

# Mathematische Programmbibliotheken im Wandel der Zeit

I. Bischoff-Gauß, R. Pietschmann, F. Schmitz, HIK

## Einleitung

Mathematische (Unter-)Programm-Bibliotheken sind Bestandteil eines vielfältigen Angebots an Programmen und Programmpaketen, welches unter dem Sammelbegriff „Mathematische Software“ für die technisch-wissenschaftlichen Anwender der Datenverarbeitung verfügbar ist. Der Begriff „Mathematical Software“ wurde 1969 von John R. Rice geprägt und als „Computer programs which implement widely applicable mathematical procedures“ definiert [2]. Das heutige Spektrum umfaßt einerseits Programm-Bibliotheken zur Bearbeitung von Aufgabenstellungen aus der Numerischen Mathematik (Funktionsberechnungen, lineare und nichtlineare Gleichungssysteme, Anfangs- und Randwertprobleme bei gewöhnlichen und partiellen Differentialgleichungen, schnelle Fourier-Transformationen etc.), aus der Statistik (beschreibende Statistik, Regression und Korrelation, Varianzanalyse, Testverfahren etc.), andererseits aber auch die sog. Computer-Algebra-Systeme, welche Ergebnisse sowohl in symbolischer als auch in numerischer Form liefern können. Zusätzlich bieten beide Arten ausführliche Online-Hilfen sowie Möglichkeiten der grafischen Darstellung und Visualisierung.

Offensichtliche Gründe für die Bereitstellung mathematischer Software sind: Minimierung des Programmieraufwandes, Nutzung von in Programme gegossenem Spezialwissen nach dem neuesten Stand der Wissenschaft, Produktion von gesicherten Ergeb-

nissen. Unter diesen Aspekten gehören Programm-Bibliotheken unverzichtbar zum Software-Angebot eines technisch-wissenschaftlichen Rechenzentrums [3].

## Historie

Geht man in der Geschichte des Rechenzentrums zurück bis etwa 1969/70, so findet man drei Quellen mathematischer Unterprogramme vor: Die SYS1.FORTLIB als Bestandteil des FORTRAN-Laufzeitsystems, welche neben Funktionen wie Sinus, Tangens etc. auch Ein-/Ausgabe- und sonstige Dienstprogramme bereitstellte. Die GFK.FORTLIB als Sammlung von FORTRAN- und Assembler-Programmen für mathematische und allgemeinere Aufgabenstellungen, welche von Mitarbeitern des Forschungszentrums der gesamten Benutzerschaft zur Verfügung gestellt wurden. Und das Scientific Subroutine Package (SSP), wohl die erste Sammlung dieser Art von über 250 mathematischen und statistischen Unterprogrammen – die meisten davon in Ausführungen für einfache und doppelte Genauigkeit – entwickelt von IBM und im FORTRAN-Quellcode verfügbar (was der Verbreitung förderlich war, weil jeder Benutzer sich die Programme auf Lochkarten ausstanzen und sie bei Bedarf auch modifizieren konnte).

Ab 1972 bot das Rechenzentrum die kostenpflichtigen Lizenzprogramme SLMATH (für FORTRAN) und PLMATH (für PL/1) als Nachfolger für SSP an, wobei PLMATH – in Ermangelung anderer Quellen für die Programmiersprache PL/1 – erst 1992 abgemietet wurde.

Einige Entwickler von SSP gründeten 1971 in den USA die Firma IMSL (International Mathematical and Statistical Libraries), welche ein Jahr später die erste rein kommerzielle mathematische Programm-Bibliothek für FORTRAN an IBM-Kunden auslieferte (komplett in je einer Version für einfache und doppelte Genauigkeit) [2]. Ab 1973 wurde die gleiche Bibliothek auch für Kunden anderer Rechnerhersteller verfügbar, was ein Novum darstellte, konnten doch jetzt Quellprogramme mit IMSL-Aufrufen zwischen verschiedenen Plattformen ausgetauscht werden, sofern nur die IMSL-Bibliothek vorhanden war. Allerdings war der Quellcode der Unterprogramme nicht verfügbar. Das Rechenzentrum lizenzierte ab 1978 diese Bibliothek und bietet sie bis heute seinen Benutzern an. Sie zeichnete sich in der Vergangenheit schon aus durch die Qualität der Algorithmen, ein umfangreiches Anwendungsspektrum, eine weite Verbreitung und gute Dokumentation in Form von Handbüchern, die in späteren Versionen auch Online verfügbar waren und durch komfortable Online-Hilfen ergänzt wurden. Ab 1991 wurde von IMSL auch eine Bibliotheksversion für die Programmiersprache C angeboten.

Mit der HARWELL Subroutine Library (HSL), einem nicht kommerziell vertriebenen Produkt von HARWELL Laboratory in England, verschaffte das Rechenzentrum 1980 seinen Benutzern wieder Zugriff auf Programme im Quellcode, welcher mit der Auflage des Zitierens in wissenschaftlichen Veröffentlichungen für Forschungszwecke frei verfügbar war. Im ma-

thematischen Teil deckte die HSL nahezu den Bereich der IMSL ab, im Statistikbereich nur einen kleinen Ausschnitt. Einige Dienstprogramme waren im IBM-Assembler-Code geschrieben (dies schränkte die Portabilität ein). Mit dem Übergang vom Betriebssystem MVS auf Unix 1999 endete die HARWELL-Ära im Forschungszentrum.

Eine neue Qualität bezüglich der Zuverlässigkeit von Ergebnissen numerischer Berechnungen brachte das IBM-Produkt ACRITH, die „High Accuracy Arithmetic Library“, mit dem Konzept der „verifizierenden Algorithmen“ bzw. der vom Programm „verifizierten Ergebnisse“: Statt eines einzigen Ergebnisses werden jetzt zwei Zahlen zurückgegeben, die das einschließende Intervall für das – unbekannte – exakte Ergebnis definieren. Die theoretische Basis für die gesicherte Fehlerschranken liefernden Algorithmen – die Anwendung der Intervall-Arithmetik sowie von Fixpunktsätzen der Analysis, der gerichteten Rundung von Zwischenergebnissen, die Verwendung eines langen Akkumulators für die exakte Berechnung von Skalarprodukten etc. – wurde schon Anfang der siebziger Jahre wesentlich von der Universität Karlsruhe gelegt. Die Firma IBM hat diese Arbeiten in das Produkt ACRITH überführt und vermarktet und sogar einen Rechner – die IBM 4361 – gebaut, bei dem die wesentlichen Arithmetik-Funktionen in Hardware implementiert worden waren. Das Rechenzentrum hat diese Bibliothek 1986 für die Benutzer verfügbar gemacht und damit – hoffentlich – ins Be-

wusstsein gerückt, nicht jedem Ergebnis bisheriger Rechnungen blindlings zu vertrauen. Die Methoden haben sich langfristig nicht durchgesetzt, vielleicht weil die Rechenzeiten gegenüber herkömmlichen Algorithmen deutlich länger waren und ACRITH nicht das gesamte Spektrum anderer Bibliotheken abdeckte.

1987 wurde im Rechenzentrum der erste Vektorrechner installiert, der Siemens-Fujitsu VP50. Um die Möglichkeiten dieser neuen Rechnergeneration – das Verarbeiten von Vektoren als Ganzes oder in Teilen gegenüber der bisherigen elementweisen Abarbeitung – nutzen zu können, mussten Programme „vektoriert“ werden. Diese Aufgabe übernahmen die Compiler der Hersteller, jedoch waren auch die Programmierer gefordert, ebenso die Anbieter von Bibliotheken. Die Herstellerbibliothek „Scientific Subroutine Library 2“ (SSL2) – sie war schon auf dem Skalarrechner im Einsatz – wurde in einer für die spezielle Hardware der VP50 optimierten und um Vektoralgorithmen erweiterten Version mitgeliefert. Der Effizienzgewinn der diese Bibliothek nutzenden Programme ging allerdings zu Lasten ihrer Portabilität.

Eine Alternative zur Effizienzsteigerung ohne Portabilitätseinbußen boten die BLAS-Programme in Verbindung mit einer plattformunabhängigen Bibliothek. BLAS („Basic Linear Algebra Subroutines“) ist das Ergebnis eines von ACM-SIGNUM geförderten Standardisierungsprojektes für Mathematische Software, in einer ersten Version 1979 bereitgestellt und in ACM Transactions On Mathematical Software (TOMS) do-

kumentiert als ein Satz von Fortran-Unterprogrammen für die elementaren Operationen mit Vektoren und Matrizen. Man hatte sich auf Namen, Parameterlisten und funktionale Beschreibungen geeinigt, ein bedeutender Schritt für die Entwicklung von Programmen der linearen Algebra.

Der Software-Hersteller IMSL paßte 1987 seine Bibliotheken an diese Entwicklung an: entsprechender Programm-Code wurde durch zum Standard konforme BLAS-Aufrufe ersetzt. Hardware-Hersteller optimierten die BLAS-Bibliothek für ihre speziellen Architekturen, so IBM mit der 3090-Vector-Facility und Siemens-Fujitsu mit dem VP400, die 1990 bzw. 1991 im Rechenzentrum installiert wurden. Dadurch konnten die Benutzer durch Einbinden dieser optimierten BLAS-Bibliotheken an Stelle der IMSL-BLAS oft in signifikantem Maße Rechenzeit einsparen.

Ein neuer Rechnertyp wurde 1995 im Rechenzentrum etabliert: der Parallel-Rechner CRAY-J916 unter dem Betriebssystem UNICOS. Die Möglichkeit, ein Programm auf mehreren Prozessoren gleichzeitig – also „parallel“ – auszuführen (schon auf der IBM3090/600VF möglich, aber nicht effizient realisierbar), war eine neue Herausforderung an Mathematiker und Entwickler mathematischer Software. Hierzu bot der Hersteller CRAY neben einer optimierten BLAS-Bibliothek und einer auf BLAS basierenden Bibliothek LAPACK, die die Funktionalität der bekannten Bibliotheken LINPACK und EISPACK abdeckte, auch eine eigene mathematische Bibliothek SCILIB an, deren wesentli-

che Programme die Vorteile der Parallel-Vektorarchitektur nutzen. Die Firma IMSL stellte für die CRAY-Rechner ebenfalls die Architektur nutzende Versionen ihrer Bibliothek bereit.

Die rasant gesteigerte Rechenleistung der Computer-Hardware zwingt die Software-Entwickler zu neuen Verfahren der Anpassung und Optimierung, die bisher sehr zeitaufwendig waren und nur von hochqualifizierten Spezialisten geleistet werden konnten. Ziel des seit 1998 laufenden Forschungsprojekts ATLAS (Automatically Tuned Linear Algebra Software) ist, empirische Methoden in Form des Software-Pakets ATLAS bereit zu stellen, um Programme der linearen Algebra – in erster Linie die BLAS-Bibliothek, aber auch aus LAPACK – für unterschiedliche Hardware-Plattformen zu optimieren [4].

### Mathematische Bibliotheken der Gegenwart

Die Anzahl weltweit angebotener freier und kommerzieller mathematischer Bibliotheken unterscheidet sich dramatisch von der Anzahl der im Forschungszentrum zentral installierten Bibliotheken. Hier einige interessante Seiten im Netz:

[www.netlib.org/index.html](http://www.netlib.org/index.html)

[math.nist.gov](http://math.nist.gov)

[www.netlib.org/lapack/index.html](http://www.netlib.org/lapack/index.html)

[www.visual-numeric.de/produkte/index.html#IMSL](http://www.visual-numeric.de/produkte/index.html#IMSL)

[www.nag.co.uk/numeric/numerical\\_libraries.asp](http://www.nag.co.uk/numeric/numerical_libraries.asp)

Im Rechenzentrum sind folgende mathematischen Unterprogramm-bibliotheken zentral verfügbar:

### BLAS (Basic Linear Algebra Subprograms)

Die BLAS-Bibliothek ist eine aus Public-Domain Routinen bestehende Sammlung von Basisprogrammen zur effizienten Lösung von Problemen der Linearen Algebra. Es werden Routinen für Vektor-Vektor, Vektor-Matrix und Matrix-Matrix Operationen angeboten. Nach und nach hat sich diese Sammlung als Standard etablieren können, da die im Quelltext verfügbaren Routinen von allen größeren Herstellern als hardwareoptimierte BLAS-Version (-> ATLAS ) zur Verfügung gestellt werden. Auch die Anbieter von Softwarepaketen bedienen sich der BLAS Routinen. Vom Rechenzentrum wurden diese Routinen erstmals auf einem Vektorrechner VP50 und einer IBM 3090/600 VF zur Verfügung gestellt.

### LAPACK (Linear Algebra PACKage)

Eine neue Art von numerischer Unterprogramm-bibliothek löste ab 1992 die Bibliotheken EISPACK (seit 1976) und LINPACK (seit 1979) ab. LAPACK bietet eine Vielzahl von sogenannten 'driver routines', die sich wieder in 'simple' und 'expert' unterteilen. Es wird aber auf die optimierten BLAS Bibliotheken des jeweiligen Herstellers zugegriffen.

### ScaLAPACK (Scalable LAPACK)

Durch Hardware-designänderungen von Hochleistungsrechnern wurde der Bedarf nach parallelen Bibliotheken geweckt. 1995 wurde die erste Version für die sogenannte MIMD-Architektur (multiple instruction streams, multiple data streams, Flynn Einteilung) zur Verfügung gestellt. Für große Probleme eröffnet diese Bibliothek die Möglichkeit, verteilten Speicher und schnelle Prozessoren über leistungsstarke Netze effizient zu nutzen. Leider steigt durch einen hohen Speicherbedarf die Anzahl der zu nutzenden Prozessoren so stark an, dass bei einer Eigenwertberechnung die Anwendung nur verteilt auf einem VPP300 mit viel lokalem Speicher gerechnet werden kann.

### IMSL

Die seit 1972 angebotene IMSL Bibliothek wird laut Benutzerbefragung auch wegen des hohen Preises vor der NAG Bibliothek (Numerical Algorithms Group) bevorzugt. Deshalb wird bis heute die IMSL auf den zentralen Rechnern angeboten, obwohl die NAG moderner ist. Die IMSL bietet mit ihren mathematischen und statistischen Teilen Fortran- als auch C-Programmierern eine einfache Schnittstelle an. Über ein MPI-Interface können verteilte Systeme gemeinsam an der Lösung einer Aufgabe arbeiten.

### Bibliotheken von morgen

Bedingt durch die Annäherung der Leistungsdaten von Hochleistungsprozessoren und die in ho-

her Stückzahl produzierten Commodity-Komponenten und den Verdrängungswettbewerb der Generalisten gegen die Spezialisten im Hochleistungsrechnerbau (IBM, SUN, HP, Intel, Cray, Thinking Machines usw.) sowie durch Leistungssteigerung mittels paralleler Strukturen ist auch ein Umdenken bei numerischen Bibliotheken notwendig.

Gerade diese parallelen Systeme können aber eine wesentlich stärkere Beschleunigung von Anwendungen bewirken, als der Intel-Mitbegründer Gordon Moore noch 1968 vorhergesagt hatte.

Adaptive Algorithmen in der Numerik haben den Vorteil, dass etwa ein Berechnungsgebiet nicht mehr wie bei herkömmlichen Verfahren a priori statisch festgelegt wird, sondern dynamisch während des Berechnungsvorganges zugeordnet wird. Dabei werden Größen wie Rechenleistung, Netzbandbreite, Latenz, usw. berücksichtigt. So ergibt sich eine dynamische Lastverteilung und optimale Nutzung der Ressourcen. Der Güte des Ergebnisses als auch dem Ressourcenmanagement gilt dabei das Hauptaugenmerk. Jedoch gibt es momentan nur Lösungen für ganz bestimmte Fragestellungen. An-

sätze sind erkennbar bei adaptiver FEM-Software, Bildanalysesoftware und in der Bioinformatik.

### Beispiel einer IMSL-Nutzung

Am Institut für Meteorologie und Klimaforschung wurde mit Hilfe umfangreicher numerischer Simulationen am VPP5000 eine Windstatistik für den Raum Baden-Württemberg berechnet ([www.lfu.baden-wuerttemberg.de/lfu/uis/energie.html](http://www.lfu.baden-wuerttemberg.de/lfu/uis/energie.html)). Dabei wurde in einem horizontalen Raster mit einer Gitterweite von 1 km die mittlere Windgeschwindigkeit in

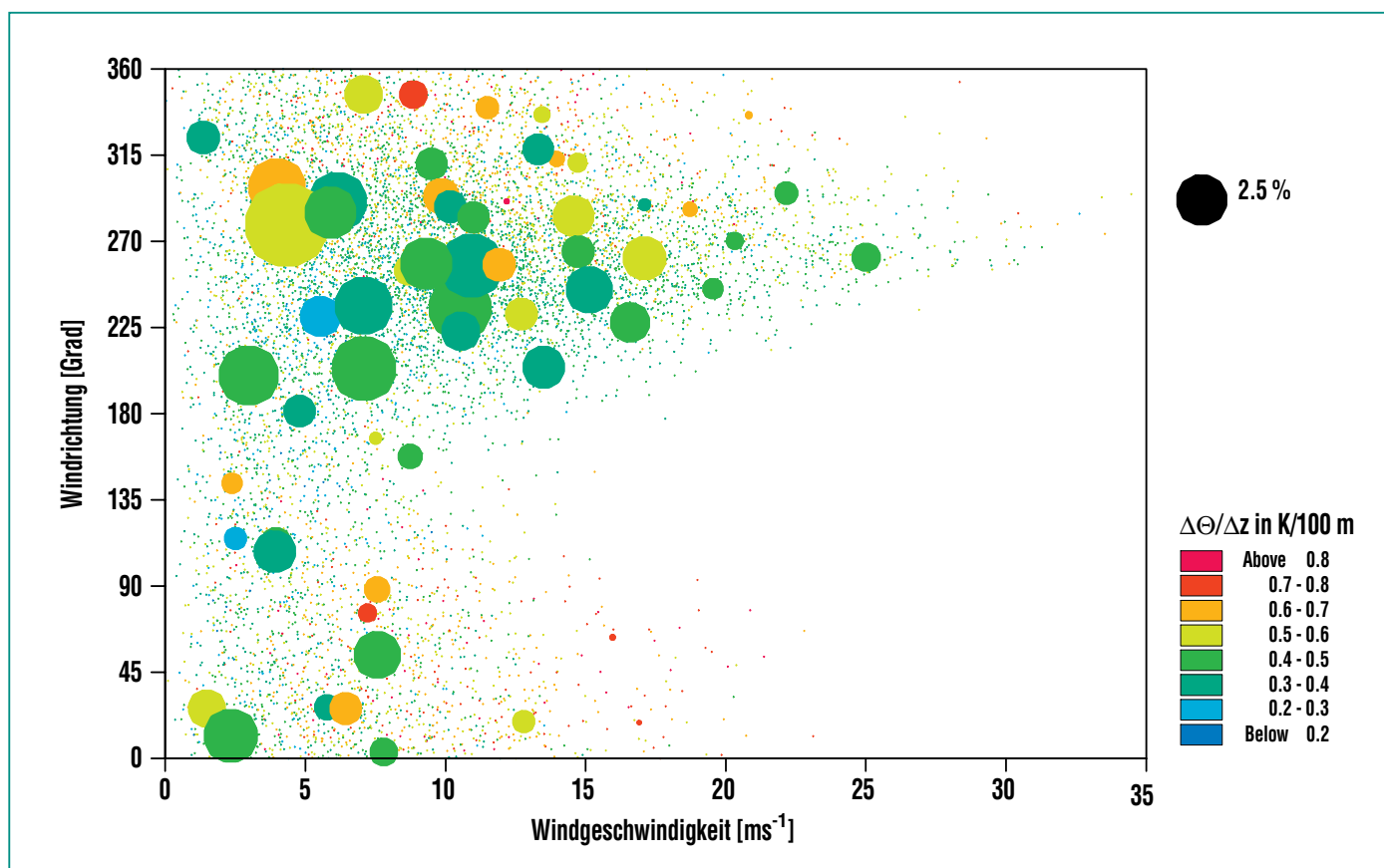


Abb. 1: Clusteranalyse für die Parameter Windrichtung, Windgeschwindigkeit und Temperaturschichtung (gefüllte Kreise). Die Größe der Kreise ist ein Maß für die Häufigkeit, die Punkte stellen die gesamten Beobachtungen dar.

verschiedenen Höhen bestimmt. Zur Berechnung ist eine Zusammenfassung der Großwetterlagen erforderlich. Diese erfolgt mit Hilfe einer Clusteranalyse, die mit der IMSL-Bibliothek durchgeführt wird, wobei die Prozeduren CDIST, CLINK und CNUMB zur Anwendung kommen. Im verwendeten numerischen Simulationsmodell KAMM [1] wird die großräumige Wetterlage charakterisiert durch die horizontalen Windkomponenten  $u$  und  $v$  in 700 hPa und einem vertikalen Temperaturgradienten  $\delta\theta/\delta z$ . Als Eingangsdaten dienen Tagesmittelwerte der NCEP/NCAR Reanalysedaten über einen Zeitraum von 25

Jahren. Es werden hierarchische Cluster nach folgendem Schema erzeugt: im ersten Schritt wird jede Beobachtung ( $u, v$  in 700 hPa,  $\delta\theta/\delta z$ ) als ein Cluster betrachtet. Es wird dann der Abstand zwischen den Clustern berechnet und die am nächsten zusammenliegenden Cluster zu einem neuen Cluster zusammengefasst. Diese Prozedur wird wiederholt, bis die gewünschte Anzahl von Clustern erreicht ist. Im vorliegenden Falle werden die ca. 9000 Beobachtungen zu 60 Clustern zusammengefasst. Die Verteilung der Cluster in der dreidimensionalen Häufigkeitsverteilung für die Parameter Windrichtung, Windgeschwindigkeit

und Temperaturschichtung ist in der Abb. 1 wiedergegeben. Für jeden Clusterpunkt wird eine numerische Simulation durchgeführt, woraus anschließend mit der entsprechenden Gewichtung die mittleren Windverhältnisse bestimmt werden.

## Literatur

- [1] G. Adrian, F. Fiedler, *Contrib. Phys. Atmos.*, 64, 27-48, 1991
- [2] Cowell W. R., *Sources and Development of Mathematical Software. Prentice-Hall (1984), Englewood Cliffs, New Jersey.*
- [3] J.-Fr. Hake, P. Jansen, *Mathematische Software: Ergebnisse und Trends. Interner Bericht, Forschungszentrum Jülich (1992).*
- [4] R.C. Whaley., A. Petitet, J. J. Dongarra, *Automated Empirical Optimization of Software and the ATLAS Project (2000). Quelle: Internet.*