitlichte Datenspezifikation für graphische Darstellungen und ihre graphische Aufbereitung.*

TR 87-7. J. Langendorf; O. Paetsch. *GRAZIL (Graphical ZIB Language).*

TR 88-1. R. Buhtz; D. A. Buhtz. *TDLG 3.1 - Ein interaktives Programm zur Darstellung dreidimensionaler Modelle auf Rastergraphikgeräten.*

TR 88-2. H. Melenk; W. Neun. *REDUCE User's Guide for the CRAY 1 / CRAY X-MP Series Running UNICOS. Version 3.3.*

TR 88-3. H. Melenk; W. Neun. *REDUCE Installation Guide for CRAY 1 / CRAY X-MP Systems Running UNICOS. Version 3.3.*

TR 88-4. D. A. Buhtz; J. Langendorf; O. Paetsch. *GRAZIL-3D. Ein graphisches Anwendungsprogramm zur Darstellung von Kurven- und Funktionsverläufen im räumlichen Koordinatensystem.*

TR 88-5. G. Maierhöfer; G. Skorobohatyj. *Parallel-TRAPEX. Ein paralleler, adaptiver Algorithmus zur numerischen Integration ; seine Implementierung für SUPRENUM-artige Architekturen mit SUSI.*

TR 89-1. *CRAY-HANDBUCH. Einführung in die Benutzung der CRAY X-MP unter UNICOS.*

TR 89-2. P. Deuffhard. *Numerik von Anfangswertmethoden für gewöhnliche Differentialgleichungen.*

TR 89-3. A. R. Walter. *Ein Finite-Element-Verfahren zur numerischen Lösung von Erhaltungsgleichungen.*

TR 89-4. R. Roitzsch. *KASKADE User's Manual.*

TR 89-5. R. Roitzsch. *KASKADE Programmer's Manual.*

TR 89-6. H. Melenk; W. Neun. *Implementation of Portable Standard LISP for the SPARC Processor.*

TR 89-7. F. A. Bornemann. *Adaptive multilevel discretization in time and space for parabolic partial differential equations.*

TR 89-8. G. Maierhöfer; G. Skorobohatyj. *Implementierung des parallelen TRAPEX auf Transputern.*

TR 90-1. K. Gatermann. *Gruppentheoretische Konstruktion von symmetrischen Kubaturformeln.*

TR 90-2. G. Maierhöfer; G. Skorobohatyj. *Implementierung von parallelen Versionen der Gleichungslöser EULEX und EULSIM auf Transputern.*

TR 90-3. *CRAY-Handbuch. Einführung in die Benutzung der CRAY X-MP unter UNICOS 5.1*

TR 90-4. H.-C. Hege. *Datenabhängigkeitsanalyse und Programmtransformationen auf CRAY-Rechnern mit dem Fortran-Präprozessor fpp.*

TR 90-5. M. Grammel; G. Maierhöfer; G. Skorobohatyj. *Trapex in POOL; Implementierung eines numerischen Algorithmus in einer parallelen objektorientierten Sprache.*

TR 90-6. P. Deuffhard; A. Hohmann. *Einführung in die Numerische Mathematik.*

TR 90-7. P. Deuffhard. *Zuses Werk weiterdenken. (Vortrag)*

TR 90-8. M. Wulkow. *Numerical Treatment of Countable Systems of Ordinary Differential Equations.*

TR 90-9. R. Roitzsch; R. Kornhuber. *BOXES - a Program to Generate Triangulations from a Rectangular Domain Description.*

TR 90-10. U. Nowak; L. Weimann. *A Family of Newton Codes for Systems of Highly Nonlinear Equations Algorithm, Implementation, Application.*

TR 90-11. U. Nowak; L. Weimann. *GIANT - A Software Package for the Numerical Solution of Very Large Systems of Highly Nonlinear Equations.*

TR 90-12. U. Nowak; U. Pöhle. *A Family of Extrapolation Codes for ODEs and DAEs - Part I.*

TR 90-13. W. K. Giloi. *Konrad Zuses Plankalkül als Vorläufer moderner Programmier-modelle. (Vortrag)*

TR 91-1. F. Bornemann; B. Erdmann; R. Roitzsch. *KASKADE - Numerical Experiments.*

TR 91-2. J. Lügger; W. Dalitz. *Verteilung mathematischer Software mittels elektronischer Netze - die elektronische Softwarebibliothek eLib -.*

TR 91-3. S. W. C. Noelle. *On the Limits of Operator Splitting: Numerical Experiments for the Complex Burgers Equation.*