



KERNFORSCHUNGSANLAGE JÜLICH GmbH

Zentralinstitut für Angewandte Mathematik

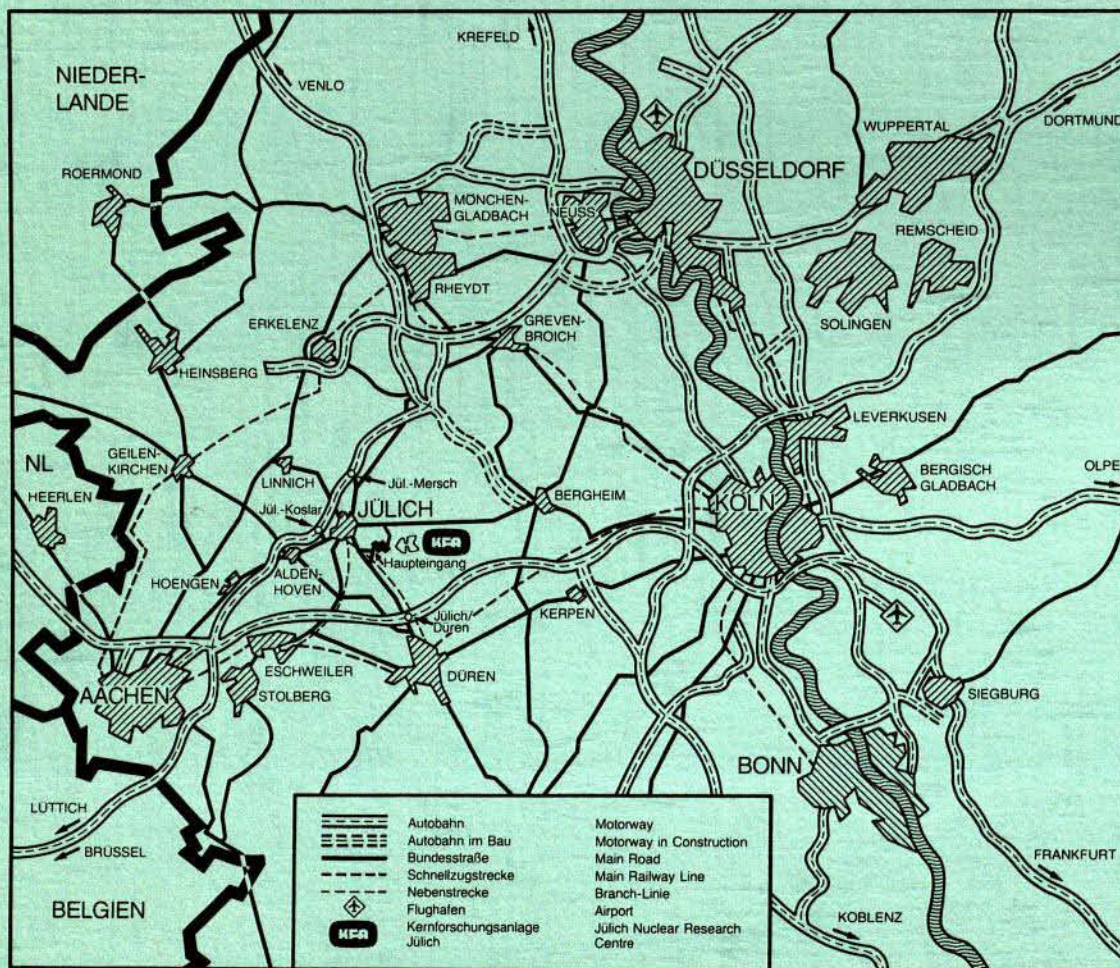
NUMETH
NUmerische METHoden
für den End-User

von
K. Kilberth

Jül - 1761

Juli 1982

ISSN 0366-0885



Als Manuskript gedruckt

Berichte der Kernforschungsanlage Jülich – Nr. 1761

Zentralinstitut für Angewandte Mathematik Jül - 1761

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH
Postfach 1913 · D-5170 Jülich (Bundesrepublik Deutschland)
Telefon: 02461/610 · Telex: 833556 kfa d

NUMETH
NUmerische METHoden
für den End-User

von
K. Kilberth

NUMETH

NUMerische METHoden für den End-User

Klaus Kilberth

Zentralinstitut für Angewandte Mathematik

Kernforschungsanlage Jülich GmbH

Zusammenfassung:

NUMETH ist ein selbsterklärendes Dialogsystem auf der Basis von APL, das am Zentralinstitut für Angewandte Mathematik der Kernforschungsanlage Jülich entwickelt wurde.

Ziel dieses Systems ist es, zur Behandlung von Experiment-Daten auch dem mathematisch oder programmiertechnisch weniger Versierten einen einfachen Zugang zu numerischen Methoden zu ermöglichen.

Konzept und einzelne Komponenten von NUMETH werden vorgestellt.

Summary:

NUMETH is a self-explanatory dialog system, based on APL, which was developed at the Zentralinstitut für Angewandte Mathematik of the Kernforschungsanlage Jülich.

The purpose of this system is to provide an easy access to numerical methods for the handling of experimental data even to users less experienced in mathematics or in programming.

Concept and several components of NUMETH are presented.

Gliederung:

1. Motivation
 - 1.1 Experiment-Daten
 - 1.2 Dialogsysteme
 - 1.3 Warum APL ?
2. Das System NUMETH
 - 2.1 Aufbau von NUMETH
 - 2.2 Die Haupt-Prozeduren
 - 2.3 Die Deskriptoren
 - 2.4 Hilfs- und Service-Prozeduren
 - 2.5 Anforderungen an den Benutzer
 - 2.6 Hinweise zur Implementierung
3. Numerische Methoden
 - 3.1 Die Verfahren der Haupt-Prozeduren
 - 3.2 Beispiele und Vergleiche
4. Hilfs- und Service-Prozeduren
 - 4.1 Konzept der APL-Funktionen
 - 4.2 NUMETH-interne Funktionen
 - 4.3 Hilfsfunktionen
5. Erfahrungen und Ausblick
6. Literatur
7. Anhang
 1. Deskriptoren
 2. NUMETH-Namen

1. Motivation

Zur Behandlung von Experiment-Daten wurde ein geeignetes Dialogsystem benötigt. Dieses System besteht aus einzelnen Komponenten, die jeweils ein selbständiges Teilsystem darstellen.

Eine dieser Komponenten enthält numerische Methoden. Für das Arbeiten mit diesem System sollen weder spezielle Kenntnisse in Numerik noch in Programmierung vorausgesetzt werden. Andererseits sollen dem Anwendungs-Experten brauchbare Programmierwerkzeuge und dem Programmier-Experten wirkungsvolle Hilfsmittel für problembezogene Anwendungen angeboten werden.

1.1 Behandlung von Experiment-Daten

Die Auswertung von Experiment-Daten, z.B. in Form von Meßreihen, vollzieht sich in mehreren Schritten. Nach der Erfassung der Daten durch den Experiment-Rechner werden sie auf dem Großrechner in einem Batchprozeß reduziert.

Diese reduzierten Daten werden im Dialog weiter ausgewertet. Dafür sind Graphik-, Statistik- und Numerik-Pakete erforderlich. Außerdem ist der Anschluß an eine Datenbank wichtig.

1.2 Das Dialog-System

Bei der Realisierung des Dialog-Systems galt es, zwei - zunächst widersprüchliche - Zielsetzungen zu verbinden: einerseits muß der Laie ausreichend unterstützt werden, andererseits soll der Experte seine Kenntnisse frei entfalten können.

Besondere Anforderungen wurden unter dem Gesichtspunkt der 'Benutzer-Freundlichkeit' an das Dialog-System gestellt:

Für den Laien: leicht erlernbar,
selbsterklärend,
nur 1 Dialogebene,
überschaubarer logischer Aufbau,
einheitliche Kommandosprache mit problembezogenen Schlüsselwörtern,
klare Fehlerdiagnostik (keine einfachen Systemfehler-Meldungen),
arbeitsfähig mit minimalem Kommandovorrat.

Für den Experten: durch den Benutzer erweiterbar,
Weiterverarbeitung von Zwischenergebnissen,
keine Einschränkung durch Dialogführung,
Unterstützung auch bei Nicht-Standardanwendungen.

Das System soll aber auch leistungsfähig sein, was bedeutet:

Auswahl moderner Standard-Verfahren,
Zugriff auf Datenbanken und höhere Programmiersprachen.

Eine gute Unterstützung des Benutzers würde man durch eine straffe Dialogführung, etwa durch eine Menü-Funktion, erreichen. Der Nachteil besteht aber in einer starken Einschränkung der Benutzeraktivitäten. Insbesondere könnte das System nicht in einfacher Weise erweitert werden.

Wir wählten einen freien Dialog. Zur Unterstützung des Benutzers stellten wir eine Fülle von Hilfsfunktionen für Information zur Verfügung. Durch Voreinstellung von Parametern wurden die Standard-Anwendungen erheblich vereinfacht.

1.3 Warum APL ?

Als Basis-Sprache entschieden wir uns für APL. Denn APL bietet bereits eine komfortable Dialog-Umgebung und verfügt über eine Fülle mächtiger Operatoren und Funktionen, die eine prägnante Formulierung numerischer Fragestellungen gestatten.

Außerdem

ermöglicht es die Definition eigener Funktionen,
verfügt über einen Editor,
enthält Testhilfen (Trace, Stop).

Der klare logische Aufbau von APL gestattet eine konsistente Erweiterung der Funktions-Aufrufe zu einer einheitlichen Kommandosprache. Dabei benötigt der Laie für die Lösung von Standard-Problemen nur Grundkenntnisse in APL. Für den APL-Experten bieten die zusätzlichen Komponenten eine beachtliche Unterstützung bei der Lösung numerischer Probleme.

Mit Hilfe des Execute-Operators, angewandt auf Zeichenketten, ist auch die Behandlung von Funktionen im Sinne einer mathematischen Funktion möglich.

Beispiel:

Die Formel für die numerische Quadratur einer Funktion F zu den Stützstellen X mit den Gewichten G läßt sich in APL sehr einfach schreiben als:

$$G +.x \int F, ' X' .$$

Für den APL-Laien ermöglicht NUMETH einen entsprechenden Aufruf mit Hilfe der Funktionen INTEGRAL und WEIGHT:

$$F \text{ INTEGRAL } G \text{ WEIGHT } X .$$

Ohne Information über die Gewichte würde auch folgender Aufruf der Funktion INTEGRAL das gewünschte Ergebnis liefern:

$$F \text{ INTEGRAL } X .$$

Ein weiterer Grund für die Wahl von APL ist, daß APL in einfacher Weise Anwendungen gestattet, die in anderen Sprachen entweder überhaupt nicht oder nur sehr umständlich formuliert werden können, z.B. die dynamische Definition von eindeutigen, katalogisierbaren Namen (vgl. Abschnitt 2.4c). Eine wichtige Rolle spielen dabei der EXECUTE-Operator und die APL-System-Funktionen und -Variablen.

2. Das System NUMETH

Das Teilsystem NUMETH (NUMerische METHoden) ist ein Workspace von APL-Funktionen für numerische Anwendungen.

Zur Unterscheidung von Funktionen im Sinne der Mathematik wird im folgenden für 'APL-Funktionen' der Begriff 'Prozeduren' verwendet.

2.1 Aufbau von NUMETH

Ziel des Systems NUMETH ist es, APL in konsistenter Weise um numerische Komponenten zu erweitern, aber diese durch eine einfache Kommandosprache auch ohne detaillierte Kenntnisse in APL verfügbar zu machen.

Das System NUMETH enthält 3 Gruppen von Prozeduren:

Haupt-Prozeduren,
Deskriptoren,
Hilfs- und Service-Prozeduren.

Unabhängig vom Gesamtsystem ist NUMETH ein eigenständiges Teilsystem.

Die Haupt-Prozeduren decken jeweils einen Bereich der Numerik ab. Ohne Zusatzinformation wird ein Standard-Verfahren angewandt. Mit Hilfe der Deskriptoren können weitere Informationen angegeben oder spezielle Verfahren ausgewählt werden.

Die Hilfs- und Service-Prozeduren ermöglichen eine präzise Information und Unterstützung des Benutzers.

2.2 Die Haupt-Prozeduren

Die Haupt-Prozeduren entsprechen in etwa dem Kern einer Methodenbank. Sie liefern ein explizites Ergebnis, das durch APL- oder NUMETH-Funktionen weiterverarbeitet werden kann.

In Analogie zum DOMINO-Operator können sie auf eine ganze Klasse von numerischen Problemen angewendet werden.

Jede Haupt-Prozedur enthält eine Reihe von Verfahren des jeweiligen Bereichs der Numerik. Hierbei wurde besonderer Wert auf Robustheit der Verfahren gelegt. Die Eingabedaten werden sorgfältig geprüft. Bei falschen oder ungeeigneten Daten wird keine Lösung, sondern eine ausführliche Fehlermeldung geliefert. Auch während der Ausführung wird die Durchführbarkeit der Verfahren kontrolliert und gegebenenfalls wird der Grund für den Abbruch ausgegeben.

Folgende Bereiche der Numerik sind abgedeckt:

A GLYST B	Lösen des linearen Gleichungs-Systems $Ax = B$,
A EWN	Berechnen von N Eigenwerten der Matrix A,
F APPROX X	Approximation der Funktion F über dem Intervall X,
Y INTERPOL X	Interpolation der Punkte (X, Y),
F INTEGRAL X	Integration der Funktion F über dem Intervall X,
F IVP X	Anfangswertproblem $Y'(T) = F(T, Y)$ im Intervall X.

Außerdem steht noch zur Verfügung:

Y PLOT X	Plot der Punkte (X, Y).
----------	-------------------------

APL-Funktionen haben höchstens 2 Argumente und das ist für viele Bereiche der Numerik ausreichend. Aber bereits bei der Lösung des Anfangswertproblems muß zusätzlich der Anfangswert übergeben werden. Auch für die Auswahl spezieller Varianten der Verfahren oder für die Beschreibung besonderer Eigenschaften der Argumente ist Zusatzinformation erforderlich.

Die zusätzlichen Angaben werden in Form von globalen Variablen an die Haupt-Prozeduren übergeben. Das Setzen dieser globalen Variablen erfolgt mit Hilfe der Deskriptoren.

Dabei stellt sich das Problem, einerseits die übliche APL-Syntax beizubehalten, andererseits diese Prozesse durch formal von APL unabhängige Kommandos zu beschreiben.

2.3 Die Deskriptoren

Zu jeder der Haupt-Prozeduren gibt es eine Reihe von Unter-Prozeduren, sog. 'Deskriptoren'.

Diese dienen zur näheren Beschreibung des Verfahrens oder spezieller Eigenschaften der Argumente.

Der Standard-Aufruf: A PROZEDUR B

wird erweitert zu: A PROZEDUR B_n DESKR_n ... B₁ DESKR₁ B₀.

Diese Deskriptoren DESKR_i werden vor Ausführung der Haupt-Prozedur PROZEDUR abgearbeitet. Sie setzen globale Variablen zur Steuerung des Ablaufs der Haupt-Prozedur oder transformieren das Argument B₀ in das Argument B.

Aus der Sicht von APL wird eine Folge von APL-Funktionen DESKR_i ausgeführt. Das explizite Ergebnis von DESKR_i ist rechtes Argument von DESKR_{i+1}. Als letzte Funktion wird die Haupt-Prozedur PROZEDUR mit dem expliziten Ergebnis von DESKR_n als rechtes Argument aufgerufen. In PROZEDUR werden alle zulässigen globalen Variablen geprüft, was mit $\square$ NC sehr einfach zu programmieren war.

Das Konzept der Deskriptoren wirkt für den Laien wie eine Erweiterung des Standard-Kommandos durch problembezogene Schlüsselwörter. Für den Kenner sind die Deskriptoren eine konsequente Beibehaltung des Konzeptes der APL-Funktionen.

Die Deskriptoren haben folgende wichtigen Eigenschaften:

- sie sind voneinander unabhängig,
- sie sind beliebig vertauschbar,
- sie können vom Benutzer abgefragt werden,
- nach Ablauf der Haupt-Prozedur werden die globalen Variablen standardmäßig gelöscht,
- der Benutzer kann sie aber auf Wunsch speichern und sich so seine eigene Default-Umgebung definieren.

Zu jeder Haupt-Prozedur gibt es zwischen 5 und 15 zulässige Deskriptoren.

Für die verschiedenen Haupt-Prozeduren haben Deskriptoren mit der gleichen Bedeutung denselben Namen. So wird z.B. die gewünschte Genauigkeit durch e PREC beschrieben. Ist $e < 1$, so ist e die Fehlerschranke, für $e \geq 1$ bedeutet e die Anzahl der richtigen Stellen. Als Fehlerschranke wird dann 0.5×10^{-e} gesetzt. Der Standard-Wert ist OCT.

Die häufig nötige Beschreibung von Intervall-Zerlegungen erfolgt mit den Deskriptoren STEP bzw. POINTS.

H STEP X beschreibt die Ausschöpfung des Intervalls X durch Teilintervalle der Länge H. Ist H ein Vektor von Schrittweiten, so ist auch eine nicht-äquidistante Zerlegung möglich.

N POINTS X definiert eine äquidistante Zerlegung des Intervalls X mit N Teilpunkten.

2.4 Hilfs- und Service-Prozeduren

Zur Unterstützung des Benutzers enthält NUMETH eine Reihe von zusätzlichen Funktionen und Prozeduren.

a) Information

Strukturierte Information über das System NUMETH erhält der Benutzer durch die Prozedur DESCRIBE. Ebenso ist die Beschreibung einzelner Komponenten von NUMETH möglich.

Einzelne Objekte werden erklärt durch EXPLAIN 'name'.

Jede Haupt-Prozedur in NUMETH erklärt sich selbst, wenn ihre Argumente durch '?' ersetzt werden. Bei den Deskriptoren erhält man so den aktuellen Wert der gespeicherten globalen Variablen. STORE '?' liefert eine Liste aller gespeicherten Deskriptoren.

Eine alphabetische Liste aller NUMETH-Namen erhält man mit NAMES (mit Hilfe von QNL 3).

Beispiele erhält man durch EXAMPLE 'name'.

Die Prozedur ZEIT liefert die seit LOGON oder dem letzten Aufruf von ZEIT verbrauchte CPU-Zeit (mit Hilfe von QAI).

b) Funktionen

Zur Behandlung von Funktionen (im Sinne einer mathematischen Funktion) stehen verschiedene Hilfs-Prozeduren zur Verfügung. Außerdem ist ein Funktionen-Kalkül für beliebige Funktionen mit den Operationen $+$ $-$ $\times$ vorgesehen.

Mathematische Funktionen $F(X)$ lassen sich mit Hilfe des EXECUTE-Operators beschreiben durch: $\$F, 'X'$. Dabei ist F eine Zeichenkette, die eine definierte APL-Funktion enthält oder direkt den zu berechnenden Ausdruck beschreibt, etwa von der Form:

$'1 + X \times X \leftarrow'$.

c) NUMETH-interne Funktionen

Für die häufig auftretenden Funktionen wie Polynome (entwickelt nach Potenzen oder in der Newton'schen Darstellung), Chebyshev- und trigonometrische Polynome sowie Spline-Funktionen ist in NUMETH eine besondere Darstellung vorgesehen, z.B. für das Polynom in X mit Koeffizienten-Vektor C : $P \leftarrow C \text{ POLY } X$.

Die entsprechende APL-Darstellung $P \leftarrow X \downarrow \Phi C$ ist für Laien weniger geeignet.

Mit Hilfe der Prozedur POLTRAN können die verschiedenen Darstellungen der Polynome in einander übergeführt werden.

Den Koeffizienten-Vektor bzw. den Namen einer Funktion F erhält man mit den Prozeduren COEFF F bzw. FNAME F .

Die Integration über solche Funktionen kann analytisch ausgeführt werden. Wird als Integrationsbereich der leere Vektor angegeben, so wird die Stammfunktion bestimmt.

Diese NUMETH-internen Funktionen werden bei der Approximation und Interpolation von diskreten Punkten erzeugt. Sie erhalten eindeutige Koeffizienten-Namen (mit Hilfe von $\square$ TS), was eine Katalogisierung dieser Größen ermöglicht.

2.5 Anforderungen an den Benutzer

Zum Arbeiten mit NUMETH sind zunächst nur wenige grundlegende Kenntnisse in APL nötig, z.B. Workspace-Handling und einfache Funktionen und Operatoren.

Über die Objekte und Möglichkeiten in NUMETH kann sich der Benutzer stufenweise informieren.

Für den versierten Benutzer sind auch komplexe Anwendungen und das Definieren eigener Objekte möglich.

Zwischenergebnisse können beliebig weiterverarbeitet werden. Durch das Speichern von Deskriptoren kann sich der Benutzer anwendungsspezifische Umgebungen schaffen.

2.6 Hinweise zur Implementierung

Bei dem zugrundeliegenden APL-System handelt es sich um VSAPL. Der Workspace NUMETH benutzt außer dem Stack-Prozessor keine speziellen Hilfsprozessoren, so daß weitgehend Portabilität gewährleistet ist. Der Stack-Prozessor wird beim Auftreten von APL-Systemfehlern zum ordnungsgemäßen Verlassen der entsprechenden Haupt-Prozedur benötigt.

NUMETH umfaßt ca. 240 APL-Funktionen. Davon sind etwa 100 für den Benutzer verfügbar, die übrigen Funktionen werden für NUMETH-interne Zwecke benötigt.

Durch Namenskonventionen wird eine einfache Trennung erreicht: die internen Namen beginnen mit A, B, ... , Z, die globalen Variablen mit 'Δ' bzw. 'Δ'.
(Note: The original text has a typo 'A' in the original image, which has been corrected to 'Δ' based on context and the subsequent line.)

Vom Benutzer definierte Objekte dürfen weder 'Δ' noch '_' enthalten.

Der Gesamtumfang beträgt ca. 200 KB.

Standard-Werte:	$\square$ IO $\leftarrow$ 1	index origin
	$\square$ CT $\leftarrow$ 5E-17	comparison tolerance
	$\square$ PP $\leftarrow$ 10	printing precision
	$\square$ PW $\leftarrow$ 79	printing width

3. Numerische Methoden

Bei den einzelnen Haupt-Prozeduren wurden moderne Standard-Verfahren ausgewählt. Zusätzlich sind jeweils noch eine Reihe von weiteren Verfahren, z.B. für spezielle Form der Daten oder bestimmte numerische Effekte vorgesehen.

Ohne Numerik-Vorkenntnisse wird ein robustes Standard-Verfahren angeboten, für den Numeriker aber stehen eine Fülle von Verfahren zur Verfügung.

Das Numerik-Paket für die 5100 von IBM erfüllte in mehrfacher Hinsicht nicht unsere Vorstellungen, so ist z.B. die Dialog-geführte Eingabe von größeren Matrizen für den Experten unzumutbar, auch eine direkte Weiterverarbeitung von expliziten Zwischenergebnissen ist nicht vorgesehen.

3.1 Die Verfahren der Haupt-Prozeduren

3.1.1 Lineare Gleichungs-Systeme

A GLSYST B

berechnet die Lösung X eines linearen Gleichungssystems

$$AX = B$$

Standard-Verfahren:

Dreiecks-Zerlegung mit Spalten-Pivotsuche und Skalierung

bei überbestimmten Systemen: Householder-Transformation

für iterative Lösung:

SOR-Verfahren mit automatischer Bestimmung des Relaxations-Parameters.

Aufruf: A GLSYST B

A NxM - Koeffizienten-Matrix mit $N \geq M$

B N-Vektor der rechten Seite oder NxK-Matrix für K rechte Seiten oder leerer Vektor

Ergebnis: N-Vektor der Lösung X oder
NxK-Matrix für K rechte Seiten oder
Zerlegung von A, falls B = leerer Vektor.

Zulässige Deskriptoren:

CHOL	Dreieckszerlegung mit Cholesky-Verfahren
DET	die Determinante wird berechnet und kann mit DET '?' abgefragt werden
DOMINO	Lösung des Gleichungssystems mit APL-Operator
ESV	Einzelschritt-Verfahren (= Gauss-Seidel-Verfahren)
GSV	Gesamtschritt-Verfahren (= Gauss-Jacobi-Verfahren)
HHTRANS	Householder-Transformation
N ITER	N Maximal-Zahl der Iterationen kann mit '?' ITER '?' abgefragt werden
NOSKAL	Matrix wird <u>nicht</u> skaliert

	NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
E	PREC	$E < 1$: relative Genauigkeit (Standardwerte: $E = 10^{-6}$) $E \geq 1$: Anzahl der richtigen Stellen
	SKAL	Matrix wird skaliert (Standard) löscht NOSKAL
P	SOR	Sucsessive-Over-(Under-) Relaxation mit Parameter P
S	START	S Startwert (oder -Vektor) für Iteration
	STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
	SYM	Matrix ist symmetrisch
	TRID	Matrix ist Tridiagonal, Eingabe als volle Matrix oder als Vektor = Nebendiagonale, Hauptdiagonale, Nebendiagonale

Beispiele für 'GLSYST'

1) Direkte Verfahren:

```

A ← A + 3  3 4  1 0  1 4  1 0  1 4
4  1  0
1  4  1
0  1  4
B ← A + x1  2  3
2  4  10

```

Standard-Verfahren:

Kompakter Gauss-Algorithmus mit Spalten-Pivotsuche und Skalierung.

Aufruf: $X \leftarrow A$ GLSYST SYM B

Ergebnis: 1 2 3

A ist tridiagonal, also ist Eingabe als Vektor möglich (Deskriptor TRID)

```

C ← 1  1  4  4  4  1  1
Aufruf: X ← C GLSYST TRID B
Ergebnis: 1 2 3

```

Die Zerlegung von $A = LxU$, z. B. beim Gauss-Algorithmus, erhält man, falls $B \leftarrow 10$

Aufruf: A GLSYST 10

Ergebnis: 4 1 0
 -0.25 3.75 -1
 0 -0.26666667 3.73333333

2) Iterative Verfahren:

Das Gleichungs-System wird iterativ gelöst, falls einer die folgenden Deskriptoren angegeben ist:

ITER, PREC, START, SOR, ESV, GSV.

Standard-Verfahren:

SOR mit automatischer Bestimmung des Relaxations-Parameters.

Es sollen höchstens 30 Iterations-Schritte durchgeführt werden (Deskriptor 30 ITER).

Aufruf: $X \leftarrow A$ GLSYST 30 ITER B

Ergebnis: 1 2 3

Die Anzahl der tatsächlich benötigten Iterationen erhält man mit:

Aufruf: '?' ITER '?'

Ergebnis: AKTUELLER WERT VON ITER IST 14.

Das Ergebnis soll auf 10 Stellen genau sein (Deskriptor 10 PREC) und als Start-Vektor soll der Vektor $V \leftarrow 1 \ 1 \ 1$ gewählt werden (Deskriptor V START):

Aufruf: $X \leftarrow A$ GLSYST 1 1 1 START 10 PREC B

Ergebnis: 1 2 3

3.1.2 Algebraisches Eigenwert-Problem

A EW N

berechnet N Eigenwerte (und Eigenvektoren) einer beliebigen reellen MxM-Matrix A.

Standard-Verfahren: Reduktion auf Hessenberg-Gestalt mit Householder-Transformationen und QR-Algorithmus mit Shift-Technik.

Für symmetrische Matrizen A kann mit 'MISES', 'WIELANDT' bzw. 'INVERS' der betragsmäßig größte, kleinste bzw. einer vorgegebenen Zahl am nächsten gelegene EW (und zugehörigen EV) berechnet werden.

Aufruf: A EW N

A MxM-Matrix

N Index (-Vektor) der gewünschten EW

$N \leftarrow 0$: Berechnung aller EW

$N \leftarrow 1$: Berechnung des größten EW

$N \leftarrow M$: Berechnung des kleinsten EW

$N \leftarrow M$: Berechnung der entsprechenden EW

Ergebnis: Reelle EW: Vektor der gewünschten EW
komplexe EW: 2-zeilige Matrix der EW
2-te Zeile enthält die Imaginär-Teile

Komplexe EV:

Treten stets paarweise auf. Die beiden Vektoren enthalten Real- und Imaginärteil. Die komplexen EV lauten:
Real + (-) Imaginärteil.

Zulässige Deskriptoren:

EV	Die Rückgabematrix enthält von der 3-ten bis letzten Zeile spaltenweise die Eigenvektoren
HESS	Eingabematrix in <u>oberer</u> Hessenberg-Form
Z INVERS	Berechnung des am nächsten bei Z gelegenen EW mit der inversen Potenz-Methode
N ITER	N Maximal-Zahl der Iterationen kann mit '?', ITER '?' abgefragt werden
MISES	Berechnung des betragsgrößten EW mit der Potenz-Methode (Vektor-Iteration=Mises-Verfahren)
NOSKAL	Matrix wird <u>nicht</u> skaliert
NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
E PREC	$E < 1$: Relative Genauigkeit (Standardwert: $E = 10^{-6}$) $E \geq 1$: Anzahl der richtigen Stellen
SKAL	Matrix wird skaliert (Standard)
STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
SYM	Matrix ist symmetrisch
WIELANDT	Berechnung des betragskleinsten EW mit der inversen Potenz-Methode (=Wielandt-Verfahren)

Beispiele für 'EW'

1) Standard-Anwendung:

Berechnung aller EW einer Matrix A

$$A = \begin{pmatrix} 3 & 17 & -5 & -8 & -5 \\ 17 & -5 & -8 & & \\ -5 & 17 & 8 & & \\ -8 & 8 & 14 & & \end{pmatrix}$$

Aufruf: $E \leftarrow A$ EW 23

Ergebnis: 30 12 6

Die Matrix A hat also die EW: 30 12 6

Die Imaginärteile sind jeweils 0.

Werden nicht alle EW benötigt, z.B. nur der erste und letzte, so lautet der Aufruf:

Aufruf: $E \leftarrow A$ EW 1 3

Ergebnis: 30 6

Die gewünschte Genauigkeit und die Maximalzahl von Iterationen kann mit den Deskriptoren PREC und ITER angegeben werden:

Aufruf: E←A EW 5 PREC 10 ITER 13
Ergebnis: 30 12.0000041 5.999995898

2) Berechnung aller EW und EV:

Sollen außerdem die zugehörigen Eigenvektoren berechnet werden, so wird dies mit Hilfe des Deskriptors EV angegeben:

Aufruf: E←A EW EV 13
Ergebnis: 30 12 6
0 0 0
1 1 0.5
-1 1 0.5
1 0 1

Die ersten beiden Zeilen enthalten Real- und Imaginärteil der EW. Die restlichen 3 Zeilen enthalten spaltenweise die Eigenvektoren. Der zum EW 12 gehörige Eigenvektor lautet also:

1
1
0

3) Berechnung einzelner EW:

Für eine symmetrische Matrix A ist auch die Berechnung einzelner EW (und zugehöriger EV möglich).

Dazu dienen die folgenden Deskriptoren:

MISES	größter EW
WIELANDT	kleinster EW
E INVERS	der am nächsten bei E liegende EW

Besonders wenn eine Näherung für einen EW bekannt ist, kann so Rechenzeit gespart werden.

Aufruf: E←A EW 5.99 INVERS 3
Ergebnis: 6

6 ist der am nächsten bei 5.99 liegende EW.

3.1.3 Approximation stetiger, reeller Funktionen

Y APPROX X

berechnet:

- Least-Square-Approximation von diskreten Kurven Y über dem Intervall X mit Chebyshev-Polynomen 'C CPOLY' (Standard)
- Beste (gleichmäßige) Approximation von Polynomen

Aufruf: Y APPROX X

Y Diskreter Vektor $Y_1, \dots, Y_N$ oder APL-Funktion
X Stütz-Stellen $X_1, \dots, X_N$ oder Intervall $[A, B]$

Ergebnis: 'C CPOLY'

Zulässige Deskriptoren

'F'	BASIS	F : Funktionen-System für Approximations-Funktion Ergebnis von APPROX: 'C + . xF'
	CHEB	Beste gleichmäßige Approximation (nur für Polynome) Ergebnis von APPROX: 'C CPOLY'
	DOMINO	Lösung des Gleichungssystems mit APL-Operator
N	GRAD	N = Grad des Approximations-Polynoms ($0 \leq N \leq 20$)
	NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
	STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
W	VALUE	W Startwert für 'IVP' bzw. Stützstelle für 'INTERPOL' oder 'APPROX'

Beispiel für 'APPROX'

1) Standard-Anwendung:

Man approximiere im quadratischen Mittel die Punkte (X_i, Y_i)
mit Chebyshev-Polynomen vom Grade N.
Sei $N = 5$. (Standard: $N \leq 20$)

Aufruf: $C \leftarrow Y \text{ APPROX } 5 \text{ GRAD } X$

Ergebnis: 'C..... CPOLY'
Chebyshev-Polynom mit Koeffizienten-Vektor C

Beispiel:

Approximation der Funktion $F(X) = \sqrt{1 + 2xX^2}$
über dem Intervall $[-1, 1]$

$Y \leftarrow \sqrt{F}, ' 5 \text{ POINTS } [-1 \ 1]$

Aufruf: $C \leftarrow Y \text{ APPROX } [-1 \ 1]$

Ergebnis: 'C..... CPOLY'

KOEFF C
0 0 1

Störung der Funktions-Werte:

$D \leftarrow (2101E^{-7})x(1)^{*}21$

$Y \leftarrow D + \sqrt{F}, ' 21 \text{ POINTS } [-1 \ 1]$

Aufruf: $C \leftarrow Y \text{ APPROX } [-1 \ 1]$

KOEFF C

$^{-9}....E^{-9} \quad ^{-4}....E^{-16} \ 0.9999999837$

- 2) Gleichmäßige Approximation eines Polynoms vom Grade N mit Koeffizienten D durch Chebyshev-Polynome vom Grade $\leq N-1$ über dem Bereich $[A,B]$.

Aufruf: 'D POLY' APPROX CHEB A,B

Ergebnis: 'C CPOLY'

Rücktransformation in POLY mit 'POLTRAN'

3.1.4 Interpolation stetiger, reeller Funktionen

Y INTERPOL X

berechnet eine Interpolations-Funktion zu den Punkten (X,Y)

Standard-Verfahren:

Interpolation mit kubischen Spline-Funktionen

Aufruf: Y INTERPOL X

Y Vektor der Stützwerte $Y_1, \dots, Y_N$ oder APL-Funktion

X Vektor der Stützstellen $X_1, \dots, X_N$ oder Intervall

Ergebnis: 'C SPLINE'

Eine Beschreibung von 'SPLINE' erhalten Sie mit

'?' SPLINE '?' oder EXPLAIN 'SPLINE'

Zulässige Deskriptoren

X	AITKEN	Berechnung des Interpolations-Polynoms an der Stelle X mit dem Verfahren von Aitken-Neville
	LAGRANGE	Polynom-Interpolation Ergebnis von INTERPOL: 'C POLY '
	NEWTON	Polynom-Interpolation (Newtonsche Darstellung) Ergebnis von INTERPOL: 'CNPOLY -(1↓X)'
	NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
N	POINTS	N Anzahl der Punkte der Zerlegung des Intervalles Beispiel: Aufruf: 5 POINTS 0 4 Ergebnis: 0 1 2 3 4
H	STEP	H Schrittweite der Zerlegung des Intervalles oder Vektor $h_1, \dots, h_N$ mit $h_i = x_{i+1} - x_i$ $i=1(1)N$ Beispiel: 1) Aufruf: 1 STEP 0 4 Ergebnis: 0 1 2 3 4 2) Aufruf: 1 3 STEP 0 4 Ergebnis: 0 1 4
	STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht

TRIGO Interpolation mit Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht

W VALUE W Startwert für 'IVP' bzw. Stützstelle für 'INTERPOL' oder 'APPROX'

Beispiele für 'INTERPOL'

1) Standard-Anwendung:

Interpolation der Funktion $F(X) = 1+3X$ zu den Stützstellen 0 1 2 (Intervall $[0,2]$ mit 3 äquidistanten Teilpunkten) mit kubischen Spline-Funktionen:

$F \leftarrow '1+3xX \leftarrow'$

Aufruf: $S \leftarrow F \text{ INTERPOL } 3 \text{ POINTS } 0,2$

Ergebnis: 'C SPLINE'

KOEFF S

0	1	3	0	0
1	4	3	0	0
2	7	0	0	0

1. Spalte: Stützstellen
2. Spalte: Funktions-Werte
3.-5. Spalte: Koeffizienten von $(X-X_i)*K$

Lokale Darstellung des Splines $S(X;F)$:

1. Intervall: $S(X;F) = 1+3x(X-0)+0x(X-0)^2+0x(X-0)^3$

2. Intervall: $S(X;F) = 4+3x(X-1)+0x(X-1)^2+0x(X-1)^3$

Der interpolierende Spline ist hier exakt.

Dasselbe Ergebnis erhält man mit diskreten Werten $Y_i = F(X_i)$:

$Y \leftarrow F, '3 \text{ POINTS } 0,2'$

Aufruf: $S \leftarrow Y \text{ INTERPOL } 0,2$

2) Newton'sches Interpolations-Polynom

Aufruf: $N \leftarrow F \text{ INTERPOL NEWTON } 3 \text{ POINTS } 0,2$

Ergebnis: 'C NPOLY -(1 ↓ X)*.-'

KOEFF N
1 3

'X' WERT N
0 1 2 (Stützstellen)

Das entspricht dem Newton'schen Polynom:

$1 + 3x(X-0) + 0x(X-0)x(X-1)$

Transformation auf Polynom-Form:

Aufruf: $P \leftarrow N \text{ POLTRAN 'POLY'}$

Ergebnis: 'C POLY'

KOEFF P
1 3

Unter numerischen Aspekten ('Auslöschung') ist die Newton'sche Form günstiger als die übliche Polynom-Darstellung.

3) Trigonometrische Interpolation der Funktion $F(X) = 1 + 2x\sin(X) + 3x\cos(X)$ über dem Intervall $[0, 2\pi]$:

```

F ← '1+(2x10X)+3x20X←'
Aufruf:  T ← F INTERPOL TRIGO 3 POINTS 0,02
Ergebnis: 'C ..... TPOLY'

KOEFF T
1 3      Koeffizienten von cos(IxX)
0 0      Koeffizienten von sin(IxX)

```

Das Ergebnis ist nicht exakt (zu wenig Punkte);
es ist die Interpolierende:
 $1x\cos(0xX) + 3x\cos(X) + 0x\sin(0xX) + 0x\sin(X)$

2 Punkte mehr:

```

Aufruf:  T ← F INTERPOL TRIGO 5 POINTS 0,02
Ergebnis: 'C ..... TPOLY'

KOEFF T
1 3
0 2

```

Die Interpolierende stimmt mit der Ausgangsfunktion überein.

3.1.5 Integration stetiger, reeller Funktionen

Y INTEGRAL X

berechnet näherungsweise das bestimmte Integral über eine stetige Funktion Y einer reellen Variablen in den Grenzen von A bis B.

Standard-Verfahren: Trapezregel mit Extrapolation
(Adaptives Romberg-Verfahren)

Aufruf: Y INTEGRAL X

Y Vektor der Stützwerte $Y_1, \dots, Y_N$ oder APL-Funktion

X Vektor der Stützstellen $X_1, \dots, X_N$ mit $A = X_1$ und $B = X_N$ oder Integrations-Bereich $[A, B]$

Falls $X \leftarrow 1.0$ und Y eine der APL-Funktionen 'POLY' oder 'TPOLY', wird die Stamm-Funktion berechnet.

Ergebnis: Wert des Integrals oder Stamm-Funktion

Zulässige Deskriptoren

AEQUI	Stützstellen äquidistant (Vektor wird nicht geprüft)
M EXTRA	M Anzahl der Extrapolationen (Standardwert: M=3)
FEHLER	Der Verfahrens-Fehler wird gespeichert und kann mit FEHLER '?' abgefragt werden
N FKTW	N Schranke für die Anzahl der Funktions-Auswertungen Die benötigte Anzahl von Auswertungen kann mit '?' FKTW '?' abgefragt werden (Standardwert: N=2*12)
N GAUSS	Gauss-Quadratur mit N Stützstellen (N=5, 9, 16, 40)

IKURVE	Integralkurve
NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
N POINTS	N Anzahl der Punkte der Zerlegung des Intervalles Beispiel: Aufruf: 5 POINTS 0 4 Ergebnis: 0 1 2 3 4
E PREC	E < 1: Relative Genauigkeit (Standardwert: E = 10 ⁻⁶ CT) E ≥ 1: Anzahl der richtigen Stellen
ROMBERG	Adaptives Romberg-Verfahren
SIMPSON	Simpson-Integration (mit 3/8 Regel)
H STEP	H Schrittweite der Zerlegung des Intervalles oder Vektor $h_1, \dots, h_N$ mit $h_i = x_{i+1} - x_i$ $i=1(1)N$ Beispiel: 1) Aufruf: 1 STEP 0 4 Ergebnis: 0 1 2 3 4 2) Aufruf: 1 3 STEP 0 4 Ergebnis: 0 1 4
STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
TRAPEZ	Trapez-Regel (Standard für diskrete Werte)
L WEIGHT	L Vektor der Gewichte der Quadraturformel

Beispiele für 'INTEGRAL'

1) Standard-Anwendung:

Integration der Funktion $F(X) = 3xX*5 + X$ über dem Intervall $[A,B] = 0,1$ auf 13 Stellen genau.

```
A ← 0
B ← 1
F ← '(3xX*5)+X' ←
```

```
Aufruf: F INTEGRAL A,B
Ergebnis: 1
```

2) Maximalzahl von Funktions-Auswertungen (hier 11)

```
Aufruf: F INTEGRAL 11 FKTW A,B
Ergebnis: 1.00008297
'FEHLERSCHRANKE NICHT ERREICHT'
```

3) Berechnung mit Gauss-Quadratur (5 Stützstellen)

```
Aufruf: F INTEGRAL 5 GAUSS A,B
Ergebnis: 1
```

Mit den Funktionen 'GAUSSX' bzw. 'GAUSSW' lassen sich die Stützstellen bzw. Gewichte auch explizit berechnen:

Y ← Φ F, ' 5 GAUSSX A,B'
W ← 5 GAUSSW A,B
Aufruf: Y INTEGRAL W WEIGHT A,B
Ergebnis: 1

- 4) Diskrete Funktionswerte $Y_i = F(X_i)$
Standard-Verfahren: Trapez-Regel

X ← 11 POINTS A,B
Y ← Φ F, ' X'
Aufruf: Y INTEGRAL A,B
Ergebnis: 1.012475

Mit Simpson-Formel:

Aufruf: Y INTEGRAL SIMPSON A,B
Ergebnis: 1.0001

- 5) Fehler-Meldungen:

Unterschiedlich lange Eingabe-Vektoren:
Aufruf: Y INTEGRAL 5 POINTS A,B

Widersprüchliche Angabe der Deskriptoren:
Aufruf: Y INTEGRAL 5 GAUSS A,B
oder:
Aufruf: F INTEGRAL 1E-3 PREC SIMPSON A,B

unzulässige Deskriptoren:

Aufruf: Y INTEGRAL 1E-3 PREC A,B

3.1.6 Anfangswert-Problem für gewöhnliche Differential-Gleichungen

F IVP YA VALUE X

berechnet eine Näherungs-Lösung für das Anfangswert-Problem:

$$Y'(T) = F(T, Y(T)) \text{ auf } (A, B)$$

$$Y(A) = YA$$

Standard-Verfahren: Midpoint-Regel mit Extrapolation (nach Gragg)
und Schrittweiten-Steuerung

Aufruf: F IVP YA VALUE X
F APL-Funktion in den Variablen T und Y(T) von der Form,
daß: $F(T, Y) \sim \Phi$ 'T', F, 'Y'
X Lösungs-Intervall [A,B] oder Zerlegung $A=X_0, \dots, X_N=B$
YA Anfangswert von Y an der Stelle A,
wird durch ... YA VALUE ... übergeben.

Ergebnis: Y(A), Y(B) oder $Y(X_0), \dots, Y(X_N)$

Zulässige Deskriptoren

EULER Euler-Verfahren (Polygonenzug-Verfahren)

M EXTRA M Anzahl der Extrapolationen (Standardwert: M=3)

	FEHLER	Der Verfahrens-Fehler wird gespeichert und kann mit Fehler '?' abgefragt werden
N	FKTW	N Schranke für die Anzahl der Funktions-Auswertungen die benötigte Anzahl von Auswertungen kann mit '?' FKTW '?' abgefragt werden (Standardwert:N=2*12)
	MIDPOINT	Midpoint-Verfahren (Stabilisiert nach Gragg) (Standard)
	NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
	NOSTSZ	keine Schrittweiten-Steuerung
E	PREC	$E < 1$: Relative Genauigkeit (Standardwert: $E = 10^{-6}$ CT) $E \geq 1$: Anzahl der richtigen Stellen
	RUNKUT	Runge-Kutta-Verfahren
H	STEP	H Schrittweite der Zerlegung des Intervalles oder Vektor $h_1, \dots, h_N$ mit $h_i = x_{i+1} - x_i$ $i=1(1)N$ Beispiel: 1) Aufruf: 1 STEP 0 4 Ergebnis: 0 1 2 3 4 2) Aufruf: 1 3 STEP 0 4 Ergebnis: 0 1 4
	STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
	STSZ	Schrittweiten-Steuerung (Standard) löscht NOSTSZ
W	VALUE	W Startwert für 'IVP' bzw. Stützstelle für 'INTERPOL' oder 'APPROX'

Beispiele für 'IVP'

1) Standard-Anwendung:

Man berechne die Lösung des AWP:

$$Y'(T) = -Y(T)$$

$$Y(0) = 1$$

an der Stelle $T = 1$

Die Darstellung der rechten Seite der Differential-Gleichung $F(T, Y(T)) = -Y(T)$ erfolgt in Form einer definierten APL-Funktion:

```

      ▽ Z ← T F Y
[1] Z ← -Y ▽

```

Aufruf: $E \leftarrow 'F'$ IVP 1 VALUE 0 1

Ergebnis: 1 0.3678794412

Die exakte Lösung ist: $Y(T) = E^{*}(-T)$

2) Berechnung der Lösung an Zwischenpunkten:

Der Standard-Aufruf liefert nur die Lösung am rechten Intervall-Endpunkt. Soll die Lösung auch an Zwischenpunkten berechnet werden, so muß eine Zerlegung des Intervalls vorgegeben werden, z.B. mit Hilfe des Deskriptors POINTS:

Aufruf: $E \leftarrow 'F' \text{ IVP } 1 \text{ VALUE } 5 \text{ POINTS } 0 \ 1$
Ergebnis: $1 \ 0.7788.. \ 0.6065.. \ 0.4723.. \ 0.367..$

3) Systeme von Differential-Gleichungen:

Eine Differential-Gleichung höherer Ordnung läßt sich nach einer Standard-Methode in ein System 1. Ordnung überführen:

Sei $Y''(T) = 1.5 \times Y^2$
 $Y'(0) = -8$
 $Y(0) = 4$

Das entsprechende System lautet dann:

$$\begin{aligned} Y_1'(T) &= Y_2(T) \\ Y_2'(T) &= 1.5 \times Y_1(T)^2 \end{aligned}$$

Mit den Anfangswerten:

$$\begin{aligned} Y_1(0) &= 4 \\ Y_2(0) &= -8 \end{aligned}$$

Für die Darstellung ist wieder eine APL-Funktion erforderlich:

$$\begin{aligned} &\nabla \ Z \leftarrow T \ F \ Y \\ [1] \quad &Z \leftarrow 1 \downarrow Y \\ [2] \quad &Z \leftarrow Z, 1.5 \times (1 \uparrow Y)^2 \quad \nabla \end{aligned}$$

Aufruf: $E \leftarrow 'F' \text{ IVP } 4 \ -8 \text{ VALUE } 0 \ 1$
Ergebnis: $4 \ 1$
 $-8 \ -1$

Die Lösung der Gleichung 2. Ordnung an der Stelle 1 lautet also:

$$Y(1) = 1$$

Die exakte Lösung ist: $Y(T) = 4/(1+T)^2$

3.1.7 Print-Plot diskreter Kurven

Y PLOT X

erstellt den Print-Plot einer diskreten Kurve (X,Y)

Aufruf: Y PLOT X

Y Vektor der Stützwerte oder APL-Funktion

X Vektor der Stützstellen oder Plot-Bereich

Sind X und Y Vektoren mit mehr als 2 Elementen, so müssen sie gleich lang sein.

Die Anzahl der Punkte auf den Achsen wird dann durch diese Anzahl von Punkten festgelegt.

Eine Interpolation bei diskreten Punkten wird nicht durchgeführt!

Ist die Anzahl der Punkte in X-Richtung kleiner als $\square PW \div 2$, so werden Leerzeichen eingefügt.

Standard-Bildgröße: $\square PW \times 17$ Punkte, falls $\square PW < 80$
 $\square PW \times \square PW \div 2$, falls $\square PW \geq 80$
 Standard-Wert: $\square PW \leftarrow 79$
 Maximale Bildgröße: 132x132 Punkte

Zulässige Deskriptoren

DIALOG	Dialog bei Plot-Prozedur (löscht 'NODIALOG')
HISTO	Ausgabe der Kurve als Histogramm
NODIALOG	kein Dialog bei Plot-Prozedur
NOSTORE	Löschen aller gespeicherter Deskriptoren nach Beendigung dieser Prozedur (Standard)
N POINTS	N Anzahl der Punkte der Zerlegung des Intervalles Beispiel: Aufruf: 5 POINTS 0 4 Ergebnis: 0 1 2 3 4
(N,M) RASTER	(N,M) Anzahl der Punkte in (X,Y)-Richtung $2 \leq N, M \leq 132$, Standardwert: (N,M) = ($\square PW$, 17)
S SPACE	S Anzahl der eingefügten Blanks in X-Richtung
H STEP	H Schrittweite der Zerlegung des Intervalles oder Vektor $h_1, \dots, h_N$ mit $h_i = x_{i+1} - x_i$ $i=1(1)N$ Beispiel: 1) Aufruf: 1 STEP 0 4 Ergebnis: 0 1 2 3 4 2) Aufruf: 1 3 STEP 0 4 Ergebnis: 0 1 4
STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
S SYMBOL	S Symbol für Print-Plot (Standard: S='*')
(S,T) YBOUND	Skalierung für die Y-Werte $S \leq Y \leq T$, Standardwert: (S,T) = (YMIN, YMAX)

Beispiele für 'PLOT'

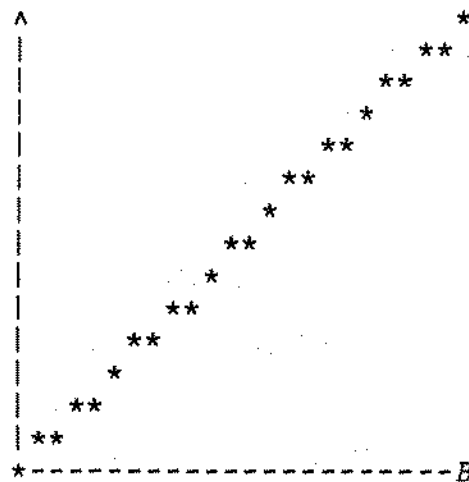
Zur Vereinfachung werden nur Beispiele in der nicht interaktiven Version von PLOT vorgeführt, Deskriptoren NODIALOG und STORE.

1) Darstellung einer Geraden mit Anstieg 1:

Aufgrund der unterschiedlichen Zeichengröße in X- und Y-Richtung wird eine Gerade mit Anstieg 1 als Winkelhalbierende dargestellt, falls die Anzahl der Punkte in X- und Y-Richtung im Verhältnis 8:5 stehen:

```
'X←' PLOT 24 15 RASTER NODIALOG STORE 0 1
```

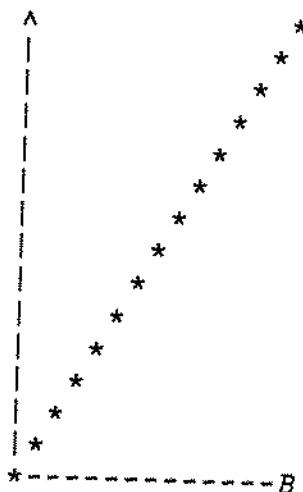
$[A,B] = [0,1]$ $YMAX = 1$



Wählt man in beiden Achsenrichtungen die gleiche Punktezahl, so ist zwar die Darstellung der Geraden deutlicher, sie ist aber nicht mehr die Winkelhalbierende:

```
'X←' PLOT 15 15 RASTER 0 1
```

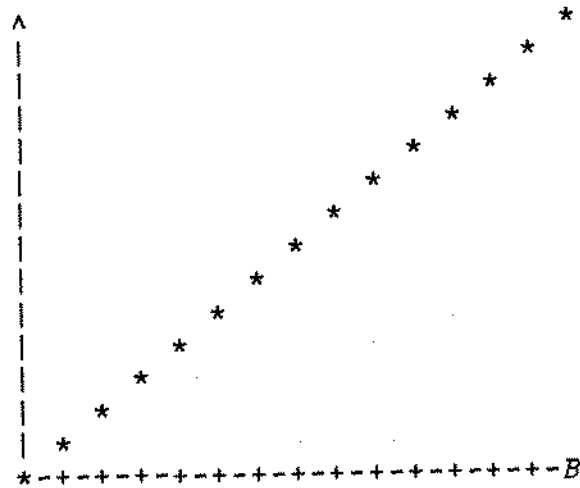
$[A,B] = [0,1]$ $YMIN = 0$ $YMAX = 1$



Durch Einfügen von je 1 Leerzeichen in X-Richtung kann dieser Effekt etwas verbessert werden (die eingefügten Zeichen werden auf der X-Achse durch '+' gekennzeichnet):

```
'X←' PLOT 1 SPACE 0 1
```

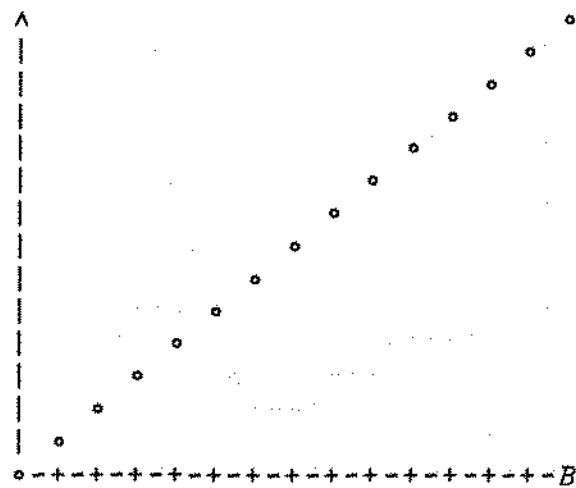
```
[A,B] = [0,1]      YMIN = 0      YMAX = 1
```



Die Wahl eines anderen Zeichen-Symbols als '*' erfolgt mit Hilfe des Deskriptors SYMBOL:

```
'X←' PLOT 'o' SYMBOL NOSTORE 0 1
```

```
[A,B] = [0,1]      YMIN = 0      YMAX = 1
```



2) Darstellung eines POLYNOMS 3. Grades:

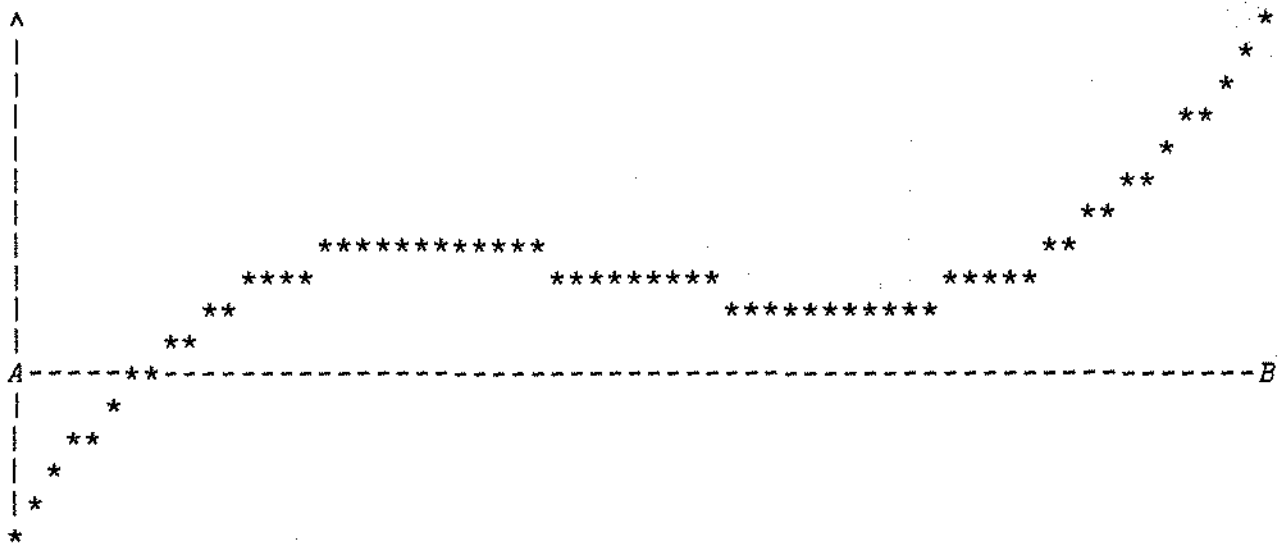
P←'3 12 -9 2 POLY'

Das Standard-Bild bei $\square$ PW←79 ergibt eine verzerrte Darstellung:

$\square$ PW←65

P PLOT NODIALOG STORE 0 3

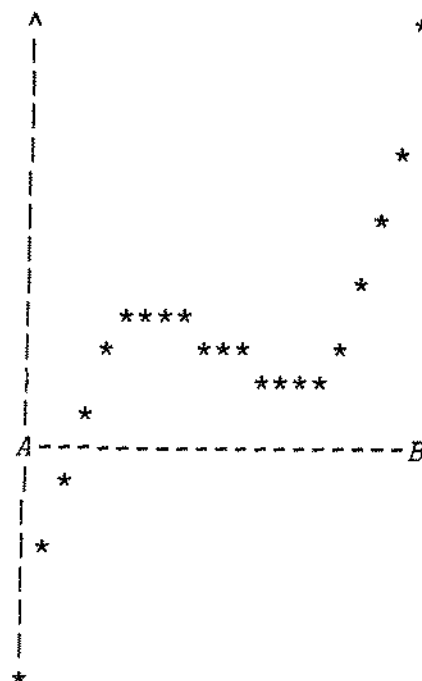
[A,B] = [0,3] YMIN = -3 YMAX = 6



Durch Wahl eines geeigneten Rasters ergibt sich eine günstigere Darstellung:

P PLOT 21 21 RASTER 0 3

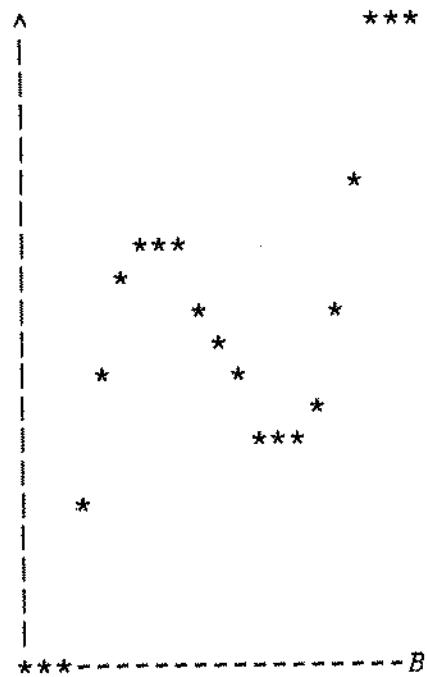
[A,B] = [0,3] YMIN = -3 YMAX = 6



Mit Hilfe des Deskriptors YBOUND lassen sich untere und obere Schranken für die Funktionswerte festlegen. Dadurch kann ein bestimmter Bildbereich vergrößert werden:

P PLOT 0 3 YBOUND 0 3

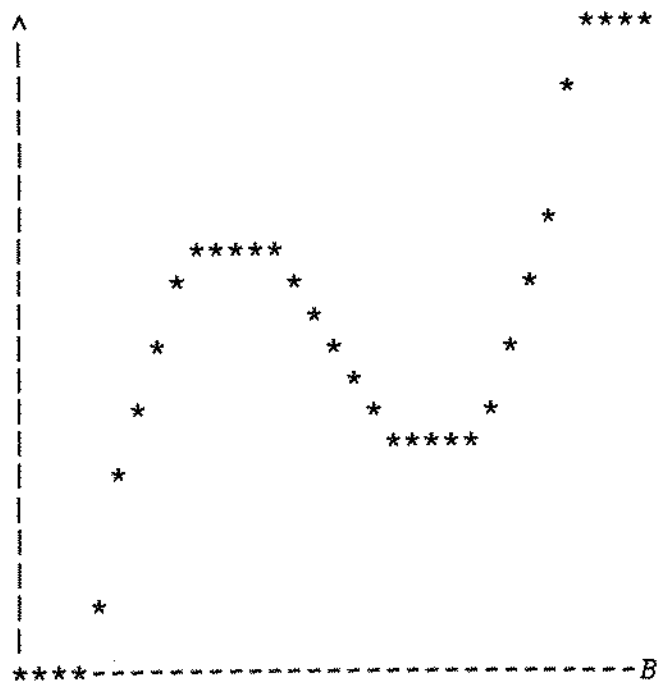
$[A,B] = [0,3]$ YMIN = 0 YMAX = 3



Durch Vergrößern der Punktzahl in X-Richtung wird die übliche Darstellung erreicht:

P PLOT 33 21 RASTER NOSTORE 0 3

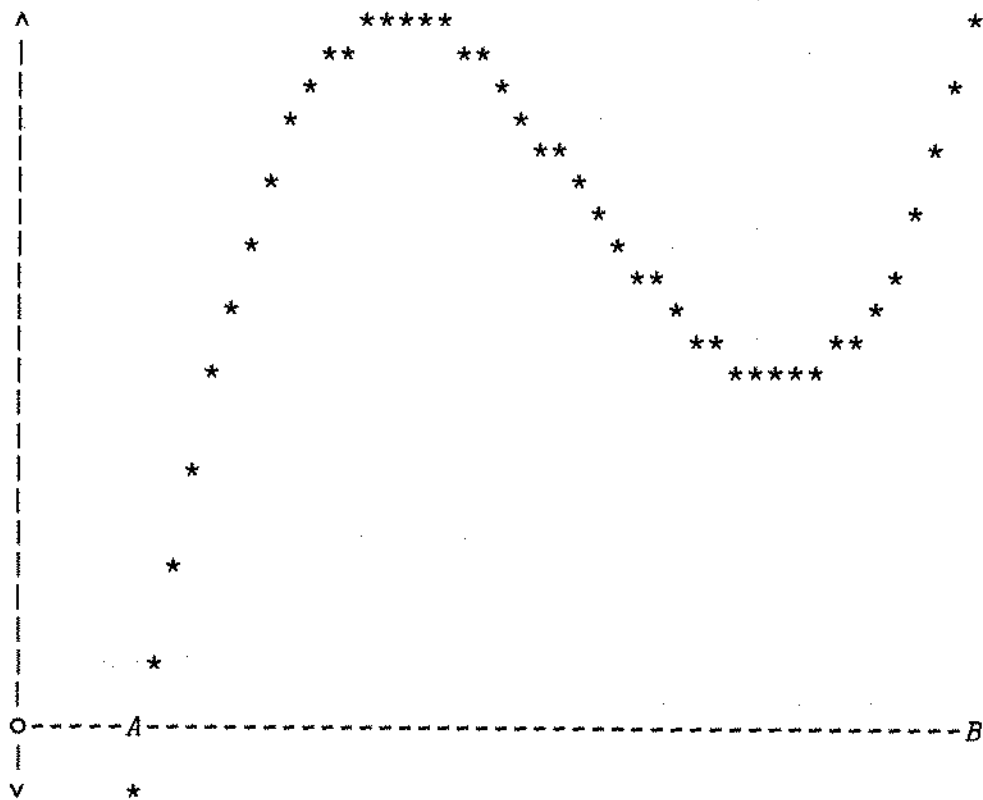
$[A,B] = [0,3]$ YMIN = 0 YMAX = 3



Noch deutlicher erhält man den Funktionsverlauf durch Wahl eines kleineren Intervalles in X-Richtung und mehr Punkte:

P PLOT 50 25 RASTER NODIALOG 0.3 2.5

[A,B] = [0.3,2.5] YMIN = -0.156 YMAX = 2



3) Histogramme:

Histogramme dienen zur Darstellung von Häufigkeits-Kurven.

Gegeben sei ein Vektor Y der Nennung über einem Bereich X:

$Y \leftarrow 1 \ 1 \ 2 \ 3 \ 4 \ 5 \ 3 \ 2 \ 1 \ 2 \ 3 \ 3 \ 4 \ 5 \ 5 \ 5 \ 3 \ 3 \ 4 \ 4 \ 2 \ 7 \ 9$

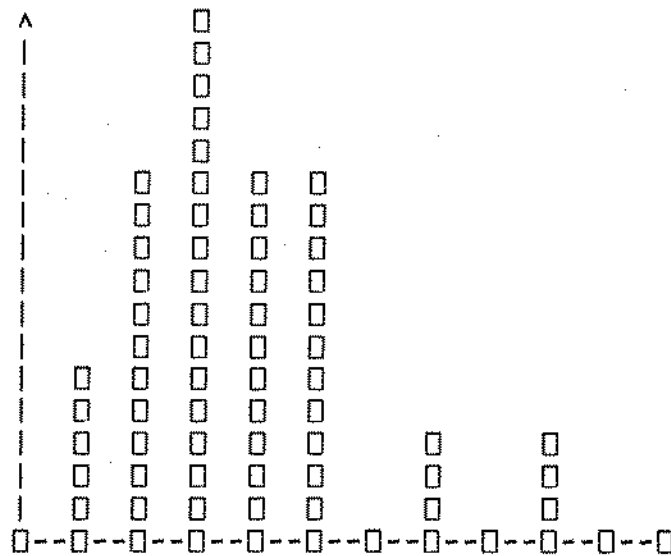
$X \leftarrow 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11$

Mit Hilfe der NUMETH-Funktion HISTOKURVE kann die Häufigkeits-Verteilung H von Y über X bestimmt werden:

$H \leftarrow Y \text{ HISTOKURVE } X$

$H \text{ PLOT HISTO NODIALOG } X$

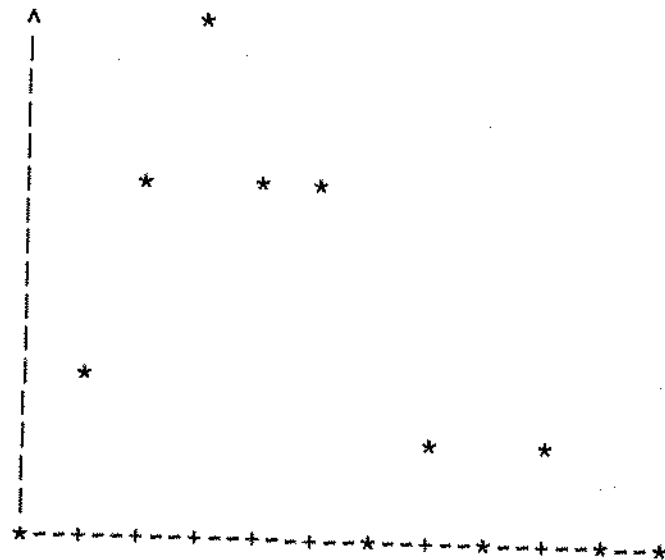
$[A,B] = [0,11] \quad YMIN = 0 \quad YMAX = 6$



Die übliche Darstellung diskreter Kurven ergibt der Aufruf:

H PLOT NODIALOG X

[A,B] = [0,11] YMIN = 0 YMAX = 6



3.2 Beispiele und Vergleiche

a) Lösen des linearen Gleichungs-Systems $Ax = B$

Standard-Aufruf: $X \leftarrow A \text{ GLSYST } B$

Falls A symmetrisch, wird das Gleichungs-System mit dem Verfahren von Cholesky gelöst, Deskriptor SYM

Aufruf: $X \leftarrow A \text{ GLSYST SYM } B$

Soll das System iterativ gelöst werden, so kann die Genauigkeit bzw. die Maximalzahl von Iterations-Schritten vorgegeben werden, Deskriptoren ITER und PREC :

Aufruf: $X \leftarrow A \text{ GLSYST } 30 \text{ ITER } 0.5E-7 \text{ PREC } B$
oder $X \leftarrow A \text{ GLSYST } 7 \text{ PREC } 30 \text{ ITER } B$

Diese Vertauschbarkeit der Deskriptoren erschien uns besonders wichtig, um Lernaufwand und Fehlermöglichkeiten nicht unnötig zu erhöhen und den Aufbau der Kommandos so einfach wie möglich zu gestalten.

Ein Vergleich der Lösung mit Hilfe des DOMINO-Operators ergab, daß dieser eine sehr effiziente und unter numerischen Gesichtspunkten sehr gute Implementierung verschiedener Algorithmen darstellt.

Wir haben ihn aber nicht als Standard-Verfahren gewählt, weil er abgesehen von APL-Systemfehlermeldungen, keine Fehler-Kontrolle während der Ausführung gestattet.

Die Ausführung von $X \leftarrow B \frac{1}{A}$ kann aber auch für den Laien mit Hilfe des Deskriptors DOMINO erreicht werden durch den Aufruf

$X \leftarrow A \text{ GLSYST DOMINO } B$

Ein Vergleich der benötigten Rechenzeit mit anderen Sprachen ist - besonders in einem Timesharing-System - nicht ganz unproblematisch. Es zeigt sich, daß für große Matrizen, etwa ab Rang 100, die reinen Ausführungszeiten für PL/I und FORTRAN um den Faktor 8 bis 10 kürzer sind, je nach Qualität der benutzten Routinen.

b) Weiterverarbeitung von Zwischenergebnissen

Die diskrete Lösung Y einer Anfangswertaufgabe soll an 21 Punkten des Intervalls $[0,1]$ auf 7 Stellen genau mit höchstens 2 Extrapolationen zum Anfangswert -1 berechnet werden, Deskriptoren POINTS, PREC, EXTRA und VALUE:

Aufruf: $Y \leftarrow F \text{ IVP } -1 \text{ VALUE } 2 \text{ EXTRA } 7 \text{ PREC } 21 \text{ POINTS } 0 \text{ } 1$

Die Werte Y sollen mit einem Polynom P interpoliert werden:

Aufruf: $P \leftarrow Y \text{ INTERPOL LAGRANGE } 0 \text{ } 1$

Das Interpolations-Polynom soll mit dem Symbol '*' und 41 Punkten auf der x-Achse, 21 Punkten auf der y-Achse geplottet werden, Deskriptoren SYMBOL und RASTER:

Aufruf: $P \leftarrow \text{PLOT } '*' \text{ SYMBOL } 41 \text{ } 21 \text{ RASTER } 0 \text{ } 1$

4. Hilfs- und Service-Prozeduren

4.1 Das Konzept der APL-Funktionen

1) Definition von APL-Funktionen

Neben den APL-Operatoren und Prozeduren sind APL-Funktionen im Sinne einer mathematischen Funktion von einer oder mehreren Variablen vorgesehen.

Der Aufbau und die Gestalt dieser Funktionen ist beliebig. Sie müssen nur der Regel

$$\underline{\phi} F, 'X' \sim F(X)$$

genügen.

Für den Aufbau dieser Funktionen bestehen etwa folgende Möglichkeiten:

- a) Die APL-Funktion F ist ein Charakter-String der Form:

$$F \leftarrow '1 \div 1 + T \leftarrow'$$

wichtig: letztes Zeichen ' $\leftarrow$ ' !

Sie ist äquivalent zu der mathematischen Funktion:

$$F(X) = 1/(1+X)$$

- b) Die APL-Funktion F ist ein Charakter-String, der wieder eine APL-Prozedur enthält:

$$F \leftarrow 'C \text{ POLY } '$$

Die äquivalente mathematische Funktion ist ein Polynom mit dem Koeffizienten-Vektor C

In beiden Fällen liefert: $\underline{\phi} F, 'X'$ den Wert der Funktion F an der Stelle X.

Mit Hilfe geeigneter APL-Operatoren lassen sich eine Reihe von mathematischen Funktionen sehr einfach beschreiben:

Beispiel:	$\sin (I \times \pi I \times X)$	$'10 I \times 0 \times \leftarrow'$
	$\exp (-X^2)$	$'* - X \times X \leftarrow'$
	$1 + X^2 + X^4$	$'1 + X + X \times X \leftarrow X \times X \leftarrow'$

In den Fällen, wo sich die mathematische Funktion nicht in der Form: $'...X \leftarrow'$ schreiben läßt, kann mit: $F(X), 0 \times X \leftarrow'$ zumeist die gewünschte Form erreicht werden:

Beispiel:	$\text{SQRT}(X)$	$'X * 0.5, 0 \times X \leftarrow'$
-----------	------------------	------------------------------------

VORSICHT:

Bei der Berechnung des Funktionswertes wird der Wert der formalen Variablen eventuell verändert!

Beispiel:	$X \leftarrow 5$
	$F \leftarrow '1 + X '$
	$W \leftarrow F, '2'$
	W erhält den Wert 3 und X hat danach den Wert 2, da der Ausdruck
	$'1 + X \leftarrow 2'$
	ausgewertet wird.

2) Verknüpfung von 2 APL-Funktionen

2 beliebige APL-Funktionen können durch eine der 3 Operationen:

+ - x

miteinander verknüpft werden. Dazu dienen die 3 Funktionen:

Plus +
Minus -
Mal x

Sind die zu verknüpfenden Funktionen vom gleichen Typ (z.B. Polynome), so ist das Ergebnis wieder von diesem Typ. Andernfalls wird ein geeigneter Charakter-String erzeugt.

Beispiel:

```
P1 ← 0 2 POLY'
P2 ← 1 0 3 POLY'
Aufruf: P1 PLUS P2
Ergebnis: '1 2 3 POLY'

Aufruf: P1 MAL P2
Ergebnis: '0 2 0 6 POLY'
```

4.2 5 NUMETH-interne Funktionen

Das System 'NUMETH' erkennt außerdem 5 Typen von Funktionen:

POLY, NPOLY, CPOLY, TPOLY und SPLINE

Diese Funktionen haben die Gestalt:

$F \leftarrow C \text{ POLY}'$ usw.

Wobei C der Koeffizienten-Vektor ist. Die Werte des Vektors erhält man mit:

KOEFF F

Von NUMETH selbst erzeugte Funktionen dieses Typs erhalten Koeffizienten-Namen: C..... . Diese Namen sind eindeutige globale Variablen. Bei Bedarf können diese Variablen mit:

$\square \text{EX 'C' } \square \text{NL 2}$ gelöscht werden.

Bei Verwendung dieser speziellen Funktionen innerhalb eines Charakter-Strings ist eine Klassifizierung nicht möglich. Obwohl die folgenden Ausdrücke formal äquivalent wären:

$'1 \div 0 0 1 \text{ POLY}' \sim 1/X^2$

Beschreibung dieser 5 Funktionen

1) Polynom 'POLY'

'C POLY': Polynom entwickelt nach Potenzen von X

C: Koeffizienten-Vektor mit C_i ist Koeffizient von $X^{(i-1)}$

Beispiel: 1 2 3 POLY X ergibt
 $1 \times X^0 + 2 \times X^1 + 3 \times X^2$

Definitions-Bereich: $(-\infty, +\infty)$

2) Newton-Polynom 'NPOLY'

'C NPOLY'-($\downarrow \underline{X0}$) $\circ$.-' : Polynom in der Newton'schen Darstellung zu den Stütz-Stellen $\underline{X0}_1, \dots, \underline{X0}_n$

C: Koeffizienten-Vektor mit C_i ist Koeffizient zu

$$(X - \underline{X0}_1) \times \dots \times (X - \underline{X0}_{i-1})$$

Beispiel:

$$\underline{X0} \leftarrow 1 \ 2 \ 3$$

$$C \leftarrow 4 \ 5 \ 6$$

$$C \text{ NPOLY } -(\downarrow \underline{X0})\circ \text{.-} X \text{ ergibt: } 4 + 5x(X-1) + 6x(X-1)x(X-2)$$

Definitions-Bereich: $[\underline{X0}_1, \underline{X0}_N]$

3) Chebyshev-Polynom 'CPOLY'

'CPOLY': Chebyshev-Polynom entwickelt nach Chebyshev-Polynomen

1. Art $T_I(X)$ mit:

$$T_1(X) = X$$

$$T_{N+1}(X) = 2X \times T_N(X) - T_{N-1}(X)$$

C: Koeffizienten-Vektor mit

C_i ist Koeffizient von $T_{i-1}(X)$.

Definitions-Bereich: $[-1, 1]$

Die Transformation von 'C POLY' in ein Polynom P, definiert auf dem Intervall $[A, B]$, erfolgt durch:

$$P \leftarrow C \text{ CPOLY' POLTRAN } A, B \text{ TRANS 'POLY'}$$

4) Trigonometrisches Polynom 'TPOLY'

'A TPOLY': Trigonometrisches Polynom entwickelt nach $\cos(IX)$ und $\sin(IX)$

A: $2 \times N$ Koeffizienten-Matrix mit

$A[1;I]$ Koeffizient von $\cos(I-1)X$

$A[2;I]$ Koeffizient von $\sin(I-1)X$

$$I = 1(1)N$$

Definitions-Bereich: $[0, 2 \times \pi]$

5) Kubischer Spline 'SPLINE'

'A SPLINE': Kubische Spline-Funktion

'Glatte' Entwicklung nach Stückweisen Polynomen vom Grade 3 zu den Interpolations-Punkten: (X_i, Y_i) , $i = 1(1)N$

A: $N \times 5$ Koeffizienten-Matrix mit

$$A[I;1] = X_i$$

$$A[I;2] = Y_i$$

$$A[I;K] = \text{Koeffizienten von } (X - X_i)^{K-2}$$

$$\text{für } K = 3, 4, 5 \text{ und } I = 1(1)N-1$$

Definitions-Bereich: $[X_1, X_N]$

4.3 Die Hilfsfunktionen

Das APL-Paket 'NUMETH' enthält eine Reihe von Hilfs- und Service-Routinen.

a) Behandlung von Zeichen-Ketten F:

KOEFF F

liefert den Wert des Koeffizienten (-Vektors) der in F enthaltenen Funktion.

Beispiel: $F \leftarrow '1\ 2\ POLY'$
Aufruf: KOEFF F
Ergebnis: 1 2

'Z' WERT F

liefert den Wert der 1-ten mit Z beginnenden Variablen in F.

Beispiel: $A \leftarrow 5$
 $F \leftarrow 'A'$
Aufruf: 'A' WERT F
Ergebnis: 5

FNAME F

liefert den Namen der in F enthaltenen Funktion (falls klassifizierbar)

Beispiel: $F \leftarrow '1\ 2\ POLY'$
Aufruf: FNAME F
Ergebnis: POLY

'A' TYP F:

untersucht, ob die in F enthaltene Funktion vom Typ 'A' ist. Ist A der leere Vektor oder besteht aus Leerzeilen, so erhält man den Funktionsname (gleiche Wirkung wie FNAME F).

Beispiel: 1) $F \leftarrow '1\ 2\ POLY'$
Aufruf: ' ' TYP F
Ergebnis: POLY
2) Aufruf: 'SPLINE' TYP F
Ergebnis: leerer Vektor und Fehler-Meldung
3) Aufruf: 'POLY' TYP F
Ergebnis: POLY

M STRING V

Umwandlung eines Vektors V aus beliebig langen Namen durch beliebige Sonderzeichen getrennt in eine Matrix M der Namen, alphabetisch sortiert und auf gleiche Länge gebracht.

b) Löschen von Variablen

ERASE

löscht nach fehlerhaftem Ende einer der Haupt-Prozeduren alle eventuell noch definierten internen und nicht mit 'STORE' gespeicherten Variablen. Sollen auch die gespeicherten Variablen gelöscht werden, so ist vor dem 'ERASE' ein 'NOSTORE 0' erforderlich.

c) Kontrolle der verbrauchten CPU-Zeit

ZEIT

liefert die seit 'LOGON' oder dem letzten Aufruf von 'ZEIT' verbrauchte CPU-Zeit in Sekunden.

d) Liste der definierten Funktionen

NAMEN

liefert Liste aller Funktionen, die nicht mit '_' oder 'Δ' beginnen.

e) Häufigkeits-Kurven ('Histogramme')

Y V HISTOKURVE X

erstellt die Häufigkeitskurve Y zu den Nennungen V über dem Bereich X

Beispiel:

```
V ← 1 2 3 2 1 2 1 2 3
X ← 0 1 2 3
Aufruf: Y ← V HISTOKURVE X
Ergebnis: 0 3 4 2
```

f) Transformation von Polynomen

Die Funktion 'POLTRAN' transformiert 'POLY', 'CPOLY', 'NPOLY' jeweils mit Koeffizienten-Vektor 'C'

```
POLY → CPOLY: Aufruf: 'C POLY' POLTRAN 'CPOLY'
Transformation des Def.-Bereichs [A,B] → [-1, 1]
Aufruf: 'C POLY' POLTRAN A, B TRANS 'CPOLY'
```

```
POLY → NPOLY: Aufruf: 'C POLY' POLTRAN X TRANS 'NPOLY'
X: Stützstellen für die Newton'sche Darstellung
Entwicklung eines Polynoms 'C POLY' um die Stelle X0:
Aufruf: 'C POLY' POLTRAN X0 TRANS 'POLY'
```

```
NPOLY → POLY: Aufruf: 'C NPOLY' POLTRAN 'POLY'
```

```
CPOLY → POLY: Aufruf: 'C CPOLY' POLTRAN 'POLY'
Transformation des Def.-Bereichs [-1, 1] → [A, B]
Aufruf: 'C CPOLY' POLTRAN A, B TRANS 'POLY'
```

g) Demonstrations-Beispiel

DEMO

liefert Demonstrations-Beispiel:

Vergleich von Spline- und Polynom-Interpolation

5. Erfahrungen und Ausblick

NUMETH ist bereits als selbständiges System für die Benutzer verfügbar. Dabei ergaben sich folgende Beurteilungen:

positiv:

- quasi kein Lernaufwand
- auch nach längerer Pause leichter Einstieg
- einfache Kommando-Struktur
- nur wenig Vorkenntnisse in Numerik erforderlich
- auch für Programmierlaien anwendbar
- auch umfangreiche Probleme lösbar
- für APL-Experten effizienter Zugriff auf Numerik-Paket

negativ:

- APL-Tastatur nötig
- kein Standard-Zugang zu bekannten Programm-Bibliotheken

Neben dem eigentlichen Ziel - die Auswertung von Experimentdaten - haben sich 2 weitere Anwendungsbereiche ergeben:

- Einsatz bei der Ausbildung der Mathematisch-Technischen Assistenten in Numerik
- Einfaches Vergleichen von numerischen Verfahren und Studieren ihrer Varianten

Viele Bereiche der Numerik sind noch nicht abgedeckt. Da NUMETH ein 'offenes' Dialogsystem ist, können sie aber bei Bedarf problemlos hinzugefügt werden.

Nach der Kopplung APL - PL/I, FORTRAN wäre der Aufruf von Programm-bibliotheken über NUMETH mit geeignetem Deskriptor möglich. Von der jeweiligen Haupt-Prozedur könnte der Aufruf der entsprechenden Bibliotheksprogramme vorbereitet werden.

Acknowledgement

Meinen Kolleginnen und Kollegen des Zentralinstituts für Angewandte Mathematik, insbesondere Herrn Homrigausen, dem Leiter der Gruppe 'Großrechner-Anwendungssysteme', sowie Herrn Wolkersdorfer und Herrn v. Seggern, möchte ich für ihr Interesse, ihre Mitwirkung und ihre Unterstützung ganz herzlich danken.

6. Literatur

Nicht alle Überlegungen - besonders im Bereich Numerik - , die bei NUMETH berücksichtigt wurden, lassen sich bibliographisch lokalisieren. Es folgt eine - unvollständige - Auswahl:

- /1/ Jenkins, M.A.: Numerical Analysis and APL, can they coexist?
APL 73, Kopenhagen, 237 - 243 (1973)
- /2/ Lattermann, D.: APL - A Tool for Personalized Computing
6th Informatik Symposium, Bad Homburg, 177 - 192 (1976)
- /3/ Macklin, D.: The APL Handbook of Techniques
IBM, (1978)
- /4/ Giloi, W.K.: Programmieren in APL
de Gruyter, Berlin (1977)
- /5/ Pakin, S.; Lochner, H.: APL - Handbuch
SRA, Stuttgart (1976) (2. Auflage)
- /6/ Sieber, D.: APL - Einführung
IBM Deutschland (1977)
- /7/ Mayforth, G.R.: APL as a Language for Application Programmers
Martin, G.A.: The Solution of Linear Systems in APL:
Towards an Extension of Matrix Divide
Erbe, R. et al.: Integrated Data Analysis and Management
Martin, B.R.: Concepts of Structure in APL
APL 80, Nordwijkerhout (1980)
- /8/ Perlis, A.J.; Rugaber, S.: Programming with Idioms in APL
APL Quote Quad 9, 232 - 235 (1979)
- /9/ Stoer, J.: Einführung in die Numerische Mathematik I
Springer, Heidelberg (1976) (2. Auflage)
- /10/ Stoer, J.; Bulirsch, R.: Einführung in die Numerische Mathematik II
Springer, Heidelberg (1978) (2. Auflage)
- /11/ IBM 5100 Math/APL User's Guide
SB 30-0064-0, IBM (1975)

Liste der zulässigen Deskriptoren

	AEQUI	Stützstellen äquidistant (Vektor wird nicht geprüft)
X	AITKEN	Berechnung des Interpolations-Polynoms an der Stelle X mit dem Verfahren Aitken-Neville
'F'	BASIS	F : Funktionen-System für Approximations-Funktion Ergebnis von APPROX: 'C+xF'

Beschreibung von 'BASIS'

Neben den Standard-Basis-Funktionen $F_i(X)$ kann ein beliebiges Funktionen-System $F_0(X), \dots, F_n(X)$ verwendet werden.

Es wird dann die Gauss-sche Normalen-Gleichung:

$$A'xAXC = A'xY$$

gelöst, mit $A[I;J] = F_{j-1}(X_i)$.

Als Lösung ergibt sich der Koeffizienten-Vektor C mit C_i ist Koeffizient von $F_{i-1}(X)$.

Hierfür wird mit dem Deskriptor 'BASIS' das Funktionen-System F

'F' BASIS ...

so angegeben, daß

$$F, 'X'$$

den Vektor $F_0(X), \dots, F_n(X)$ liefert.

Ergebnis: 'C+.xF'

Standard-mäßig stehen 3 Funktionen-Systeme zur Verfügung:

$$1) F_i(X) = X^i \quad i = 0(1)N$$

Aufruf: Y APPROX 'N XPOTENZ' BASIS X

$$2) F_i(X) = \sin(ix\pi X)$$

... 'N SINNP' BASIS ...

$$3) F_i(X) = \cos(ix\pi X)$$

... 'N COSNP' BASIS ...

Ein beliebiges Basis-System F mit den Basis-Funktionen $F_1, \dots, F_{10}$ könnte folgende Gestalt haben:

$$\begin{aligned} \nabla Z &\leftarrow F \quad X \\ [1] \quad Z &\leftarrow \underline{1} F_1, 'X' \\ &\vdots \\ [10] \quad Z &\leftarrow Z, \underline{1} F_{10}, 'X' \quad \nabla \end{aligned}$$

CHEB	Beste gleichmäßige Approximation (nur für Polynome) Ergebnis von APPROX: 'C CPOLY'
CHOLES	Dreieckszerlegung mit Cholesky-Verfahren
DET	Die Determinante wird berechnet und kann mit DET '?' abgefragt werden
DIALOG	Dialog bei Plot-Prozedur (löscht 'NODIALOG')
DOMINO	Lösung des Gleichungssystems mit APL-Operator
ESV	Einzelschritt-Verfahren (= Gauss-Seidel-Verfahren)
EULER	Euler-Verfahren (Polygonenzug-Verfahren)
EV	Die Rückgabematrix enthält von der 3-ten bis letzten Zeile spaltenweise die Eigenvektoren
M EXTRA	M Anzahl der Extrapolationen (Standardwert: M=3)
FEHLER	Der Verfahrensfehler wird gespeichert und kann mit FEHLER '?' abgefragt werden
N FKTW	N Schranke für die Anzahl der Funktions-Auswertungen die benötigte Anzahl von Auswertungen kann mit '?' FKTW '?' abgefragt werden (Standardwert: N=2*12)
N GAUSS	Gauss-Quadratur mit N Stützstellen (N=5,9,16,40)
N GRAD	N = Grad des Approximations-Polynoms ($0 \leq N \leq 20$)
GSV	Gesamtschritt-Verfahren (= Gauss-Jacobi-Verfahren)
HESS	Eingabematrix in <u>oberer</u> Hessenberg-Form
HHTRANS	Householder-Transformation
HISTO	Ausgabe der Kurve als Histogramm
IKURVE	Integralkurve
Z INVERS	Berechnung des am nächsten bei Z gelegenen EW-S mit der inversen Potenz-Methode
N ITER	N Maximal-Zahl der Iterationen kann mit '?' ITER '?' abgefragt werden
LAGRANGE	Polynom-Interpolation Ergebnis von INTERPOL: 'C POLY'
MIDPOINT	Midpoint-Verfahren (stabilisiert nach Gragg) (Standard)
MISES	Berechnung des betrags-größten EW mit der Potenz- Methode (=Vektor-Iteration=Mises-Verfahren)
NEWTON	Polynom-Interpolation (Newtonsche Darstellung) Ergebnis von INTERPOL: 'C NPOLY - (1↓ X.....)%, -'
NODIALOG	Kein Dialog bei Plot-Prozedur
NOSKAL	Matrix wird <u>nicht</u> skaliert
NOSTORE	Löschen aller gespeicherten Deskriptoren nach Beendigung dieser Prozedur (Standard)
NOSTSZ	Keine Schrittweisen-Steuerung
N POINTS	N Anzahl der Punkte der Zerlegung des Intervalles Beispiel: Aufruf 5 POINTS 0 4 Ergebnis: 0 1 2 3 4
E PREC	$E < 1$: Relative Genauigkeit (Standardwert: $E = \square CT$) $E \geq 1$: Anzahl der richtigen Stellen
(N,M) RASTER	(N,M) Anzahl der Punkte in (X,Y)-Richtung $2 \leq N, M \leq 132$, Standardwert: (N,M) = ($\square PW$, 17)
ROMBERG	Adaptives Romberg-Verfahren (Standard)
RUNKUT	Runge-Kutta-Verfahren
SIMPSON	Simpson-Integration (mit 3/8 Regel)
SKAL	Matrix wird skaliert (Standard) löscht NOSKAL

P	SOR	Sucsessive-Over-(Under-) Relaxation mit Parameter P
S	SPACE	S Anzahl der eingefügten Blanks in X-Richtung
S	START	S Startwert (oder -Vektor) für Iteration
H	STEP	H Schrittweite oder Zerlegung des Intervalles oder Vektor $h_1, \dots, h_N$ mit $h_i = x_{i+1} - x_i$ $i=1(1)N$ Beispiel: 1) Aufruf: 1 STEP 0 4 Ergebnis: 0 1 2 3 4 2) Aufruf: 1 3 STEP 0 4 Ergebnis: 0 1 4
	STORE	Speichern aller Deskriptoren für diesen und alle weiteren Aufrufe; wird durch 'NOSTORE' gelöscht
	STSZ	Schrittweiten-Steuerung (Standard) löscht NOSTSZ
	SYM	Matrix ist symmetrisch
S	SYMBOL	S Symbol für Print-Plot (Standard: S='*')
(A,B)	TRANS	Transformation von Polynomen auf Intervall [A,B]
	TRAPEZ	Trapez-Regel (Standard für diskrete Werte)
	TRID	Matrix ist tridiagonal, Eingabe als volle Matrix oder als Vektor = Nebendiagonale, Hauptdiagonale, Nebendiagonale
	TRIGO	Interpolation mit trigonometrischen Funktionen Ergebnis von INTERPOL: 'C TPOLY'
W	VALUE	W Startwerte für 'IVP' bzw. Stützstelle für 'INTERPOL' oder 'APPROX'
L	WEIGHT	L Vektor der Gewichte der Quadraturformel
	WIELANDT	Berechnung des betrags-kleinsten EW mit der inversen Potenz-Methode (=Wielandt-Verfahren)
(S,T)	YBOUND	Skalierung für die Y-Werte $S \leq Y \leq T$, Standardwert: (S,T) = (YMIN,YMAX)

NUMETH-Namen

AEQUI	GLSYST	POLTRAN
AITKEN	GRAD	POLY
ALL	GRID	PREC
ALLE	GSV	PROCEDURES
APPROX	HELP	PROZEDUREN
AUFBAU	HESS	RASTER
BASIS	HHTRANS	ROMBERG
CHEB	HILFSFUNKTIONEN	RUNKUT
CHOLES	HISTO	SCALE
COEFF	HISTOCURVE	SERVICEFUNCTIONS
COSNP	HISTOKURVE	SIMPSON
CPOLY	ICURVE	SINNP
DEGREE	IKURVE	SKAL
DEMO	INFO	SOR
DESCRIBE	INTEGRAL	SPACE
DESCRIPTORS	INTERPOL	SPLINE
DESIGN	INVERS	START
DESKRIPTOREN	ITER	STEP
DET	IVP	STORE
DIALOG	KOEFF	STRING
DOMINO	LAGRANGE	STSZ
EQUI	MAL	SYM
ERASE	MIDPOINT	SYMBOL
ERROR	MINUS	TIME
ESV	MISES	TPOLY
EULER	MITTEL	TRANS
EV	MULTIPLY	TRAPEZ
EW	NAMEN	TRID
EXAMPLE	NAMES	TRIGO
EXPLAIN	NEWTON	TYP
EXTRA	NODIALOG	TYPE
ECTV	NOSCALE	VALUE
FEHLER	NOSKAL	VARVALUE
FKTW	NOSTORE	WEIGHT
FNAME	NOSTSZ	WERT
FUNCTIONS	NPOLY	WIELANDT
FUNKTIONEN	NUMETH	XPOTENZ
GAUSS	PLOT	YBOUND
GAUSSW	PLUS	ZEIT
GAUSSX	POINTS	

ABFRAGE
ADAPT
ADAPTFEHL
AITK
ALL
ALLE
APLERR
APLOT
APPROX
AUFBAU
BALBAK
BALNC
BASIS
BAPPROX
BDESCRIPTORS
BDESKRIPTOREN
BEW
BGLSYST
BINTEGRAL
BINTERPOL
BIVP
BNUMETH
BPLOT
CDIV
CHOLESKY
CMULT
COEFF
COSNP
CPINT
CPOLY
DEMO
DESCRIBE
DESCRIPTORS
DESIGN
DESKR
DESKRIPTOREN
DIASUB
DIRVERF
DISKINT
DRUCK
ELMHES
ELTRAN
ERASE
EW
EXAMPLE
EXPLAIN
FINT

FNAM
FNAME
FSPLINE
FTYP
FUNCTIONS
FUNKEX
FUNKTIONEN
FUNKT131
FUNKT132
FUNKT133
GAUSSW
GAUSSX
GLSYST
GWGAUSS
HELP
HILFSFUNKTIONEN
HISTOCURVE
HISTOKURVE
HORNER
HOUSEHLD
HQR
HQR2
INFO
INTEGRAL
INTERPOL
INVR
ITERVERF
IVP
IVPSTART
IVPVERF
KFNORM
KNAME
KOEF
KOEFF
KOMPGAUSS
KOPF
KPLOT
LPLOT
LSTSQR
MAL
MINUS
MIS
MULTIPLY
NAME
NAMEN
NAMES
NEWT

NEXT
NEXTEV
NPINT
NPOLY
NUMETH
PLOT
PLUS
POINT
POLTRAN
POLY
POLYKOF
PLOT
PROCEDURES
PROZEDUREN
ROMINT
RUECKW
SERVICEFUNCTIONS
SINNP
SPINT
SPLINE
SPLOT
STACKOFFER
STAMM
STGAUSS
STRING
TIME
TPOLY
TRIDIAG
TRIG
TRISUM
TRNS
TYP
TYPE
VARNAME
VARVALUE
VERK
VEULER
VMIDP
VORW
VRUNK
WERT
WIELAN
XPLOT
XPOTENZ
YPLOT
ZEIT