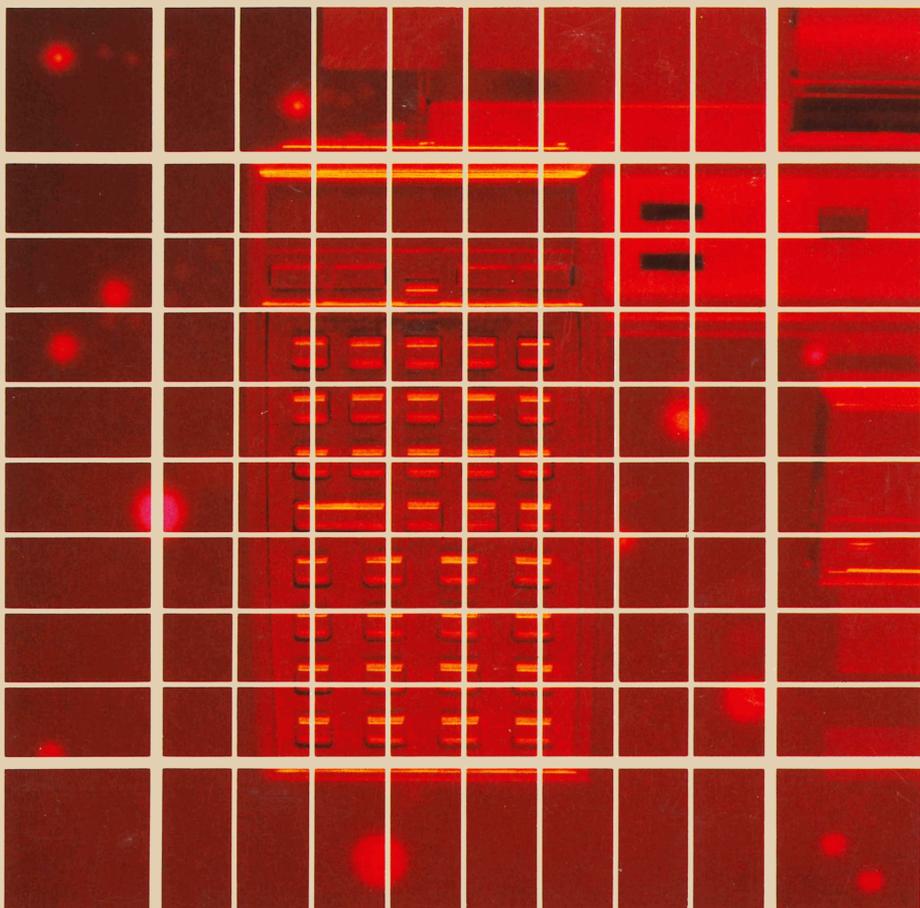


HEWLETT-PACKARD

HP-41

ADVANTAGE PAC



Tastenfolge	Bedeutung
$\Sigma+$	Tastefeld-Funktion. Drücken Sie $\Sigma+$.
$\Sigma-$	Umgeschaltete Tastefeld-Funktion. Drücken Sie nacheinander $\Sigma+$.
A (FX)	Programmspezifische Funktion eines jeweiligen Programms. Drücken Sie $\Sigma+$. (Dies entspricht der Taste mit dem Buchstaben "A".) In diesem Beispiel erscheint für A die Funktionsbezeichnung FX in der Anzeige.
XEQ TVM	TVM ist eine der Funktionen, die normalerweise nicht mit einer einzelnen Taste aufgerufen werden kann (Alpha-Funktion). Zum Ausführen dieser Funktion drücken Sie XEQ ALPHA TVM ALPHA oder weisen Sie diese Funktion einer User-Taste zu, um sie mit einer einzelnen Taste ausführen zu können.
XEQ SIZE 013 ABC	XEQ ALPHA SIZE ALPHA 0 1 3 Zeichen des Alpha-Tastefeldes, entsprechend den blauen Buchstaben auf den jeweiligen Tasten. Drücken Sie zu Beginn und am Ende ALPHA .
123	Zeichen des umgeschalteten Alpha-Tastefeldes. (Die Zuweisungstabelle befindet sich auf der Rückseite des Taschenrechners.)



HP-41
Advantage Pac

August 1985

00041-90562

Printed in Singapore

ANMERKUNG

Die in diesem Paket enthaltenen Matrix-Operationen stützen sich auf das CCD ROM, das von W & W Software Products GmbH, 5060 Bergisch Gladbach 2 entwickelt wurde.

Die Programme für die Bestimmung der Nullstellen einer Funktion bzw. eines Polynoms und für die numerische Integration wurden von den für den Taschenrechner HP-15C von Firmware Specialists Inc., Corvallis, Oregon 97330 USA geschriebenen Programme übernommen.

Das ursprüngliche Konzept über den Inhalt und das Benutzer-Interface wurde von Chris Bunsen, Corvallis, Oregon USA für dieses Paket aufgestellt.

HINWEIS

Hewlett-Packard übernimmt weder ausdrücklich noch stillschweigend irgendwelche Haftung für die in diesem Handbuch dargestellten Programm-Materialien—weder für ihre Funktionsfähigkeit noch für ihre Eignung für irgendeine spezielle Anwendung. Die Programm-Materialien dienen lediglich der beispielhaften Illustration; der Anwender trägt das volle Risiko bezüglich ihrer Güte und Durchführbarkeit. Sollten sich Programm-Materialien als fehlerhaft erweisen, trägt der Anwender (und nicht Hewlett-Packard oder irgendeine dritte Seite) die Kosten für Korrekturen und alle Neben- und Folgeschäden. Hewlett-Packard haftet für keinerlei Neben- und Folgeschäden im Zusammenhang mit oder als Folge aus der Lieferung, Benutzung oder Leistung der Programm-Materialien.

INHALTSVERZEICHNIS

Einleitung	5
Einstecken und Entfernen der Anwender-Module	7
Verwendung dieses Handbuchs	11
Das Matrix-Programm	19
Dieses Programm vereinfacht das Anlegen von reellen oder komplexen Matrizen. Es enthält Matrixinversion und -transposition sowie Determinantenberechnung und Lösung von Gleichungssystemen.	
Matrix-Operationen	30
Es sind über fünfundvierzig Funktionen zum Speichern und Verändern von Matrizen bzw. Teilen von Matrizen eingebaut, die über alle Fähigkeiten des Matrix-Programms, der Matrizenrechnung und der Suche nach Matrixelementen verfügen.	
Nullstellen einer Funktion	63
Mit diesem Programm können die reellen Nullstellen einer Gleichung $f(x) = 0$ bestimmt werden.	
Nullstellen und Funktionswerte eines Polynoms	73
Dieses Programm berechnet die reellen und komplexen Nullstellen von Polynomen mit reellen Koeffizienten vom zweiten bis zum fünften Grad.	
Numerische Integration	81
Dieses Programm berechnet das bestimmte Integral einer Gleichung $f(x) = 0$ in einem vorgegebenen Intervall.	
Differentialgleichungen	89
Dieses Programm löst Differentialgleichungen erster und zweiter Ordnung.	
Operationen mit komplexen Zahlen	95
Dies beinhaltet komplexe Rechnung und andere gebräuchliche Operationen wie z.B. Trigonometrie, Betrag, Inversion, natürlicher Logarithmus und andere Logarithmen.	
Vektorrechnung	103
Dieses Programm ermöglicht folgende gebräuchliche Operationen mit 2- und 3-dimensionalen Vektoren: Addition, Subtraktion, Skalarprodukt, Vektorprodukt, Abstand, Winkel, Betrag und Einheitsvektor.	

4 Inhaltsverzeichnis

Koordinatentransformation	119
Dieses Programm leistet 2- und 3-dimensionale Koordinatentransformation und/oder Achsenrotation.	
Umwandlung zwischen Zahlensystemen sowie Boolesche Logik	129
Dieses Paket ermöglicht das Umwandeln von Zahlen zwischen Zweier-, Oktal-, Dezimal- und Hexadezimalsystem und verfügt über Boolesche Logik, Bit-Test und Bit-Rotation.	
Kurvenanpassung	135
Dieses Programm liest Datenpunkte ein und paßt ihnen eine Gerade, eine Exponentialkurve, eine logarithmischen Kurve, eine Potenzkurve oder die am besten passende Kurve an.	
Finanztechnische Berechnungen	145
Dieses Programm löst finanzielle Aufgaben, die sich mit Zeit, Geld und Zins beschäftigen.	
Programm-Index	Innenseite des Rückumschlags

EINLEITUNG

Das Advantage Pac für den HP-41 stellt Ihnen eine Auswahl von Programmen und Funktionen zur Verfügung, die Ihnen das Lösen von fortgeschrittenen mathematischen und ingenieurmäßigen Aufgaben, statistischen Kurvenanpassungsaufgaben und einfachen finanztechnischen Berechnungen ermöglichen. Es bietet eine umfangreiche Lösungssammlung für den technischen Studenten und Fachmann.

Viele der von diesem Paket intern benutzten Programme sind Ihnen in Ihren eigenen Programmen als Unterprogramme zugänglich.

Dieses Benutzerhandbuch enthält eine Beschreibung für jedes Programm und jede Funktionssammlung, die wichtige Gleichungen, schrittweise die Anweisungen zur Ausführung und Beispiele für die zur Lösung benötigten Tastenfolge enthält. Ebenso sind die für den Anwender zugreifbare Unterprogramme beschrieben.

Hinweis: *Schalten Sie Ihren Taschenrechner immer aus, bevor Sie ein Modul einstecken, und vergewissern Sie sich, daß Sie den Abschnitt "Einstecken und Entfernen der Anwender-Module" verstanden haben.*

EINSTECKEN UND ENTFERNEN DER ANWENDER-MODULE

Lesen Sie die nachfolgende Information, bevor Sie das erste Mal ein Anwender-Modul in Ihren Taschenrechner einstecken.

Sie können bis zu vier Anwender-Module in die Einschubschächte Ihres HP-41 einstecken. In Katalog 2 (**CATALOG** 2) werden alle Namen der in einem eingesteckten Modul enthaltenen Programme angezeigt.

VORSICHT

Schalten Sie Ihren Taschenrechner immer vor dem Einstecken oder Entfernen von Zubehörteilen aus; ansonsten kann sowohl der Rechner als auch das Zubehör beschädigt werden.

Einstecken eines Moduls:

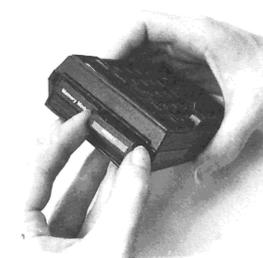
Schalten Sie den Taschenrechner aus!



Entfernen Sie die Schutzkappe. Bewahren Sie die Schutzkappe auf, damit Sie einen unbenutzten Schacht wieder verschließen können.



Wenn Sie einen HP-41C besitzen, stecken Sie das Anwender-Modul in einen Einschubschacht *nach* dem letzten Speichererweiterungs-Modul ein. (HP-41CV und HP-41CX verwenden keine Speichererweiterungs-Module.) *Stecken Sie niemals ein Anwender-Modul in einen Einschubschacht mit niedrigerer Nummer als der eines Speichererweiterungs-Moduls.* (Die Nummerierung der Einschubschächte befindet sich auf der Rückseite des Taschenrechners.)



8 Einstecken und Entfernen der Anwender-Module

Stecken Sie ebenso jedes zusätzliche Modul nach dem letzten Speichererweiterungs-Modul ein, und verschließen Sie die unbelegten Einschubschächte mit einer Schutzkappe.

Nun können Sie auf die im Anwender-Modul enthaltenen Programme zugreifen!

Entfernen der Anwender-Module:

Schalten Sie Ihren Taschenrechner aus! Versäumen Sie dies, so können Rechner und Modul beschädigt werden.

Nehmen Sie das entsprechende Modul und ziehen Sie es heraus, wie auf der Abbildung dargestellt.



Stecken Sie eine Schutzkappe in den unbelegten Einschubschacht.

Stecken Sie jedes Zubehörteil, wie z.B. den Magnetkartenleser HP 82104A oder den Lesestift HP 82153A, entsprechend den Anwender-Modulen in die Einschubschächte ein.

Sie können auch Schächte überspringen. Beispielsweise kann in Einschubschacht 1 ein Speichererweiterungs-Modul und in Einschubschacht 4 ein Anwender-Modul stecken, wobei Schacht 2 und 3 freibleibt.

VERWENDUNG DIESES HANDBUCHS

Aufbau der Kapitel

Jedes Kapitel dieses Handbuchs beschreibt ein bestimmtes Programm oder eine Funktionssammlung. Die Kapitel sind, mit Ausnahme der beiden Kapitel über Matrizen, voneinander unabhängig.

Jedes Kapitel beginnt mit der **Beschreibung** des Anwendungszweckes des entsprechenden Programms bzw. der entsprechenden Funktionssammlung. Es werden die dem Programm zugrundeliegenden **Gleichungen** und, falls zweckmäßig, **Literaturhinweise** angegeben. Wenn notwendig, wird auch der **zulässige Wertebereich von Daten** angegeben. In einigen Anwendungsfällen kann das Programm außerhalb des zulässigen Wertebereichs ablaufen und ein Ergebnis liefern, das für Sie jedoch nicht ausreichend genau sein könnte.

Die Anweisungstabelle

Der Hauptteil jedes Kapitels umfaßt die jeweiligen **Anweisungen** und die entsprechende **Anweisungstabelle**. Dies bietet Ihnen einen Überblick und eine stufenweise Anleitung, wie das Programm bzw. die Funktion anzuwenden ist. Ebenso wird angegeben, welche Arten von Datenwerten Sie in den Taschenrechner eingeben können und welche Tasten Sie drücken müssen, um ein Ergebnis zu erhalten.

Im Kopf der Anweisungstabelle wird die minimale Anzahl der vom Programm benötigten Datenspeicher-Register angegeben. (Siehe nachfolgender Abschnitt "Festlegen der Speicherplatzverteilung".)

		Umfang: 016
Anweisungen	Eingabe	Anzeige
: 3. Geben Sie ein Datenpaar ein: Wiederholen Sie die Eingabe für jedes Datenpaar :	y <input type="button" value="ENTER"/> x <input type="button" value="A"/> ($\Sigma+$)	$\Sigma+$ CL Σ FIT

Diese Spalte enthält durchzuführende Schritte und welche Art von Datenwerten Sie eingeben müssen.

Diese Spalte sagt Ihnen, welche Taste(n) Sie zur Eingabe von Daten oder zur Berechnung eines Ergebnisses drücken müssen.

Diese Spalte zeigt, was in der Anzeige des Taschenrechners nach der Ausführung der eingegebenen Anweisungen stehen soll. Meistens steht in der Anzeige ein Ergebnis, eine informelle Textausgabe oder ein Menü. (Die von den meisten Programmen in diesem Modul verwendete *Menütechnik* wird im Abschnitt "Verwenden eines Menüs" beschrieben.)

Der Anweisungstabelle folgen einige **Bemerkungen** über das Programm.

Jedes Kapitel beinhaltet **Beispiele** für das Benutzen des entsprechenden Programms bzw. der entsprechenden Funktionen.

Zuletzt werden **Programmier-Informationen** für den Aufruf der benutzerzugänglichen Unterprogramme gegeben.

Festlegen der Speicherplatzverteilung ()

In der Anweisungstabelle wird die minimale Anzahl von Datenspeicher-Registern, die Sie für den Ablauf des entsprechenden Programms benötigen, angegeben. Verwenden Sie zur Festlegung des Speicherplatzes für die *nnn* Speicherregister die Funktion (drücken Sie *nnn*). Schlagen Sie für weitere Informationen im Benutzerhandbuch des HP-41 nach. Wenn Sie ein Programm ausführen wollen, aber die Meldung

SIZE > = nnn

erhalten, müssen Sie mit mindestens *nnn* eingeben. Drücken Sie anschließend , um das Programm ablaufen zu lassen.

Bezeichnung der Rechner Tasten

Der HP-41 verfügt, wie im Benutzerhandbuch erklärt, über zwei verschiedene Funktionstypen, die *Tastefeld-Funktionen* und die *Alpha-Funktionen*. Diese beiden Funktionstypen werden auf zwei unterschiedliche Weisen aufgerufen (*ausgeführt*). Den Tastefeld-Funktionen ist eine entsprechende Taste (z.B. $\boxed{\Sigma+}$ und $\boxed{x^2}$) zugeordnet. Alpha-Funktionen, *einschließlich Programme*, werden aufgerufen, indem Sie zuerst $\boxed{\text{XEQ}}$ drücken und anschließend ihren Namen (*Alpha-Namen* genannt) eintippen.*

Tastensequenz	Zu drückende Tasten
$\boxed{\Sigma+}$	$\boxed{\Sigma+}$ Dies ist eine Tastefeld-Funktion.
$\boxed{\Sigma-}$	$\boxed{\Sigma+}$ (Drücken Sie die Tasten nacheinander.) Dies ist eine <i>umgeschaltete</i> Tastefeld-Funktion.
\boxed{A} (FX)	$\boxed{\Sigma+}$ ($\Sigma+$ ist auf der Oberseite aufgedruckt; A ist auf der Vorderseite aufgedruckt.) Dies ist eine "programmspezifische" Funktion eines jeweiligen Programms. Zum Beispiel würde FX in der Anzeige über \boxed{A} stehen. FX ist das <i>Menü-Label</i> für \boxed{A} .
$\boxed{\text{XEQ}}$ $\boxed{\text{TVM}}$	$\boxed{\text{XEQ}}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{TVM}}$ $\boxed{\text{ALPHA}}$ (Mit der $\boxed{\text{ALPHA}}$ -Taste schalten Sie auf das Alpha-Tastefeld um und wieder zurück.) Dies ist eine <i>Alpha-Funktion</i> . Sie kann ebenso über eine User-Taste ausgeführt werden. (Schlagen Sie dazu im Benutzerhandbuch des HP-41 nach.)
$\boxed{\text{XEQ}}$ $\boxed{\text{SIZE}}$ 013	$\boxed{\text{XEQ}}$ $\boxed{\text{ALPHA}}$ $\boxed{\text{SIZE}}$ $\boxed{\text{ALPHA}}$ $\boxed{0}$ $\boxed{1}$ $\boxed{3}$

Dieses Modul verwendet die Tasten in den beiden oberen Reihen des Tastefeldes als spezielle, neudefinierte Funktionen. Diese Tasten werden dann durch \boxed{A} bis \boxed{J} und nicht durch $\boxed{\Sigma+}$ usw. dargestellt.

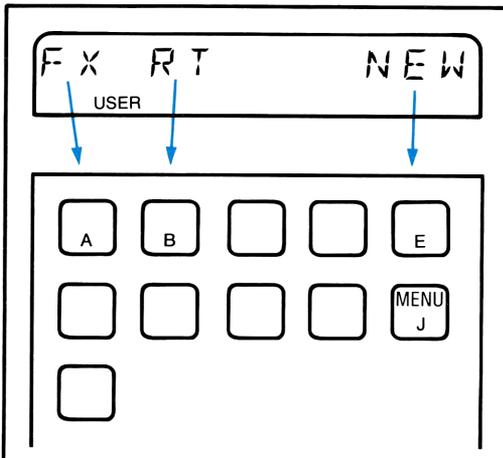
Verwenden eines Menüs

Mit diesem Modul stehen Ihnen sowohl neue *Funktionen* als auch neue *Programme* zur Verfügung. Eine Funktion können Sie wie eine gewöhnliche Alpha-Funktion verwenden. Die Programme sind anspruchsvoller *und* dabei einfach zu verwenden: Sie vereinigen mehrere neue Funktionen und ein *Anwender-Interface* mit Eingabeaufforderungen und Menüs, die Sie von der Dateneingabe bis zur Ergebnisausgabe führen.

* Die andere, schnellere Möglichkeit zur Ausführung von Alpha-Funktionen besteht darin, den Funktionsnamen (Alpha-Namen) einer Taste des User-Tastefeldes zuzuordnen. Schlagen Sie dazu im Benutzerhandbuch des HP-41 nach.

Beachten Sie, daß jedes Programm einige Tasten in den beiden oberen Reihen des Tastenfeldes neu definiert, um mit einem einzigen Tastendruck im Programm definierte Operationen auszuführen. *Damit ein Programm einwandfrei funktioniert, müssen Sie alle existierenden Tastenzuweisungen in den beiden oberen Reihen löschen.* Für das Benutzen der neu definierten Tasten muß das User-Tastenfeld aktiv sein. Alle in diesem Modul enthaltenen Programme, die diese Eigenschaft aufweisen, aktivieren beim Starten des Programms *automatisch* das User-Tastenfeld. Wenn Sie aus irgend einem Grunde das User-Tastenfeld inaktivieren, müssen Sie es wieder aktivieren (drücken Sie **USER**), um die neu definierten Tasten zu verwenden. Die Menüs besitzen *Labels*, die die neu definierten Tasten identifizieren. Hier ist ein Beispiel:

Menü zur Bestimmung der Nullstellen eines Polynoms



Das Menü-Label **FX** zeigt Ihnen, wenn Sie das Programm PLY ausführen, daß die Taste oben links, **A**, neu definiert wurde, um den Wert des Polynoms $f(x)$ für x zu berechnen. **B** (mit dem Label **RT** identifiziert) ist neu definiert, um die Nullstellen zu berechnen. **E** (mit dem Label **NEW** identifiziert) initialisiert das Programm, um die Koeffizienten eines neuen Polynoms einzulesen.

Die Taste **[J]** hat eine besondere Bedeutung. Das Drücken der Taste **[J]** bewirkt in allen Menüs dieses Moduls die erneute Anzeige des Menüs.* Sie können diese Taste zu jedem Zeitpunkt und beliebig oft drücken: Das Menü ist nur eine Hilfe zur Identifikation der Tasten.

Fehlermeldungen. Wenn Sie eine Fehlermeldung während eines Programmablaufs erhalten, ist es vorteilhaft, die Taste **[J]** (nachdem Sie den Fehler beseitigt haben) zu drücken. Eine Erklärung der Fehlermeldungen finden Sie im Benutzerhandbuch des HP-41. Fehlermeldungen, die im Zusammenhang mit Matrix-Operationen auftreten können, werden in Kapitel "Matrix-Operationen" erläutert.

Wenn sich der Rechner abschaltet. Tritt dies während der Ausführung eines Programms auf, wird in der Anzeige nach Einschalten des Rechners etwas anderes stehen. Es wird dann nur der Inhalt des X-Registers angezeigt und keine Aufforderung oder Menü, was eventuell vor dem Ausschalten angezeigt wurde. Obwohl das Programm immer noch aktiv ist, sollten Sie durch Drücken der Taste **[J]** das Menü anzeigen und dadurch das Programm neu starten.

Wenn Sie ein Programm ablaufen lassen, das *kein* Menü beinhaltet, kann das Setzen von Flag 21 notwendig sein, um eine korrekte Darstellung der Ergebnisse wieder herzustellen.

Der Programmablauf-Indikator

Wenn das Programm abläuft, erscheint der Programmablauf-Indikator (↗) und läuft quer durch die Anzeige. Führen Sie also eine Operation eines Programms aus, zeigt Ihnen der Indikator den Programmablauf an.

Auflisten des Modul-Inhalts

Katalog 2 zeigt Ihnen alle Namen von Programmen und Unterprogrammen, die in den eingesteckten Modulen enthalten sind, an. Drücken Sie **CATALOG** 2.

Verwendung von Programmen als Unterprogramme

Sie können die in diesem Modul enthaltenen Programme (und einige Unterprogramme) zur Verwendung in Ihrem eigenen Programm als Unterprogramme aufrufen. Schlagen Sie dazu in den Abschnitten "Programmier-Informationen" nach.

* Die Taste **[J]** ist dieselbe Taste wie die Taste **[TAN]**. Die Buchstabenbezeichnung wird verwendet, damit keine Verwechslung von der "alten" Funktion (Tangens) mit der neuen auftritt.

Verwenden eines Druckers

Wenn Sie während der Verwendung dieses Moduls einen Drucker an den HP-41 angeschlossen haben, setzen Sie ihn in den MAN-Modus, um die am besten lesbare Ausdrücke Ihrer Eingaben und der Ergebnisse automatisch zu erhalten. (Einige Programme verlangen NORMAL-Modus.) Im NORMAL-Modus werden ebenso alle Eingabedaten und alle Tastendrucke gedruckt.

Kopieren von Programmen

Viele Programme dieses Moduls können Sie mit der Funktion **COPY** kopieren. *Es ist jedoch nicht notwendig, ein Programm in den Hauptspeicher zu kopieren, um es starten zu können.* Ebenso ist es nicht notwendig, ein Unterprogramm zu kopieren, um es in Ihrem eigenen Programm aufrufen zu können.

Verwenden von Labels

Vermeiden Sie in Ihren eigenen Programmen das Benutzen von Labels, die identisch mit denen dieses Anwender-Moduls sind. Im Zweifelsfall hat das im Programmspeicher vorkommende Label Vorrang vor dem in diesem Anwender-Modul benutzten Label. In Katalog 2 werden alle in diesem Paket verwendeten Labels aufgelistet.

Störungen mit anderen Anwender-Modulen

Hinweis: Stecken Sie das HP-41 Advantage Pac und das HP-IL Modul nicht gleichzeitig in den HP-41 ein. Die beiden Module verwenden die gleichen ROM-Identifikationsnummern. Die gleichzeitige Verwendung der beiden Module würde zu Betriebschwierigkeiten des Taschenrechners führen.

Bestimmte Funktionsnamen werden vom HP-41 Advantage Pac sowie vom HP-41 Mathematik-Paket und HP-41 Real Estate Pac verwendet. Bevor Sie diese Funktionen benutzen, sollten Sie diejenigen Module entfernen, deren Funktionen Sie nicht verwenden möchten.

Doppelte Funktionen

Mathematik-Paket

Alle Funktionen mit komplexen Zahlen. N, PV, PMT, FV und *I
Alle Funktionen in DIFEQ.

Real Estate Pac

Unterstützung

Wenn Sie irgendwelche Fragen zum Betrieb Ihres Taschenrechners haben, schlagen Sie zunächst im Benutzerhandbuch Ihres HP-41 nach. Wenn Sie technische Schwierigkeiten mit diesem Paket haben, und das Handbuch Ihnen nicht weiterhilft, können Sie sich an die Kundenberatung von Hewlett-Packard wenden. Entnehmen Sie die Adresse und die Telefonnummer Ihrem HP-41 Benutzerhandbuch.

DAS MATRIX-PROGRAMM

Das Advantage Pac ermöglicht Ihnen, reelle und komplexe Matrizen anzulegen, abzuspeichern oder mit ihnen zu rechnen. Diese Operationen können Sie entweder über *individuelle Funktionen* oder über ein *Programm* mit Menüs und Eingabe-Aufforderungen ausführen. Dasselbe gilt für viele andere Themenbereiche in diesem Modul. Das Thema Matrizenrechnung ist, im Gegensatz zu anderen, wegen seiner Größe und Komplexität in zwei Kapitel unterteilt.

Dieses Kapitel beschreibt das Matrix-Programm **MATRIX**. Es ist ein einfaches, "benutzerfreundliches" Programm, mit dem Sie die gebräuchlichsten Matrix-Operationen auf eine neu angelegte Matrix anwenden können. Bei Verwendung von **MATRIX** müssen Sie nicht im Detail wissen, wo und wie der Rechner die Matrizen in seinem Speicher abspeichert. Im nächsten Kapitel, "Matrix-Operationen", werden alle in diesem Modul benutzten Matrix-Funktionen (einschließlich der von **MATRIX** aufgerufenen) aufgelistet und definiert. Das Verwenden dieser einzelnen Funktionen erfordert jedoch ein genaueres Wissen, wie und wo der Rechner Matrizen abspeichert.

Fähigkeiten dieses Programms

Betrachten Sie die Gleichungen

$$3.8x_1 + 7.2x_2 = 16.5$$

$$1.3x_1 - 0.9x_2 = -22.1$$

für die Sie x_1 und x_2 bestimmen müssen. Diese Gleichungen können in Matrix-Form als $\mathbf{AX} = \mathbf{B}$ geschrieben werden. Dabei ist

$$\mathbf{A} = \begin{bmatrix} 3.8 & 7.2 \\ 1.3 & -0.9 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{und} \quad \mathbf{B} = \begin{bmatrix} 16.5 \\ -22.1 \end{bmatrix}$$

\mathbf{A} ist die *Koeffizientenmatrix* des Gleichungssystems, \mathbf{B} die *Spalten- bzw. Konstantenmatrix* und \mathbf{X} die *Lösungs- bzw. Ergebnismatrix*.

Für ein derartiges Matrixsystem legt (dimensioniert) das Programm **MATRIX** eine quadratische reelle bzw. komplexe Matrix \mathbf{A} und eine Spaltenmatrix \mathbf{B} an. Dann können Sie:

- die Elemente in \mathbf{A} und \mathbf{B} eingeben, ändern ("editieren") oder einfach anschauen
- \mathbf{A} invertieren

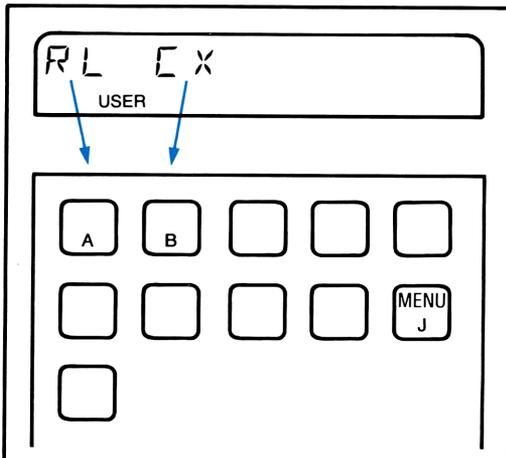
- **A** transponieren, falls **A** reell ist
- die Determinante zu **A** finden, falls **A** reell ist
- das Gleichungssystem lösen, indem Sie die Lösung zu $\mathbf{AX} = \mathbf{B}$ finden

Die Dimension der Matrix ist nur durch den zur Verfügung stehenden Speicherplatz begrenzt. (Für jedes Element einer reellen Matrix wird ein Speicherregister benötigt. Zusätzlich wird ein weiteres Speicherregister je Matrix benötigt.) Wenn Sie mehr als eine Matrix abspeichern wollen, müssen Sie die im nächsten Kapitel beschriebene Matrix-Funktion **MATDIM** verwenden. Das Programm **MATRIX** speichert Matrizen nicht ab und ruft Matrizen nicht wieder auf. Es verwendet nur eine quadratische Matrix **A** und eine Spaltenmatrix **B**. Wenn Sie neue Elemente in **A** eingeben, überschreiben Sie die alten Elemente.

Anweisungen

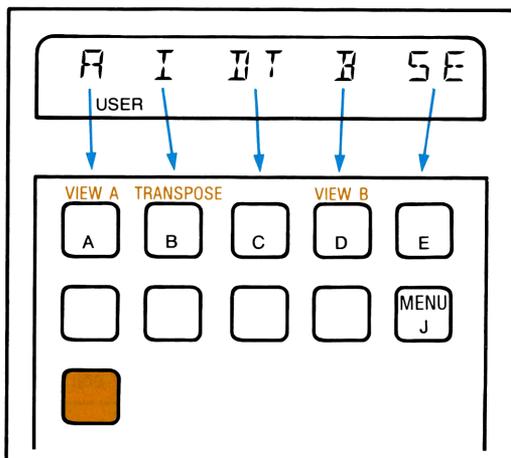
MATRIX verfügt über zwei Menüs, die Ihnen zeigen, welche Funktion über welche Taste zugänglich ist. Das nachfolgend abgebildete *Startmenü* erlaubt Ihnen, zwischen reeller und komplexer Matrix zu wählen:

Startmenü



Nachdem Sie diese Wahl getroffen haben, geben Sie die Dimension der Matrix ein und drücken **[R/S]**. Dann sehen Sie das *Hauptmenü*:

Hauptmenü



Dieses Menü gibt Ihnen an, welche Matrixoperationen Sie mit dem Programm MATRIX ausführen können. Sie können sich dieses Menü zu jeder Zeit durch Drücken der Taste **[J]** anzeigen lassen. Der Programmablauf wird dadurch in keiner Weise gestört.

Durch Drücken der Taste **[←]** können Sie das Menü jederzeit löschen. Dann erscheint in der Anzeige der Inhalt des X-Registers. Das Programm wird dadurch aber nicht beendet. Sie können nun Berechnungen durchführen und dann das Menü durch Drücken von **[J]** wieder in die Anzeige holen. (Sie müssen die Anzeige des Programms *nicht* löschen, bevor Sie Rechnungen ausführen.)

- Das Programm beginnt mit der Frage nach dem Typ der neuen Matrix. Es läßt Sie zwischen reeller und komplexer Matrix wählen und die Dimension einer quadratischen Matrix für **A** angeben.
- Das Programm löscht *nicht* die vorhergehenden Matrix-Daten. Die vorhergehenden, u.U. bedeutungslosen Daten bleiben in den neuen Matrizen **A** und **B** solange stehen, bis Sie für deren Elemente neue Werte eingeben.
- Jedes Element einer komplexen Matrix hat zwei Werte (Real- und Imaginärteil) und benötigt den vierfachen Speicherplatz im Vergleich zu einer reellen Matrix. Die Eingabeaufforderungen für die Realteile x_{11} , x_{12} , etc. lauten **1:1 = ?**, **1:2 = ?**, etc. Die Eingabeaufforderungen für die komplexen Elemente $x_{11} + iy_{11}$, $x_{12} + iy_{12}$, etc. lauten **RE.1:1 = ?**, **IM.1:1 = ?**, **RE.1:2 = ?**, **IM.1:2 = ?**, etc.

Das folgende Kapitel ("Matrix-Operationen") enthält im Abschnitt "Abspeichern einer Matrix" ausführliche Erläuterungen zu den speziellen Anforderungen für das Abspeichern einer Matrix. Sie sollten darüber Bescheid wissen, da das Programm Sie nach der richtigen Speicherplatzzuweisung durch **SIZE** $\geq nnn$ fragt, wenn der Speicherplatz nicht ausreichend groß ist. (In diesem Fall setzen Sie mit **SIZE** nnn die erforderliche Größe des Speicherplatzes.)

In der folgenden Tabelle werden die Tastenfolgen für das Ausführen von Matrix-Operationen mit dem Programm **MATRIX** angegeben. Alle diese Operationen sind ebenso als einzelne HP-41 Funktionen ausführbar (siehe nächstes Kapitel).

Anweisungstabelle für MATRIX

		Umfang: variabel*
Anweisungen	Eingabe	Anzeige
<p>1. Starten Sie das Programm MATRIX.</p>	<p>XEQ MATRIX †</p>	<p>RL CX</p>
<p>2. Wählen Sie eine neue reelle (RL) oder komplexe (CX) Matrix.</p>	<p>A (RL) oder B (CX)</p>	<p>ORDER=?</p>
<p>3. Geben Sie die Dimension n Ihrer quadratischen Matrix A ein.</p>	<p>n R/S</p>	<p>A I DT B SE</p>
<p>4. Geben Sie die Elemente Ihrer Matrix ein. Nach Erscheinen des ? können Sie das entsprechende Element ändern, falls gewünscht. Nach der Eingabe des Elements drücken Sie R/S, um das nächste Element eingeben zu können.</p>	<p>A R/S : R/S R/S</p>	<p>1:1=a₁₁? RE.1:1=a₁₁? 1:2=a₁₂? IM.1:1=y₁₁? : n:n=a_{nn}? IM.n:n=y_{nn}? A I DT B SE</p>
<p>Um die Matrix A wieder anzusehen oder zu ändern, wiederholen Sie den vorangegangenen Ablauf. Drücken Sie einfach R/S, um ein Element unverändert zu lassen.</p>		
<p>5. Um ein spezielles Element $a_{i,j}$ zu ändern, gehen Sie zuerst in den Edit-Modus und wählen Sie das Element mit $iii.jjj$ an. †</p>	<p>A $iii.jjj$ A</p>	<p>$i:j=a_{i,j}$? oder RE.i:j=a_{i,j}?</p>
<p>Wenn das Element $iii.jjj$ nicht existiert, geht das Programm vom Edit-Modus zum Hauptmenü zurück.</p>		
<p>Verwenden Sie die Taste R/S, um von einem Element zum nächsten zu gelangen. Vom letzten Element erhalten Sie nach R/S das Hauptmenü.</p>	<p>: R/S</p>	<p>: A I DT B SE</p>

Anweisungen	Eingabe	Anzeige
<p>6. Wenn Sie die Matrix A <i>nur ansehen</i> möchten:</p> <p>Beachten Sie, daß in diesem Fall kein ? erscheint. Sie können die Matrixelemente nicht ändern.</p> <p>7. Um die Konstantenmatrix B einzugeben, zu ändern oder anzusehen, folgen Sie genau den Schritten 4, 5 bzw. 6 (unter Verwendung von <input type="text" value="D"/> (B) und <input type="text" value="D"/> (<input type="text" value="B"/>)). B ist automatisch richtig auf 1 Spalte und n Zeilen dimensioniert (Schritt 3).</p> <p>8. Um den Edit-Modus zu beenden und in das Hauptmenü zurückzukehren:</p> <p>9. Ausführen einer Matrix-Operation:</p> <ul style="list-style-type: none"> ■ Inversion von A nach A^{-1}. Matrix A wird ersetzt. Ansehen von A^{-1}. ■ Transponieren von A (falls reell) nach A^T. Matrix A wird ersetzt. (Falls Sie A invertiert haben, invertieren Sie sie zuerst wieder zurück. Wenn Sie $\det(A)$ berechnet oder ein Gleichungssystem mit A gelöst haben, müssen Sie vor dem Transponieren A zuerst <i>zweimal</i> invertieren, um A wiederherzustellen. Weitere Informationen dazu finden Sie im Abschnitt "Bemerkungen".) Ansehen von A^T. ■ Berechnung der Determinante von A (falls reell), $\det(A)$. (Wenn Sie A invertiert haben, invertieren Sie A zuerst zurück. Diese Operation ersetzt A durch seine LR-zerlegte Form. Siehe "Bemerkungen".) 	<p><input type="text" value="A"/></p> <p><input type="text" value="R/S"/> §</p> <p>⋮</p> <p><input type="text" value="R/S"/> §</p> <p><input type="text" value="J"/></p> <p><input type="text" value="B"/> (I)</p> <p><input type="text" value="A"/></p> <p><input type="text" value="R/S"/> §</p> <p>⋮</p> <p><input type="text" value="B"/></p> <p><input type="text" value="A"/></p> <p><input type="text" value="R/S"/> §</p> <p>⋮</p> <p><input type="text" value="C"/> (DT)</p> <p><input type="text" value="R/S"/> §</p>	<p>1:1=a_{11} oder RE.1:1=a_{11} 1:2=a_{12} oder IM.1:1=y_{11} ⋮ A I DT B SE</p> <p>A I DT B SE</p> <p>A I DT B SE</p> <p>A I DT B SE</p> <p>DET=Ergebnis A I DT B SE</p>

Anweisungen	Eingabe	Anzeige
<ul style="list-style-type: none"> ■ Lösen des durch $AX = B$ beschriebenen Gleichungssystems. Die gefundene Lösung X ersetzt B. (Ebenso wird A durch die LR-zerlegte Form ersetzt. Siehe "Bemerkungen".) <p>Ansehen von X (ersetzt B).</p>	<p><input type="text" value="E"/> (SE)</p> <p><input type="text" value="D"/> (<input type="text" value="B"/>)</p> <p><input type="text" value="R/S"/> §</p> <p>⋮</p>	<p>A I DT B SE</p>
<p>* Die benötigte Anzahl der Speicherregister hängt von der Dimension der einzugebenden Matrizen ab. Sie ist $(Dimension^2 + Dimension + 2)$ für reelle Matrizen A und B; $[4(Dimension^2) + 2(Dimension) + 2]$ für komplexe Matrizen A und B. Das Programm wird Ihnen jedoch mitteilen, welchen Wert Sie für SIZE eingeben müssen, wenn der Speicherplatz nicht ausreichend groß ist.</p> <p>† Zur Ausführung eines Programms drücken Sie entweder <input type="text" value="XEQ"/> <input type="text" value="ALPHA"/> Alpha-Name <input type="text" value="ALPHA"/> oder eine entsprechende User-Taste.</p> <p>‡ Sie können im i-Teil führende und im j-Teil nachfolgende Nullen weglassen. Ist ein Teil Null, so wird er automatisch auf 1 umgeändert. Beispiel: 0.000 wird zu 1.001.</p> <p>§ Wenn Sie einen Drucker angeschlossen haben, erscheint in der Anzeige nach dem Drucken des Ergebnisses automatisch das Hauptmenü.</p>		

Eine Liste von Fehlermeldungen, die beim Ausführen von Matrix-Operationen in der Anzeige erscheinen können, ist im Abschnitt "Fehlermeldungen" im nächsten Kapitel aufgeführt.

Bemerkungen

Verändern der ursprünglichen Matrix. Die eingegebene Matrix A wird durch die Operationen Inversion, Bestimmung der Determinante, Transposition und Lösen eines Gleichungssystems verändert. Sie können A^{-1} zurückinvertieren und A^T zurücktransponieren, um die ursprüngliche Matrix A wieder zu erhalten. Wenn Sie jedoch die Determinante berechnet oder das Gleichungssystem gelöst haben, ist A in der LR-zerlegten Form gegeben. Um wieder A zu erhalten, müssen Sie die vorliegende Matrix *zweimal invertieren*. Die LR-Zerlegung stört *außer* Transponieren und Ändern* keine der im Programm MATRX nachfolgenden Operationen. Schlagen Sie für zusätzliche Informationen über die LR-Zerlegung im Abschnitt "LR-Zerlegung" des nächsten Kapitels ("Matrix-Operationen") nach.

* Ändern Sie in einer LR-zerlegten Matrix immer nur alle Elemente.

Abspeichern einer Matrix. Das Programm MATRX speichert eine Matrix **A** (beginnend in R_0) im Hauptspeicher ab; sie wird **R0** bezeichnet. Danach wird die Konstantenmatrix **B** abgespeichert. Die Ergebnismatrix **X** überschreibt **B**. Wie Matrizen genannt, wo sie abgespeichert werden und wieviel Speicherplatz sie benötigen, erfahren Sie im Kapitel "Matrix-Operationen".

MATRX kann außer auf die vorhergehende Matrix **R0** und die zugehörige Konstantenmatrix auf keine anderen Matrizen zugreifen.

Neudefinierte Tasten. Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten **A**–**E**, **J**, **A**, **B** und **D** zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung überschrieben. *Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten.* Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Beispiele

Das zu Beginn dieses Kapitels gegebene Gleichungssystem definiert die Matrix-Gleichung $\mathbf{AX} = \mathbf{B}$ bzw.

$$\begin{bmatrix} 3.8 & 7.2 \\ 1.3 & -0.9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 16.5 \\ -22.1 \end{bmatrix}$$

Bestimmen Sie zuerst die inverse Matrix, die Determinante und die Transponierte zu **A**. Berechnen Sie anschließend die Ergebnismatrix **X**.

Tastensequenz

FIX 4
XEQ **SIZE** 008

XEQ **MATRX**

A (**RL**)

2 **R/S**

Anzeige

RL CX

ORDER=?

A I DT B SE

Stellt das hier benutzte Anzeigeformat ein.
 Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $\text{SIZE} \geq 008$ ist.
 Beginnt das Programm MATRX.
 Wählt eine reelle Matrix.
 Dimensioniert eine 2×2 Matrix.

Tastensequenz

A

3.8 R/S

7.2 R/S

1.3 R/S

.9 CHS R/S

A

R/S *

R/S *

R/S *

R/S *

B (I)

A

R/S *

R/S *

R/S *

R/S *

B (I)

B

A

R/S *

R/S *

R/S *

R/S *

B

C (DT)

D (B)

16.5 R/S

22.1 CHS R/S

Anzeige1:1 = a_{11} ?1:2 = a_{12} ?2:1 = a_{21} ?2:2 = a_{22} ?

A I DT B SE

1:1 = 3.8000

1:2 = 7.2000

2:1 = 1.3000

2:2 = -0.9000

A I DT B SE

A I DT B SE

1:1 = 0.0704

1:2 = 0.5634

2:1 = 0.1017

2:2 = -0.2973

A I DT B SE

A I DT B SE

A I DT B SE

1:1 = 3.8000

1:2 = 1.3000

2:1 = 7.2000

2:2 = -0.9000

A I DT B SE

A I DT B SE

DET = -12.7800

1:1 = b_{11} ?2:1 = b_{21} ?

A I DT B SE

Beginn des Editierens von **A** und Anzeige des (alten) Elements a_{11} .

Eingabe von 3.8 für a_{11} .

Eingabe für a_{22} und Rückkehr zum Hauptmenü.

Anzeige des momentanen Inhalts von **A**.

Inversion von **A**.

Anzeige des momentanen Inhalts von **A**, nun A^{-1} .

Zurückinvertieren von A^{-1} nach **A**.

Transposition von **A**.

Anzeige des momentanen Inhalts von **A**, nun A^T .

Zurücktransponieren von A^T nach **A**.

Det(**A**).

Beginn des Editierens von **B** und Anzeige des (alten) Elements b_{11} .

Eingabe von 16.5 für b_{11} .

Eingabe für b_{21} und Rückkehr zum Hauptmenü.

Tastensequenz

D (**B**)
 R/S *
 R/S *
 E (**SE**)

D (**B**)
 R/S *
 R/S *

Berechne die zu dieser komplexen Matrix inverse Matrix:

$$\begin{bmatrix} 1 + 2i & 3 + 3i \\ 4 + 5i & 6 + 7i \end{bmatrix}$$

Tastensequenz

XEQ **SIZE** 017

XEQ **MATRIX**

B (**CX**)
 2 **R/S**

A (**A**)
 1 **R/S**
 2 **R/S**
 3 **R/S**
 4 **R/S**

1.002 **A**

R/S

Anzeige

1:1=16.5000
2:1=-22.1000
A I DT B SE
A I DT B SE

Anzeige des momentanen Inhalts von **B**.

Lösen des Gleichungssystems $\mathbf{AX} = \mathbf{B}$ und Abspeichern von \mathbf{X} in **B**.

1:1=-11.2887
2:1=8.2496
A I DT B SE

Anzeige der Ergebnismatrix (in **B**).

Anzeige

RL CX

ORDER=?
A I DT B SE

RE.1:1=a₁₁?
IM.1:1=y₁₁?
RE.1:2=a₁₂?
IM.1:2=y₁₂?
RE.2:1=a₂₁?

Für eine komplexe Matrix **A**:
 $2^2 \times 4 + 1 = 17$.

Neues Starten des Programms.

Dimensionierung einer 2×2 komplexen Matrix.

RE.1:2=3.0000?

Falsche Eingabe für y_{12} . Sollte 3 und nicht 4 sein.

IM.1:2=4.0000?

Zurücksetzen des Editors auf das Element a_{12} .

Der Imaginärteil. (Falscher Wert.)

* Wenn ein Drucker angeschlossen ist, muß die Taste **R/S** nicht gedrückt werden.

Tastensequenz

3

4

5

6

7

(I)

2.002

(oder)

Anzeige

RE.2:1 = a_{21} ?

IM.2:1 = y_{21} ?

RE.2:2 = a_{22} ?

IM.2:2 = y_{22} ?

A I DT B SE

A I DT B SE

RE.1:1 = -0.9663

RE.2:2 = -0.2360

IM.2:2 = -0.0225

A I DT B SE

Eingabe des korrekten Wertes für y_{12} . Weitere Dateneingaben folgen.

Eingabe für das letzte Element und Rückkehr zum Hauptmenü.

Inversion von A.

Anzeigen von A^{-1} .*

Anzeige von $a_{22} + iy_{22}$.*

Rücksprung zum Hauptmenü.

* Wenn Sie einen Drucker angeschlossen haben, wird beim Anzeigen der Matrix automatisch die vollständige Matrix gedruckt und danach das Hauptmenü wieder angezeigt.

MATRIX-OPERATIONEN

Inhalt

Anlegen einer Matrix	31
Zuweisen eines Namens für eine Matrix	32
Dimensionieren einer Matrix	32
Wie wird eine Matrix abgespeichert?	33
Benutzen des Matrix-Editors	34
Anzeigen der Matrixelemente	35
Spezifizieren einer Matrix	36
Voreingestellte Matrix-Parameter	37
Fehlermeldungen	38
Abspeichern und Abrufen einzelner	
Matrixelemente	39
Einzelner Zugriff auf Matrixelemente	39
Sequentieller Zugriff auf Matrixelemente	41
Matrixfunktionen	42
Matrix-Arithmetik	43
Wichtige Matrix-Operationen	44
Andere Matrixfunktionen ("Dienstfunktionen")	46
Arbeiten mit komplexen Matrizen	50
Darstellung komplexer Matrixelemente	50
Benutzen von Funktionen mit komplexen Matrizen	52
LR-Zerlegung	52
Beispiele	54
Alphabetisch geordnete Funktionstabelle	59

MATRIX-OPERATIONEN

Dieses Kapitel steht mit dem vorhergehenden Kapitel "Das Matrix-Programm" in Verbindung. Es werden alle in diesem Paket verfügbaren Funktionen, die Matrix-Operationen betreffen, ausführlich beschrieben.

Sie können reelle und komplexe Matrizen anlegen, verändern und abspeichern. Die Dimension und Anzahl der Matrizen ist nur durch den in Ihrem Rechner verfügbaren Speicherplatz begrenzt. Wenn Sie über einen Zusatzspeicher (ein erweitertes Funktionen/Speicher-Modul HP 82180A oder einen HP-41CX) verfügen, können Sie dort ebenso Matrizen abspeichern.

Die in diesem Paket angebotenen Matrix-Operationen beinhalten Inversion, Transposition, Bestimmung der Determinante, Lösen eines Gleichungssystems und Ausführung von Matrix-Arithmetik. Zusätzlich können Sie einzelne Elemente von und zwischen Matrizen ändern.

Anlegen einer Matrix

Zum Anlegen einer Matrix müssen Sie ihren Namen und Dimension angeben. Die Funktion `MATDIM` verwendet zum Anlegen einer Matrix den Namen aus dem Alpha-Register und die Dimensionen `mmm.nnn` aus dem X-Register.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & a_{ij} & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Die Elemente einer neuen Matrix im Hauptspeicher werden *nicht* gelöscht (mit 0 initialisiert), sondern es wird der Inhalt der Elemente der alten Matrix bzw. der Register beibehalten. Die Elemente einer neuen Matrix im erweiterten Speicher *werden* jedoch gelöscht.

Geben Sie dann die Werte—numerische Werte oder Alpha-Werte—in die Matrix mit Hilfe des Matrix-Editors ein (Seite 34).

Zuweisen eines Namens für eine Matrix

Der Name, den Sie einer Matrix zuweisen, bestimmt darüber, wo die Matrix abgespeichert wird. Damit eine Matrix im Hauptspeicher (nicht im erweiterten Speicherbereich) abgelegt wird, muß der Name das Format

$$\mathbf{R}_{xxx}$$

aufweisen, wobei xxx bis zu drei Ziffern darstellt. (Sie können führende Nullen weglassen.) Mit der Speicherung der Matrix wird in \mathbf{R}_{xxx} *begonnen*. Beispielsweise ist $\mathbf{R007}$ gleichbedeutend mit $\mathbf{R7}$, wodurch die Matrix, beginnend in \mathbf{R}_{07} , abgespeichert wird.

Soll eine Matrix beginnend in $\mathbf{R0}$ abgespeichert werden, können Sie kurz \mathbf{R} als Matrixnamen angeben.

Eine Matrix A können Sie durch Angabe eines Namens mit bis zu sieben Alpha-Zeichen im erweiterten Speicher ablegen. Der Name darf jedoch nicht die Buchstaben "X" (reserviert zur Benennung des X-Register) und "R", gefolgt von bis zu drei Ziffern (reserviert zur Benennung der Matrizen im Hauptspeicher), beinhalten. Sie müssen den Dateityp nicht spezifizieren; es wird der Matrix automatisch ein für Matrizen einheitlicher Dateityp zugewiesen.

Verwenden Sie das Alpha-Register, um Matrixnamen zu spezifizieren. Wenn Sie mehr als einen Matrixnamen (als Parameter für gewisse Funktionen) angeben wollen, trennen Sie die Namen durch Kommas.

MNAME? sendet den Namen der momentanen Matrix zum Alpha-Register zurück.

Dimensionieren einer Matrix

Spezifizieren Sie die Dimensionen einer neuen Matrix mit $mmm.nnn$, wobei m die Anzahl der Zeilen und n die Anzahl der Spalten ist. Sie können führende Nullen in m und nachfolgende Nullen in n weglassen.

Für eine komplexe Matrix spezifizieren Sie $mmm.nnn$ als *doppelte* Anzahl der Zeilen und Spalten. (Schlagen Sie im Abschnitt "Arbeiten mit komplexen Matrizen" nach.)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$mmm.nnn = 2.003$$

$$\begin{bmatrix} 1 + i & 2 + 3i \\ 4 + 5i & 6 + 7i \end{bmatrix}$$

$$mmm.nnn = 4.004$$

Einem Dimensionsteil, der Null ist, wird 1 zugewiesen. Daher ist 0 gleichbedeutend mit 1.001, 3 mit 3.001 und .023 mit 1.023.

MATDIM dimensioniert eine neue Matrix oder dimensioniert eine bestehende Matrix um.

DIM? Sendet die Dimensionen *mmm.nnn* einer im Alpha-Register spezifizierten Matrix zum X-Register. (Durch ein leeres Alpha-Register wird die momentane Matrix spezifiziert.)

Wie wird eine Matrix abgespeichert?

Die Matrixelemente werden nacheinander zeilenweise von links nach rechts, beginnend mit der ersten Zeile, im Speicher ablegt. Jedes Element benötigt ein Datenspeicher-Register. Eine komplexe Zahl wird in vier Speicherregistern abgelegt.

Speicherplatz. Eine im Hauptspeicher abgelegte Matrix benötigt $(m \times n) + 1$ Datenspeicher-Register. Davon wird ein Register als Status-Register benutzt. Eine komplexe Matrix benötigt $(2m \times 2n) + 1$ Speicherregister, wobei m die Anzahl der Zeilen und n die Spaltenzahl der komplexen Matrix bedeuten.

Hinweis: Um eine im Hauptspeicher abzulegende Matrix erfolgreich zu dimensionieren, muß der Speicherbereich groß genug gewählt werden. Ansonsten wird beim Ausführen der Funktion **MATDIM** die Meldung **NONEXISTENT** angezeigt. Ordnen Sie mit **SIZE** *nnn* mehr Datenregister zu und führen Sie die Funktion erneut aus.

Eine im erweiterten Speicher abgelegte Matrix hat eine Dateilänge von $m \times n$. ($2m \times 2n$ für komplexe Matrizen.) Der Dateityp wird nur für Matrizen verwendet. Verwenden Sie nie die Funktion **CLFL** im Zusammenhang mit einer im erweiterten Speicher abgelegten Matrix: Dies würde teilweise die Daten im Dateivorsatz löschen. Verwenden Sie stattdessen die Funktion **PURFL** zum vollständigen Löschen der Matrix.

Ändern der Matrix-Dimensionen. Wenn Sie eine Matrix umdimensionieren, werden die bestehenden Elemente den neuen Elementen entsprechend der neuen Dimension zugeordnet. Überschüssige alte Elemente gehen verloren; zusätzlichen neuen Elementen werden die Werte der entsprechenden Speicherregister (außer im erweiterten Speicher, wo die Elemente mit 0 initialisiert werden) zugewiesen.

VORSICHT

Wenn Sie mit `MATDIM` eine im erweiterten Speicher abgelegte Matrix umdimensionieren, wird die Position des Matrix-Zeigers nicht angeglichen. Falls der Zeiger auf ein Element außerhalb der neuen Matrixgrenzen zeigt, müssen Sie den Zeiger durch Ausführen von `MSIJ` oder `MSIJA` mit gültigen Indizes auf ein Element innerhalb der neuen Grenzen ausrichten.

Im erweiterten Speicher abgelegte Matrizen können nicht umdimensioniert werden, sodaß sie den erweiterten Speicher vollständig ausfüllen. Die maximal zulässige Größe einer umdimensionierten Matrix ist ein Register kleiner als der momentan verfügbare Speicher. Eine *neue* Matrix kann jedoch so dimensioniert werden, daß sie den ganzen verfügbaren erweiterten Speicher ausfüllt.

Umdimensionieren von 2×3 auf 2×2

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{matrix} \text{verloren} \\ 5 \quad 6 \end{matrix}$$

Umdimensionieren von 2×3 auf 2×4

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & ? & ? \end{bmatrix}$$

Dies geschieht jedesmal, wenn Sie eine neue Matrix dimensionieren. Die alten Elemente der vorhergehenden Matrix bleiben solange erhalten, bis Sie sie ändern.

Benutzen des Matrix-Editors

Es gibt zwei Matrix-Editoren: `MEDIT` für reelle Matrizen und `CMEDIT` für komplexe Matrizen.

Die Matrix-Editoren werden für drei Zwecke verwendet:

- Eingeben neuer Werte für die Elemente einer Matrix.
- Ansehen und ändern ("editieren") von Matrixelementen; entweder der Reihenfolge nach oder durch "wahlfreien Zugriff" auf spezifizierte Elemente.
- Ansehen (ohne Änderungen vornehmen zu können) von Matrixelementen (Flag 08 gesetzt).

Anzeigen der Matrixelemente

Wenn Sie **MEDIT** oder **CMEDIT** ausführen, zeigt der Editor das Element 1,1 der im Alpha-Register spezifizierten Matrix an (oder der momentanen Matrix, falls das Alpha-Register leer ist). Durch Drücken der Taste **R/S** können Sie schrittweise alle Elemente einer Matrix ansehen. Für eine komplexe Matrix werden beide Teile der Matrix nacheinander angezeigt.

Funktion	Anzeige	Funktion	Anzeige
MEDIT	1:1 = 1.0000?	CMEDIT	RE.1:1 = 1.0000?
R/S	1:2 = 2.0000?	R/S	IM.1:1 = 1.0000?
		R/S	RE.1:2 = 2.0000?
⋮	⋮	⋮	⋮
R/S	(X-Register)	R/S	(X-Register)

Das ? am Ende der Anzeigezeile sagt Ihnen, daß Sie diesen Wert ändern können. Wenn Sie den Wert des angezeigten Elements ändern möchten, geben Sie einfach den neuen Wert ein und drücken dann **R/S**. Dies machen Sie für eine neue Matrix und ebenso, um einzelne Elemente einer bestehende Matrix zu ändern oder zu korrigieren.

Wenn Sie **R/S** drücken und keinen neuen Wert eingegeben haben, bleibt der alte Wert unverändert.

Ansehen ohne zu editieren. Wenn Sie Flag 08 setzen, können Sie im Editor die Matrixelemente nur ansehen und nicht ändern. In der Anzeige erscheint dann am Ende der Zeile kein ?.

$$1:1 = 1.0000$$

Wenn Sie einen Drucker angeschlossen haben und Flag 08 gesetzt ist, druckt er alle Matrixelemente ohne Unterbrechung.

Direkter Zugriff auf ein Matrixelement. Sie können, während der Editor aktiv ist (ebenso das Benutzer-Tastenfeld), auf ein Matrixelement direkt zugreifen. Um auf das Element in der *iten* Zeile und in der *jten* Spalte zuzugreifen, geben Sie *iii.jjj* ein und drücken anschließend **A**. (Dies ist wie im Programm MATRX.) Sie können in *iii* führende Nullen und in *jjj* nachfolgende Nullen weglassen.

Bei einer komplexen Matrix können Sie direkt auf den *Realteil* des Elements *i, j* zugreifen. Drücken Sie dann **R/S**, um auf den *Imaginärteil* zugreifen zu können.

Tastensequenz

[ALPHA] Matrix Name [ALPHA]

[XEQ] [MEDIT]

3.003 [A]

[ALPHA] Name d. komplexen Matrix [ALPHA]

[XEQ] [CMEDIT]

3.003 [A]

[R/S]

Anzeige

1:1 = 1.0000?

3:3 = 6.0000?

RE.1:1 = 1.0000?

RE.3:3 = 6.0000?

IM.3:3 = 7.0000?

Sie können im i -Teil führende und im j -Teil nachfolgende Nullen weglassen. Einem Teil, der Null ist, wird 1 zugewiesen.

Verlassen des Editors. Um den Editor zu verlassen, bevor er beim letzten Element angelangt ist, haben Sie zwei Möglichkeiten:

- Drücken Sie [J].
- Versuchen Sie, auf ein nicht existierendes Element zuzugreifen. Drücken Sie beispielsweise für eine 4×4 Matrix 5 [A].

Spezifizieren einer Matrix

Für die Matrizenmultiplikation $\mathbf{AB} = \mathbf{C}$ sind \mathbf{A} und \mathbf{B} gegeben, und Sie wollen das Matrizenprodukt \mathbf{C} berechnen. Damit der Rechner diese Operation ausführen kann, müssen Sie ihm die Namen der bestehenden Matrizen \mathbf{A} und \mathbf{B} sowie den Namen der Ergebnismatrix \mathbf{C} mitteilen. (Die Ergebnismatrix kann eine der Eingabe-Matrizen sein.) Alle angegebenen Matrizen müssen bereits einen Namen aufweisen und dimensioniert sein. Natürlich müssen nur \mathbf{A} und \mathbf{B} gültige Daten enthalten.

Einige Funktionen verwenden nur eine Eingabe-Matrix. Ebenso verwenden einige Funktionen *automatisch* eine der Eingabe-Matrizen als Ergebnismatrix. Die minimale Anzahl der zu spezifizierenden Matrizen ist also eins; die maximale Anzahl ist drei.

Eine Matrix-Funktion entnimmt dem Alpha-Register die Namen (d.h. die Orte) der Eingabe-Matrizen. Bevor Sie eine derartige Funktion ausführen, müssen Sie alle benötigten Parameter durch Kommas getrennt im Alpha-Register *spezifizieren*.

Alpha-Register

Eingabe-Matrix[,Eingabe-Matrix][,Ergebnismatrix]

Alpha-Register A,B,C

XEQ M*M

beispielsweise multipliziert die Matrizen **A** und **B** und legt das Ergebnis in die bestehende Matrix **C** ab.

Skalare Operationen. Skalare Ein- und Ausgabe muß über das X-Register erfolgen. Dieser Bereich muß daher nicht spezifiziert werden, *außer*, daß die betreffende Funktion *entweder* einen Skalar *oder* eine Matrix als gleichen Eingabe-Parameter verwendet. Benutzen Sie X, um das X-Register zu spezifizieren.

Beispielsweise *verlangt* MATDIM eine skalare Eingabe und einen Matrixnamen. Sie müssen daher das X-Register nicht spezifizieren. Andererseits verwenden die arithmetischen Funktionen, wie z.B. MAT*, *entweder* zwei Matrizen *oder* einen Skalar und eine Matrix als Eingabe. In diesem Fall müssen Sie X spezifizieren.

Die momentane Matrix. Die *momentane Matrix* ist diejenige, auf die zuletzt durch eine Matrix-Operation zugegriffen wurde (die zuletzt benutzt wurde). Wenn das Alpha-Register gelöscht ist und Sie eine Matrix-Funktion ausführen, die eine Matrix-Spezifikation erfordert, wird die momentane Matrix *als Standardvorgabe* benutzt. (Wenn keine momentane Matrix vorhanden ist, wird UNDEF ARRAY angezeigt.)

Die Ergebnismatrix wird nach der Ausführung der entsprechenden Operation automatisch die momentane Matrix.

Um den Namen der momentanen Matrix im Alpha-Register anzuzeigen, führen Sie MNAME? aus.

Voreingestellte Matrix-Parameter

Wenn Sie *nicht* alle oder keine von einer Matrix-Funktion benötigten Matrizen spezifizieren, werden die Standardvorgaben übernommen. (Standardvorgabe bedeutet, daß ein Parameter automatisch festgelegt wird, wenn Sie ihn nicht selbst festlegen.) Die von Ihnen am meisten benutzte Standardvorgabe ist wahrscheinlich die momentane Matrix. Wenn Sie keine bestimmte Matrix im Alpha-Register spezifizieren und *das Alpha-Register leer ist*, wird die Standardvorgabe automatisch zur momentanen Matrix.

Die folgende Tabelle enthält zur Interpretation der Parameter die Konventionen für Matrix-Operationen, die bis zu drei Matrixnamen verlangen.

Matrix-Spezifikationen

Inhalt des Alpha-Registers	spezifizierte Matrizen
A,B,C	A, B, C
A,B	A, B, B
A	A, A, A
A,,B	A, A, B
,A,B	momentane, A, B
,A	momentane, A, A
,,A	momentane, momentane, A
X,A,B	X-Register, A, B
X,A	X-Register, A, A
A,X	A, X-Register, A
A,,X	A, A, A (ignoriert X)
X	X-Register, momentane, momentane
(leer)	moment., moment., moment.

Fehlermeldungen

Für Fehlermeldungen, die Sie in diesem Handbuch nicht finden, schlagen Sie im Benutzerhandbuch Ihres HP-41 nach.

ALPHA DATA wird angezeigt, wenn die spezifizierte Matrix Alpha-Zeichen enthält. Die Matrix wird nicht verändert.

DATA ERROR wird angezeigt, wenn der Wert im X-, Y- oder Z-Register nicht zulässig ist.

DATA ERROR X wird angezeigt, wenn der Wert im X-Register nicht zulässig ist.

DATA ERROR Y wird angezeigt, wenn der Wert im Y-Register nicht zulässig ist.

DIM ERROR wird angezeigt, wenn die Dimension der spezifizierten Matrix für die momentane Operation nicht zulässig ist.

END OF ARRAY wird angezeigt, wenn Sie eine Funktion ausführen wollen, die den Matrix-Zeiger verwendet und er auf ein Element außerhalb der definierten Matrixgrenzen zeigt.

NAME ERROR wird angezeigt, wenn ein falscher Matrixname angegeben wird (z.B. "X") oder wenn die Anzahl von verschiedenen Matrixnamen für eine bestimmte Funktion nicht zulässig ist.

NO ROOM wird angezeigt, wenn zum Abspeichern einer Matrix im erweiterten Speicher nicht genügend Speicherplatz zur Verfügung steht.

NO X-MEMORY wird angezeigt, wenn Sie versuchen, eine Matrix im erweiterten Speicher anzulegen, den Ihr Taschenrechner nicht aufweist (z.B. ein HP-41C/CV ohne erweiterte Funktionen/Speicher-Modul HP 82180A).

NONEXISTENT wird angezeigt, wenn nicht genügend Speicherregister im Hauptspeicher verfügbar sind, um die Matrix abzuspeichern. Vergrößern Sie den Speicherbereich mit `[SIZE]nnn`.

NOT ARRAY FL wird angezeigt, wenn Sie versuchen, eine Matrix-Operation auf eine Datei im erweiterten Speicher anzuwenden, die keine Matrix-Datei ist.

NOT CPX wird angezeigt, wenn Sie `[CMEDIT]` in Verbindung mit einer reellen Matrix ungerader Ordnung benutzen.

TRY AGAIN wird angezeigt, wenn Sie `[MATDIM]` mit weniger als zwei verfügbaren Registern als Programmspeicher benutzen. *Verkleinern* Sie entweder den Speicherbereich für Speicherregister oder benutzen Sie `[CLP]`, um ein Programm zu löschen.

UNDEF ARRAY wird angezeigt, wenn Sie eine Matrix-Funktion ausführen, die eine Matrix-Spezifikation benötigt und dabei das Alpha-Register keine gültige Matrix-Spezifikation enthält.

Abspeichern und Abrufen einzelner Matrixelemente

Der Matrix-Editor ermöglicht Ihnen das Abspeichern und Abrufen von Matrixelementen. Sie können in einem Programm folgende Funktionen zum Ändern einzelner Matrixelemente verwenden.

Ein einzelnes Element wird durch die Angabe des Wertes *iii.jjj* für die *ite* Zeile und *jte* Spalte identifiziert. Sie können im *i*-Index führende und im *j*-Index nachfolgende Nullen weglassen.

Einzelner Zugriff auf Matrixelemente

Um ein einzelnes Element abzuspeichern oder anzuzeigen, geben Sie zuerst den Wert *iii.jjj* des Element-Zeigers (Zeile und Spalte) an und speichern dann den Wert (aus dem X-Register) des Elements ab bzw. zeigen den Wert des Elements im X-Register an. Um zu einem anderen Element weiterzuschalten, inkrementieren Sie den Zeiger oder setzen ihn zurück.

Der Wert des Zeigers definiert das *momentane Element*.

Einstellen und Abrufen des Zeigers

Funktion	Wirkung
MSIJA (Einstellen des Zeigers durch Alpha)	Stellt den Element-Zeiger der spezifizierten Matrix auf <i>iii.jjj</i> ein. Eingabe: Matrix-Name in Alpha-Reg. <i>iii.jjj</i> in X-Reg.
MSIJ (Einstellen des Zeigers)	Stellt den Element-Zeiger der momentanen Matrix auf <i>iii.jjj</i> ein. Eingabe: <i>iii.jjj</i> in X-Reg.
MRIJA (Abrufen des Zeigers durch Alpha)	Ruft den Element-Zeiger der spezifizierten Matrix zur Anzeige im X-Reg. ab. Eingabe: Matrix-Name in Alpha-Reg. Ausgabe: <i>iii.jjj</i> in X-Reg.
MRIJ (Abrufen des Zeigers)	Ruft den Element-Zeiger der momentanen Matrix zur Anzeige im X-Reg. ab. Ausgabe: <i>iii.jjj</i> in X-Reg.

Die folgenden Funktionen inkrementieren und dekrementieren den Element-Zeiger zeilenweise (*iii*) bzw. spaltenweise (*jjj*). Wenn der Zeiger das Ende einer Spalte (mit dem *i*-Index) oder das Ende einer Zeile (mit dem *j*-Index) erreicht, wird der Spalten- bzw. Zeilenindex inkrementiert bzw. dekrementiert und Flag 09 gesetzt. Wenn ein Index die Dimension der Matrix überschreitet, werden Flag 09 und 10 gesetzt. Diese Funktionen setzen oder löschen immer Flag 09 und 10. Wenn die o.a. Bedingungen nicht auftreten, werden die Flags nach jeder Ausführung zurückgesetzt.

Inkrementieren und Dekrementieren des Zeigers

Funktion	Wirkung
I+	<i>iii</i> des Zeigers wird um 1 inkrementiert.
I-	<i>iii</i> wird um 1 dekrementiert.
J+	<i>jjj</i> des Zeigers wird um 1 inkrementiert.
J-	<i>jjj</i> wird um 1 dekrementiert.

Abspeichern und Abrufen eines Elements

Funktion	Wirkung
MS (Matrix abspeichern)	Speichert den Wert des X-Reg. im momentanen Element der momentanen Matrix ab. Eingabe: Wert in X-Reg.
MR (Matrix abrufen)	Ruft den Wert des momentanen Elements der momentanen Matrix zur Anzeige im X-Reg. ab. Ausgabe: Wert in X-Reg.

Sequentieller Zugriff auf Matrixelemente

Die folgenden Funktionen bieten einen schnelleren und automatisierteren Zugriff auf Matrixelemente. Sie beinhalten das Abspeichern bzw. das Abrufen von Matrixelementen und anschließendes automatisches Inkrementieren oder Dekrementieren des i - oder j -Index, sodaß der Zeiger auf das nächste Element ausgerichtet ist.

Abspeichern und Abrufen eines Elements

Funktion	Wirkung
MSC+ (Spaltenweises Abspeichern)	Speichert den Wert des X-Reg. im momentanen Element ab und inkrementiert den Zeilenindex. Eingabe: Wert in X-Reg.
MSR+ (Zeilenweises Abspeichern)	Speichert den Wert des X-Reg. im momentanen Element ab und inkrementiert den Spaltenindex. Eingabe: Wert in X-Reg.
MRC+ (Spaltenweises Abrufen)	Ruft den Wert des momentanen Elements zur Anzeige im X-Reg. ab und inkrementiert den Zeilenindex. Ausgabe: Wert in X-Reg.
MRR+ (Zeilenweises Abrufen)	Ruft den Wert des momentanen Elements zur Anzeige im X-Reg. ab und inkrementiert den Spaltenindex. Ausgabe: Wert in X-Reg.
MRC- (Spaltenweises Abrufen (rückwärts))	Ruft das momentane Element zur Anzeige im X-Reg. ab und dekrementiert den Zeilenindex. Ausgabe: Wert in X-Reg.
MRR- (Zeilenweises Abrufen (rückwärts))	Ruft das momentane Element zur Anzeige im X-Reg. ab und dekrementiert den Spaltenindex. Ausgabe: Wert in X-Reg.

Wenn der Zeiger das Ende der Spalte bzw. Zeile erreicht hat, wird der Zeigerindex auf die nächste (oder vorhergehende) Spalte bzw. Zeile erhöht (oder erniedrigt). Wenn der Zeigerindex auf einen Wert außerhalb der Matrixdimension erhöht (oder erniedrigt) wird, können Sie den Zeiger mit diesen Funktionen nicht mehr zurücksetzen. Verwenden Sie in diesem Fall **MSIJ** oder **MSIJA**.

Mit der folgenden Tastenfolge legen Sie die Matrix **ABC** (im erweiterten Speicher) an.

$$ABC = \begin{bmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$$

Tastenfolge

FIX 4

ALPHA **ABC** **ALPHA**

2.003 **XEQ** **MATDIM**

0 **XEQ** **MSIJ**

5 **XEQ** **MSR+**

6 **XEQ** **MSR+**

7 **XEQ** **MSR+**

8 **XEQ** **MSR+**

9 **XEQ** **MSR+**

10 **XEQ** **MS**

SF 08

XEQ **MEDIT**

Anzeige

2.0030

0.0000

5.0000

6.0000

7.0000

8.0000

9.0000

10.0000

1:1 = 5.0000

1:2 = 6.0000

1:3 = 7.0000

2:1 = 8.0000

2:2 = 9.0000

2:3 = 10.0000

Stellt das hier benutzte Anzeigeformat ein.

Matrixname im erweiterten Speicher.

Dimensioniert Matrix **ABC** zu 2 Zeilen × 3 Spalten.

Setzt den Element-
Zeiger auf 1.001.

Gibt Element ein, setzt
Zeiger auf nächste
Spalte für nächste Ein-
gabe und setzt Flag 09.

Der Zeiger geht auto-
matisch zur 2. Zeile.

Dadurch können die
Elemente im Editor
nur angesehen wer-
den. Wenn Sie einen
Drucker angeschlossen
haben, ist dies eine
schnellere Möglichkeit,
die Matrixelemente
anzusehen.

Anzeigen von **ABC**.
(**ABC** steht immer
noch im Alpha-Regi-
ster.) Wenn Sie keinen
Drucker angeschlossen
haben, drücken Sie
[R/S], um die Ele-
mente nacheinander
ansetzen zu können.
Editor wird beendet.

Matrixfunktionen

In diesem Abschnitt werden die nicht zum Abspeichern und zum Abrufen benutzten Matrixfunktionen kurz definiert. Auf Seite 59 befindet sich eine Funktionstabelle, die alle in diesem Paket vorhandenen Matrixfunktionen enthält.

Beachten Sie, daß die Anwendung der meisten dieser Funktionen auf Matrizen mit Alpha-Zeichen oder komplexe Matrizen nicht sinnvoll ist. Außer der Funktion **CMEDIT** behandeln alle anderen eine komplexe Matrix wie eine reelle. Schlagen Sie für Informationen über das Benutzen dieser Funktionen mit komplexen Matrizen im Abschnitt "Arbeiten mit komplexen Matrizen" nach.

Matrix-Arithmetik

Die arithmetischen Operationen betreffende Matrixfunktionen ermöglichen skalare Addition, Subtraktion, Multiplikation und Division sowie Matrixmultiplikation. Diese Funktionen können zwei Matrizen oder eine Matrix und einen Skalar als Operanden führen. Die Matrizen müssen nicht die gleichen Dimensionen aufweisen, jedoch die gleiche Anzahl von Elementen haben. Dasselbe gilt für die Ergebnismatrix. (Beachten Sie, daß die in der unten aufgeführten Tabelle benutzte i - j Bezeichnung gleiche Dimensionen der Matrizen voraussetzt, ansonsten ist die i - j Bezeichnung nicht anwendbar.)

Das Programm für Matrixmultiplikation berechnet jedes neue Element, indem die Zeilenelemente der ersten Matrix mit den Spaltenelementen der zweiten multipliziert werden und die Summe der Produkte dem neuen Element zugeordnet wird. Die Anzahl der Spalten in der ersten Matrix muß gleich der Zeilenanzahl in der zweiten Matrix sein. Die Ergebnismatrix muß die gleiche Zeilenanzahl wie in der ersten Matrix und die gleiche Spaltenanzahl wie in der zweiten Matrix aufweisen.

Ein eventueller skalarer Operand muß sich im X-Register befinden, und X muß im Alpha-Register spezifiziert sein.

Funktion	Wirkung
MAT+ (Matrix-Addition)	<p style="text-align: center;">Skalare Arithmetik</p> <p>Addiert einen Skalar oder ein Matrixelement zu jedem Element.</p> <p>Eingabe: <i>Matrixname A</i> oder X, <i>Matrixname B</i> oder X, <i>Name C</i> der <i>Ergebnismatrix</i> in Alpha-Reg.</p> <p>Ausgabe: $c_{ij} = a_{ij} + x$ oder $c_{ij} = x + b_{ij}$ oder $c_{ij} = a_{ij} + b_{ij}$ für alle i, j in C.</p>
MAT- (Matrix-Subtraktion)	<p>Subtrahiert einen Skalar oder ein Matrixelement von jedem Element.</p> <p>Eingabe: <i>Matrixname A</i> oder X, <i>Matrixname B</i> oder X, <i>Name C</i> der <i>Ergebnismatrix</i> in Alpha-Reg.</p> <p>Ausgabe: $c_{ij} = a_{ij} - x$ oder $c_{ij} = x - b_{ij}$ oder $c_{ij} = a_{ij} - b_{ij}$ für alle i, j in C.</p>

Funktion	Wirkung
<p>MAT* (skalare Matrix-Multiplikation)</p>	<p>Multipliziert jedes Element mit einem Skalar oder einem Matrixelement. Eingabe: Matrixname A oder X, Matrixname B oder X, Name C der Ergebnismatrix in Alpha-Reg. Ausgabe: $c_{ij} = a_{ij} \times x$ oder $c_{ij} = x + b_{ij}$ oder $c_{ij} = a_{ij} \times b_{ij}$ für alle i, j in C.</p>
<p>MAT/ (Matrix-Division)</p>	<p>Dividiert jedes Element durch einen Skalar oder ein Matrixelement. Eingabe: Matrixname A oder X, Matrixname B oder X, Name C der Ergebnismatrix in Alpha-Reg. Ausgabe: $c_{ij} = a_{ij} \div x$ oder $c_{ij} = x \div b_{ij}$ oder $c_{ij} = a_{ij} \div b_{ij}$ für alle i, j in C.</p>
<p>M*M (Matrizenmultiplikation)</p>	<p style="text-align: center;">Nichtskalare Arithmetik</p> <p>Berechnet jedes neue Element i, j durch Multiplikation der iten Zeile in A mit der jten Spalte in B. Eingabe: Matrixname A, Matrixname B, Name C der Ergebnismatrix in Alpha-Reg., wobei C von A und B verschieden sein muß. Ausgabe: $c_{ij} = \sum_{k=1}^p a_{ik} \times b_{kj}$, wobei A p Spalten und B p Zeilen hat.</p>

Wichtige Matrix-Operationen

Die wichtigen Matrix-Operationen sind: Inversion, Bestimmung der Determinante, Transposition und Lösen eines linearen Gleichungssystems.

Ein lineares Gleichungssystem

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

kann durch die Matrixgleichung $\mathbf{AX} = \mathbf{B}$ dargestellt werden, wobei

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{und} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

A die Koeffizientenmatrix des Gleichungssystems, **B** die Spalten- bzw. Konstantenmatrix und **X** die Ergebnismatrix sind. (Matrix **B** wird nach Lösen des Gleichungssystems mit Matrix **X** überschrieben.)

Funktion	Wirkung
MDET (Determinante)	Berechnet die durch eine reelle quadratische Matrix gegebene Determinante. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Determinante in X-Reg. (Ersetzt Matrix durch LR-zerlegte Form).
MINV (<i>Inversion</i>)	Invertiert und ersetzt die gegebene quadratische Matrix. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Ersetzt Matrix durch die inverse.
MSYS (Gleichungssystem)	Löst ein lineares Gleichungssystem. Eingabe: <i>Matrixname A, Matrixname B</i> in Alpha-Reg. Ausgabe: Ergebnismatrix <i>X</i> ersetzt <i>B</i> ; <i>A</i> wird durch die LR-zerlegte Form ersetzt.)
TRNPS (Transposition)	Transponiert und ersetzt die gegebene reelle Matrix. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Ersetzt Matrix durch die transponierte.

Hinweis: Sie können eine Matrix *A*, für die Sie die Determinante bestimmt oder ein Gleichungssystem gelöst haben, nicht transponieren und keines ihrer Elemente ändern, da die Funktionen **MDET** und **MSYS** Matrix *A* in die *LR-zerlegte Form* umgewandelt haben. (Schlagen Sie für weitere Informationen im Abschnitt "LR-Zerlegung" nach.) Sie können jedoch die ursprüngliche Form von *A* dadurch zurückerhalten, indem Sie die LR-zerlegte Form *zweimal* invertieren (führen Sie **MINV** zweimal aus). Die LR-Zerlegung beeinflusst *nicht* die Ausführung der Funktionen **MINV**, **MSYS** und **MDET**.

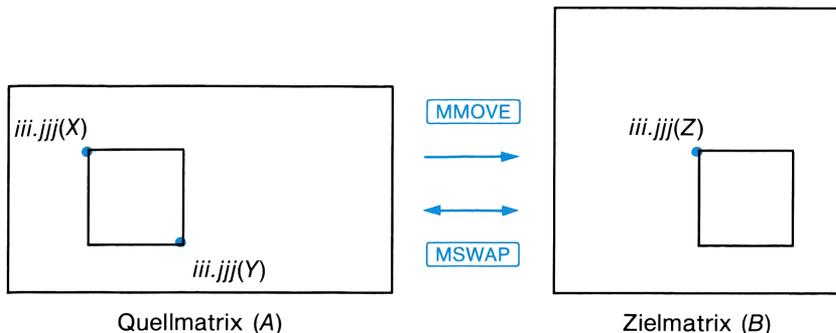
Andere Matrixfunktionen ("Dienstfunktionen")

Die restlichen Matrixfunktionen, auch *Dienstfunktionen* genannt, sind Funktionen zum Kopieren und Austauschen von Matrixteilen sowie zum Ausführen verschiedener zusätzlicher arithmetischer Operationen: Summenbildung, Normen, Optimierung und Matrixreduktion.

Verlagern und Austauschen von Matrixteilen

Funktion	Wirkung
C<>C (vertausche Spalten)	Vertauscht die Spalten k und l in einer Matrix. Eingabe: <i>Matrixname</i> in Alpha-Reg. $kkk.lll$ in X-Reg.
R<>R (vertausche Zeilen)	Vertauscht die Zeilen k und l in einer Matrix. Eingabe: <i>Matrixname</i> in Alpha-Reg. $kkk.lll$ in X-Reg.
MMOVE (verlagere Matrix)	Kopiert eine in der Quellmatrix durch Zeiger definierte Untermatrix in den durch einen Zeiger definierten Teil der Zielmatrix. Eingabe: <i>Name A der Quellmatrix, Name B der Zielmatrix</i> in Alpha-Reg. in X-Reg: $iii.jjj$ für Anfangselement in A ; in Y-Reg: $iii.jjj$ für letztes Element in A ; in Z-Reg: $iii.jjj$ für Anfangselement in B .
MSWAP (Matrix austauschen)	Vertauscht die in der Quellmatrix durch Zeiger definierte Untermatrix mit dem durch einen Zeiger definierten Teil der Zielmatrix. Eingabe: <i>Matrixname A, Matrixname B</i> in Alpha-Reg. in X-Reg: $iii.jjj$ für Anfangselement in A ; in Y-Reg: $iii.jjj$ für letztes Element in A ; in Z-Reg: $iii.jjj$ für Anfangselement in B .

Wenn die Funktionen **MMOVE** und **MSWAP** ausgeführt werden, wobei A und B dieselbe Matrix darstellen und die Quell-Untermatrix und Ziel-Untermatrix sich überlappen, werden die Elemente in folgender Reihenfolge verarbeitet: umgekehrte Reihenfolge der Spalten und umgekehrte Reihenfolge der Zeilenelemente in jeder Spalte.



Wenn im X-Register eine Eingabe in der Form *iii.jjj* erwartet wird, interpretieren diese Funktionen einen Null-Index als 1. (0 allein wird zu 1.001.) Dies gilt für die von **MMOVE** und **MSWAP** im X- und Z-Register erwarteten *iii.jjj*-Werte, jedoch *nicht für den Zeigerwert im Y-Register*.

Bei der Eingabe aus dem Y-Register wird ein Nullwert für den *i*-Index als *m* (letzte Zeile) und ein Nullwert für den *j*-Index als *n* (letzte Spalte) interpretiert. Beispiel für eine 4×5 Matrix:

Y-Register	Zeigerwert
0.000	4.005
3.000	3.005
0.003	4.003

Durch diese Konvention können Sie sehr einfach Matrizen vollständig kopieren (oder austauschen): Durch Löschen des Stacks (**CLST**) oder durch Eingabe von drei Nullen spezifizieren Sie die Elemente 1.001 (X) und *mmm.nnn* (Y) für die erste Matrix sowie das Element 1.001 (Z) für die zweite Matrix, d.h., Sie definieren zwei vollständige Matrizen.

Durch die folgenden Anweisungen würden Sie die Matrix **R0** mit un spezifizierten Dimensionen in eine neue Matrix **R30** kopieren:

```

ALPHA R0 ALPHA
XEQ DIM?
ALPHA R30 ALPHA
XEQ MATDIM
ALPHA R0,R30 ALPHA
XEQ CLST
XEQ MMOVE

```

Verschiedene arithmetische Funktionen

Funktion	Wirkung
Maxima und Minima	
MAX (<i>Maximum</i>)	Bestimmt das Element einer Matrix mit maximalem Wert. Stellt den Element-Zeiger auf das Element ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: maximaler Wert in X-Reg.
MIN (<i>Minimum</i>)	Bestimmt das Element einer Matrix mit minimalem Wert. Stellt den Element-Zeiger auf das Element ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: minimaler Wert in X-Reg.
MAXAB (<i>Max. Absolutbetrag</i>)	Bestimmt max. Absolutbetrag in Matrix. Stellt Zeiger auf das Element ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: max. Absolutbetrag in X-Reg.
CMAXAB (<i>max. Absolutbetrag in Spalte</i>)	Bestimmt max. Absolutbetrag in kter Spalte. Stellt Zeiger auf das Element ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. <i>kkk</i> in X-Reg. Ausgabe: max. Absolutbetrag in X-Reg.
RMAXAB (<i>max. Absolutbetrag in Zeile</i>)	Bestimmt max. Absolutbetrag in kter Zeile. Stellt Zeiger auf das Element ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. <i>kkk</i> in X-Reg. Ausgabe: max. Absolutbetrag in X-Reg.
Normen	
CNRM (<i>Spaltensummennorm/ Eins-Norm</i>)	Bestimmt die größte Summe der Absolutbeträge der Elemente in jeder Spalte der Matrix. Stellt den Zeiger auf das erste Element in der Spalte mit der größten Summe ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Spaltensummennorm in X-Reg.
FNRM (<i>Frobenius-Norm</i>)	Bestimmt die Quadratwurzel der Summe der Quadrate der Werte aller Elemente einer Matrix. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Frobenius-Norm in X-Reg.

Verschiedene arithmetische Funktionen (Fortsetzung)

Funktion	Wirkung
RNRM (Zeilen- summennorm/ Unendlich-Norm)	Bestimmt die größte Summe der Absolutbeträge der Elemente in jeder Zeile der Matrix. Stellt den Zeiger auf das erste Element in der Zeile mit der größten Summe ein. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Zeilensummennorm in X-Reg.
SUM	Summen
	Summiert alle Matrixelemente. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Summe in X-Reg.
SUMAB (Summe aller Absolut- beträge)	Summiert die Absolutbeträge aller Matrixelemente. Eingabe: <i>Matrixname</i> in Alpha-Reg. Ausgabe: Summe der Absolutbeträge in X-Reg.
CSUM (Spalten- summe)	Bestimmt die Summe jeder Spalte und speichert sie im Ergebnisvektor ab. Eingabe: <i>Matrixname, Name der Ergebnismatrix</i> in Alpha-Reg. Anzahl der Elemente in Ergebnismatrix muß mit Anzahl der Spalten in Eingabe-Matrix übereinstimmen.
RSUM (Zeilen- summe)	Bestimmt die Summe jeder Zeile und speichert sie im Ergebnisvektor ab. Eingabe: <i>Matrixname, Name der Ergebnismatrix</i> in Alpha-Reg. Anzahl der Elemente in Ergebnismatrix muß mit Anzahl der Zeilen in Eingabe-Matrix übereinstimmen.
	Sonstige Funktionen
YC+C (<i>Y mal Spalte plus Spalte</i>)	Multipliziert jedes Matrixelement in Spalte <i>k</i> mit dem im Y-Reg. stehenden Wert, addiert das Produkt zum entsprechenden Element in Spalte <i>l</i> und ändert dadurch die Elemente in Spalte <i>l</i> . D.h., es ändert a_{il} in $a_{il} + y \times a_{ik}$. Eingabe: <i>Matrixname</i> in Alpha-Reg. <i>kkk.lll</i> in X-Reg. <i>y</i> in Y-Reg.
PIV (<i>Pivot</i>)	Bestimmt den Wert des Pivotelements in Spalte <i>k</i> ; d.h. den maximalen Absolutbetrag eines Elements auf oder unterhalb der Diagonalen. Eingabe: <i>Matrixname</i> in Alpha-Reg. <i>kkk</i> in X-Reg. Ausgabe: Pivot-Wert in X-Reg.; Stellt Zeiger auf Pivotelement ein.

Verschiedene arithmetische Funktionen (Fortsetzung)

Funktion	Wirkung
<p>R>R? (vergleiche Zeilen)</p>	<p>Vergleicht die Elemente in der <i>k</i>ten und <i>l</i>ten Zeile. Falls (und nur falls) das erste ungleiche Element in <i>k</i> größer als das entsprechende Element in <i>l</i> ist, ergibt sich ein positiver Vergleich (entsprechend der Programmierregel "do if true"). Eingabe: <i>Matrixname</i> in Alpha-Reg. <i>kkk.lll</i> in X-Reg. Ausgabe: YES, falls das erste ungleiche Element in Zeile <i>k</i> größer als das entsprechende in Zeile <i>l</i> ist. NO in allen übrigen Fällen.</p>
<p>AIP (Alpha-Ab-ruf des ganz-zahligen Teils)</p>	<p>Fügt den ganzzahligen Teil der im X-Reg. stehenden Zahl dem Inhalt des Alpha-Reg. an. Für $x < 0$ fügt AIP den Absolutbetrag an.</p>
<p>MP (Alpha-Ab-ruf der Matrix-Ein-gabeaufforde-rung)</p>	<p>Fügt die Matrix-Eingabeaufforderung <i>rrr.ccc</i> = dem Inhalt im Alpha-Reg. an.</p>

Arbeiten mit komplexen Matrizen

Wenn Sie mit komplexen Matrizen arbeiten, ist es sehr wichtig sich zu erinnern, daß im Taschenrechner eine komplexe Matrix einfach durch eine reelle Matrix mit der vierfachen Anzahl von Elementen dargestellt wird. Nur das Programm MATRX und der Editor für komplexe Matrizen (**CMEDIT**) "erkennen" eine komplexe Matrix und verarbeiten die Elemente entsprechend. Alle anderen Funktionen behandeln die Real- und Imaginärteile der komplexen Zahlen als *getrennte* reelle Elemente.

Darstellung komplexer Matrixelemente

Intern wird eine komplexe Matrix durch die doppelte Anzahl von Zeilen und Spalten, als "normalerweise" notwendig wäre, repräsentiert.

Die komplexe Zahl $100 + 200i$ wird folgendermaßen abgespeichert:

$$\begin{bmatrix} 100 & -200 \\ 200 & 100 \end{bmatrix}$$

Die 2×1 komplexe Matrix

$$\begin{bmatrix} 1 + 2i \\ 3 - 4i \end{bmatrix} \text{ wird abgespeichert als } \begin{bmatrix} 1 & -2 \\ 2 & 1 \\ 3 & 4 \\ -4 & 3 \end{bmatrix}$$

Es gibt eine wichtige Ausnahme: die Spaltenmatrix (ein Vektor) in einem Gleichungssystem.

Lösen eines komplexen Gleichungssystems. Das Arbeiten mit komplexen Matrizen ist am einfachsten, wenn Sie das Programm **MATRIX** benutzen. Es dimensioniert automatisch komplexe Ein- und Ausgabematrizen. Dagegen kann **MSYS** kompliziertere Gleichungssysteme als **MATRIX** lösen.

Weiterhin kann eine komplexe Ergebnismatrix des Programms **MATRIX** bei nachfolgenden Operationen nicht *außerhalb* von **MATRIX** verwendet werden. Ursache dafür ist, daß **MATRIX** eine komplexe *Spaltenmatrix* anders als $2m \times 2$ dimensioniert. *Stattdessen* verwendet es die Dimensionen $2m \times 1$, wobei die Real- und Imaginärteile einer Zahl nacheinander in der Einzelspalte stehen.

Diese Form hat den Vorteil, Speicherplatz zu sparen und die Ausführungsgeschwindigkeit zu vergrößern. Der Editor für komplexe Matrizen sowie **MSYS** können ebenfalls diese $2m \times 1$ -Form verarbeiten, obwohl sie diese Form nicht voraussetzen. Das bedeutet, daß Sie die Funktion **MSYS** auf ein Matrixsystem aus **MATRIX** anwenden können.

Sie können aus einer komplexen $2m \times 2$ Spaltenmatrix die $2m \times 1$ -Form erhalten, indem Sie die Matrix zuerst transponieren, dann auf $1 \times 2m$ neu dimensionieren und zuletzt zurücktransponieren. Es gibt jedoch keinen einfachen umgekehrten Weg.

Zugriff auf komplexe Elemente. Wenn Sie den Editor für komplexe Matrizen (**CMEDIT**) oder den Editor im Programm **MATRIX** benutzen, können Sie auf komplexe Elemente wie auf komplexe Zahlen zugreifen. Andernfalls (wie z.B. bei Verwendung von Zeiger setzenden Funktionen) müssen Sie auf komplexe Elemente wie auf reelle, entsprechend dem $2m \times 2n$ Schema abgespeicherte, Elemente zugreifen.

Speicherplatz im Speicher. Da die benötigte Größe für komplexe Matrizen viermal größer als die tatsächliche Anzahl komplexer Elemente ist (eine komplexe $m \times n$ Matrix wird als $2m \times 2n$ dimensioniert), wird die Anzahl der benötigten Speicherregister im Vergleich zu einer reellen Matrix mit gleicher Elementenzahl viermal größer. Mit anderen Worten, stellen Sie sich die Speichergröße für eine komplexe Matrix entsprechend ihrer **MATDIM** - bzw. **DIM?** -Dimensionen vor und nicht entsprechend der Anzahl ihrer komplexen Elemente.

Benutzen von Funktionen mit komplexen Matrizen

Die meisten Funktionen liefern bei Anwendung auf komplexe Matrizen kein sinnvolles Ergebnis: Da diese Funktionen die Real- und Imaginärteile einer komplexen Zahl nicht als eine Zahl erkennen, entspricht das Ergebnis nicht den komplexen Elementen.

Zulässige komplexe Operationen. Bestimmte Matrixfunktionen lassen sich gleich gut auf reelle und komplexe Matrizen anwenden. Diese sind:

- MSYS** Lösen von Gleichungssystemen
- MINV** Matrix-Inversion
- MAT+** Matrix-Addition
- MAT-** Matrix-Subtraktion
- MAT*** skalare Matrix-Multiplikation, *jedoch nur mit einem reellen Skalar im X-Reg.*
- M*M** Matrizenmultiplikation

Sowohl die Eingabe-Matrizen als auch die Ergebnismatrix müssen komplex sein.

LR-Zerlegung

Die *LR-Zerlegung* ist eine unerkennbare, veränderte Form der Matrix, die oft Alpha-Zeichen enthält. Diese Zerlegung findet statt bei der Bestimmung der:

- Lösung eines Gleichungssystems (**MSYS**; **SE** im MATRX-Programm).
- Determinante (**MDET**; **DT** im MATRX-Programm).
- Inversion (**MINV**; **I** im MATRX-Programm).

Die ersten beiden Operationen wandeln die Matrix in die LR-Form um und *behalten diese Form bei*. Bei der Inversion dagegen liegt die Matrix am Ende der Operation in invertierter Form vor. Wenn Sie Funktionen benutzen, die eine LR-Zerlegung herbeiführen, müssen Sie folgende Punkte beachten:

- Sie können eine LR-zerlegte Matrix nicht editieren, es sei denn, Sie ändern alle Elemente. (Schlagen Sie für weitere Informationen im Abschnitt "Editieren und Ansehen LR-zerlegter Matrizen" nach.)
- Sie können keine Operation (außer **MINV**) auf eine LR-zerlegte Matrix anwenden, da dadurch der LR-Status der Matrix gelöscht würde und die Matrix nicht mehr als LR-zerlegt erkennbar wäre. Operationen mit dieser Eigenschaft sind: **R<>R**, **C<>C**, **MS**, **MSR+**, **MSC-**, **MSC+**, **MMOVE** (innerhalb Matrix), **MSWAP** und **TRNPS**.
- Sie müssen sehr vorsichtig sein, wenn Sie eine LR-zerlegte Matrix *ansehen*. Gewisse Operationen können ohne Ihr Wissen Elemente ändern. (Schlagen Sie dazu im Abschnitt "Editieren und Ansehen LR-zerlegter Matrizen" nach).
- Die LR-Zerlegung zerstört die ursprüngliche Form der Matrix. Wenn Sie daher die Funktion **MSYS** oder **MDET** ausführen und anschließend die Matrix ansehen wollen (**A** im Programm **MATRIX**), werden Sie nur die *veränderte, zerlegte Form* vorfinden.
- Sie können eine LR-zerlegte Matrix nicht transponieren (**TRNPS**; **B** im Programm **MATRIX**). Eine LR-Zerlegung *hindert Sie nicht, eine Matrix zu invertieren, die Determinante zu bestimmen oder ein Gleichungssystem zu lösen*, da diese Operationen die LR-Zerlegung ohnehin benötigen.

Umkehrung der LR-Zerlegung. Um eine Matrix wieder in ihre ursprüngliche Form zu bringen, *invertieren Sie sie einfach zweimal*. Natürlich muß die Matrix invertierbar (nichtsingulär) sein. Das Ergebnis kann wegen Rundungen etwas von der Ausgangsform abweichen.

Editieren und Ansehen LR-zerlegter Matrizen. LR-zerlegte Matrizen werden anders als die ursprüngliche Form abgespeichert:

- Bestimmte Elemente enthalten Alpha-Daten.
- Das Status-Register für die Matrix wird dahingehend abgeändert, daß es die LR-Zerlegung anzeigt.

54 Matrix-Operationen

Das Editieren *irgendeines* Elementes hat das Löschen des LR-Flags im Status-Register zur Folge. Dadurch wird die LR-Zerlegung unerkennbar. Deshalb müssen Sie, wenn Sie ein Element editieren wollen, alle Elemente editieren, damit Sie die Matrix weiter verwenden können. Beachten Sie, daß die Matrix nach dieser Ausführung nicht länger in LR-zerlegter Form vorliegt.

Sie können die Elemente einer LR-zerlegten Matrix folgendermaßen *ansetzen*:

- Drücken Sie im Hauptmenü **[A]** (A ansehen), um einzelne Elemente anzusehen (ohne sie dabei zu ändern).
- Setzen Sie vor der Ausführung von **[MEDIT]** oder **[CMEDIT]** Flag 08. Dadurch können Sie die Elemente ansehen, ohne sie dabei zu ändern.

Beispiele

Bestimmen Sie die Determinante der invertierten, transponierten, gegebenen Matrix:

$$\begin{bmatrix} 6 & 3 & -2 \\ 1 & 4 & -3 \\ 2 & 3 & -1 \end{bmatrix}$$

Die Größe des Datenspeichers muß mindestens 10 Register betragen (**[SIZE]** 010).

Tastensequenz

[FIX] 4

[XEQ] **[SIZE]** 010

[ALPHA] **R0** **[ALPHA]**

3.003 **[XEQ]** **[MATDIM]**

[CF] 08

Anzeige



Stellt das hier benutzte Format ein.

Benennt die im Hauptspeicher von R_{00} – R_{10} abzulegende Matrix als **R0**.

Dimensioniert **R0** zu 3×3 .

Setzt Editor in den Edit-Modus.

Tastensequenz

XEQ MEDIT

6 R/S
 3 R/S
 2 CHS R/S
 1 R/S
 4 R/S
 3 CHS R/S
 2 R/S
 3 R/S
 1 CHS R/S
 XEQ TRNPS
 XEQ MINV
 XEQ MDET

Anzeige

1:1 = ?

1:2 = ?

1:3 = ?

2:1 = ?

2:2 = ?

2:3 = ?

3:1 = ?

3:2 = ?

3:3 = ?

0.0400

Der Matrix-Editor fragt Sie nach den neuen Elementen ab, indem er die alten Elemente bzw. den alten Inhalt der Register anzeigt.

Verläßt den Editor.

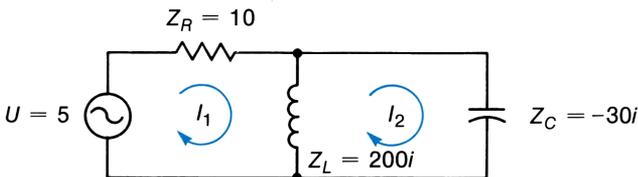
R0 wird transponiert.

R0 (bereits transponiert) wird invertiert.

Die Determinante der transponierten, invertierten Ausgangsmatrix.

Beachten Sie, daß wenn Sie die *Ausgangsmatrix* nach der Bestimmung der Determinante transponieren würden, Sie die Matrix zweimal invertieren müßten, um aus der LR-Zerlegung wieder die Ausgangsmatrix zu erhalten.

Berechnen Sie die im abgebildeten elektrischen Netzwerk angegebenen Ströme I_1 und I_2 . Die Impedanzen der Bauelemente sind in komplexer Form angegeben.



56 Matrix-Operationen

Dieses System kann durch eine komplexe Matrixgleichung

$$\begin{bmatrix} 10 + 200i & -200i \\ -200i & (200 - 30)i \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

bzw.

$$\mathbf{A} \mathbf{X} = \mathbf{B}.$$

dargestellt werden. Die Anzahl der Speicherregister muß mindestens 26 (**SIZE** 026) betragen, um die beiden komplexen Matrizen abspeichern zu können.

Tastensequenz

ALPHA **R** **ALPHA**
4.004 **XEQ** **MATDIM**

XEQ **CMEDIT**

10 **R/S** 200 **R/S**
0 **R/S** 200 **CHS** **R/S**
0 **R/S** 200 **CHS** **R/S**
0 **R/S** 170 **CHS** **R/S**

ALPHA **R** 17 **ALPHA**
4.002 **XEQ** **MATDIM**

XEQ **CMEDIT**

5 **R/S** 0 **R/S**
0 **R/S** 0 **R/S**

ALPHA **R,R** 17 **ALPHA**
XEQ **MSYS**

SF 08

Anzeige

4.0040

Dimensioniert die komplexe Koeffizientenmatrix **R0** zu 4×4 für 2 Zeilen und 2 Spalten. Es werden dafür 17 Register benötigt.

RE.1:1= ?

Editor für komplexe Matrizen.

RE.1:2= ?

Eingabe der Real- und Imaginärteile der Elemente in **R0**, die Koeffizientenmatrix (**A**).

RE.2:1= ?

RE.2:2= ?

-170.0000

Dimensioniert die Spaltenmatrix **R17** zu 4×2 für 2 komplexe Zeilen und 1 komplexe Spalte. Es werden dafür 9 Register benötigt.

4.0020

RE.1:1= ?

Editor für komplexe Matrizen.

RE.2:1= ?

0.0000

Eingabe der Real- und Imaginärteile der Elemente in **R17**, die Spaltenmatrix (**B**).

0.0000

Berechnet die Ergebnismatrix (**X**) und speichert sie in **R17** ab.

Stellt den Editor in den Modus "nur ansehen" ein.

Tastenfolge

ALPHA R 17 ALPHA
 XEQ CMEDIT
 R/S
 R/S
 R/S

Anzeige

RE.1:1=0.0372
 IM.1:1=0.1311
 RE.2:1=0.0437
 IM.2:1=0.1543

Zeigt das in R17 abgespeicherte komplexe Ergebnis für I_1 und I_2 an. Wenn Sie einen Drucker angeschlossen und Flag 08 vor der Ausführung von CMEDIT gesetzt haben, werden automatisch alle Elemente gedruckt.

Das Ergebnis lautet:

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0.0372 + 0.1311i \\ 0.0437 + 0.1543i \end{bmatrix}$$

Im letzten Beispiel ist ein Gleichungssystem mit zwei Unbekannten zu lösen. Dazu wird die Funktion MSYS benötigt.

Landwirt Maier hat die folgende Liste über den Verkauf von Kohl und Spargel in drei unterschiedlichen Wochen. Er kennt das Gesamtgewicht des innerhalb jeder Woche verkauften Gemüses, die Gesamteinnahmen für jede Woche und den Kilopreis der beiden Gemüse. Berechnen Sie das Gewicht des in jeder Woche verkauften Kohls und Spargels.

	1.Woche	2.Woche	3.Woche
Ges.-gewicht	274 kg	233 kg	331 kg
Ges.-einnahme	120.32 DM	112.96 DM	151.36 DM

Der Preis für Kohl beträgt 0.24 DM/kg, für Spargel 0.86 DM/kg.

Das folgende lineare Gleichungssystem beschreibt die beiden Unbekannten (das Gewicht von Kohl und Spargel) für alle drei Wochen. Die erste Zeile in der Konstantenmatrix enthält das Gewicht des Kohls, und die zweite Zeile enthält das Gewicht des Spargels in den drei Wochen. Da die Konstantenmatrix keine Spaltenmatrix ist, müssen Sie die Funktion MSYS und nicht die Funktion SE im Programm MATRX verwenden.

$$\begin{bmatrix} 1 & 1 \\ 0.24 & 0.86 \end{bmatrix} \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \end{bmatrix} = \begin{bmatrix} 274 & 233 & 331 \\ 120.32 & 112.96 & 151.36 \end{bmatrix}$$

Die Anzahl der verfügbaren Speicherregister muß mindestens 12 betragen (SIZE 012), um die beiden reellen Matrizen abspeichern zu können.

Tastenfolge

ALPHA R ALPHA
2.002 XEQ MATDIM

ALPHA R5 ALPHA
2.003 XEQ MATDIM

CF 08

XEQ MEDIT

274 R/S 233 R/S
331 R/S 120.32 R/S
112.96 R/S 151.36 R/S

ALPHA R ALPHA
XEQ MEDIT

1 R/S 1 R/S
.24 R/S .86 R/S

ALPHA R,R5 ALPHA

XEQ MSYS

SF 08

ALPHA R5 ALPHA
XEQ MEDIT
R/S
R/S
R/S
R/S
R/S
R/S
R/S

Anzeige

2.0020

2.0030

1:1 = ?

1:3 = ?

2:2 = ?

3.0010

1:1 = ?

2:1 = ?

3.0010

1:1 = 186.0000

1:2 = 141.0000

1:3 = 215.0000

2:1 = 88.0000

2:2 = 92.0000

2:3 = 116.0000

3.0010

Dimensioniert die Koeffizientenmatrix R0 zu 2 x 2.

Dimensioniert die Konstantenmatrix R5 zu 2 x 3.

Setzt Editor in den Editieren-Modus.

Ruft die momentane Matrix (R5) zum Editieren auf.

Eingabe in R5, die Konstantenmatrix.

Editieren von R0.

Eingabe von R0, die Koeffizientenmatrix.

Spezifiziert die Eingabematrizen (Koeffizienten- und Konstantenmatrix). Die Lösung wird in R5 abgespeichert.

Berechnet die Ergebnismatrix.

Stellt den Editor in den Modus "nur ansehen" ein.

Zeigt die Ergebnismatrix an.

Das Ergebnis lautet

	1.Woche	2.Woche	3.Woche
Kohl (kg)	186	141	215
Spargel (kg)	88	92	116

Alphabetisch geordnete Funktionstabelle

Außer bei einem Vermerk wird jede Funktion auf die im Alpha-Register spezifizierte(n) Matrix (Matrizen) ausgeführt. Wenn das Alpha-Register leer ist, wird die Funktion auf die momentane Matrix angewendet.

Matrixfunktionen

Funktionsname	Beschreibung
AIP (S. 50)	Fügt ganzzahligen Teil von x dem Alpha-Reg. an.
C<>>C (S. 46)	Vertauscht Spalten k und l .
CMAXAB (S. 48)	Bestimmt max. Absolutbetrag in Spalte k .
CMEDIT (S. 34)	Ruft Editor für komplexe Matrizen auf.
CNRM (S. 48)	Bestimmt Spaltensummennorm.
CSUM (S. 49)	Bestimmt Spaltensummen und legt sie in eine Zeilenmatrix ab.
DIM? (S. 33)	Gibt die $mmm.nnn$ -Dimensionen an.
FNRM (S. 48)	Bestimmt Frobenius-Norm.
I+ (S. 40)	Inkrementiert den Zeilenteil des Zeigers.
I- (S. 40)	Dekrementiert den Zeilenteil des Zeigers.
J+ (S. 40)	Inkrementiert den Spaltenteil des Zeigers.
J- (S. 40)	Dekrementiert den Spaltenteil des Zeigers.
M*M (S. 44)	Multiplikation zweier Matrizen (keine Skalare).
MAT+ (S. 43)	Addiert Skalar oder Matrix zu Matrix.
MAT- (S. 43)	Subtrahiert Skalar oder Matrix von Matrix.
MAT* (S. 44)	Multipliziert Matrix mit Skalar oder Matrix elementweise.
MAT/ (S. 44)	Dividiert Matrix durch Skalar oder Matrix elementweise.
MATDIM (S. 31)	Dimensioniert Matrix zu $mmm.nnn$.

Matrixfunktionen (Fortsetzung)

Funktionsname	Beschreibung
<code>MAX</code> (S. 48)	Bestimmt Element mit maximalem Wert.
<code>MAXAB</code> (S. 48)	Bestimmt max. Absolutbetrag der Elemente.
<code>MDET</code> (S. 45)	Bestimmt Determinante.
<code>MEDIT</code> (S. 34)	Ruft Editor für reelle Matrizen auf.
<code>MIN</code> (S. 48)	Bestimmt minimales Element.
<code>MINV</code> (S. 45)	Invertiert gegebene Matrix.
<code>MMOVE</code> (S. 46)	Kopiert Quellmatrix bzw. Untermatrix in Zielmatrix.
<code>MNAME?</code> (S. 37)	Zeigt Name der momentanen Matrix im Alpha-Reg. an.
<code>MP</code> (S. 50)	Fügt eine Matrixabfrage <code>rrr:ccc=</code> dem Alpha-Reg. an.
<code>MR</code> (S. 40)	Ruft momentanes Element ab.
<code>MRC+</code> (S. 41)	Ruft Elemente spaltenweise ab.
<code>MRC-</code> (S. 41)	Ruft Elemente spaltenweise rückwärts ab.
<code>MRIJ</code> (S. 40)	Ruft Zeigerwert <code>iii.jjj</code> der momentanen Matrix ab.
<code>MRIJA</code> (S. 40)	Ruft Zeigerwert <code>iii.jjj</code> der spezifizierten Matrix ab.
<code>MRR+</code> (S. 41)	Ruft Elemente zeilenweise ab.
<code>MRR-</code> (S. 41)	Ruft Elemente zeilenweise rückwärts ab.
<code>MS</code> (S. 40)	Speichert momentanes Element ab.
<code>MSC+</code> (S. 42)	Speichert momentanes Element spaltenweise ab.
<code>MSIJ</code> (S. 40)	Stellt Zeiger der momentanen Matrix auf Element <code>iii.jjj</code> ein.
<code>MSIJA</code> (S. 40)	Stellt Zeiger der spezifizierten Matrix auf Element <code>iii.jjj</code> ein.
<code>MSR+</code> (S. 41)	Speichert momentanes Element zeilenweise ab.
<code>MSWAP</code> (S. 46)	Vertauscht zwei Matrizen oder Untermatrizen.
<code>MSYS</code> (S. 45)	Löst ein Gleichungssystem.
<code>PIV</code> (S. 49)	Bestimmt max. Absolutwert auf oder unterhalb der Diagonalen.
<code>R<>R</code> (S. 46)	Vertauscht Zeilen <code>k</code> und <code>l</code> .
<code>R>R?</code> (S. 50)	Prüft elementweise, ob Zeile <code>k</code> größer als Zeile <code>l</code> ist.
<code>RMAXAB</code> (S. 48)	Bestimmt max. Absolutbetrag in Zeile <code>k</code> .
<code>RNRM</code> (S. 49)	Bestimmt Zeilensummennorm.
<code>RSUM</code> (S. 49)	Bestimmt Zeilensummen und speichert Sie in einer Spaltenmatrix ab.

Matrixfunktionen (Fortsetzung)

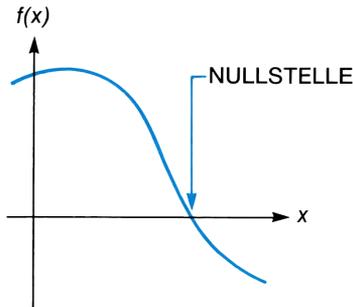
Funktionsname	Beschreibung
SUM (S. 49)	Bestimmt Summe aller Elemente.
SUMAB (S. 49)	Bestimmt Summe der Absolutbeträge aller Elemente.
TRNPS (S. 45)	Transponiert gegebene Matrix.
YC+C (S. 49)	Multipliziert jedes Element in Spalte k mit y -Wert und addiert Produkt zu Element in Spalte l . Ersetzt Element in Spalte l .

NULLSTELLEN EINER FUNKTION

Das Programm SOLVE bestimmt die Nullstellen einer Funktion der Form

$$f(x) = 0,$$

wobei x eine *reelle Nullstelle* darstellt.*

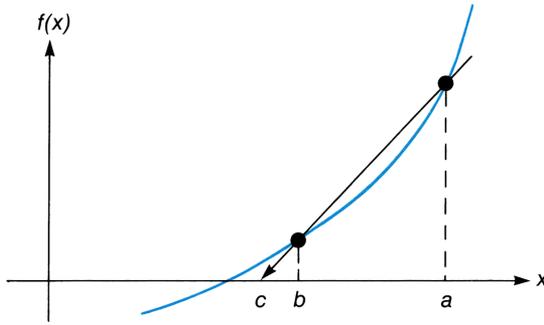


Das Programm SOLVE ([SOLVE](#)) stellt Ihnen eine fortgeschrittene Methode zur numerischen Bestimmung der reellen Nullstellen von einer Vielzahl von Funktionen zur Verfügung. Sie geben die Funktionsgleichung (in einem Programm) und die beiden Anfangsnäherungen ein. Das Programm SOLVE bestimmt dann die Nullstellen.

Arbeitsweise von SOLVE

In der Regel wendet das Programm SOLVE das Sekantenverfahren an, um iterativ x -Werte als mögliche Nullstellen zu bestimmen und zu testen. Die Programmdauer beträgt zwischen einigen Sekunden und einigen Minuten.

* Beachten Sie, daß jede Gleichung mit einer Variablen auf diese Weise geschrieben wird. Beispielsweise ist $f(x) = a$ gleichbedeutend mit $f(x) - a = 0$ und $f(x) = g(x)$ mit $f(x) - g(x) = 0$.



Wenn c keine Nullstelle ist, aber $f(c)$ näher an Null liegt als $f(b)$, dann wird b in a und c in b umbenannt und der Näherungsprozess wiederholt. Angenommen, der Graph von $f(x)$ ist glatt und die anfänglichen Werte von a und b sind in der Nähe einer einfachen Nullstelle, dann konvergiert das Sekantenverfahren sehr schnell zu einer Nullstelle.

Wenn die berechnete Sekante beinahe waagrecht ist, modifiziert SOLVE das Sekantenverfahren, um $|c - b| \leq 100 |a - b|$ sicherzustellen. (Dies ist besonders wichtig, da dadurch auch die Gefahr verringert wird, daß das Sekantenverfahren irreführende Ergebnisse erzeugt, wenn der Rundungsfehler in der Nähe der Nullstelle signifikant wird.)

Wenn SOLVE bereits Werte a und b bestimmt hat, für die $f(a)$ und $f(b)$ verschiedenes Vorzeichen haben, wird das Sekantenverfahren so modifiziert, daß c immer in dem Intervall liegt, welches den Vorzeichenwechsel enthält. Damit wird sichergestellt, daß das Suchintervall mit jeder Iteration kleiner wird.

Falls die bisherigen Schritte nicht zu einer Nullstelle führen, legt SOLVE eine Parabel durch die Funktionswerte an den Stellen a , b und c und bestimmt ihr Maximum bzw. Minimum d . Die Näherung wird mit dem Sekantenverfahren fortgesetzt, indem a durch d ersetzt wird.

Wenn drei nacheinanderfolgende Kurvenanpassungen mit einer Parabel zu keiner Nullstelle führen oder $d = b$ ist, zeigt der Taschenrechner **NO** an. Das X- und Z-Register enthält dann b bzw. $f(b)$. a oder c ist im Y-Register abgespeichert. An dieser Stelle können Sie: den Näherungsprozess an der gleichen Stelle wieder aufnehmen; die Näherung mit anderen Werten fortsetzen; $f(b)$ als vernachlässigbar annehmen, sodaß $x = b$ eine Nullstelle ist; die Gleichung in eine einfacher zu lösende Gleichung umwandeln; oder daraus schließen, daß keine Nullstelle existiert.

Anweisungen

Das Programm SOLVE führt das *von Ihnen* geschriebene Programm zur Berechnung von $f(x)$ wiederholt aus. Ebenso müssen Sie SOLVE die beiden Anfangsnäherungen für x , die einen *Bereich* für das Suchen einer Nullstelle definieren, angeben.

Realistische Näherungswerte vergrößern die Geschwindigkeit der Nullstellenberechnung sowie die Genauigkeit. Wenn die Variable x nur in einem begrenzten Bereich sinnvoll und darin eine Nullstelle enthalten ist, dann sollte dieser Bereich als Anfangsintervall gewählt werden. (Beispielsweise sind negative Nullstellen in physikalischen Aufgaben oft unrealistisch.)

- SOLVE benötigt 13 unbenutzte Programm-Register. Falls nicht genügend Register zur Verfügung stehen, wird SOLVE nicht ausgeführt und stattdessen die Fehlermeldung **NO ROOM** angezeigt. Führen Sie im Programm-Modus **GTO** \square \square aus, um zu sehen, wieviele Register zur Verfügung stehen.
- Vor der Ausführung von SOLVE muß bereits ein Programm zur Berechnung von $f(x)$ im Speicher oder in einem eingeschobenen Modul vorliegen. Dieses Programm muß mit einem *globalen Label* bezeichnet sein.* SOLVE ruft dann Ihr Programm iterativ auf, um nacheinander genauere Näherungswerte für x zu bestimmen. Ihr Programm kann davon Gebrauch machen, daß SOLVE die laufenden Näherungswerte für x bei jedem Aufruf Ihres Programms im Stack ablegt.
- Geben Sie dann die beiden Anfangsnäherungen a und b für die Nullstelle ins X- bzw. Y-Register ein.
- Zuletzt geben Sie den Namen Ihres Programms (das die Funktion berechnet) in das Alpha-Register ein und führen dann **SOLVE** aus.

* Dieses Programm sollte *nicht* die Funktionen **PASN**, **PSIZE**, **AK**, über den Kartenleser (HP 82104A) zugängliche Funktionen sowie jede andere Funktion, die den Speicher, die User-Tastenbelegung oder Alarme des Taschenrechners verändert, enthalten.

Wenn das Programm anhält und eine Zahl angezeigt wird, befindet sich im Stack:

- Z** = der Funktionswert an der Stelle x = Nullstelle (sollte Null sein).*
- Y** = die vorige Näherung der Nullstelle (sollte nahe an der Nullstelle liegen).
- X** = die Nullstelle (in der Anzeige zu sehen).

Wenn die zu analysierende Funktion mehrere Nullstellen im gegebenen Intervall aufweist, bricht SOLVE die Berechnung ab, sobald eine der Nullstellen gefunden wurde. Um zusätzliche Nullstellen zu bestimmen, geben Sie andere Anfangswerte ein und führen **SOLVE** erneut aus.

Anweisungstabelle für SOLVE

Anweisungen	Eingabe	Anzeige
1. Schalten Sie den Programm-Modus ein und packen Sie zur Vorbereitung für das Eingeben eines neuen Programms den Speicher. (In der Anzeige erscheint dann die Anzahl der verfügbaren Programm-Register.)	PRGM GTO $\left[\begin{smallmatrix} \cdot \\ \cdot \end{smallmatrix} \right]$	00 REG nnn
2. Geben Sie ein <i>globales</i> Alpha-Label für das Programm, das $f(x)$ für $f(x) = 0$ beschreibt, ein.	LBL <i>globales Label</i>	01 LBL^TLabel
3. Geben Sie das Programm ein und beenden Sie es mit der Anweisung RTN .	: RTN	
4. Überprüfen Sie, ob der Programmspeicherbereich für das Ausführen von SOLVE groß genug ist ($nnn \geq 13$). [*] Schalten Sie dann den Programm-Modus aus.	GTO $\left[\begin{smallmatrix} \cdot \\ \cdot \end{smallmatrix} \right]$ PRGM	00 REG nnn
5. Geben Sie den Namen Ihres Programms aus Schritt 2 ins Alpha-Register ein.	ALPHA <i>globales Label</i> ALPHA	
6. Geben Sie den Anfangsbereich für die Bestimmung von x ein.	a ENTER \uparrow b	a b

^{*} Wenn der Inhalt des Z-Registers *ungleich Null* ist, enthält das X-Register nicht den exakten Wert für die Nullstelle. Stattdessen sind die Inhalte des X- und Y-Registers dicht an der Nullstelle liegende Näherungswerte und umschließen einen Vorzeichenwechsel im Funktionswert.

Anweisungstabelle für SOLVE (Fortsetzung)

Anweisungen	Eingabe	Anzeige
7. Führen Sie SOLVE aus. Die Rechenzeit beträgt bis zu einigen Minuten. Wenn eine Nullstelle gefunden wurde, wird sie in die Anzeige ausgegeben. Andernfalls erscheint in der Anzeige NO . 8. Um eine andere Nullstelle zu bestimmen, wiederholen Sie die Schritte 6 und 7.	SOLVE †	x
* Falls <i>nnn</i> nicht ≥ 13 ist, verwenden Sie SIZE , um mehr Speicherplatz den Programm-Registern zuzuweisen, oder löschen Sie bestehende Programme. Schlagen Sie für entsprechende Anweisungen im Benutzerhandbuch des HP-41 nach. † Zur Ausführung eines Programms drücken Sie entweder XEQ ALPHA <i>Alpha-Name</i> ALPHA oder eine entsprechende benutzerspezifische Taste.		

Bemerkungen

Durch Drücken von **R/S** wird das Programm SOLVE abgebrochen.

Beispiele

Bestimmen Sie die Nullstellen der Gleichung $f(x) = x^2 - 3x - 10 = 0$.

Schreiben Sie zur Definition der Funktion zuerst ein Programm. (Der Name sei TEST). Geben Sie dann vor der Ausführung von **SOLVE** den Namen Ihres Programms in das Alpha-Register sowie die Anfangsnäherungen für die Nullstelle ein.

Mit Hilfe des Hornerchemas können Sie die Funktion zur Erhaltung größerer Programmeffizienz wie folgt darstellen: $f(x) = (x - 3)x - 10$. (Beachten Sie, daß Sie diese Nullstellen ebenso algebraisch bestimmen können.) Da das Programm SOLVE die momentanen Näherungswerte für x vor dem Aufruf von TEST im Stack ablegt, kann TEST bei seiner Ausführung x aus dem Stack entnehmen.

Tastenfolge

FIX 4

PRGM **GTO** **.** **.**

LBL **ALPHA** TEST
ALPHA
 3

Anzeige

00 REG *nnn*

 01 LBL^TTEST
 02 3_

Stellt das hier benutzte Anzeigeformat ein. Programm-Modus; für die Eingabe zur Berechnung von $(x - 3)x - 10$ bereit. Globales Alpha-Label "TEST".

Tastensequenz

[-]

[x]

10

[-]

[RTN]

[GTO] [·] [·]

[PRGM]

[ALPHA] TEST

[ALPHA]

0 [ENTER↑] 10

[XEQ] [SOLVE]

Anzeige03 - $(x - 3)$ 04 * $(x - 3)x$

05 10_

06 - $(x - 3)x - 10$ 07 RTN Ende des $f(x)$ definierenden Programms.00 REG *nnn* Anzahl der verfügbaren Programm-Register (sollte ≥ 13 sein).

TEST_ Verlassen Sie den Programm-Modus.

TEST_ Geben Sie "TEST" (Ihr Programmname) ins Alpha-Reg. ein.

Dies ist der erste notwendige Schritt für das Ausführen von SOLVE.

10_ Eingabe der Anfangsnäherungen 0 und 10. Nun können Sie

[SOLVE] ausführen.

5.0000 Führt SOLVE aus; bestimmt die Nullstelle zu $x = 5.0000$ (nach ca. 12 Sekunden).

Überprüfen Sie, ob 5.0000 wirklich eine Nullstelle von $f(x) = 0$ ist, indem Sie den Inhalt des Z-Registers ansehen. Bestimmen Sie dann die zweite Nullstelle (die bei quadratischen Gleichungen üblich ist), indem Sie die neuen anfänglichen Näherungswerte 0 und -10 für eine negative Nullstelle eingeben.

Tastensequenz

[R↓] [R↓]

0 [ENTER↑] 10 [CHS]

[XEQ] [SOLVE]

[R↓] [R↓]

Anzeige

0.0000

Zuerst wird das Y-Register und dann das Z-Register angezeigt. Da $f(5) = 0$ ist, ist 5 eine genaue Berechnung.

-10_ Neue Anfangsnäherungen für das Bestimmen der zweiten Nullstelle.

-2.0000

0.0000

Zweite Nullstelle. Die zweite Nullstelle ist ebenso genau.

Hier ist eine Aufgabe, bei der Sie die Nullstellen einer Gleichung nicht algebraisch berechnen können. Ein Hammerwerfer wirft seinen Hammer mit einer Geschwindigkeit von 50 Meter/Sekunde in die Luft. Wenn die Höhe des Hammers über dem Erdboden durch den Ausdruck

$$h = 5000(1 - e^{-t/20}) - 200t$$

gegeben ist, wie lange dauert es, bis der Hammer wieder zur Erde fällt? In dieser Gleichung bedeutet h die Höhe in Metern und t die Zeit in Sekunden. Das gewünschte Ergebnis ist die *positive* Nullstelle t .

Sie können wie im vorherigen Beispiel in Ihrem die Funktion definierenden Programm den Vorteil ausnutzen, daß vor dem Aufruf Ihres Programms immer der momentane Näherungswert für x im Stack abgelegt wird.

Tastensequenz

PRGM GTO . .
 LBL ALPHA HOCH
 ALPHA

20 CHS
 ÷
 e^x
 CHS
 1
 +
 5000
 ×
 x↔y
 200
 ×
 -
 RTN

GTO . .
 PRGM

ALPHA HOCH ALPHA

Anzeige

00 REG nnn

01 LBL^THOCH

02 -20_

03 /

04 E^TX

05 CHS

06 1_

07 +

08 5000_

09 *

10 X<>Y

11 200_

12 *

13 -

14 RTN

00 REG nnn

Kennzeichnen Sie das folgende Programm mit dem globalen Label "HOCH".

$-t/20$

$-e^{-t/20}$

$1 - e^{-t/20}$

$5000(1 - e^{-t/20})$

$200t$

Nun ist die vollständige Gleichung programmiert:
 $5000(1 - e^{-t/20}) - 200t$

Ist $nnn \geq 13$?

Beendet den Programm-Modus.

Geben Sie den Namen Ihres Programms ins Alpha-Register ein.

70 Nullstellen einer Funktion

Tastenfolge

5 6

Anzeige

6_

9.2843

0.0000

Beispiel eines anfänglichen Näherungswerts für t .

Die Nullstelle $t = 9.2843$ Sekunden.

Zeigt, daß $h(9.2843) = 0$ ist.

Wenn keine Nullstelle gefunden wird

Es ist möglich, daß eine Gleichung keine reellen Nullstellen hat. In diesem Fall zeigt der Rechner anstelle eines numerischen Ergebnisses **NO** an. Dies würde beispielsweise bei der Gleichung

$$|x| = -1,$$

auftreten, da die Betragsfunktion nie negativ wird.

Es gibt drei allgemeine Fehlerarten, für die die Ausführung von SOLVE abgebrochen wird:

- Wenn aufeinanderfolgende Iterationen einen konstanten Funktionswert ungleich Null ergeben, bricht die Funktionsausführung mit der Anzeige **NO** ab.
- Wenn die Funktionswerte andeuten, daß das *betragsmäßige Minimum* der Funktion in dem untersuchten Bereich nicht gleich Null ist, wird die Ausführung mit der Anzeige **NO** abgebrochen.
- Wenn innerhalb des Unterprogramms ein unerlaubtes Argument in einer mathematischen Operation verwendet wird, wird die Ausführung mit der Anzeige **DATA ERROR** abgebrochen.

Programmier-Informationen

Sie können das Programm SOLVE in ein von Ihnen geschriebenes Programm einbauen. Vergewissern Sie sich, daß Ihr Programm die beiden Anfangsnäherungen ins X- bzw. Y-Register eingibt, bevor es ausführt. Denken Sie auch daran, daß SOLVE den Namen des Programms im Alpha-Register erwartet.

Wenn SOLVE, das in Ihrem Programm aufgerufen wurde, eine Nullstelle berechnet hat, wird die nächste Zeile in Ihrem Programm ausgeführt. Wenn *keine* Nullstelle gefunden wurde, wird die nächste Programmzeile übersprungen. (Dies ist die bei der Programmierung des HP-41 praktizierte "do if true"-Regel.) Mit Hilfe dieser Regel können Sie den Fall, in dem SOLVE keine Nullstelle findet, in Ihr Programm einbauen, indem dann neue Anfangsnäherungen gewählt werden oder ein Funktionsparameter geändert wird.

SOLVE verwendet einen der sechs verschachtelbaren Rücksprungbefehle. Es verbleiben demnach fünf Rücksprungbefehle für ein Programm, welches SOLVE aufruft.

Beachten Sie, daß SOLVE nicht rekursiv (selbstaufrufend) benutzt werden kann. Wenn Sie es rekursiv benutzen, hält das Programm an, und es wird **RECURSION** angezeigt. Sie *können* SOLVE in Verbindung mit INTEG, dem Integrationsprogramm, verwenden.

Literaturhinweise

"Wirkungsvolle Verwendung von SOLVE", *HP-15C Fortgeschrittene Funktionen*, Hewlett-Packard Co., 1982.

Kahan, W.M., "Personal Calculator Has Key to Solve Any Equation $f(x)=0$," *Hewlett-Packard Journal*, 30:12, December 1979.

NULLSTELLEN UND FUNKTIONS- WERTE EINES POLYNOMS

Das Programm PLY kann zur Bestimmung der Nullstellen von Polynomen mit reellen Koeffizienten vom zweiten bis zum fünften Grad und zur Bestimmung der Funktionswerte für Polynome vom zweiten bis zum zwanzigsten Grad verwendet werden.

Die Gleichung lautet demnach:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0,$$

mit $n = 2, 3, 4$ oder 5 .

Es können die Funktionswerte eines Polynoms für beliebige x -Werte berechnet werden. Dies ist für das Plotten von Polynomen und für das Benutzen von auf Polynomen basierenden Datenkorrelationen nützlich.

Beim Starten des Programms ist der Polynomgrad (n) vom Benutzer einzugeben. Der Rechner erfragt dann die Koeffizienten a_n, \dots, a_1, a_0 . Sind die Koeffizienten gleich Null, so ist dies anzugeben. Die Koeffizienten sind in den Registern 00-05 abgespeichert. (Bei der Bestimmung von *Funktionswerten* eines Polynoms bis zum zwanzigsten Grad sind die Koeffizienten in den Registern 00-20 abgespeichert.)

Gleichungen

Für das Bestimmen der Nullstellen werden zuerst alle Koeffizienten durch a_n dividiert, um die Gleichungsform $x^n + a'_{n-1} x^{n-1} + \dots + a'_0 = 0$ zu erhalten. Der Divisor wird dann in Register a_n zur Verwendung für die Berechnung von Funktionswerten für beliebige x -Werte abgespeichert.

Die Routinen für Gleichungen dritten und fünften Grades verwenden ein Iterationsverfahren, um eine reelle Lösung der Gleichung zu finden. Für diesen Fall darf a_0 nicht gleich Null sein. (Ist $a_0 = 0$, dann ist Null eine reelle Lösung, und damit kann der Grad des Polynoms durch Abdivision von x um 1 reduziert werden.) Sobald eine Nullstelle gefunden ist, wird die Gleichung durch synthetische Division auf eine Gleichung zweiten oder vierten Grades reduziert.

74 Nullstellen und Funktionswerte eines Polynoms

Zur Lösung der Gleichung vierten Grades muß zuerst die kubische Gleichung

$$y^3 + b_2y^2 + b_1y + b_0 = 0,$$

gelöst werden, wobei $b_2 = -a_2$
 $b_1 = a_3a_1 - 4a_0$
 $b_0 = a_0(4a_2 - a_3^2) - a_1^2$ ist.

Es sei y_0 die größte Lösung der obigen Gleichung dritten Grades.

Damit kann die Gleichung vierten Grades in zwei quadratische Gleichungen zerlegt werden:

$$x^2 + (A + C)x + (B + D) = 0$$

$$x^2 + (A - C)x + (B - D) = 0$$

wobei $A = \frac{a_3}{2}$, $B = \frac{y_0}{2}$, $D = \sqrt{B^2 - a_0}$, $C = \sqrt{A^2 - a_2 + y_0}$.

Lösungen für die Gleichung vierten Grades ergeben sich aus der Lösung der beiden quadratischen Gleichungen.

Eine quadratische Gleichung $x^2 + a_1x + a_0 = 0$ wird nach der Formel

$$x_{1,2} = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} - a_0} \text{ gelöst.}$$

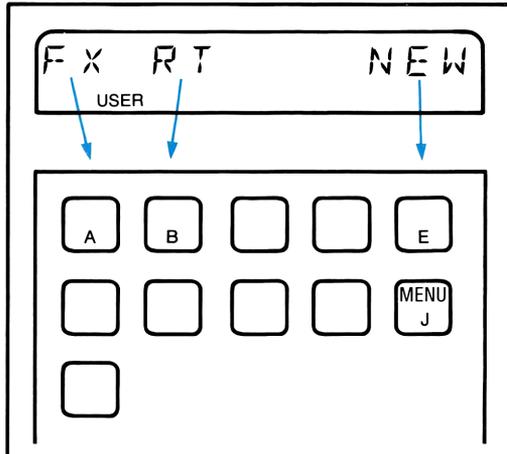
Wenn $D = a_1^2/4 - a_0 > 0$, sind die Nullstellen reell; wenn $D < 0$, sind die Nullstellen komplex entsprechend $u \pm iv = -(a_1/2) \pm i\sqrt{-D}$.

Eine reelle Lösung wird als einzelne Zahl ausgegeben. Komplexe Nullstellen erscheinen immer als Paare der Form $u \pm iv$ und sind in der Ausgabe bezeichnet.

Für die Gleichungen 3., 4. und 5. Grades sind lange Programmlaufzeiten zu erwarten, da diese ein- oder mehrfach eine Iteration durchlaufen.

Anweisungen

Sobald Sie die Variablen eingegeben haben, erscheint in der Anzeige das entsprechende Menü zur Kennzeichnung der in PLY enthaltenen Tastenfeld-Funktionen. Drücken Sie \boxed{J} , um das Menü zu jeder Zeit abzurufen. Der Programmablauf wird dadurch nicht gestört.



Durch Drücken der Taste \leftarrow können Sie das Menü jederzeit löschen. Dadurch wird der Inhalt des X-Registers angezeigt, das Programm jedoch nicht beendet. Nun können Sie Berechnungen ausführen und dann durch Drücken der Taste $\boxed{R/S}$ das Programm fortsetzen. (Sie müssen die Programmanzeige *nicht* löschen, bevor Sie die Berechnungen ausführen.)

Anweisungstabelle für PLY

		Umfang: 023
Anweisungen	Eingabe	Anzeige
1. Starten Sie das Programm PLY.	\boxed{XEQ} \boxed{PLY}^*	DEGREE=?
2. Geben Sie den Grad des Polynoms ($n = 2,3,4,5$ bei Nullstellenbestimmung; bis zu 20 für Berechnung von Funktionswerten) ein.	n $\boxed{R/S}$	$a_n=?$
3. Geben Sie die Koeffizienten a_n des Polynoms ein. (Koeffizient = 0 muß ebenfalls eingegeben werden.) Wiederholen Sie diese Anweisung bis Sie zur Eingabe von a_0 gefragt werden.	a_n $\boxed{R/S}$: a_1	$a(n-1)=?$: $a_0=?$
4. Geben Sie Koeffizient a_0 ein.	a_0 $\boxed{R/S}$	FX RT NEW
5. Verwenden Sie FX , um Funktionswerte des Polynoms für x zu berechnen. Sie können diesen Schritt für neue x -Werte wiederholen.	x \boxed{A} (FX) $\boxed{R/S}$ †	F<X>=f(x) FX RT NEW

Anweisungstabelle für PLY (Fortsetzung)

Anweisungen	Eingabe	Anzeige
6. Verwenden Sie RT , um die Nullstellen des Polynoms zu bestimmen und [R/S] , um die Nullstellen nacheinander anzuzeigen.	[B] (RT) [R/S] : [R/S] [R/S] [R/S] [R/S] [R/S] † [E] (NEW)	ROOT=1. Stelle ROOT=2. Stelle : U=u-Wert V=v-Wert U=u-Wert V=-v-Wert FX RT NEW DEGREE=?
* Zur Ausführung eines Programms drücken Sie entweder [XEQ] [ALPHA] <i>Alpha-Name</i> [ALPHA] oder eine entsprechende benutzerspezifische Taste. † Diese Tastenfolge ist nicht notwendig, wenn Sie einen Drucker angeschlossen haben, da er automatisch alle Ergebnisse druckt und dann das Menü angezeigt wird.		

Hinweis: Dieses Programm kann wegen Rundungsfehlern bei Zwischenergebnissen Nullstellen fehlerhaft berechnen. Dies kommt normalerweise nur bei *reellen* Nullstellen vor. Um die berechnete Nullstelle auf ihre Richtigkeit hin zu überprüfen, berechnen Sie mit PLY für diese Nullstelle den Funktionswert des Polynoms (Schritt 5). Wenn Sie als Ergebnis eine sehr kleine Zahl nahe Null erhalten, ist die Nullstelle korrekt.

Bemerkungen

Wenn Sie vor der Ausführung von Schritt 6 Flag 06 setzen (**[SF]**06), werden die der Reihe nach berechneten Nullstellen, angefangen in R_{24} , in der Reihenfolge Real-/Imaginärteil abgespeichert. (Der Imaginärteil von reellen Nullstellen ist Null.)

Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten **[A]**, **[B]**, **[E]** und **[J]** zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung *überschrieben*. *Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten*. Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Beispiele

Gesucht sind die Nullstellen von:

$$x^5 - x^4 - 101x^3 + 101x^2 + 100x - 100 = 0.$$

Tastensequenz

FIX 4

XEQ SIZE 023

XEQ PLY

5 R/S

1 R/S

1 CHS R/S

101 CHS R/S

101 R/S

100 R/S

100 CHS R/S

B (RT)

R/S

R/S

R/S

R/S

Anzeige

DEGREE=?

a5=?

a4=?

a3=?

a2=?

a1=?

a0=?

FX RT NEW

ROOT=10.0000

ROOT=1.0000

ROOT=1.0000

ROOT=-1.0000

ROOT=-10.0000

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $SIZE \geq 023$ ist.

1. Nullstelle
2. Nullstelle
3. Nullstelle
4. Nullstelle
5. Nullstelle

Es ist $4x^4 - 8x^3 - 13x^2 - 10x + 22 = 0$ zu lösen.

Tastensequenz

J

E (NEW)

4 R/S

4 R/S

8 CHS R/S

13 CHS R/S

10 CHS R/S

22 R/S

B (RT)

R/S

R/S

Anzeige

FX RT NEW

DEGREE=?

a4=?

a3=?

a2=?

a1=?

a0=?

FX RT NEW

U=-1.0000

V=1.0000

U=-1.0000

Zeigt das Menü an (wahlfreier Schritt).

Fragt nach einem neuen Polynom (nach dem im 1. Beispiel benutzten).

Zeigt das Menü an.

1. und 2. Nullstelle sind $-1.00 \pm 1.00i$.

Tastensequenz

R/S

R/S

R/S

Anzeige

V = -1.0000

ROOT = 3.1180

3. Nullstelle

ROOT = 0.8820

4. Nullstelle

Es sind die Funktionswerte des Polynoms $f(x) = x^5 + 5x^4 - 3x^2 - 7x + 11$ an den Stellen $x = 2.5$ und $x = -5$ zu bestimmen.

Tastensequenz

J

E (NEW)

5 R/S

1 R/S

5 R/S

0 R/S

3 CHS R/S

7 CHS R/S

11 R/S

2.5 A (FX)

5 CHS R/S

Anzeige

FX RT NEW

DEGREE = ?

a5 = ?

a4 = ?

a3 = ?

a2 = ?

a1 = ?

a0 = ?

FX RT NEW

F < X > = 267.7188

F < X > = -29.0000

Zeigt neues Menü an
(wahlfreier Schritt).

Fragt für ein neues
Polynom.

Programmier-Informationen

Sie können das Unterprogramm RTS in Ihren eigenen Programmen verwenden. Es bestimmt die reellen und komplexen Nullstellen eines Polynoms vom zweiten bis fünften Grad.

Speicherminimum: 23 Register, außer wenn Flag 6 gesetzt ist. Wenn die berechneten Nullstellen abgespeichert werden sollen, muß die Anzahl der Speicherregister $24 + 2(\text{Grad})$ betragen.

Benutzte Flags: 00, 03, 05, 06, 21

Unterprogramm: RTS

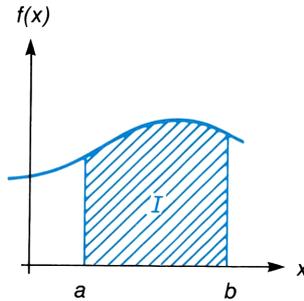
alter Registerinhalt	neuer Registerinhalt	Zu initialisierende Flags
$R_{00} = a_0$	$R_{00} = a_0/a_5$	SF 00
$R_{01} = a_1$	$R_{01} = a_1/a_5$	CF 03
$R_{02} = a_2$	$R_{02} = a_2/a_5$	CF 05
$R_{03} = a_3$	$R_{03} = a_3/a_5$	SF 06 um die Nullstellen abzuspeichern
$R_{04} = a_4$	$R_{04} = a_4/a_5$	CF 06 um die Nullstellen nicht abzuspeichern
$R_{05} = a_5$	$R_{05} = a_5$	SF 21 um anzuhalten, wenn Ergebnisse angezeigt werden
	$R_{06} \dots R_{21}$ undefiniert	CF 21 um nicht anzuhalten, wenn Ergebnisse angezeigt werden
$R_{22} = \text{Grad der Gleichung}$	$R_{22} = \text{Grad der Gleichung}$	
$R_{23} = \text{Zeiger}$		
Wenn Flag 06 gesetzt ist:	$R_{24}, R_{25} = 1. \text{ Nullstelle}$	
	$R_{26}, R_{27} = 2. \text{ Nullstelle}$	
	$R_{28}, R_{29} = 3. \text{ Nullstelle}$	
	$R_{30}, R_{31} = 4. \text{ Nullstelle}$	
	$R_{32}, R_{33} = 5. \text{ Nullstelle}$	

Anmerkungen. Wenn Sie RTS benutzen, geben Sie die Koeffizienten in R_{00} – R_{05} und den Grad in R_{22} ein. Setzen Sie Flag 06, um die Nullstellen abzuspeichern, löschen Sie Flag 03 und 05 und setzen Sie Flag 00. Wenn die Nullstellen abgespeichert werden, so werden sie immer als komplexe Zahl mit Real- und Imaginärteil abgespeichert. Eine reelle Nullstelle hat als Imaginärteil Null.

NUMERISCHE INTEGRATION

Das Programm INTEG berechnet das bestimmte Integral I einer Funktion $f(x)$ mit Integrationsgrenzen a und b . Dies wird mathematisch und graphisch folgendermaßen ausgedrückt:

$$I = \int_a^b f(x) dx.$$



INTEG ([INTEG](#)) stellt Ihnen eine fortgeschrittene Methode zur Bestimmung des bestimmten Integrals einer Funktion zur Verfügung. Sie geben die Funktionsgleichung (in einem Programm) und die Integrationsgrenzen ein. Das Programm INTEG berechnet dann das Integral.

Verfahren

Das Programm INTEG verwendet das Rombergverfahren zur Bestimmung des Integrals. Der Algorithmus berechnet Funktionswerte $f(x)$ für viele x -Werte (Stützstellen) innerhalb der Integrationsgrenzen. Die Programmlaufzeit beträgt demnach zwischen einigen Sekunden und einigen Minuten.

Verschiedene zusätzliche Verfeinerungen machen den Algorithmus noch effektiver. Beispielsweise werden in INTEG an Stelle von äquidistanten Stützstellen, die bei periodischen Funktionen zu einer Art Resonanz und daher zu fehlerhaften Ergebnissen führen können, Stützstellen mit unterschiedlichem Abstand verwendet. Eine weitere Verfeinerung ist, daß INTEG zur internen Summenbildung eine höhere Genauigkeit (13 signifikante Stellen) benutzt. Dadurch können tausende von Stützstellen mit großer Genauigkeit aufaddiert werden (falls nötig), ohne daß durch Rundung Informationen verloren gehen.

Ein Taschenrechner kann mit numerischer Integration ein Integral fast nie exakt berechnen. Sie können jedoch auf bequeme Weise die Größe des zulässigen Fehlers spezifizieren. Stellen Sie dazu das Anzeigeformat entsprechend der Anzahl der genauen Stellen im Integrand ein. Wenn Sie das Anzeigeformat mit **FIX**2 einstellen, weiß der Rechner, daß das Ergebnis nur mit einer Genauigkeit von zwei Stellen berechnet werden muß. Dadurch ergibt sich eine wesentliche Zeitersparnis. Schlagen Sie für weitere Informationen im Abschnitt "Genauigkeit von INTEG" nach.

Anweisungen

Das Programm INTEG führt das *von Ihnen* geschriebene Programm zur Berechnung von $f(x)$ wiederholt aus. Ebenso müssen Sie INTEG die beiden Integrationsgrenzen für x angeben.

- INTEG benötigt 32 unbenutzte Programm-Register. Falls nicht genügend freie Register zur Verfügung stehen, wird INTEG nicht ausgeführt und stattdessen die Fehlermeldung **NO ROOM** angezeigt. Führen Sie im Programm-Modus **GTO** \cdot \square aus, um zu sehen, wieviele Programm-Register zur Verfügung stehen.
- Vor der Ausführung von INTEG muß bereits ein Programm zur Berechnung von $f(x)$ im Speicher oder in einem eingeschobenen Modul vorliegen. Dieses Programm muß mit einem *globalen Label* bezeichnet sein.* Ihr Programm kann davon Gebrauch machen, daß INTEG die laufenden Näherungswerte für x bei jedem Aufruf Ihres Programms im Stack ablegt.
- Geben Sie dann die beiden Grenzen a und b ins X- bzw. Z-Register ein.
- Zuletzt geben Sie den Namen Ihres Programms (das die Funktion berechnet) in das Alpha-Register ein und führen dann **INTEG** aus.

Wenn das Programm anhält und der Wert des Integrals angezeigt wird, befindet sich im Stack:

- T** = die untere Integrationsgrenze a .
- Z** = die obere Integrationsgrenze b .
- Y** = die Fehlerabschätzung für das Integral.
- X** = der Näherungswert für das Integral (steht in der Anzeige).

* Dieses Programm sollte *nicht* die Funktionen **PASN**, **PSIZE**, **AK**, über den Kartenleser (HP 82104A) zugängliche Funktionen sowie andere Funktionen, die den Speicher, die User-Tastenbelegung oder Alarmer des Taschenrechners verändern, enthalten.

Anweisungstabelle für INTEG

Anweisungen	Eingabe	Anzeige
1. Schalten Sie in den Programm-Modus um und packen Sie zur Eingabe eines neuen Programms den Programmspeicher.	<code>PRGM</code> <code>GTO</code> <code>□</code> <code>□</code>	00 REG <i>nnn</i>
2. Geben Sie das globale Alpha-Label als Programmnamen für das $f(x)$ berechnende Programm ein.	<code>LBL</code> Globales Label	01 LBL ^T Label
3. Geben Sie Ihr Programm ein und beenden Sie die Eingabe mit der Anweisung <code>RTN</code> .	: <code>RTN</code>	
4. Überprüfen Sie, ob der Programmspeicher für das Ausführen von INTEG ($nnn \geq 32$)* groß genug ist. Verlassen Sie dann den Programm-Modus.	<code>GTO</code> <code>□</code> <code>□</code> <code>PRGM</code>	00 REG <i>nnn</i>
5. Geben Sie den Namen des Programms aus Schritt 2 in das Alpha-Register ein.	<code>ALPHA</code> <i>globalesLabel</i> <code>ALPHA</code>	
6. Geben Sie die Integrationsgrenzen ein:	a <code>ENTER</code> <code>▶</code> b	a b
7. Stellen Sie das Anzeigeformat entsprechend der gewünschten Genauigkeit des Ergebnisses ein.	<code>FIX</code> <i>n</i> oder <code>SCI</code> <i>n</i> oder <code>ENG</code> <i>n</i>	
8. Führen Sie <code>INTEG</code> aus. Das Programm läuft bis zu einigen Minuten, bevor es den Wert des Integrals anzeigt.	<code>INTEG</code> <code>†</code>	Integral
9. Um die Berechnung des Integrals mit anderer Genauigkeit zu wiederholen, stellen Sie ein neues Anzeigeformat ein, vertauschen Sie den Stackinhalt zyklisch nach "unten" (um die ursprüngliche Integrationsgrenzen wiederherzustellen) und führen Sie <code>INTEG</code> erneut aus.	<code>FIX</code> <i>n</i> oder <code>SCI</code> <i>n</i> oder <code>ENG</code> <i>n</i> <code>R</code> <code>↓</code> <code>R</code> <code>↓</code> <code>INTEG</code>	b Integral

* Falls *nnn* nicht ≥ 32 ist, verwenden Sie `SIZE`, um mehr Speicherplatz den Programm-Registern zuzuweisen, oder löschen Sie bestehende Programme. Schlagen Sie für entsprechende Anweisungen im Benutzerhandbuch des HP-41 nach.

† Zur Ausführung eines Programms drücken Sie entweder `XEQ` `ALPHA` *Alpha-Name* `ALPHA` oder eine entsprechende benutzerspezifische Taste.

Bemerkungen

Durch Drücken von `R/S` wird das Programm INTEG abgebrochen.

1. Beispiel

Die Besselsche Funktion erster Art 0-ter Ordnung hat folgende Form:

$$J_0(x) = 1/\pi \int_0^\pi \cos(x \sin \theta) d\theta.$$

Berechnen Sie

$$I_0(1) = 1/\pi \int_0^\pi \cos(\sin \theta) d\theta.$$

Schreiben Sie zuerst ein Programm zur Definition des Integranden. Vergewissern Sie sich, daß der Rechner für die Berechnung der trigonometrischen Funktionen auf den Winkelmodus RAD eingestellt ist. Geben Sie dann vor der Ausführung von **INTEG** den Namen Ihres Programms in das Alpha-Register sowie die Integrationsgrenzen ein. Vergessen Sie nicht, die angezeigte Lösung mit $1/\pi$ zu multiplizieren.

Tastenfolge

FIX 4

PRGM **GTO** . .

LBL **ALPHA** **J01** **ALPHA**

SIN

COS

RTN

GTO . .

PRGM

ALPHA **J01**

ALPHA

0 **ENTER** π

RAD

Anzeige

00 REG nnn

01 LBL^TJ01

02 SIN

03 COS

04 RTN

00 REG nnn

J01

3.1416

3.1416

Stellt das hier benutzte Anzeigeformat ein.

Programm-Modus; Für die Eingabe des Programms zur Berechnung von $\cos(\sin \theta)$ bereit.

Globales Alpha-Label "J01".

$\sin \theta$.

$\cos(\sin \theta)$.

Ende des Programms für $f(x)$.

Anzahl der zur Verfügung stehenden Programm-Register; ist $nnn \geq 32$?

Verläßt den Programm-Modus.

Geben Sie "J01" (der Name Ihres Programms) in das Alpha-Register ein. Dies ist der erste notwendige Schritt für das Ausführen von **INTEG**.

Eingabe der Integrationsgrenzen 0 und π .

Stellt den Winkelmodus RAD ein. Nun können Sie **INTEG** ausführen.

Tastensequenz

XEQ INTEG

Anzeige

2.4040

3.1416

0.7652

Führt INTEG aus und gibt den Wert des Integrals (nach ca. 25 Sekunden) aus. Vergessen Sie nicht, das Ergebnis mit der Konstanten vor dem Integral zu multiplizieren.

 $J_0(1)$.

π

÷

Genauigkeit von INTEG

Da der Rechner den Wert eines Integrals nicht exakt berechnen kann, bestimmt er einen *Näherungswert*. Die Genauigkeit dieser Näherung hängt von der Genauigkeit der in Ihrem Programm berechneten Funktion (Integrand) ab*. Die Genauigkeit der Funktionsberechnung wird durch Rundungsfehler und der Genauigkeit von empirischen Konstanten beeinflusst.

Um die Genauigkeit der Funktionsberechnung zu spezifizieren, stellen Sie das Anzeigeformat entsprechend ein (FIX n , SCI n oder ENG n), wobei n nicht größer als die Anzahl der signifikanten Stellen ist. Für kleinere n -Werte ist die Rechenzeit kürzer. Der Rechner setzt jedoch voraus, daß die Funktion nicht genauer als die angegebene Stellenzahl ist.†

Das Programm INTEG gibt den berechneten Näherungswert des Integrals ins X-Register (Anzeige) und die *Fehlerabschätzung* der Näherung ins Y-Register aus.‡ Um die Fehlerabschätzung anzuzeigen, drücken Sie $\boxed{x\ y}$.

Wenn die Fehlerabschätzung einer Näherung größer als gewünscht ist, können Sie die Fehlerabschätzung verkleinern, indem Sie die Anzahl der in der Anzeige erscheinenden Stellen vergrößern und anschließend das Programm INTEG erneut ausführen.

* Es ist möglich, daß Integrale von Funktionen mit gewissen Charakteristika (wie scharfe Zacken oder sehr schnelle Oszillationen) ungenau berechnet werden. Dies kommt jedoch nur selten vor.

† Bei Benutzen von SCI und ENG wird eine zum Betrag der Funktion *proportionale* Fehlerabschätzung angegeben. Bei FIX ist die Fehlerabschätzung *unabhängig* vom Betrag der Funktion.

‡ Kein Algorithmus, der bei der numerischen Integration verwendet wird, kann den genauen Unterschied zwischen seiner Approximation und dem tatsächlichen Integral feststellen. Der hier verwendete Algorithmus schätzt aber eine *obere Grenze* für diese Differenz, die als *Fehlerabschätzung* der Approximation bezeichnet wird.

Wenn Sie das Programm INTEG mit einer anderen Stellengenauigkeit erneut ausführen wollen, müssen Sie die Integrationsgrenzen nicht noch einmal eingeben (falls Sie in der Zwischenzeit keine Berechnungen ausgeführt haben). Da die Integrationsgrenzen sich nun im T- und Z-Register befinden (wie in "Anweisungen" erwähnt), drücken Sie einfach $\boxed{R\downarrow} \boxed{R\downarrow}$, um den alten Zustand wiederherzustellen und führen $\boxed{\text{INTEG}}$ erneut aus.

2. Beispiel

Berechnen Sie das im 1. Beispiel für $J_0(1)$ angegebene Integral mit dem auf $\boxed{\text{SCI}}$ 2 eingestellten Anzeigeformat. Überprüfen Sie dann die Fehlerabschätzung der Näherung. Berechnen Sie anschließend das Integral auf vier (anstatt zwei) Stellen genau und überprüfen Sie die Fehlerabschätzung. (Vergewissern Sie sich, daß der Winkelmodus immer noch RAD (**RAD**-Indikator ein) ist. Sie müssen die Integrationsgrenzen nur für die erste Ausführung des Programms eingeben.

Tastensequenz

$\boxed{\text{SCI}}$ 2

0 $\boxed{\text{ENTER}\uparrow} \boxed{\pi}$

$\boxed{\text{XEQ}} \boxed{\text{INTEG}}$

$\boxed{x\text{y}}$

$\boxed{\text{SCI}}$ 4

$\boxed{R\downarrow} \boxed{R\downarrow}$

$\boxed{\text{XEQ}} \boxed{\text{INTEG}}$

$\boxed{x\text{y}}$

Anzeige

3.14 00

2.40 00

1.57 -03

1.5708 -03

3.1416 00

2.4039 00

1.5708 -05

Stellt wissenschaftliches Anzeigeformat ein; zwei Dezimalstellen.

Eingabe der unteren (0) und oberen (π) Integrationsgrenze.

Das Integral auf zwei Stellen genau.

Die Fehlerabschätzung der Näherung.

Stellt Anzeigeformat auf vier Dezimalstellen ein.

Schieben Sie den Stackinhalt nach "unten", bis die obere Grenze erscheint.

Das Integral auf vier Stellen genau.

Fehlerabschätzung (viel kleiner).

Programmier-Informationen

Sie können das Programm INTEG in ein von Ihnen geschriebenes Programm einbauen. Vergewissern Sie sich, daß Ihr Programm die beiden Integrationsgrenzen ins X- bzw. Y-Register eingibt, bevor es `INTEG` ausführt. Denken Sie auch daran, daß INTEG den Namen der Funktion, für die das Integral zu berechnen ist, im Alpha-Register erwartet.

INTEG verwendet einen der sechs verschachtelbaren Rücksprungbefehle. Es verbleiben demnach fünf Rücksprungbefehle für ein Programm, das INTEG aufruft.

Beachten Sie, daß INTEG nicht rekursiv (selbstaufrufend) benutzt werden kann. Wenn Sie es rekursiv benutzen, hält das Programm an, und es wird **RECURSION** angezeigt. Sie *können* INTEG in Verbindung mit SOLVE verwenden. Ein Programm, das INTEG und SOLVE verbindet, benötigt für seine Ausführung 32 verfügbare Programm-Register.

Literaturhinweise

“Arbeiten mit \int ”, *HP-15C Fortgeschrittene Funktionen*, Hewlett-Packard Co., 1982.

Kahan, W.M., “Handheld Calculator Evaluates Integrals,” *Hewlett-Packard Journal*, 31:8, August 1980.

DIFFERENTIALGLEICHUNGEN

Das Programm DIFEQ berechnet Ableitungen erster und zweiter Ordnung nach der Methode von RUNGE-KUTTA. Eine Ableitung erster Ordnung ist von der Form $y' = f(x, y)$, mit den Anfangswerten x_0, y_0 ; eine Ableitung zweiter Ordnung ist von der Form $y'' = f(x, y, y')$, mit den Anfangswerten x_0, y_0, y_0' .

In beiden Fällen kann die Funktion $f(x)$ in den Programmspeicher unter einem beliebigen, maximal 6 Zeichen langen, *globalen* Label eingegeben werden. Die Werte x und y werden im X- bzw. Y-Register erwartet; y' wird im Z-Register bei Ableitungen zweiter Ordnung erwartet. Das Programm DIFEQ belegt die Register 00-07. Die verbleibenden Register stehen für die Definition der Funktion zur Verfügung.

Die Lösung ist ein numerischer Wert, bei dem y_i für $x_i = x_0 + ih$ ($i = 1, 2, 3, \dots$) für ein vom Benutzer zu bestimmendes Inkrement h berechnet wird. Der Wert für h kann zu jeder Zeit während der Programmausführung geändert werden; man speichert dazu $h/2$ in Register 01 ab. Dies erlaubt Lösungen der Gleichung beliebig nahe an einem ihrer Pole ($y \rightarrow \pm\infty$).

Gleichungen

Ableitung 1. Ordnung:

$$y_{i+1} = y_i + \frac{1}{6} (c_1 + 2c_2 + 2c_3 + c_4)$$

mit

$$c_1 = hf(x_i, y_i)$$

$$c_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{c_1}{2}\right)$$

$$c_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{c_2}{2}\right)$$

$$c_4 = hf(x_i + h, y_i + c_3)$$

Ableitungen 2. Ordnung:

$$y_{i+1} = y_i + h \left[y'_i + \frac{1}{6} (k_1 + k_2 + k_3) \right]$$

$$y'_{i+1} = y'_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(x_i, y_i, y'_i)$$

$$k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} y'_i + \frac{h}{8} k_1, y'_i + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} y'_i + \frac{h}{8} k_2, y'_i + \frac{k_2}{2}\right)$$

$$k_4 = hf\left(x_i + h, y_i + h y'_i + \frac{h}{2} k_3, y'_i + k_3\right)$$

Anweisungen

Werden die Werte für eine Ableitung 2. Ordnung eingegeben, so müssen die Werte von x_0 und y_0 vor dem Wert für y_0' eingegeben werden. Alle Werte, einschließlich Null, müssen eingegeben werden.

Beachten Sie, daß ein zu großer Wert für h , die Schrittweite, zu falschen Ergebnissen führen kann.*

* Sie können ein Ergebnis überprüfen, indem Sie vom Ergebnis aus zur Anfangsbedingung unter Verwendung von $-h$ zurückgehen. Wenn Sie nicht die korrekte Anfangsbedingung erhalten, führen Sie das Programm DIFEQ für ein kleineres h erneut aus.

Anweisungstabelle für DIFEQ

		Umfang: 008
Anweisungen	Eingabe	Anzeige
1. Funktionseingabe $f(x, y, y')$ vorbereiten.	$\boxed{\text{GTO}} \boxed{\cdot} \boxed{\cdot}$	
2. Schalten Sie den Programm-Modus ein.	$\boxed{\text{PRGM}}$	
3. Geben Sie die Funktion unter dem gewünschten, globalen Alpha-Label ein. Fügen Sie $\boxed{\text{RTN}}$ hinzu.	$\boxed{\text{LBL}}$ Funktions-Label : $\boxed{\text{RTN}}$	
4. Verlassen Sie den Programm-Modus.	$\boxed{\text{PRGM}}$	
5. Starten Sie das Programm.	$\boxed{\text{XEQ}} \boxed{\text{DIFEQ}}^*$	NAME?
6. Geben Sie das <i>Funktions-Label</i> (aus Schritt 3) ein.	Funktions-Label $\boxed{\text{R/S}}$	ORDER=?
7. Geben Sie die Ordnung der Differentialgleichung (1. oder 2.) ein.	Ordnung $\boxed{\text{R/S}}$	STEP SIZE=?
8. Geben Sie die Schrittweite (h) ein.	$h \boxed{\text{R/S}}$	X0=?
9. Geben Sie den Anfangswert für x ein.	$x_0 \boxed{\text{R/S}}$	Y0=?
10. Geben Sie den Anfangswert für y ein.	$y_0 \boxed{\text{R/S}}$	x_1 (Differentialgleichung 1. Ordnung) oder Y0.=? (Differentialgleichung 2. Ordnung)
11. Für eine Differentialgleichung 2. Ordnung: Geben Sie den Anfangswert für y' ein.	$y_0' \boxed{\text{R/S}}$	x_1
12. Geben Sie nacheinander die Werte für x und y aus.	$\boxed{\text{R/S}}$ $\boxed{\text{R/S}}$ $\boxed{\text{R/S}}$	y_1 x_2 y_2 etc.
* Zur Ausführung eines Programms drücken Sie entweder $\boxed{\text{XEQ}} \boxed{\text{ALPHA}}$ <i>Alpha-Name</i> $\boxed{\text{ALPHA}}$ oder eine entsprechende benutzerspezifische Taste.		

Beispiele

Unter Verwendung des Funktions-Labels FX soll die Differentialgleichung erster Ordnung

$$y' = \frac{\sin x + \tan^{-1}(y/x)}{y - \ln(\sqrt{x^2 + y^2})}$$

mit $x_0 = y_0 = 1$ und $h = 0.5$ numerisch bestimmt werden. Der Winkelmodus muß auf RAD gesetzt werden. Außerdem werden drei weitere Speicherregister zur Definition der Funktion benötigt.

Tastensequenz

FIX 4

XEQ **SIZE** 011

PRGM

GTO ◻ ◻

LBL **ALPHA** **FX** **ALPHA**

XEQ **RAD**

STO 08

x↔y

STO 09

x↔y

R→P

LN

STO 10

R↓

RCL 08

SIN

+

RCL 09

RCL 10

-

÷

XEQ **DEG**

RTN

PRGM

Anzeige

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $\text{SIZE} \geq 011$ ist.

Tastensequenz

XEQ DIFEQ
 ALPHA FX ALPHA R/S
 1 R/S
 .5 R/S
 1 R/S
 1 R/S
 R/S
 R/S
 R/S
 R/S
 R/S

Anzeige

NAME?
 ORDER=?
 STEP SIZE=?
 X0=?
 Y0=?
 1.5000 x_1
 2.0553 y_1
 2.0000 x_2
 2.7780 y_2
 2.5000 x_3
 3.2781 y_3
 etc.

Unter Verwendung des Funktions-Labels DIF soll die Differentialgleichung 2. Ordnung

$$(1 - x^2)y'' + xy' = x,$$

mit $x_0 = y_0 = y_0' = 0$ und $h = 0.1$ gelöst werden.

Man formuliert die Gleichung um zu:

$$y'' = \frac{x(1 - y')}{1 - x^2} = \frac{x(y' - 1)}{x^2 - 1} \quad x \neq 1$$

Tastensequenz

PRGM
 GTO . .
 LBL ALPHA DIF ALPHA
 STO 08
 R↓ R↓
 1 -
 RCL 08
 x
 LASTx
 x²
 1 - ÷
 RTN
 PRGM

Anzeige

Tastenfolge

XEQ DIFEQ
 ALPHA DIF ALPHA R/S
 2 R/S
 .1 R/S
 0 R/S
 0 R/S
 0 R/S
 R/S
 R/S
 R/S
 R/S
 R/S
 R/S
 R/S
 R/S

Anzeige

NAME?
 ORDER=?
 STEP SIZE=?
 X0=?
 Y0=?
 Y0.=?
 0.1000 x_1
 0.0002 y_1
 0.2000 x_2
 0.0013 y_2
 0.3000 x_3
 0.0046 y_3
 0.4000 x_4
 0.0109 y_4
 etc.

OPERATIONEN MIT KOMPLEXEN ZAHLEN

Dieses Programm gilt für verkettete Operationen mit komplexen Zahlen in kartesischer Form. Neben den vier Grundrechenarten für komplexe Zahlen ($+$, $-$, \times , \div) sind die gebräuchlichsten Funktionen der komplexen Zahlen z und w ($|z|$, $1/z$, z^n , $z^{1/n}$, e^z , $\ln z$, $\sin z$, $\cos z$, $\tan z$, a^z , $\log_a z$, $z^{1/w}$ und z^w) enthalten. Funktionen und Operationen können gemischt werden, um Ausdrücke wie $z_3/(z_1 + z_2)$, $e^{z_1 z_2}$, $|z_1 + z_2| + |z_2 - z_3|$, etc. zu berechnen, wobei z_1 , z_2 und z_3 komplexe Zahlen der Form $x + iy$ sind.

Für den wiederholten Gebrauch dieser Operationen kann der Benutzer die einzelnen Programme ausgewählten User-Tasten zuweisen und eine entsprechende Maske selbst herstellen. Eine mögliche Zuweisung könnte beispielsweise sein:

ASN	SINZ	SIN
ASN	LNZ	LN
ASN	C+	+
ASN	C-	-
ASN	CINV	1/x

Das Logik-System für diese Funktionen ist eine Variante des regulären Stacks des HP-41. Anstatt vier reelle Zahlen abzuspeichern, sind in diesem Stack zwei komplexe Zahlen abgelegt. Das untere und das obere Register des komplexen Stacks sei als ξ bzw. τ bezeichnet. Diese entsprechen somit dem X- und T-Register des Vier-Register Stacks des Rechners.* Eine komplexe Zahl z wird durch die Tastefolge z_y $\boxed{\text{ENTER}\uparrow}$ z_x ins ξ -Register eingegeben. Durch die Eingabe einer zweiten komplexen Zahl w ($\boxed{\text{ENTER}\uparrow}$ w_y $\boxed{\text{ENTER}\uparrow}$ w_x) wird z ins Register τ geschoben und w im Register ξ abgelegt. Der vorige Inhalt von τ ist verloren.

* Jedes komplexe Stack-Register muß zwei reelle Zahlen enthalten—den Realteil und den Imaginärteil des komplexen Inhalts. Folglich werden zwei Stack-Register für eine komplexe Zahl benötigt. Komplexe Stack-Register werden in diesem Kapitel wie ein einziges Register behandelt.

Speicher-Stacks

T	t
Z	z
Y	y
X	x

Regulärer Stack

τ	iz_y	}	z
	z_x		
ξ	iw_y	}	w
	w_x		

Komplexer Stack

Funktionen entnehmen den Eingabewert dem ξ -Register und legen das Ergebnis (außer für $|z|$, die eine reelle Zahl ausgibt) im ξ -Register ab. Arithmetische Operationen verwenden sowohl das ξ - als auch das τ -Register; das Ergebnis der Operation wird in ξ abgelegt.

Diese Funktionen belegen die Register 00-04.

Gleichungen

Es sei

$$z_k = x_k + iy_k = r_k e^{i\theta_k}, \quad k = 1, 2$$

$$z = x + iy = r e^{i\theta}$$

Die Lösung sei in beiden Fällen von der Form $u + iv$.

$$z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2)$$

$$z_1 - z_2 = (x_1 - x_2) + i(y_1 - y_2)$$

$$z_1 z_2 = r_1 r_2 e^{i(\theta_1 + \theta_2)}$$

$$z_1 / z_2 = \frac{r_1}{r_2} e^{i(\theta_1 - \theta_2)}$$

$$|z| = \sqrt{x^2 + y^2}$$

$$1/z = \frac{x}{r^2} - i \frac{y}{r^2}$$

$$z^n = r^n e^{in\theta}$$

$$z^{1/n} = r^{1/n} e^{i\left(\frac{\theta}{n} + \frac{360k}{n}\right)}, \quad k = 0, 1, \dots, n-1$$

(Alle n Wurzeln werden ausgegeben, $k = 0, 1, \dots, n-1$.)

$$e^z = e^x (\cos y + i \sin y), \quad \text{mit } y \text{ in Rad}$$

$$\ln z = \ln r + i\theta, \quad \text{mit } z \neq 0$$

$$a^z = e^{z \ln a}, \quad \text{mit } a > 0 \text{ und reell}$$

$$\log_a z = \frac{\ln z}{\ln a}, \text{ mit } a > 0 \text{ und reell, } z \neq 0$$

$$z^w = e^{w \ln z}, \text{ mit } z \neq 0, w \text{ ist komplex}$$

$$z^{1/w} = e^{\ln z/w}, \text{ mit } z \neq 0, w \text{ ist komplex und } w \neq 0$$

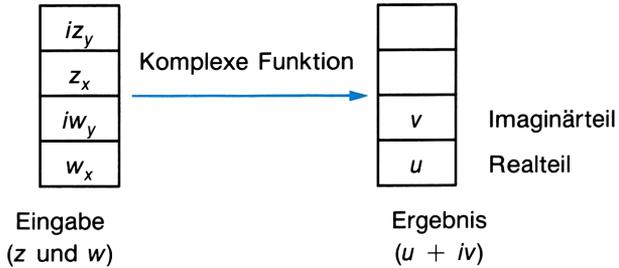
$$\sin z = \sin x \cosh y + i \cos x \sinh y, \text{ Winkel in Rad}$$

$$\cos z = \cos x \cosh y - i \sin x \sinh y, \text{ Winkel in Rad}$$

$$\tan z = \frac{\sin 2x + i \sinh 2y}{\cos 2x + \cosh 2y}, \text{ Winkel in Rad}$$

Anweisungen

Typische Ein- und Ausgabe



Anweisungstabelle für komplexe arithmetische Funktionen

		Umfang: 005
Anweisungen	Eingabe	Anzeige
Komplexe arithmetische Funktionen 1. Geben Sie die erste komplexe Zahl ($z_x + iz_y$) ein. 2. Geben Sie die zweite komplexe Zahl ($w_x + iw_y$) ein. 3. Wählen Sie eine der vier Operationen:	z_y <input type="button" value="ENTER+"/> z_x <input type="button" value="ENTER+"/> w_y <input type="button" value="ENTER+"/> w_x	z_y z_x w_y w_x

**Anweisungstabelle für komplexe arithmetische Funktionen
(Fortsetzung)**

Anweisungen	Eingabe	Anzeige
<ul style="list-style-type: none"> ■ Addition ■ Subtraktion ■ Multiplikation ■ Division 	<p>XEQ C+ R/S</p> <p>XEQ C- R/S</p> <p>XEQ C× R/S</p> <p>XEQ C+ R/S</p>	<p>U = u-Wert V = v-Wert</p>
<p>4. Das Ergebnis der Operation ist im Stack abgelegt; gehen Sie für weiter arithmetische Berechnungen zu Schritt 2 zurück.</p> <p style="text-align: center;">Komplexe Funktionen mit einer komplexen Zahl</p>		
<p>1. Geben Sie die komplexe Zahl ($z_x + iz_y$) ein.</p>	<p>z_y ENTER+ z_x</p>	<p>z_y z_x</p>
<p>2. Wählen Sie eine der folgenden Operationen:</p>		
<ul style="list-style-type: none"> ■ SINZ (sin z) ■ COSZ (cos z) ■ TANZ (tan z) ■ MAGZ (Betrag, z) ■ CINV (1/z) ■ e+z (e^z) ■ LNz (ln z) ■ Z+N (zⁿ, wobei n ganzzahlig ist) ■ Z+1/N (z^{1/n}) 	<p>XEQ SINZ R/S</p> <p>XEQ COSZ R/S</p> <p>XEQ TANZ R/S</p> <p>XEQ MAGZ R/S</p> <p>XEQ CINV R/S</p> <p>XEQ e+z R/S</p> <p>XEQ LNz R/S</p> <p>ENTER+ n XEQ Z+N R/S</p> <p>ENTER+ n XEQ Z+1/N R/S</p>	<p>U = u-Wert V = v-Wert</p> <p>U = u-Wert V = v-Wert</p> <p>U = u-Wert V = v-Wert</p> <p>R = Betrag</p> <p>U = u-Wert V = v-Wert</p> <p>n U = u-Wert V = v-Wert</p> <p>n U = u-Wert V = v-Wert</p>
<p>Beachten Sie, daß n Wurzeln ($u + iv$) bestimmt werden.</p>		

**Anweisungstabelle für komplexe arithmetische Funktionen
(Fortsetzung)**

Anweisungen	Eingabe	Anzeige
<ul style="list-style-type: none"> ■ a+Z (a^z, wobei a reell ist) 	ENTER+ a XEQ a+Z R/S	a U=u-Wert V=v-Wert
<ul style="list-style-type: none"> ■ LOGZ ($\log_a z$, wobei a reell ist) 	ENTER+ a XEQ LOGZ R/S	a U=u-Wert V=v-Wert
Komplexe Funktionen mit zwei komplexen Zahlen		
1. Geben Sie die erste komplexe Zahl ($z_x + iz_y$) ein.	z_y ENTER+ z_x ENTER+	z_y z_x
2. Geben Sie die zweite komplexe Zahl ($w_x + iw_y$) ein.	w_y ENTER+ w_x	w_y w_x
3. Wählen Sie eine der beiden Operationen:		
<ul style="list-style-type: none"> ■ Z+W (z^w) 	XEQ Z+W R/S	U=u-Wert V=v-Wert
<ul style="list-style-type: none"> ■ Z+1/W ($z^{1/w}$) 	XEQ Z+1/W R/S	U=u-Wert V=v-Wert

Bemerkungen

Wenn Flag 04 gesetzt ist, können Sie die einzelnen komplexen Operationen (eigentlich Programme) als Unterprogramme in Ihr eigenes Programm einbauen. Das komplexe Ergebnis wird dann in das X-Register (Realteil) und Y-Register (Imaginärteil) abgelegt.

Beispiele

Es soll der Ausdruck

$$\frac{z_1}{z_2 + z_3}$$

mit $z_1 = 23 + 13i$, $z_2 = -2 + i$ und $z_3 = 4 - 3i$ berechnet werden.

Vorschlag: Da nur jeweils zwei Zahlen vom Programm bearbeitet werden können, sollte die Gleichung umformuliert werden zu

$$z_1 \times [1/(z_2 + z_3)].$$

Tastenfolge

FIX 4

XEQ **SIZE** 005

1 **ENTER**↑
 2 **CHS** **ENTER**↑
 3 **CHS** **ENTER**↑ 4

XEQ **C+**

R/S

XEQ **CINV**

R/S

13 **ENTER**↑ 23

XEQ **C×**

R/S

Anzeige

1.0000
 -2.0000
 4_
 U=2.0000
 V=-2.0000

 U=0.2500
 V=0.2500
 23_
 U=2.5000
 V=9.0000

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $SIZE \geq 005$ ist.

Realteil ($z_2 + z_3$).

Imaginärteil ($z_2 + z_3$).

$1/(z_2 + z_3)$

$z_1/(z_2 + z_3)$

Gesucht sind die drei Kubikwurzeln von 8.

Tastenfolge

0 **ENTER**↑
 8 **ENTER**↑ 3

XEQ **Z↑1/N**

R/S

R/S

R/S

R/S

R/S

Anzeige

0.0000
 3_
 U=2.0000
 V=0.0000
 U=-1.0000
 V=1.7321
 U=-1.0000
 V=-1.7321

Berechnen Sie e^{z-2} mit $z = (1 + i)$.

Tastenfolge

1 **ENTER** ↑
 1 **ENTER** ↑ 2
XEQ **Z↑N**
R/S
XEQ **CINV**
R/S
XEQ **e↑Z**
R/S

Anzeige

1.0000

2_

U=0.0000

 z^2

V=2.0000

U=0.0000

 z^{-2}

V=-0.5000

U=0.8776

 e^{z-2}

V=-0.4794

Berechnen Sie $\sin(2 + 3i)$.

Tastenfolge

3 **ENTER** ↑ 2
XEQ **SINZ**
R/S

Anzeige

2_

U=9.1545

V=-4.1689

VEKTORRECHNUNG

Das Programm VC simuliert einen Ihrem normalen Rechner darübergelegten "Vektor-Rechner". Er definiert die User-Tasten in den beiden obersten Reihen entsprechend den Vektoroperationen: Addition, Subtraktion, Abstand, Skalarprodukt (dot product), Vektorprodukt (cross product), Winkel zwischen zwei Vektoren, Betrag und Einheitsvektor neu. Dieses Modul stellt Ihnen diese Operationen ebenso als reguläre Funktionen (*nicht über* den Vektor-Rechner) zur Verfügung, die Sie wie jede andere Alpha-Funktion des HP-41 ausführen können. Die Alpha-Namen sind im Abschnitt "Zusammenfassung der Vektoroperationen" aufgelistet.

Die Vektoroperationen werden mit *3-dimensionalen* Vektoren, die in *kartesischen Koordinaten* angegeben werden, ausgeführt. Mit anderen Worten, jeder Vektor hat drei Komponenten, V_x , V_y und V_z . Für einen 2-dimensionalen Vektor setzen Sie V_z gleich Null.

Eine Ergänzung zu VC ist das Programm TR, das zur Koordinatentransformation verwendet wird. Das bedeutet, daß Sie auf dieselben Daten sowohl Vektoroperationen als auch Koordinatentransformationen ausführen können, da Sie jeweils das andere Programm in einem der beiden Programme aufrufen können. Das Programm zur Koordinatentransformation wird im nächsten Kapitel, "Koordinatentransformation", beschrieben.

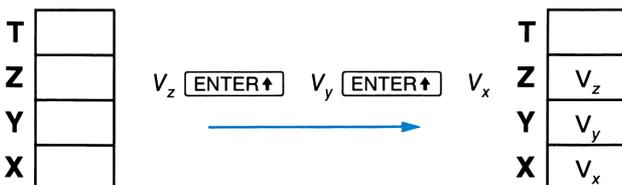
Verfahren

Der Vektor-Stack

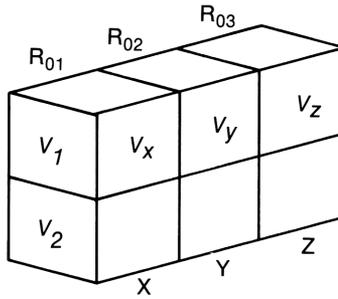
\vec{V}_1	x_1, y_1, z_1
\vec{V}_2	x_2, y_2, z_2

Der Vektor-Rechner (Programm VC) legt einen *Vektor-Stack*, der mit dem regulären RPN Stack (X-, Y-, Z- und T-Register) zusammenarbeitet, an. Wenn Sie die drei Komponenten eines Vektors in der Reihenfolge V_z , V_y , V_x eingeben, sieht die Stackbelegung folgendermaßen aus:

Der RPN-Stack



Welche Beziehung besteht zwischen den beiden Stacks? Grundsätzlich ist die "untere" Ebene des Vektor-Stacks (V_2) in den Registern X, Y und Z des Stacks und die "obere" Ebene des Vektor-Stacks (V_1) in den Speicherregistern R_{01} , R_{02} und R_{03} abgelegt. Sie können sich die Verteilung der erwähnten Register folgendermaßen in einem 3-dimensionalen Stack vorstellen:



Der Vektor-Stack besteht aus zwei Vektorebenen und kann damit zwei Vektoren abspeichern. Beachten Sie jedoch, daß jede Ebene *drei* Komponenten, die x -, y - und z -Komponente, für jeden Vektor enthält.

Das auf der nächsten Seite stehende Diagramm zeigt Ihnen die Vektoreingabe und die Umschichtungen im Vektor-Stack vom Standpunkt des Vektor-Stacks und vom Standpunkt des RPN zusammen mit dem Vektor-Stack.

Wenn Sie zwei Vektoren eingeben (so wie Sie es vor der Ausführung einer typischen Vektoroperation normalerweise machen), wird der erste eingegebene Vektor V_1 und der zweite V_2 . Durch eine "Vektoreingabe" (mit der Funktion \boxed{VE} oder durch Drücken von $\boxed{R/S}$ im Vektor-Rechner) wird der untere Vektor (V_2) in den oberen Vektor (V_1) kopiert. Dann wird der untere Vektor (V_2) mit dem nächsten eingegebenen Vektor überschrieben. Der erste Vektor befindet sich nun in V_1 und der zweite in V_2 .

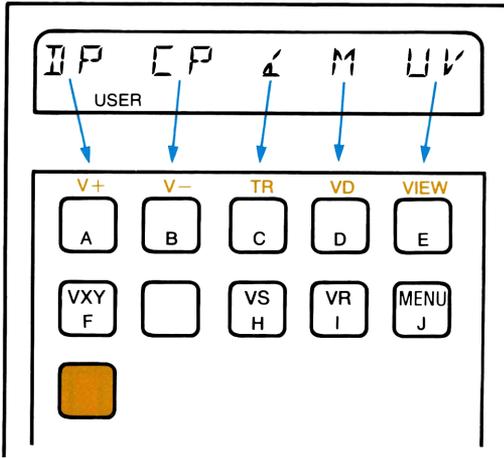
Vektor-Stack Lift

	Geben Sie die Komponenten des Vektors ein:	“Vektor-eingabe:”	Geben Sie die Komponenten des 2. Vektors ein:
Vektor-Stack	V_1 <input type="text"/> V_2 <input type="text" value="x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>"/>	V_1 <input type="text" value="x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>"/> V_2 <input type="text" value="x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>"/> ↑	V_1 <input type="text" value="x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>"/> V_2 <input type="text" value="x<sub>2</sub>, y<sub>2</sub>, z<sub>2</sub>"/>
Eingabe	V_z <input type="text" value="ENTER↑"/> V_y <input type="text" value="ENTER↑"/> V_x	<input type="text" value="VE"/> (oder <input type="text" value="R/S"/>) im Vektor-Rechner)	V_z <input type="text" value="ENTER↑"/> V_y <input type="text" value="ENTER↑"/> V_x
Vektor- und RPN-Stack	V_2 <input type="text" value="x<sub>1</sub> y<sub>1</sub> z<sub>1</sub>"/> X Y Z	R_{01} R_{02} R_{03} V_1 <input type="text" value="x<sub>1</sub> y<sub>1</sub> z<sub>1</sub>"/> ↑ ↑ ↑ V_2 <input type="text" value="x<sub>1</sub> y<sub>1</sub> z<sub>1</sub>"/> X Y Z	R_{01} R_{02} R_{03} V_1 <input type="text" value="x<sub>1</sub> y<sub>1</sub> z<sub>1</sub>"/> V_2 <input type="text" value="x<sub>2</sub> y<sub>2</sub> z<sub>2</sub>"/> X Y Z ↑ ↑ ↑

Ein eventueller Ergebnisvektor wird sowohl in V_1 als auch in V_2 abgelegt und überschreibt damit die früheren Inhalte. Dadurch werden verkettete Vektorrechnungen erleichtert. Beim Abrufen eines Vektors wird V_2 in V_1 kopiert und der abgerufene Vektor in V_2 abgelegt.

Anweisungen

- Durch das Starten von VC (Einschalten des Vektor-Rechners) wird der Vektor-Stack *nicht* gelöscht. Damit können Sie weiterhin die früher abgespeicherten Vektoren für Berechnungen verwenden.
- Vergewissern Sie sich, jedem Vektor 3 Dimensionen zuzuweisen. Bei 2-dimensionalen Vektoren geben Sie für V_z Null ein.
- Geben Sie die Koordinaten eines Vektors in *kartesischer* Form ein. Polarkoordinaten (Betrag und Winkel) können Sie für einen 2-dimensionalen Vektor mit Hilfe der Funktion (*Polar in kartesisch*) in kartesische Koordinaten umwandeln.
- Für Winkel betreffende Funktionen wird die Einheit des eingestellten Winkelmodus verwendet (DEG, RAD oder GRAD).
- Die Funktion “Ansehen” () ist für das nachträgliche Ansehen des im Stack abgelegten Vektors V_2 sehr hilfreich.
- Mit V_1 bezeichnet man den “oberen” Vektor; in R_{01} , R_{02} und R_{03} abgespeichert. Mit V_2 bezeichnet man den “unteren” Vektor; in X, Y und Z abgespeichert.



Dieses Menü gibt Ihnen an, welche Vektoroperationen Sie mit dem Programm VC ausführen können. Sie können sich dieses Menü jederzeit durch Drücken der Taste \boxed{J} in der Anzeige ansehen.

Durch Drücken der Taste $\boxed{\leftarrow}$ können Sie das Menü jederzeit löschen. Dann erscheint in der Anzeige der Inhalt des X-Registers. Das Programm wird dadurch aber nicht beendet. Sie können nun Rechnungen durchführen und dann das Menü durch Drücken von \boxed{J} wieder in die Anzeige holen. (Sie müssen die Anzeige des Programms *nicht* löschen, bevor Sie Rechnungen ausführen.)

Der Vektor-Rechner stellt Ihnen zwei Verfahren zur Eingabe von Vektoren in den Vektor-Stack zur Verfügung. Die Vektoreingabe-Funktion (\boxed{VE}) ist analog zur Taste $\boxed{ENTER\uparrow}$. Eine kürzere Methode ist die Eingabe des Vektors mit Hilfe der Taste $\boxed{R/S}$. Die Eingabe der Komponenten über die Taste $\boxed{R/S}$ hat die gleiche Funktion wie \boxed{VE} , wenn zuletzt das Menü angezeigt wurde.

Die folgende Tabelle enthält die für das Ausführen von Vektoroperationen im Vektor-Rechner (Programm VC) notwendigen Tastenfolgen. Jede Operation wird im Abschnitt "Zusammenfassung der Vektoroperationen" (steht nach der Anweisungstabelle) beschrieben.

Anweisungstabelle für VC

		Umfang: 004
Anweisungen	Eingabe	Anzeige
1. Starten Sie das Programm VC für den Vektor-Rechner.	<input type="button" value="XEQ"/> <input type="button" value="VC"/> *	DP CP Δ M UV
2. Geben Sie die drei Komponenten des ersten Vektors (V_1) ein. Trennen Sie die Eingabe zweier Vektoren nach den drei Komponenten durch "Vektoreingabe": Führen Sie <input type="button" value="VE"/> aus oder—nur falls vor der Eingabe der ersten Komponente zuletzt das Menü angezeigt wurde—drücken Sie <input type="button" value="R/S"/> .	z_1 <input type="button" value="ENTER+"/> y_1 <input type="button" value="ENTER+"/> x_1 <input type="button" value="R/S"/>	z_1 y_1 DP CP Δ M UV
3. Geben Sie den zweiten Vektor (V_2) ein. Drücken Sie nicht <input type="button" value="R/S"/> .	z_2 <input type="button" value="ENTER+"/> y_2 <input type="button" value="ENTER+"/> x_2	z_2 y_2 x_2
4. Anzeigen des Menüs (wahlfrei).	<input type="button" value="J"/>	DP CP Δ M UV
5. Führen Sie eine Vektoroperation aus:		
■ Skalares Produkt, $V_1 \cdot V_2$	<input type="button" value="A"/> (DP)	DOT=Ergebnis
■ Vektorprodukt, $V_1 \times V_2$	<input type="button" value="B"/> (CP) <input type="button" value="R/S"/> † <input type="button" value="R/S"/> †	X=x Ergebnis Y=y Ergebnis Z=z Ergebnis
■ Winkel zwischen V_1 und V_2	<input type="button" value="C"/> (Δ)	Δ =Ergebnis
■ Betrag (Länge) von V_2 (Dadurch wird ebenso der Einheitsvektor zu V_2 in Y, Z, und T abgelegt.)	<input type="button" value="D"/> (M)	M=Ergebnis
■ Einheitsvektor zu V_2 (Dadurch wird ebenso der Betrag ins T-Register abgelegt.)	<input type="button" value="E"/> (UV) <input type="button" value="R/S"/> † <input type="button" value="R/S"/> †	X=x Ergebnis Y=y Ergebnis Z=z Ergebnis
■ Vektoraddition, $V_1 + V_2$	<input type="button" value="A"/> <input type="button" value="A"/> <input type="button" value="R/S"/> † <input type="button" value="R/S"/> †	X=x Ergebnis Y=y Ergebnis Z=z Ergebnis
■ Vektorsubtraktion, $V_1 - V_2$	<input type="button" value="B"/> <input type="button" value="B"/> <input type="button" value="R/S"/> † <input type="button" value="R/S"/> †	X=x Ergebnis Y=y Ergebnis Z=z Ergebnis
■ Koordinatentransformation—schlagen Sie dazu im Kapitel "Koordinatentransformation" nach. <input type="button" value="USER"/> <input type="button" value="C"/> schaltet den Vektor-Rechner wieder ein.	<input type="button" value="C"/> <input type="button" value="C"/> <input type="button" value="USER"/> <input type="button" value="C"/>	Z0,Y0,X0 ? DP CP Δ M UV
■ Abstand zwischen V_1 und V_2	<input type="button" value="D"/> <input type="button" value="D"/>	d=Ergebnis

Anweisungstabelle für VC (Fortsetzung)

Anweisungen	Eingabe	Anzeige
6. Lassen Sie sich das Hauptmenü nach oder während den Operationen anzeigen, wenn gewünscht.	\boxed{J} (oder $\boxed{R/S}$)	DP CP Δ M UV
7. Um die Komponenten von V_2 , im Vektor-Stack, anzusehen:	$\blacksquare \boxed{E}$ $\boxed{R/S} \dagger$ $\boxed{R/S} \dagger$	X=x-Koordinate Y=y-Koordinate Z=z-Koordinate
8. Um V_1 und V_2 zu vertauschen (die Vektor-komponenten in R_{01} , R_{02} und R_{03} werden mit denen in X, Y und Z vertauscht):	\boxed{F}	DP CP Δ M UV
9. Um die Komponenten von V_2 als Vektor-Register n in R_{3n+1} , R_{3n+2} und R_{3n+3} abzuspeichern ($n \geq 0$):	$n \boxed{H}$	DP CP Δ M UV
10. Um den Inhalt des Vektor-Registers n in V_2 (X, Y und Z) abzurufen. V_2 wird in V_1 verlagert:	$n \boxed{I}$ $\boxed{R/S} \dagger$ $\boxed{R/S} \dagger$	X=x-Koordinate Y=y-Koordinate Z=z-Koordinate
<p>* Zur Ausführung eines Programms drücken Sie entweder \boxed{XEQ} \boxed{ALPHA} Alpha-Name \boxed{ALPHA} oder eine entsprechende benutzerspezifische Taste.</p> <p>† Diese Tastenfolge ist nicht notwendig, wenn Sie einen Drucker angeschlossen haben, da er automatisch alle Ergebnisse druckt und dann das Menü angezeigt wird.</p>		

Bemerkungen

Sie können durch Setzen des Flags 04 die Anzeige von Ergebnissen im Vektor-Rechner unterdrücken. Aufeinanderfolgende Berechnungen werden dann schneller ausgeführt. Sie können Ergebnisse jedoch jederzeit anzeigen lassen, indem Sie $\blacksquare \boxed{E}$ drücken.

Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten \boxed{A} - \boxed{F} , \boxed{H} - \boxed{J} und $\blacksquare \boxed{A}$ - $\blacksquare \boxed{E}$ zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung überschrieben. Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten. Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Zusammenfassung der Vektoroperationen

Auf Vektoroperationen kann auf zwei Arten zugegriffen werden:

- Entweder benutzen Sie den Vektor-Rechner und seine neu definierten User-Tasten wie anfangs erklärt.
- Oder Sie führen eine Vektorfunktion direkt durch Verwenden des Alpha-Namens wie jede andere Alpha-Funktion des HP-41 aus.
- Mit V_1 bezeichnet man den ersten (bzw. "oberen") Vektor; in R_{01} , R_{02} und R_{03} abgespeichert. Mit V_2 bezeichnet man den zweiten (bzw. "unteren") Vektor; in X , Y und Z abgespeichert.

Die Ausführung der Operationen hängt nicht von der Art ihrer Ausführung ab. Alle Operationen sind unten zusammen mit entsprechendem Alpha-Namen und Beschreibung in der Tabelle aufgeführt.* Sie können diese Operationen im Vektor-Rechner ebenfalls mit ihren Alpha-Namen aufrufen. In diesem Fall ist es jedoch bequemer, die im Vektor-Rechner neu definierten User-Tasten zu benutzen.

Wenn Sie Vektoroperationen ohne Vektor-Rechner benutzen,—d.h., wenn Sie Alpha-Namen (wie unten aufgeführt) verwenden—ist es am besten, wenn **USER** nicht eingeschaltet ist (User-Tastenfeld inaktiv). Dadurch werden Konflikte zwischen vom Vektor-Rechner für User-Tasten gemachte Zuweisungen und den ursprünglichen Funktionen der Tastatur (wie z.B. \boxed{xy}) vermieden.

Tabelle für Vektoroperationen

Funktion	Wirkung
$\boxed{\text{CROSS}}$ (Vektorprodukt)	$V_1 \times V_2$. Berechnet das 3-dimensionale Produkt und legt es in V_2 (in X, Y, Z) ab. Eine Kopie wird ebenso in V_1 abgelegt. R_0 wird nicht erhalten. Der Vektor-Rechner benutzt auch $\boxed{\text{B}}$ (CP).
$\boxed{\text{DOT}}$ (Skalarprodukt)	$V_1 \cdot V_2$. Legt das Skalarprodukt im X-Register ab. (Der Inhalt von V_2 wird zerstört; V_1 wird nicht beeinflusst.) Der Vektor-Rechner benutzt auch $\boxed{\text{A}}$ (DP).
$\boxed{\text{TR}}$ (Koordinatentransformation)	Ruft das Programm zur Koordinatentransformation TR auf. Schlagen Sie dazu im nächsten Kapitel nach. Der Vektor-Rechner benutzt auch $\boxed{\text{C}}$.

* Die Operationen "Vektor ansehen" und "Betrag" können nur im Vektor-Rechner ausgeführt werden. Der Betrag wird jedoch auch bei Ausführung der Operation "Einheitsvektor" berechnet.

Tabelle für Vektoroperationen (Fortsetzung)

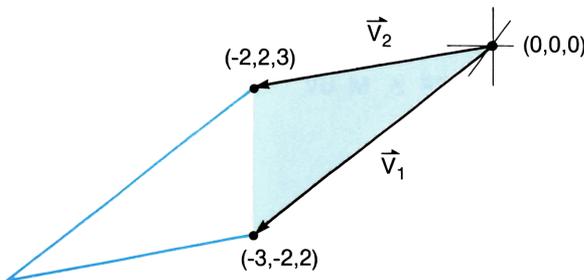
Funktion	Wirkung
<p>UV (Einheitsvektor)</p>	<p>Wandelt V_2 (in X, Y, Z) in seinen entsprechenden Einheitsvektor um und speichert seinen Betrag im T-Register ab. (V_1 wird nicht beeinflusst.) Der Vektor-Rechner benutzt auch E (UV). Beachte: der Einheitsvektor zu (0,0,0) ist (0,0,1) mit dem Betrag gleich Null.</p>
<p>V+ (Vektoraddition)*</p>	<p>$V_1 + V_2$. Speichert die Summe sowohl in V_1 als auch in V_2 ab. Der Vektor-Rechner benutzt auch A.</p>
<p>V- (Vektorsubtraktion)*</p>	<p>$V_1 - V_2$. Speichert die Differenz sowohl in V_1 als auch in V_2 ab. Der Vektor-Rechner benutzt auch B.</p>
<p>V* (Vektor multipliziert mit einem Skalar)</p>	<p>$V_2 * a$. Multipliziert V_2 (in Y, Z, T) mit a in (X-Register) und speichert Ergebnis in X, Y und Z ab.</p>
<p>VΔ (Winkel zwischen Vektoren)</p>	<p>Speichert den Winkel im X-Register ab. Die Einheit ist entsprechend dem eingestellten Winkelmodus. V_1 und V_2 werden nicht erhalten; der Einheitsvektor zu V_2 wird in V_1 abgespeichert. Der Vektor-Rechner benutzt auch C (Δ). Beachte: es wird angenommen, daß Vektor (0,0,0) die gleiche Richtung wie (0,0,1) hat.</p>
<p>VD (Abstand zwischen Vektoren)</p>	<p>Speichert den skalaren Abstand zwischen V_1 und V_2 im X-Register ab. Ebenso wird der Differenzvektor ($V_1 - V_2$) in V_1 abgespeichert. V_2 wird nicht erhalten. Der Vektor-Rechner benutzt auch D.</p>
<p>VE (Vektoreingabe)</p>	<p>Analog zu ENTER↑. Wird verwendet, um die Eingabe von zwei Vektoren vor der Ausführung einer Operation zu trennen (V_1, dann V_2). (Vektoreingabe kopiert den ersten Vektor aus X, Y, Z in R_{01}, R_{02}, R_{03}.) Im Vektor-Rechner können Sie stattdessen R/S drücken, jedoch nur, wenn zuletzt das Menü angezeigt wurde.</p>
<p>VR (Vektorabruf)</p>	<p>Mit n ($n > 0$)† im X-Register wird V_2 in V_1 kopiert und dann ein 3-dimensionaler Vektor vom Vektor-Register n in V_2 (X, Y und Z) aus den Speicherregistern R_{3n+1}, R_{3n+2} und R_{3n+3} abgerufen. Analog zu RCL. (Der vorherige Inhalt von V_2 wird nach V_1 verlagert und überschreibt damit V_1.) Der Vektor-Rechner benutzt auch I.</p>

Tabelle für Vektoroperationen (Fortsetzung)

Funktion	Wirkung
<p>[VS] (Vektor abspeichern)</p> <p>[VXY] (Vektoren vertauschen)</p>	<p>Mit n ($n > 0$)† im X-Register wird V_2 (nun in Y, Z und T) als Vektor-Register n in den Speicherregistern R_{3n+1}, R_{3n+2} und R_{3n+3} abgelegt. Analog zu [STO]. (V_2 wird nicht beeinflusst.) Der Vektor-Rechner benutzt auch [H].</p> <p>V_1 vertauscht Werte mit V_2. Die Koordinaten x_1, y_1 und z_1 werden aus R_{01}, R_{02} und R_{03} im X-, Y- und Z-Register abgelegt, während x_2, y_2 und z_2 aus X, Y, und Z in R_{01}, R_{02} und R_{03} geschrieben werden. Der Vektor-Rechner benutzt auch [F] ([xsy]).</p>
<p>* Denken Sie daran, daß + und – umgeschaltete Alpha-Zeichen sind.</p> <p>† Falls $n = 0$ ist, kopieren sowohl [VR] als auch [VS] V_2 in V_1 (wie [VE]). Verwenden Sie nicht $n < 5$, wenn Sie die Vektoren für das Benutzen im Programm TR abspeichern ([C]).</p>	

Beispiele

Berechnen Sie die Fläche des durch die Vektoren $V_1 = (-3, -2, 2)$ und $V_2 = (-2, 2, 3)$ definierten Dreiecks. Erinnern Sie sich, daß die Fläche des durch die Vektoren V_1 und V_2 definierten Parallelogramms dem Betrag von $V_1 \times V_2$ entspricht.



Tastenfolge

FIX 4

XEQ SIZE 004

XEQ VC

2 ENTER↑
2 CHS ENTER↑
3 CHS R/S3 ENTER↑
2 ENTER↑
2 CHS
JB (CP)
R/S
R/S

J or R/S

D (M)

2 ÷

Anzeige

DP CP Δ M UV

2.0000
-2.0000
DP CP Δ M UV3.0000
2.0000
-2_
DP CP Δ M UVX = -10.0000
Y = 5.0000
Z = -10.0000

DP CP Δ M UV

M = 15.0000

7.5000

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $SIZE \geq 004$ ist.

Startet den Vektor-Rechner. (Sie können ebenso die Operationen direkt, ohne den Vektor-Rechner, benutzen.)

Geben Sie zuerst z_1 , dann y_1 und zuletzt x_1 ein. Beenden Sie die Eingabe mit "Vektoreingabe".

Geben Sie der Reihe nach z_2 , y_2 und x_2 ein.

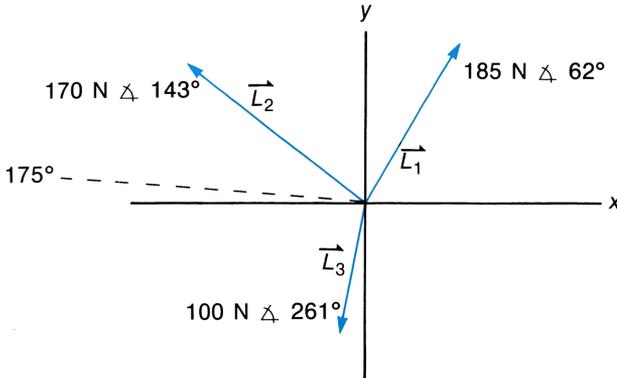
Zeigt das Vektormenü erneut an (optional).

Das Ergebnis lautet $(-10, 5, -10)$.

Für das Bestimmen des Betrags bereit. Betrag, der der Fläche des Parallelogramms entspricht.

Dies ist die Dreiecksfläche, die halb so groß wie die Fläche des Parallelogramms ist.

Bestimmen Sie aus drei gegebenen Kräften die Kraft in die 175° -Richtung. Benutzen Sie dazu das Skalarprodukt des Summenvektors mit dem Einheitsvektor in 175° -Richtung. Dabei müssen Sie zuerst die Polarkoordinaten in kartesische Koordinaten umwandeln. Vergessen Sie nicht, $z = 0$ zu setzen.



Speichern Sie die kartesischen Koordinaten von L_3 und des Einheitsvektors in 175° -Richtung so ab, damit Sie sie zur Bestimmung der Kraft in 175° -Richtung (Skalarprodukt) nach Verdopplung von L_3 wieder verwenden können. In diesem Beispiel werden diese Vektoren in Vektor-Register 1 und 2 abgelegt.

Die Lösung wurde mit Hilfe der Ausführung von Alpha-Funktionen (manuell) und nicht mit Hilfe des Vektor-Rechners gefunden. Sie hätten jedoch genauso wie im vorigen Beispiel den Vektor-Rechner verwenden können. Vergewissern Sie sich, daß das User-Tastenfeld nicht aktiv ist.

Tastenfolge

XEQ **SIZE** 010

Anzeige

Optional—gibt die Anzahl der für dieses Beispiel benötigten Speicherregister (einschließlich Vektorab-speicherung) an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $\text{SIZE} \geq 010$ ist.

Tastenfolge

XEQ DEG*

0 ENTER↑

62 ENTER↑

185 P→R

x↔y †

x↔y †

XEQ VE

0 ENTER↑

143 ENTER↑

170 P→R

XEQ V+

Anzeige

0.0000

62.0000

86.8522

163.3453

86.8522

86.8522

0.0000

143.0000

-135.7680

-48.9158

Vergewissern Sie sich, daß der Winkelmodus auf DEG eingestellt ist.

Geben Sie für die z-Koordinate Null ein (in Vorbereitung auf Vektoroperationen nach der Koordinatentransformation).

Um L_1 in kartesische Koordinaten umzuwandeln, geben Sie zuerst θ und dann r ein.

x -Koordinate von L_1 .

y -Koordinate von L_1 . Dieser Schritt ist optional—dadurch können Sie y ansehen.

x wird in X und y in Y zurückgespeichert—nur notwendig, falls Sie sie zuvor vertauscht haben (im vorigen Schritt).

Kein Menü; es wird das vorherige Ergebnis angezeigt.

Zeigt x_2 an. L_2 ist in kartesische Koordinaten umgewandelt.

x -Koordinate des Summenvektors (in V_1 und V_2).

* Falls der **USER**-Indikator angezeigt wird, drücken Sie **USER**, um den User-Modus auszuschalten.

† Beachten Sie, daß Sie **x↔y** innerhalb des Vektor-Rechners für das Vertauschen von X und Y nicht benutzen können, da diese Taste im Vektor-Rechner für das Vertauschen von V_1 und V_2 neu definiert ist. Verwenden Sie stattdessen **R+**.

Tastenfolge

0 [ENTER↑]
 261 [ENTER↑]
 100 [P→R]

1 [XEQ] [VS]

[XEQ] [V+]

0 [ENTER↑]
 175 [ENTER↑]
 1 [P→R]

2 [XEQ] [VS]

[XEQ] [DOT]

[XEQ] [VXY]

1 [XEQ] [VR]

[XEQ] [V+]

2 [XEQ] [VR]

[XEQ] [DOT]

Anzeige

0.0000
 261.0000
 -15.6434

-15.6434

-64.5592

0.0000
 175.0000
 -0.9962

-0.9962

78.8586

-64.5592

-15.6434

-80.2027

-0.9962

85.8342

x_3 , L_3 ist in kartesische Koordinaten umgewandelt.

Speichert L_3 in Vektor-Register 1 ab (R_4 , R_5 , R_6).

x -Koordinate des Summenvektors $L_1 + L_2 + L_3$. In V_1 und V_2 abgelegt.

x -Koordinate des Einheitsvektors in 175° -Richtung.

Speichert kartesische Koordinaten des Einheitsvektors in 175° -Richtung im Vektor-Register 2 (R_7 , R_8 , R_9) ab.

Das Skalarprodukt ist die Projektion des resultierenden Vektors L auf die 175° -Achse.

Legt die Projektion des Summenvektors $L_1 + L_2 + L_3$ in V_2 (X , Y , Z) ab.

Ruft L_3 ab.

Addiert L_3 zum vorherigen Summenvektor (eigentlich Verdopplung von L_3).

Ruft Koordinaten des Einheitsvektor in 175° -Richtung ab.

Bestimmt das neue Skalarprodukt für die Projektion des neuen Summenvektors auf die 175° -Achse.

Programmier-Informationen

Die nachfolgend aufgeführten Unterprogramme in VC können Sie in Ihren eigenen Programmen verwenden. Es sind 3-dimensionale Vektor-Operationen mit einem oder zwei Vektoren.

Speicherminimum: SIZE 004, Vektorabspeichern und Vektorabrufen nicht eingeschlossen.

Unterprogramme

Name des Unterprogramms	alter Registerinhalt	neuer Registerinhalt
GROSS (Vektorprodukt)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = (V_1 \times V_2)_x$ $Y = (V_1 \times V_2)_y$ $Z = (V_1 \times V_2)_z$ $R_{00} = \text{Zwischenspeicher}$ $R_{01} = (V_1 \times V_2)_x$ $R_{02} = (V_1 \times V_2)_y$ $R_{03} = (V_1 \times V_2)_z$
DOT (Skalarprodukt)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = V_1 \cdot V_2$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$
V+ (Vektoraddition)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = V_{1x} + V_{2x}$ $Y = V_{1y} + V_{2y}$ $Z = V_{1z} + V_{2z}$ $R_{01} = V_{1x} + V_{2x}$ $R_{02} = V_{1y} + V_{2y}$ $R_{03} = V_{1z} + V_{2z}$
V- (Vektorsubtraktion)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = V_{1x} - V_{2x}$ $Y = V_{1y} - V_{2y}$ $Z = V_{1z} - V_{2z}$ $R_{01} = V_{1x} - V_{2x}$ $R_{02} = V_{1y} - V_{2y}$ $R_{03} = V_{1z} - V_{2z}$
V* (Vektormultiplikation S _i)	$X = a$ $Y = V_x$ $Z = V_y$ $T = V_z$	$X = V_x * a$ $Y = V_y * a$ $Z = V_z * a$
VΔ (Winkel zwischen ; Vekt)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = \text{spitzer Winkel zwischen } V_1 \text{ und } V_2$ $R_{01} = \text{Einheitsvektor } V_{2x}$ $R_{02} = \text{Einheitsvektor } V_{2y}$ $R_{03} = \text{Einheitsvektor } V_{2z}$

Unterprogramme (Fortsetzung)

Name des Unterprogramms	alter Registerinhalt	neuer Registerinhalt
VD (Vektorabstand)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X = \text{Abstand zwischen } V_1 \text{ und } V_2$ $R_{01} = V_{1x} - V_{2x}$ $R_{02} = V_{1y} - V_{2y}$ $R_{03} = V_{1z} - V_{2z}$
VE (Vektoreingabe)	$X\text{-Register} = V_x$ $Y\text{-Register} = V_y$ $Z\text{-Register} = V_z$	$X\text{-Register} = V_x$ $Y\text{-Register} = V_y$ $Z\text{-Register} = V_z$ $R_{01} = V_x$ $R_{02} = V_y$ $R_{03} = V_z$
VR (Vektorabruf)	$X = n$ $Y = V_x$ $Z = V_y$ $T = V_z$	$X = R_{3n+1}$ $Y = R_{3n+2}$ $Z = R_{3n+3}$ $R_{01} = V_x$ $R_{02} = V_y$ $R_{03} = V_z$
VS (Vektor abspeichern)	$X = n$ $Y = V_x$ $Z = V_y$ $T = V_z$	$X = V_x$ $Y = V_y$ $Z = V_z$ $R_{3n+1} = V_x$ $R_{3n+2} = V_y$ $R_{3n+3} = V_z$
VXY (Vektoren vertauschen)	$X\text{-Register} = V_{2x}$ $Y\text{-Register} = V_{2y}$ $Z\text{-Register} = V_{2z}$ $R_{01} = V_{1x}$ $R_{02} = V_{1y}$ $R_{03} = V_{1z}$	$X\text{-Register} = V_{1x}$ $Y\text{-Register} = V_{1y}$ $Z\text{-Register} = V_{1z}$ $R_{01} = V_{2x}$ $R_{02} = V_{2y}$ $R_{03} = V_{2z}$
UV (Einheitsvektor)	$X = V_x$ $Y = V_y$ $Z = V_z$	$X = \text{Einheitsvektor } x$ $Y = \text{Einheitsvektor } y$ $Z = \text{Einheitsvektor } z$ $T = \text{Betrag}$

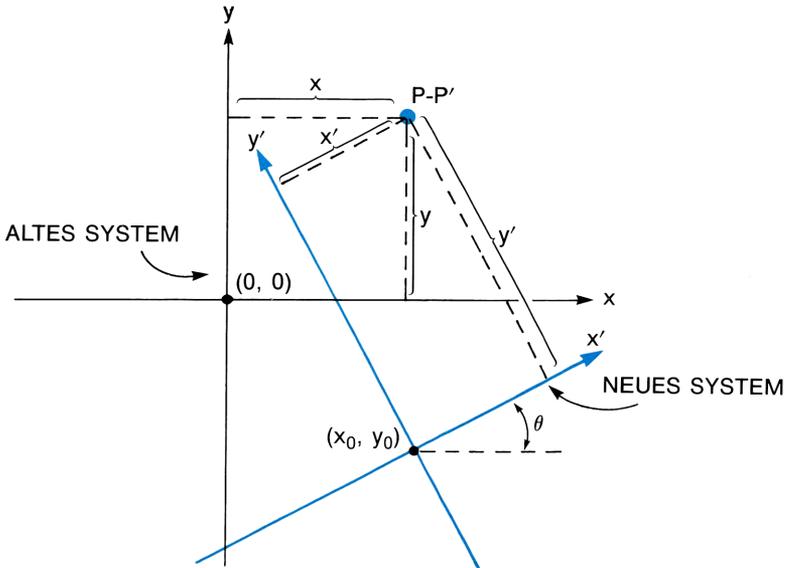
Anmerkungen. Die Vektoroperationen werden auf einen bzw. zwei Vektoren ausgeführt. Der eine wird im Stack (X-, Y- und Z-Register), der andere in R_{01} , R_{02} und R_{03} abgespeichert. Bei einer Operation mit zwei Vektoren wird V_1 in R_{01} – R_{03} und V_2 im Stack erwartet. Die Vektorkomponenten sind der Reihe nach abgespeichert, d.h. V_x , V_y und V_z in X, Y und Z bzw. in R_{01} , R_{02} und R_{03} .

KOORDINATENTRANSFORMATION

Das Programm TR führt drei-dimensionale Koordinaten-Verschiebung und/oder Rotation aus. Es verwendet Teile des Programms VC für Vektoroperationen. Sie können auf TR entweder direkt oder über VC zugreifen. (VC und der Vektor-Rechner wurden im vorausgehenden Kapitel "Vektorrechnung" beschrieben.)

Das Programm erfragt die Koordinaten des Ursprungs des *neuen* Systems (x_0, y_0, z_0) , den Rotationswinkel des neuen Systems in Bezug auf das *ursprüngliche* System und die Rotationsachse. Dann können Sie die Koordinaten der Punkte im ursprünglichen System (x, y, z) eingeben, für die Sie die Koordinaten im neuen System (x', y', z') bestimmen möchten, bzw. die Koordinaten im neuen System zur Berechnung der Koordinaten im ursprünglichen System. Für eine 2-dimensionale Koordinatentransformation geben Sie für z_0 Null ein.

Zwei-dimensionale Drehung um die Achse $(0, 0, 1)$



Nachdem Sie den neuen Ursprung (x, y, z) spezifiziert haben, geben Sie den Rotationswinkel ein. Im 3-dimensionalen Fall, mit einem Rotationswinkel ungleich Null spezifizieren Sie zusätzlich den *Rotationsvektor* (a, b, c) . Der Rotationsvektor definiert die Rotationsachse; der Betrag kann jeden Wert ungleich Null annehmen.

Gleichungen

$$\vec{P}' = [(\vec{P} - \vec{T}) \cdot \vec{n}] \vec{n} (1 - \cos\theta) + (\vec{P} - \vec{T}) \cos\theta + [(\vec{P} - \vec{T}) \times \vec{n}] \sin\theta$$

$$\vec{P} = [(\vec{P}' \cdot \vec{n}) \vec{n} (1 - \cos\theta) + \vec{P}' \cos\theta + (\vec{P}' \times \vec{n}) \sin(-\theta)] + \vec{T}$$

mit

\vec{P}' = Vektor im neuen System

\vec{P} = Vektor im alten System

\vec{T} = Ursprung des neuen Systems

\vec{n} = Einheits-Rotationsvektor (a, b, c)

θ = Rotationswinkel

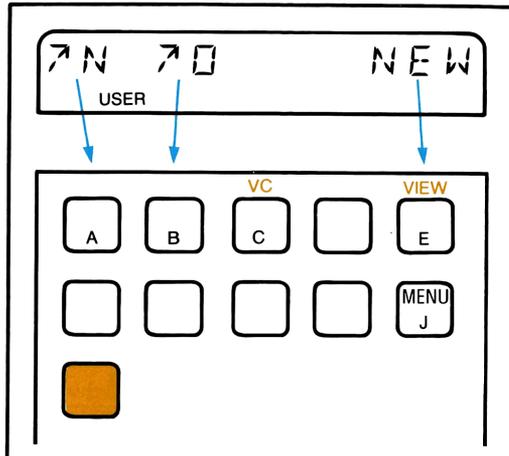
Zwei-dimensionale Transformationen werden als Spezialfall der drei-dimensionalen Transformationen behandelt. Dabei wird (a, b, c) zu $(0, 0, 1)$ gesetzt.

Anweisungen

Sie können TR entweder direkt ($\boxed{\text{XEQ}} \boxed{\text{TR}}$) oder vom Vektor-Rechner aus ($\boxed{\text{C}}$) in VC starten. Der Vektor-Rechner wird im Kapitel "Vektorrechnung" beschrieben.

Geben Sie die Koordinaten in kartesischer Form ein und spezifizieren Sie die Winkel entsprechend des eingestellten Winkelmodus (DEG-, RAD- oder GRAD-Modus).

- Bei zwei Dimensionen geben Sie für die z-Koordinate Null ein.
- Bei reiner Verschiebungen ist für den Rotationswinkel Null einzugeben.
- Bei reiner Rotation sind x_0, y_0 und z_0 gleich Null zu setzen.
- Das Vorzeichen des Rotationswinkels wird nach der Rechte-Hand-Regel und der Richtung des Rotationsvektors bestimmt. Bei zwei Dimensionen wird eine Drehung gegen den Uhrzeigersinn als positive Drehung bezeichnet.
- Sie können den Vektor-Rechner ein- bzw. ausschalten, indem Sie $\boxed{\text{C}}$ ("C" für Calculator und Coordinate transformations) drücken. Im Vektor-Rechner können Sie Operationen auf die im Stack und in Speicherregistern abgelegten Koordinaten von Vektoren ausführen. (Schlagen Sie für Informationen über die Speicherorte von Vektorkoordinaten im Abschnitt "Bemerkungen" nach.)
- Die Funktion "Anzeigen" ($\boxed{\text{E}}$) ist für das Ansehen der im Stack abgelegten Vektorkoordinaten sehr nützlich.



Nachdem Sie die Werte der Variablen eingegeben haben, wird das Menü angezeigt. Es gibt an, welche User-Tasten mit welchen Funktionen in TR belegt sind. Sie können das Menü jederzeit, wenn der **USER-Indikator** angezeigt wird (falls er nicht angezeigt wird, drücken Sie die [USER]-Taste), durch Drücken von [J] anzeigen. Ebenso können Sie, wenn der Rechner Ergebnisse anzeigt, [R/S] sofort drücken, bis das Menü angezeigt wird. Der Programmablauf wird dadurch in keiner Weise gestört.

Durch Drücken der Taste [←] können Sie das Menü jederzeit löschen. Dadurch wird der Inhalt des X-Registers angezeigt, das Programm jedoch nicht beendet. Nun können Sie Berechnungen ausführen und dann durch Drücken der Taste [R/S] das Programm fortsetzen. (Sie müssen die Programmanzeige *nicht* löschen, bevor Sie die Berechnungen ausführen.)

Anweisungstabelle für TR

		Umfang: 017
Anweisungen	Eingabe	Anzeige
1. Starten Sie das Programm TR. Die Anzeige des Menüs gibt die neue Belegung der obersten Reihe des User-Tastenfelds mit entsprechenden Funktionen an.	[XEQ] [TR]*	Z0,Y0,X0 ?
2. Geben Sie den Ursprung des neuen Koordinatensystems ein.	z ₀ [ENTER↑] y ₀ [ENTER↑] x ₀ [R/S]	z ₀ y ₀ ROT Δ ?
3. Geben Sie den Rotationswinkel für das neue Koordinatensystem ein.	θ [R/S]	c,b,a ?

Anweisungstabelle für TR (Fortsetzung)

Anweisungen	Eingabe	Anzeige
4. Für ein 3-dimensionales System: Geben Sie die Koordinaten des Rotationsvektors ein. Für ein 2-dimensionales System: Drücken Sie $\boxed{R/S}$.	c $\boxed{ENTER\uparrow}$ b $\boxed{ENTER\uparrow}$ a $\boxed{R/S}$	c b †N †O NEW
5. Um die Koordinaten eines Punktes vom alten ins neue System zu transformieren (†N), geben Sie die drei Koordinaten des Punktes ein und drücken Sie †N. (Setzen Sie für 2 Dimensionen z=0.)	z $\boxed{ENTER\uparrow}$ y' $\boxed{ENTER\uparrow}$ x \boxed{A} (†N) $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$	z y' X= x' Y= y' Z= z' †N †O NEW
6. Um die Koordinaten eines Punktes vom neuen ins alte System zu transformieren (†O), geben Sie die drei Koordinaten des Punktes ein und drücken Sie †O. (Setzen Sie für 2 Dimensionen z=0.)	z' $\boxed{ENTER\uparrow}$ y' $\boxed{ENTER\uparrow}$ x' \boxed{B} (†O) $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$	z' y' X= x Y= y Z= z †N †O NEW
7. Um die im Stack abgelegten Koordinaten des Punktes anzusehen:	\boxed{E} $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$ $\boxed{R/S}\uparrow$	X=x-Koordinate Y=y-Koordinate Z=z-Koordinate †N †O NEW
8. Für das Transformieren weiterer Koordinaten gehen Sie zu Schritt 5 bzw. 6 zurück.		
9. Um ein neues transformiertes Koordinatensystem einzugeben, wählen Sie NEW und gehen zu Schritt 2 zurück.	\boxed{E} (NEW)	Z0,Y0,X0 ?
10. Um Vektoroperationen anzuwenden, schalten Sie den Vektor-Rechner ein. Dann sind alle im Kapitel "Vektorrechnung" beschriebenen Funktionen für Sie verfügbar.	\boxed{C} (USER -Indikator muß an sein)	DP CP Δ M UV
11. Um aus dem VC ins Programm TR zurückzukehren:	\boxed{C}	Z0,Y0,X0 ?
12. Um einen Vektor V_2 als Ergebnis aus VC zu transformieren, übergeben Sie die Eingabewerte und rufen das Hauptmenü auf (vorausgesetzt, daß ein transformiertes Koordinatensystem bereits definiert wurde).	\boxed{USER} \boxed{J}	†N †O NEW

* Zur Ausführung eines Programms drücken Sie entweder \boxed{XEQ} \boxed{ALPHA} Alpha-Name \boxed{ALPHA} oder eine entsprechende benutzerspezifische Taste.

† Diese Tastenfolge ist nicht notwendig, wenn Sie einen Drucker angeschlossen haben, da er automatisch alle Ergebnisse druckt und dann das Menü angezeigt wird.

Bemerkungen

Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten **[A]**, **[B]**, **[E]**, **[C]** und **[J]** zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung *überschrieben*. *Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten.* Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Diese lokalen Alpha-Labels sind jedoch nur aktiv, wenn der **USER**-Indikator an ist. Sie können, wenn der **USER**-Indikator ausgeschaltet ist, die arithmetischen Funktionen in den beiden oberen Reihen benutzen. (Solange **USER** angezeigt wird, sind den oben erwähnten User-Tasten andere Funktionen zugewiesen. Sie können daher die ursprüngliche Funktionen nicht ausführen.)

Datenspeicher. Der zu transformierende Vektor bzw. Punkt wird in R_{04} , R_{05} , R_{06} , dem Vektor-Speicherregister 1 (anfänglich vom X-, Y- und Z-Register), abgespeichert. Der Rotationsvektor wird in R_{07} , R_{08} , R_{09} , dem Vektor-Speicherregister 2, abgespeichert. Der Ursprung des neuen Systems wird in R_{10} , R_{11} , R_{12} abgespeichert, das das Vektor-Speicherregister 3 ist. Der Rotationswinkel wird in R_{16} abgespeichert, und R_{13} , R_{14} und R_{15} werden als Zwischenspeicher verwendet.

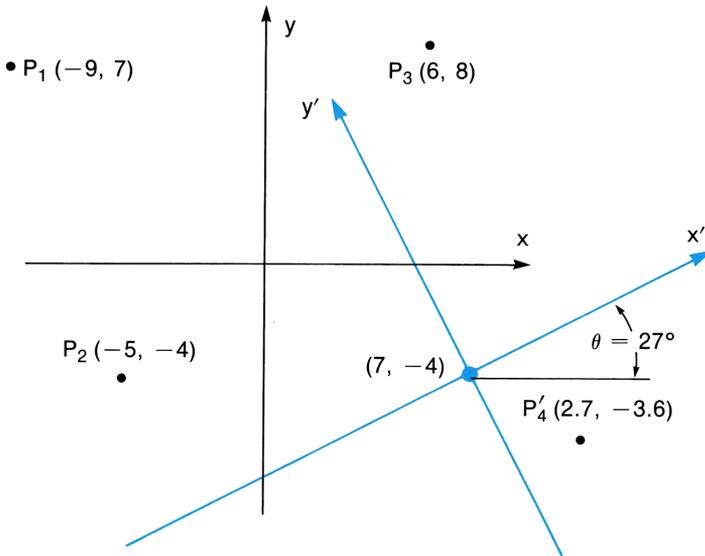
Wenn Sie Operationen für Vektoren (**[VS]**, **[VR]** und den Vektor-Rechner) sowie TR benutzen, denken Sie daran, daß TR R_0 – R_{16} benötigt, wenn es initialisiert wird (**[XEQ]** **[TR]**). Das bedeutet, daß Sie Vektoren nicht in den Vektor-Registern 1-5 abspeichern sollen (falls Sie für Ihre Vektorrechnungen TR verwenden).

Flags. Flag 01 wird benutzt, um anzuzeigen, ob die Transformation ins neue oder ins alte System erfolgen soll. Flag 1 gesetzt, bedeutet, daß die Transformation ins neue System erfolgt.

Flag 05 wird gesetzt, wenn das neue System rotiert wurde.

Beispiele

Gegeben seien die beiden untenstehenden Koordinatensysteme (x, y) und (x', y') :



Die Punkte P_1 , P_2 und P_3 sollen in die äquivalenten Koordinaten des (x', y') -Systems überführt werden. Der Punkt P_4' ist in sein Äquivalent im (x, y) -System umzuwandeln.

Tastensequenz

FIX 4

XEQ **SIZE** 017

XEQ **TR**

0 **ENTER**

4 **CHS** **ENTER**

7 **R/S**

Anzeige

Z0, Y0, X0 ?

0.0000

-4.0000

ROT Δ ?

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $SIZE \geq 017$ ist.

Erfragt z_0 , y_0 und x_0 des neuen Systems.

Eingabe von Null für z_0 .

Erfragt den Rotationswinkel.

Tastensequenz

 27 **R/S**
R/S

 0 **ENTER** 7 **ENTER**

 9 **CHS** **A** (**↑N**)

R/S
R/S
R/S

 0 **ENTER** 4 **CHS**
ENTER

 5 **CHS** **A** (**↑N**)

R/S
R/S
R/S

 0 **ENTER** 8 **ENTER**

 6 **A** (**↑N**)

R/S
R/S
R/S

 0 **ENTER** 3.6 **CHS**
ENTER

 2.7 **B** (**↑O**)

R/S
R/S
Anzeige
c,b,a ?
↑N ↑O NEW
7.0000
X = -9.2622
Y = 17.0649
Z = 0.0000
↑N ↑O NEW
-4.0000
X = -10.6921
Y = 5.4479
Z = 0.0000
↑N ↑O NEW
8.0000
X = 4.5569
Y = 11.1461
Z = 0.0000
↑N ↑O NEW
-3.6000
X = 11.0401
Y = -5.9818
Z = 0.0000

Erfragt die Koordinaten des Rotationsvektors. Überspringen Sie diesen Schritt für ein 2-dimensionales System. Erfragt P_1 .

 x_1'
 y_1'
 z_1'

Zur Eingabe von P_2 bereit. Dieser Schritt ist optional—er bringt Sie ins Hauptprogramm zurück.

 x_2' von P_2 .

 y_2'
 z_2'

Zeigt erneut das Menü an.

 x_3' von P_3 .

 y_3'
 z_3'

Zeigt erneut das Menü an.

 x_4' von P_4' .

 y_4
 z_4

Ein 3-dimensionales Koordinatensystem wird in den Punkt (2.45, 4.00, 4.25) verschoben. Danach wird eine Drehung von 62.5° um die Achse (0, -1, -1) ausgeführt. Im ursprünglichen System hat ein bestimmter Punkt die Koordinaten (3.9, 2.1, 7.0). Was sind die Koordinaten dieses Punktes im neuen System?

Tastensequenz

[J]
 [E] (NEW)
 4.25 [ENTER↑] 4 [ENTER↑]
 2.45 [R/S]
 62.5 [R/S]
 1 [CHS] [ENTER↑]
 1 [CHS] [ENTER↑]
 0 [R/S]
 7 [ENTER↑] 2.1 [ENTER↑]
 3.9 [A] (↑N)
 [R/S]
 [R/S]

Anzeige

↑N ↑O NEW
 Z0,Y0,X0 ?
 4.0000
 ROT Δ ?
 c,b,a ?
 -1.0000
 -1.0000
 ↑N ↑O NEW
 2.1000
 X=3.5861
 Y=0.2609
 Z=0.5891

Zeigt Menü erneut an (falls **USER**-Indikator an ist).
 Erfragt die Koordinaten des neuen Systems.

Zur Eingabe der Koordinaten von *P* bereit.

x'
y'
z'

In dem neuen (verschobenen und gedrehten) Achsensystem hat der Punkt die Koordinaten (1, 1, 1). Was waren seine Koordinaten im ursprünglichen System?

Tastensequenz

[R/S]
 1 [ENTER↑] 1 [ENTER↑]
 1 [B] (↑O)
 [R/S]
 [R/S]

Anzeige

↑N ↑O NEW
 1.0000
 X=2.9117
 Y=4.3728
 Z=5.8772

Zeigt Hauptmenü erneut an. Optionaler Schritt.

x
y
z

Programmier-Informationen

Sie können das Unterprogramm CT in Ihren eigenen Programmen verwenden. Es führt eine Koordinatentransformation (Verschiebung und Rotation) in drei Dimensionen aus. Die *x*-, *y*- und *z*-Koordinaten werden dem Stack (X-, Y- und Z-Register) entnommen und in ein neues System bzw. von einem neuen ins ursprüngliche System transformiert.

Speicherminimum: SIZE 017.

Verwendete Flags: 01, 05.

Unterprogramm: CT

alter Registerinhalt	neuer Registerinhalt	Zu initialisierende Flags
X-Register = x-Koordinate	X-Register = transformierte x-Koordinate	SF 01, um ins neue System zu transformieren
Y-Register = y-Koordinate	Y-Register = transformierte y-Koordinate	CF 01, um ins ursprüngliche System zu transformieren
Z-Register = z-Koordinate	Z-Register = transformierte z-Koordinate	SF 05, um die Koordinaten zu drehen
	$R_{00} = (1 - \cos\theta)(N \cdot P)$	CF 05, um die Koordinaten nicht zu drehen
	$R_{01} = X\text{-Registerinhalt}$	
	$R_{02} = Y\text{-Registerinhalt}$	
	$R_{03} = Z\text{-Registerinhalt}$	
	$R_{04} = P_x$ (oder $P_x - T_x$ falls Flag 01 gesetzt ist)	
	$R_{05} = P_y$ (oder $P_y - T_y$ falls Flag 01 gesetzt)	
	$R_{06} = P_z$ (oder $P_z - T_z$ falls Flag 01 gesetzt)	
$R_{07} = a$ (N_x , der Einheits-Rotationsvektor)	$R_{07} = a$ (N_x , der Einheits-Rotationsvektor)	
$R_{08} = b$ (N_y)	$R_{08} = b$ (N_y)	
$R_{09} = c$ (N_z)	$R_{09} = c$ (N_z)	
$R_{10} = T_x$, der Verschiebungsvektor	$R_{10} = T_x$, der Verschiebungsvektor	
$R_{11} = T_y$	$R_{11} = T_y$	
$R_{12} = T_z$	$R_{12} = T_z$	
$R_{16} = \text{Rotationswinkel}$	$R_{16} = \text{Rotationswinkel}$	

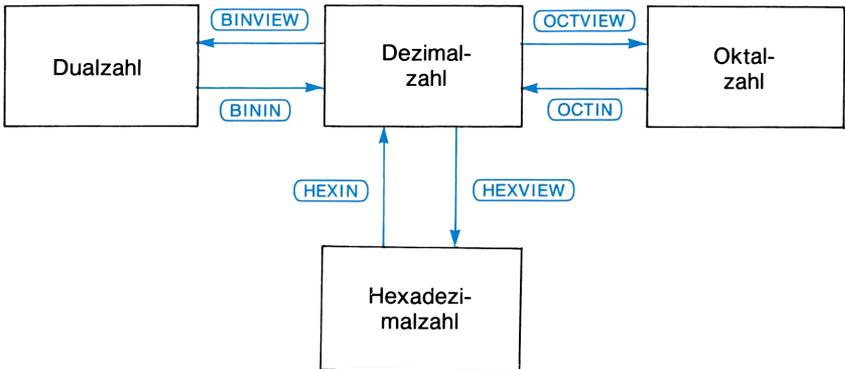
Bemerkungen. Um CT auszuführen, geben Sie die Koordinaten des Verschiebungsvektors (T), des Einheits-Rotationsvektors (N) sowie den Rotationswinkel ein. Setzen Sie dann Flag 01, um ins neue System bzw. löschen Sie Flag 01, um ins alte System zu transformieren. Setzen Sie zuletzt Flag 05, um den Vektor P zu drehen. Das Ergebnis befindet sich im X-, Y- und Z-Register sowie in R_{01} , R_{02} und R_{03} .

UMWANDLUNG ZWISCHEN ZAHLENSYSTEMEN SOWIE BOOLESCHE LOGIK

In diesem Modul sind mehrere Funktionen zum Berechnen und Manipulieren von Dual-, Oktal- und Hexadezimalzahlen enthalten. Es gibt sechs Funktionen zur Basisumwandlung von Zahlen, vier Boolesche Funktionen und zwei Funktionen zur Bit-Manipulation. Alle Funktionen verwenden eine Wortlänge von 32 Bits.

Funktion zur Basisumwandlung

Sechs Funktionen stehen für das Umwandeln von Zahlen zwischen dem Dezimalsystem und dem Dual-, Oktal- oder Hexadezimalsystem zur Verfügung. Das unten aufgeführte Bild zeigt die Ausführungen der sechs Funktionen.



Zulässiger Bereich für Daten

- Die binäre Eingabe für **BININ** muß aus den Ziffern Null und Eins bestehen und darf maximal 10 Ziffern betragen.
- Die dezimale Eingabe für **BINVIEW** muß eine Integerzahl zwischen 0 und 1023 sein. Bei Eingabe anderer Zahlen wird der Integerteil verwendet. Ebenso wird nur der Absolutbetrag verwendet.

130 Umwandlung zwischen Zahlensystemen sowie Boolesche Logik

- Die oktale Eingabe für **OCTIN** muß aus Ziffern 0 bis 7 bestehen und darf maximal 10 Ziffern betragen.
- Die dezimale Eingabe für **OCTVIEW** muß eine Integerzahl zwischen 0 und 1 073 741 823 sein. Bei Eingabe anderer Zahlen wird der Integerteil verwendet. Ebenso wird nur der Absolutbetrag benutzt.
- Die hexadezimale Eingabe für **HEXIN** muß aus Ziffern von 0 bis 9 und "Buchstaben" von A bis F bestehen und darf maximal 8 Zeichen betragen.
- Die dezimale Eingabe für **HEXVIEW** muß eine Integerzahl zwischen 0 und 4 294 967 295 sein. Bei Eingabe anderer Zahlen wird der Integerteil verwendet. Ebenso wird nur der Absolutbetrag benutzt.

Anweisungen

- Die "VIEW"-Funktionen (BINVIEW, OCTVIEW, HEXVIEW) wandeln die in der Anzeige (X-Register) stehende Dezimalzahl um. (Die Zahl wird in dezimaler Form im Stack abgelegt.) Drücken Sie **↵**, um den Inhalt des X-Registers erneut anzuzeigen.
- Das momentane, mit **FIX** eingestellte Anzeigeformat entscheidet über die Anzahl der durch Kommas getrennten angezeigten Ziffern einer nichtdezimalen Zahl.
- Die "IN"-Funktionen (BININ, OCTIN, HEXIN) sind "Voraus"-Funktionen: *Zuerst* führen Sie die Funktion aus, und *dann* geben Sie den entsprechenden Wert ein. Drücken Sie **ENTER**, um das Ergebnis anzuzeigen.
- Um die Ausführung einer "IN"-Funktion abzubrechen, drücken Sie **ALPHA** **ALPHA**.
- Wenn eine "IN"-Funktion in einem Programm ausgeführt wird, hält das Programm anschließend an.

Funktionen zur Basisumwandlung

Funktion	Wirkung
BININ (binär in dezimal)	<p>Wandelt eine binäre Eingabe in den entsprechenden dezimalen Wert im X-Register um.</p> <ol style="list-style-type: none"> 1. Führen Sie BININ aus. Es wird _ B angezeigt. 2. Geben Sie die Dualzahl ein. 3. Drücken Sie für das Ergebnis ENTER↑.
BINVIEW (dezimal in binär)	<p>Zeigt vorübergehend den äquivalenten binären Wert der Dezimalzahl im X-Register an.</p> <ol style="list-style-type: none"> 1. Geben Sie die umzuwandelnde Dezimalzahl ein. 2. Führen Sie BINVIEW aus. 3. Zeigt <i>Ergebnis B</i> an. 4. Drücken Sie ←, um den Inhalt des X-Registers erneut anzuzeigen.
OCTIN (oktal in dezimal)	<p>Wandelt eine oktale Eingabe in den entsprechenden dezimalen Wert im X-Register um.</p> <ol style="list-style-type: none"> 1. Führen Sie OCTIN aus. Es wird _ O angezeigt. 2. Geben Sie die Oktalzahl ein. 3. Drücken Sie für das Ergebnis ENTER↑.
OCTVIEW (dezimal in oktal)	<p>Zeigt vorübergehend den äquivalenten oktalen Wert der Dezimalzahl im X-Register an.</p> <ol style="list-style-type: none"> 1. Geben Sie die umzuwandelnde Oktalzahl ein. 2. Führen Sie OCTVIEW aus. 3. Zeigt <i>Ergebnis O</i> an. 4. Drücken Sie ←, um den Inhalt des X-Registers erneut anzuzeigen.
HEXIN (hexadezimal in dezimal)	<p>Wandelt eine hexadezimale Eingabe in den entsprechenden dezimalen Wert im X-Register um.</p> <ol style="list-style-type: none"> 1. Führen Sie HEXIN aus. Es wird _ H angezeigt. 2. Geben Sie die Hexadezimalzahl ein. 3. Drücken Sie für das Ergebnis ENTER↑.
HEXVIEW (dezimal in hexadezimal)	<p>Zeigt vorübergehend den äquivalenten hexadezimalen Wert der Dezimalzahl im X-Register an.</p> <ol style="list-style-type: none"> 1. Geben Sie die umzuwandelnde Hexadezimalzahl ein. 2. Führen Sie HEXVIEW aus. 3. Zeigt <i>Ergebnis H</i> an. 4. Drücken Sie ←, um den Inhalt des X-Registers erneut anzuzeigen.

Boolesche Funktionen

Diese Gruppe von Funktionen umschließt Boolesche Logik sowie Bit-Prüfung und Bit-Rotation.

Zulässiger Eingabebereich für Daten

Der zulässige Eingabebereich dieser Funktionen ist eine Integerzahl zwischen 0 und 4 294 967 295 (32-Bit, Integer ohne Vorzeichen). Bei Eingabe anderer Zahlen wird der Integerteil verwendet. Bei negativen Zahlen wird der Absolutbetrag verwendet.

Anweisungen

Das Ergebnis einer booleschen Operation wird im X-Register angezeigt. Der ursprüngliche Inhalt des X-Registers wird im LAST X-Register abgelegt (*außer* für `BIT?`, wodurch das LAST X-Register und der Stack nicht verändert werden). Alle anderen Funktionen mit zwei Parametern verschieben den Stack nach "unten".

Boolesche Funktionen

Funktion	Wirkung
<code>AND</code>	Berechnet das logische AND von x und y .
<code>OR</code>	Berechnet das logische inklusive OR von x und y .
<code>XOR</code>	Berechnet das logische exklusive OR von x und y .
<code>NOT</code>	Bestimmt das Einer-Komplement zu $ x $.
<code>BIT?</code> (<i>Teste Bit</i>)	Überprüft die Bitdarstellung einer im Y-Register spezifizierten Dezimalzahl. Die zu prüfende Bitposition wird im X-Register angegeben. Wenn das Bit 1 ist, zeigt der Rechner YES an; wenn das Bit 0 ist, zeigt er NO an. In einem Programm ist <code>BIT?</code> eine bedingte Funktion entsprechend der "do if true"-Regel: ein "1"-Bit bedingt das Ausführen des nächsten Programmschrittes, während bei einem "0"-Bit der nächste Schritt übersprungen wird.
<code>ROTX</code> (<i>Rotiere Y um X</i>)	Rotiert die binäre Darstellung des Dezimalwerts im Y-Register um die im X-Register spezifizierte Anzahl von Bits nach rechts und gibt das Rotationsergebnis als Dezimalwert im X-Register an. Das Rotieren nach rechts um $(32-x)$ Bits ist gleichbedeutend mit der Rotation nach links um x Bits.

Beispiele

Was ist der binäre, oktal und hexadezimale Wert von 65_{10} ? Stellen Sie **FIX** 4 ein, damit durch Kommas immer vier Ziffern getrennt werden.

Tastensequenz

FIX 4

65

XEQ **BINVIEW**

XEQ **OCTVIEW**

XEQ **HEXVIEW**

Anzeige

65_

100,0001 B

101 O

41 H

Stellt das hier benutzte Anzeigeformat ein.

Binär.

Oktal.

Hexadezimal.

Was ist das oktale Ergebnis der Rotation von $FA407_{16}$ um sechs Bits nach rechts, Addition von 100100_2 zum Zwischenergebnis und anschließender logischer AND-Verknüpfung mit 25_{10} ?

Tastensequenz

XEQ **HEXIN**

FA407

ENTER↑

6

XEQ **ROTX**

XEQ **BININ**

100100

+

25

XEQ **AND**

XEQ **OCTVIEW**

Anzeige

_ H

FA407_ H

1,025,031.000

6_

469,778,064.0

_ B

10,0100_ B

469,778,100.0

25_

16.0000

20 O

Dezimales Äquivalent von $FA407_{16}$.

Rotiert Wert um sechs Bits nach rechts.

Addiert duale Eingabe zum vorigen Ergebnis.

Verknüpft voriges Ergebnis und 25 mit logisch AND.

Oktales Ergebnis.

KURVENANPASSUNG

Das Programm CFIT liest statistische Daten (x_i, y_i) ein und paßt ihnen eine Kurve an, die aus den folgenden vier Kurven vorgegeben wird (oder diejenige, die am besten paßt). Die Kurve mit dem größten Bestimmtheitsmaß r^2 ist die am besten den Daten anpassende Kurve.

- Gerade (lineare Regression), $y = a + bx$
- Exponentialkurve, $y = ae^{bx}$ (wobei $a > 0$ ist)
- Logarithmische Kurve, $y = a + b(\ln x)$
- Potenzkurve, $y = ax^b$ (wobei $a > 0$ ist)

Das Programm berechnet a , b , r^2 und \hat{y} , ein linearer Schätzwert (ein Näherungswert für y).

Gleichungen

Die Regressionskoeffizienten a und b werden durch Lösen der folgenden linearen Gleichungen bestimmt, wobei n die Gesamtzahl von Datenpaaren ist.

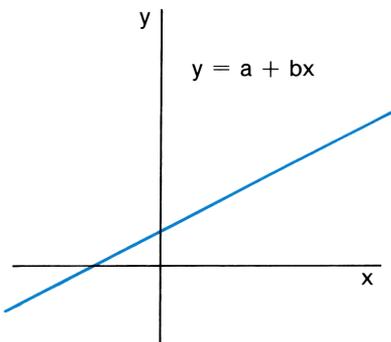
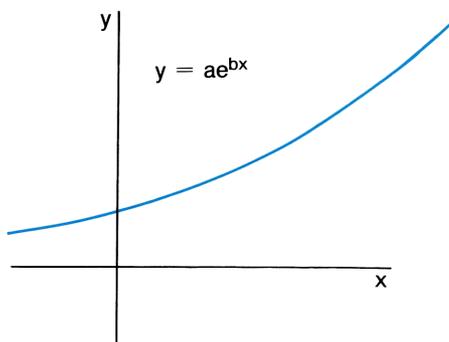
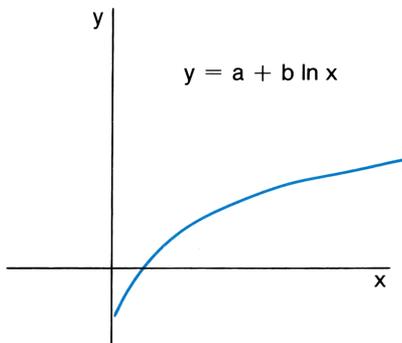
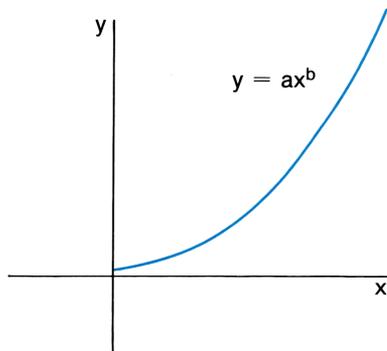
$$\begin{aligned}An + b\Sigma X_i &= \Sigma Y_i \\A\Sigma X_i + b\Sigma(X_i)^2 &= \Sigma(Y_i X_i)\end{aligned}$$

Definition der Regressionsvariablen

Regression	A	X_i	Y_i
Linear	a	x_i	y_i
Exponentiell	$\ln a$	x_i	$\ln y_i$
Logarithmisch	a	$\ln x_i$	y_i
Potenz	$\ln a$	$\ln x_i$	$\ln y_i$

Das Bestimmtheitsmaß lautet

$$r^2 = \frac{A\Sigma Y_i + b\Sigma(X_i Y_i) - \frac{1}{n} (\Sigma Y_i)^2}{\Sigma(Y_i)^2 - \frac{1}{n} (\Sigma Y_i)^2}$$

Lineare Regression**Exponentiell****Logarithmisch****Potenz****Zulässiger Eingabebereich für Daten**

Das Programm CFIT berechnet die Kurve nach der Methode der kleinsten Quadrate, indem es entweder die ursprüngliche Gleichung (Gerade und logarithmische Kurve) oder die transformierte Gleichungen (Exponential- und Potenzkurve) benutzt.

Alle Datenwerte (x_i, y_i) müssen positiv und ungleich Null sein. Andernfalls wird **DATA ERROR** angezeigt.

Mit kleiner werdendem Unterschied zwischen den x - und y -Werten nimmt die Genauigkeit der Regressionskoeffizienten ab.

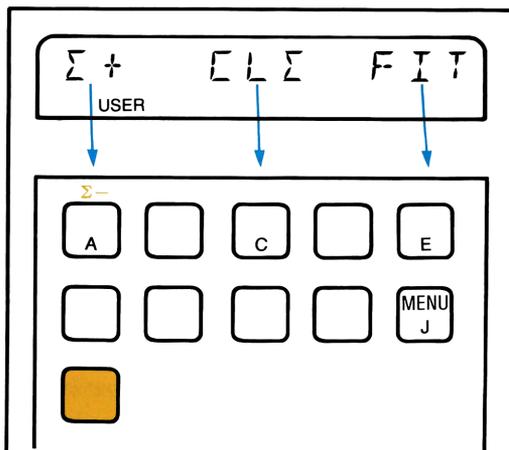
Beachten Sie ebenfalls, daß ungenaue Ergebnisse erzielt werden können, wenn eine Variable sehr viel größer als die andere ist oder sich viel schneller verändert. (Dies ist der Fall, wenn der Rechner auf mehr als zehn Stellen genau rechnen soll, was er nicht kann.) In diesem Fall sollten Sie Datenwerte entsprechend normieren, um ein genaues Ergebnis zu erhalten. Normierungsmethoden werden in vielen Statistikbüchern beschrieben.

DATA ERROR wird angezeigt, wenn Sie versuchen, eine Kurve einem einzelnen Punkt anzupassen oder wenn Ihre Daten negative Werte oder Nullwerte enthalten.

Anweisungen

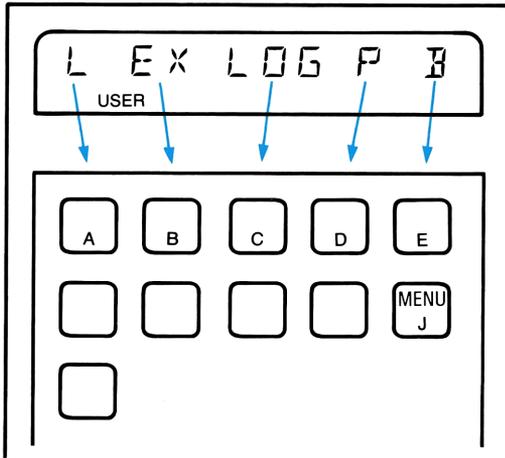
- Das Programm CFIT startet mit seinem Hauptmenü $\Sigma+$ CL Σ FIT. Es ermöglicht Ihnen die Eingabe Ihrer statistischen Daten: $\Sigma+$ um einzugeben (zuerst y , dann x), $\Sigma+$ um zu löschen und CL Σ um alte statistische Daten zu löschen. Durch Drücken von FIT wird das Kurvenmenü angezeigt.
- Das Kurvenmenü L EX LOG P B ermöglicht Ihnen die Wahl der Kurve, die Sie Ihren Daten anpassen wollen: Linear, EXponentiell, LOGarithmisch, Potenz und Bestanpassung. Bestanpassung wählt eine für die Daten am besten passende Kurve aus.
- Nachdem Sie die Wahl der Kurve getroffen haben, werden durch Drücken von $\boxed{R/S}$ nacheinander die Regressionsvariablen angezeigt. Durch Drücken von \boxed{J} können Sie jederzeit das Hauptmenü in die Anzeige zurückholen.

Hauptmenü



Dieses Menü zeigt die Neubelegung der User-Tasten mit Funktionen aus CFIT an. Sie können sich dieses Menü jederzeit durch Drücken der Taste **J** in der Anzeige ansehen. Der Programmablauf wird dadurch in keiner Weise gestört.

Kurvenmenü



Durch Drücken der Taste **←** können Sie das Menü jederzeit löschen. Dann erscheint in der Anzeige der Inhalt des X-Registers. Das Programm wird dadurch nicht beendet. Sie können nun Berechnungen durchführen und dann das Menü durch Drücken von **J** wieder in die Anzeige holen. (Sie müssen die Anzeige des Programms *nicht* löschen, bevor Sie Berechnungen ausführen.)

Anweisungstabelle für CFIT

		Umfang: 018
Anweisungen	Eingabe	Anzeige
1. Starten Sie das Programm CFIT. Die Anzeige des Menüs gibt die neue Belegung der beiden oberen Reihen des User-Tastenfelds mit entsprechenden Funktionen an.	XEQ CFIT *	$\Sigma+$ CL Σ FIT
2. Löschen Sie alte statistische Daten. (Dies ist nicht notwendig, wenn Sie gerade CFIT ausgeführt haben, wodurch automatisch alte Daten gelöscht werden.)	C (CL Σ)	$\Sigma+$ CL Σ FIT

Anweisungstabelle für CFIT (Fortsetzung)

Anweisungen	Eingabe	Anzeige
3. Geben Sie ihr Datenpaar ein. Wiederholen Sie dies für jedes Paar.	y <input type="text" value="ENTER"/> x <input type="text" value="A"/> (Σ+) [†]	y Σ+ CLΣ FIT
4. Löschen Sie die Anzeige, um zu sehen, wieviele Datenpaare Sie bereits eingegeben haben (optional).	<input type="text" value="←"/> <input type="text" value="J"/>	n Σ+ CLΣ FIT
5. Um ein Datenpaar zu korrigieren, geben Sie es zuerst erneut ein und löschen Sie es an- schließend. Geben Sie dann das korrekte Paar ein (Schritt 3).	y _k <input type="text" value="ENTER"/> x _k <input type="text" value="A"/> (<input type="text" value="Σ-"/>)	y _k Σ+ CLΣ FIT
6. Holen Sie das Kurvenmenü in die Anzeige.	<input type="text" value="E"/> (FIT)	L EX LOG P B
7. Wählen Sie gewünschte anzupassende Kurve aus.	<input type="text" value="A"/> (L) <input type="text" value="B"/> (EX) <input type="text" value="C"/> (LOG) <input type="text" value="D"/> (P) <input type="text" value="E"/> (B)	LIN EXP LOG POW (die "Bestanpas- sung" aus den oben aufgeführten Kurven)
8. Bestimmen Sie die Werte für a, b, und r ² .	<input type="text" value="R/S"/> ‡ <input type="text" value="R/S"/> ‡ <input type="text" value="R/S"/> ‡	a= Ergebnis b= Ergebnis R ² = Ergebnis
9. Bestimmen Sie den linearen Schätzwert \hat{y} . Wiederholen Sie, falls gewünscht.	<input type="text" value="R/S"/> ‡ x <input type="text" value="R/S"/> <input type="text" value="R/S"/> ‡	X=? Y=Ergebnis X=?
10. Um erneut zu starten; zeigen Sie das Hauptmenü an):	<input type="text" value="J"/>	Σ+ CLΣ FIT

* Zur Ausführung eines Programms drücken Sie entweder Alpha-Name
 oder eine entsprechende benutzerspezifische Taste.

† Wenn Sie einen Drucker angeschlossen haben, können Sie mit diesem Schritt einen Aus-
druck der gerade eingegebenen Werte erhalten. Schlagen Sie für weitere Informationen im
Handbuch Ihres Druckers nach.

‡ Diese Tastenfolge ist nicht notwendig, wenn Sie einen Drucker angeschlossen haben, da er
automatisch alle Ergebnisse druckt und dann das Menü angezeigt wird.

Bemerkungen

Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten -, und zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung überschrieben. Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten. Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Hinweis: Das Programm CFIT verändert die Position der Statistik-Register. Wenn Sie *nach Ausführung dieses Programms* auf Statistik-Register zugreifen wollen, müssen Sie mit der Funktion **ΣREG** den Statistik-Registern neue bekannte Positionen zuweisen (schlagen Sie dazu im Benutzerhandbuch des HP-41 nach). Dies gilt ebenso, wenn die Position der Statistik-Register der Voreinstellung (R_{11} – R_{16}) entsprechen soll. Für das Zugreifen auf statistische Daten, die *durch* dieses Programm abgespeichert wurden, schlagen Sie im Abschnitt "Programmier-Informationen" am Ende dieses Kapitels nach.

Beispiele

Passen Sie den folgenden Daten eine Gerade an, und berechnen Sie \hat{y} für $x = 37$ und $x = 35$.

x	40.5	38.6	37.9	36.2	35.1	34.6
y	104.5	102	100	97.5	95.5	94

Tastensequenz

FIX 4

XEQ **SIZE** 018

XEQ **CFIT**

104.5 **ENTER** 40.5
A ($\Sigma+$)

102 **ENTER** 38.6
A ($\Sigma+$)

100 **ENTER** 37.9
A ($\Sigma+$)

Anzeige

$\Sigma+$ **CL** Σ **FIT**

40.5_ $\Sigma+$ **CL** Σ **FIT**

38.6_ $\Sigma+$ **CL** Σ **FIT**

37.9_ $\Sigma+$ **CL** Σ **FIT**

Stellt das hier benutzte Anzeigeformat ein.

Optional—gibt die Anzahl der vom Programm benötigten Speicherregister an. Dies ist nicht notwendig, wenn Ihre Einstellung bereits $SIZE \geq 018$ ist.

Durch Starten des Programms werden die alten statistischen Daten gelöscht.

Eingabe des ersten Datenpaares; zuerst y .

Zweites Paar.

Usw.

Tastensequenz

97.5 **ENTER** 36.2

A ($\Sigma+$)

95.5 **ENTER** 35.2

A ($\Sigma+$)

95.5 **ENTER** 35.2

A ($\Sigma-$)

95.5 **ENTER** 35.1

A ($\Sigma+$)

94 **ENTER** 34.6

A ($\Sigma+$)

E (FIT)

A (L)

R/S

R/S

R/S

R/S

37 **R/S**

R/S

35 **R/S**

R/S

R/S

R/S

Anzeige

36.2_

$\Sigma+$ CL Σ FIT

35.2_

$\Sigma+$ CL Σ FIT

35.2_

$\Sigma+$ CL Σ FIT

35.1_

$\Sigma+$ CL Σ FIT

34.6_

$\Sigma+$ CL Σ FIT

L EX LOG P B

LIN

a=33.5271

b= 1.7601

R²=0.9909

X=?

Y=98.6526

X=?

Y=95.1323

X=?

L EX LOG P B

$\Sigma+$ CL Σ FIT

Falsche Eingabe für x!

Löschen Sie das fehlerhafte Paar.

Geben Sie korrektes Paar ein.

Das Kurvenmenü. Auswählen der linearen Kurve.

Erfragt x-Wert für den zu bestimmen den Schätzwert für y. \hat{y} .

Zeigt das Kurvenmenü an. Zur Anpassung einer anderen Kurve bereit.

Zeigt Hauptmenü an. Zum Lösen einer anderen Aufgabe bereit.

Geben Sie die folgenden Daten ein und passen Sie ihnen die am besten passende Kurve an. Berechnen Sie dann \hat{y} für $x = 1.5$ und $x = 2$.

x	0.72	1.31	1.95	2.58	3.14
y	2.16	1.61	1.16	0.85	0.5

Tastensequenz

R/S

C (CL Σ)

Anzeige

$\Sigma+$ CL Σ FIT

$\Sigma+$ CL Σ FIT

Vergewissern Sie sich, daß das Hauptmenü angezeigt wird.

Daten aus dem 1. Beispiel werden gelöscht.

Tastensequenz

2.16 [ENTER↑] .72
 [A] (Σ +)
 1.61 [ENTER↑] 1.31
 [A] (Σ +)
 1.16 [ENTER↑] 1.95
 [A] (Σ +)
 .85 [ENTER↑] 2.58
 [A] (Σ +)
 .5 [ENTER↑] 3.14
 [A] (Σ +)
 [E] (FIT)
 [E] (B)

[R/S]

[R/S]

[R/S]

[R/S]

1.5 [R/S]

[R/S]

2 [R/S]

[R/S]

[R/S]

Anzeige

.72_
 Σ + CL Σ FIT
 1.31_
 Σ + CL Σ FIT
 1.95_
 Σ + CL Σ FIT
 2.58_
 Σ + CL Σ FIT
 3.14_
 Σ + CL Σ FIT
 L EX LOG P B
 LOG

a= 1.8515
 b= -1.1021
 R \uparrow 2=0.9893
 X=?
 Y=1.4046
 X=?
 Y=1.0875
 X=?
 L EX LOG P B

Eingabe des ersten
 Datenpaares.

Am besten paßt eine
 logarithmische Kurve.

Programmier-Informationen

Sie können die Unterprogramme A Σ , D Σ , FIT und BFIT in Ihren eigenen Programmen verwenden.

- A Σ addiert das im X- und Y-Register abgelegte Datenpaar zu einem Block von Statistik-Registern für statistische Auswertungen.
- D Σ löscht das im X- und Y-Register abgelegte Datenpaar aus dem Block von Statistik-Registern.
- FIT paßt eine Kurve des Typs 1-4 den statistischen Daten an, die durch das Programm CFIT oder durch die Unterprogramme A Σ und D Σ abgespeichert wurden.
- BFIT bestimmt diejenige Kurve aus den Kurven 1-4, die zu den durch das Programm CFIT oder durch die Unterprogramme A Σ und D Σ abgespeicherten statistischen Daten am besten paßt.

Speicherminimum: SIZE 018

Benutzte Flags: BFIT und FIT benutzen 01, 02, 03, 04, 06, 07.
 A Σ und D Σ benutzen keine Flags.

Unterprogramme: AΣ und DΣ

alter Registerinhalt	neuer Registerinhalt	Zu initialisierende Flags
Y-Register: y-Wert X-Register: x-Wert		
$R_{04} = 0$	$R_{04} = \Sigma(y \ln x)$	
$R_{05} = 0$	$R_{05} = \Sigma(x \ln y)$	
$R_{06} = 0$	$R_{06} = \Sigma y$	
$R_{07} = 0$	$R_{07} = \Sigma y^2$	
$R_{08} = 0$	$R_{08} = \Sigma x$	
$R_{09} = 0$	$R_{09} = \Sigma x^2$	
$R_{10} = 0$	$R_{10} = \Sigma(xy)$	
$R_{11} = 0$	$R_{11} = n$	
$R_{12} = 0$	$R_{12} = \Sigma(\ln y)$	
$R_{13} = 0$	$R_{13} = \Sigma(\ln y)^2$	
$R_{14} = 0$	$R_{14} = \Sigma(\ln x)$	
$R_{15} = 0$	$R_{15} = \Sigma(\ln x)^2$	
$R_{16} = 0$	$R_{16} = \Sigma(\ln x)(\ln y)$	
$R_{17} = 0$	$R_{17} = n$, und vorübergehend Σy	

Unterprogramm: FIT

alter Registerinhalt	neuer Registerinhalt	Zu initialisierende Flags
X-Register = 1 = linear 2 = exponentiell 3 = logarithmisch 4 = Potenzfunktion		CF 01
	$R_{00} = 1, 2, 3 \text{ oder } 4$	CF 02
	$R_{01} = a$	CF 03
	$R_{02} = b$	CF 04
	$R_{03} = r^2$	
R_{04} – R_{17} : alle Statistik-Register sind mit den obigen in AΣ identisch.		

Anmerkungen. Nachdem Sie unter Benutzung von A Σ und D Σ die Daten in die Statistik-Register eingegeben haben, tippen Sie in die Anzeige (X-Register) die Nummer der anzupassenden Kurve (1, 2, 3, 4). Führen Sie dann FIT aus. FIT setzt Flag 07 und setzt ein der Nummer des Kurventyps entsprechendes Flag (01–04). Es speichert a , b und r^2 in R₀₁, R₀₂ und R₀₃ ab.

Unterprogramm: BFIT

alter Registerinhalt	neuer Registerinhalt	Zu initialisierende Flags
	R ₀₀ = 1, 2, 3 oder 4	CF 01
	R ₀₁ = a	CF 02
	R ₀₂ = b	CF 03
	R ₀₃ = r^2	CF 04
R ₀₄ –R ₁₇ : alle Statistik-Register sind mit den obigen in A Σ und FIT identisch.		

Anmerkungen. Nachdem Sie die Daten unter Benutzung von A Σ und D Σ in die Statistik-Register eingegeben haben, führen Sie BFIT aus. Es bestimmt die den Daten am besten anzupassende Kurve aus den Typen: lineare, exponentielle und logarithmische Kurve sowie Potenzkurve. BFIT setzt Flag 01 (lineare Kurve), 02 (exponentielle Kurve), 03 (logarithmische Kurve) oder 04 (Potenzkurve), speichert die entsprechende Nummer des Kurventyps in R₀₀ ab und legt a , b und r^2 in R₀₁, R₀₂ und R₀₃ ab.

FINANZTECHNISCHE BERECHNUNGEN

Das Programm TVM löst verschiedene finanztechnische Aufgaben, die sich mit Zeit, Geld und Zins beschäftigen—die Zinseszins-Funktionen. Folgende Variablen können Eingaben oder Ergebnisse sein.

- N* Die Anzahl der Verzinsungsperioden bzw. Annuitäten. (Für ein Darlehen mit dreißigjähriger Laufzeit und monatlicher Zahlung ist $N = 12 \times 30 = 360$.)
- I* Der Periodenzinssatz in Prozent. (Wenn die Verzinsung nicht jährlich erfolgt, stellt dies den jährlichen Prozentsatz dividiert durch die Anzahl der Verzinsungsperioden pro Jahr dar. Beispielsweise ergibt eine jährliche Verzinsung mit 9% eine nominale monatliche Verzinsung mit $9 \div 12 = 0.75\%$.)
- PV* Der Barwert ist die ursprüngliche Zahlung oder der Gegenwartswert einer Reihe zukünftiger Zahlungen.
- PMT* Die wiederkehrende Zahlung bzw. Annuität.
- FV* Die letzte Zahlung oder der berechnete Endwert einer Reihe von früheren Zahlungen.

Die Zahlungen können zum *Anfang* (*vorschüssig* bzw. *Beginn-Modus*) oder zum *Ende* (*nachschüssig* bzw. *Ende-Modus*) der Verzinsungsperiode festgelegt werden. Im Beginn-Modus ist Flag 00 gesetzt. Zahlungen am Ende der Verzinsungsperioden kommen häufig bei Hypotheken und direkt abzuzahlenden Darlehen vor. Vorschüssige Zahlungen treten häufig bei Leasing auf.

Gleichung

$$0 = PV + (1 + ip) PMT \left[\frac{1 - (1 + i)^{-N}}{i} \right] + FV (1 + i)^{-N}$$

mit i = der periodische Zinssatz als *Bruchteil* ($i = I/100$),
 $p = 1$ in Beginn-Modus bzw. 0 in Ende-Modus.

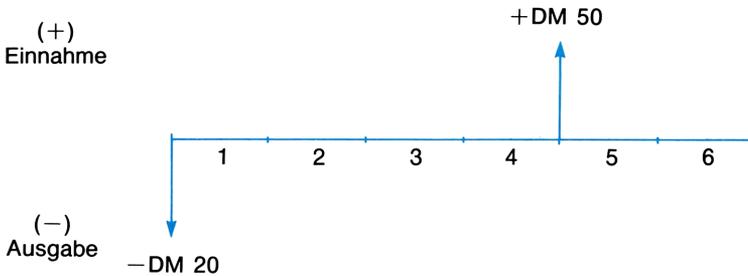
Zulässiger Eingabebereich für Daten

Verwenden Sie ein *Cash-Flow-Diagramm*, um die Cash-Flow-Eingaben zu bestimmen und das Vorzeichen (*positiv* oder *negativ*) zu spezifizieren.

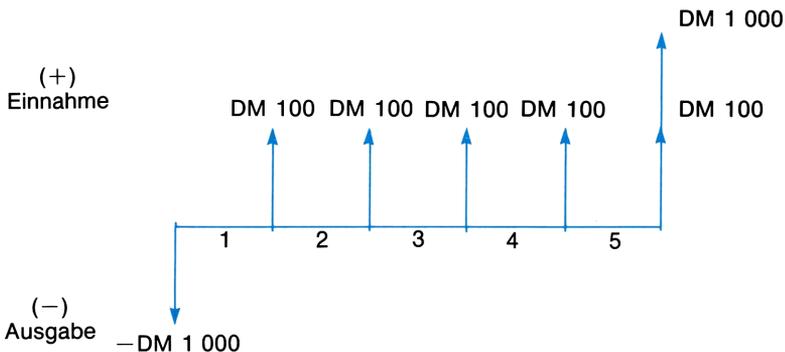
146 Finanztechnische Berechnungen

Das Cash-Flow-Diagramm ist eine in Zeitperioden eingeteilte Zeitachse. Cash-Flows (Transaktionen) werden durch senkrechte Pfeile eingetragen: ein nach *oben* gerichteter Pfeil (*Einnahme*) ist *positiv*, ein nach *unten* gerichteter Pfeil (*Ausgabe*) ist *negativ*.

Beispielsweise zeigt diese in sechs Verzinsungsperioden eingeteilte Zeitachse eine anfängliche Ausgabe von DM 20 und eine Einnahme von DM 50 am Ende der vierten Periode.



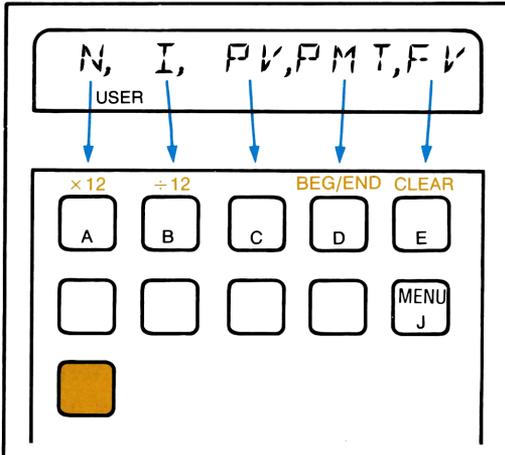
Diese in fünf Verzinsungsperioden eingeteilte Zeitachse zeigt eine anfängliche Ausgabe von DM 1 000 und eine Einnahme von DM 100 am Ende jeder Periode sowie eine zusätzliche Einnahme von DM 1 000 am Ende der fünften Periode.



Anweisungen

- Das Programm TVM kann jede der fünf Variablen N , I , PV , PMT oder FV berechnen, wenn die anderen drei bzw. vier Variablen (N oder I muß enthalten sein) gegeben sind. Die Reihenfolge der Variableneingabe ist beliebig.*
- Löschen Sie vorherige Daten (■[E]), bevor Sie mit einer neuen Berechnung anfangen. Andernfalls werden vorherige Daten, welche nicht überschrieben wurden, für die Berechnung mit verwendet. Vorherige Daten werden durch das erneute Ausführen des Programms automatisch gelöscht.
- Denken Sie daran, Einnahmen als positive Werte (Pfeile nach oben) und Ausgaben durch negative Werte (Pfeile nach unten) zu kennzeichnen. Das Ergebnis wird ebenfalls als positiver bzw. negativer Wert ausgegeben, wodurch eine Einnahme bzw. eine Ausgabe gekennzeichnet wird.
- Überprüfen Sie, ob der Zahlungsmodus richtig gesetzt ist. Wenn der Indikator für Flag 00 in der Anzeige erscheint (eine kleine 0 unterhalb der Haupt-Anzeigezeile), befinden Sie sich im Beginn-Modus. Andernfalls ist der Ende-Modus eingestellt. Um den Modus zu ändern, drücken Sie ■[D] (Umschalter). Die Anzeige läßt dann den momentan eingestellten Modus erkennen: **BEGIN MODE** oder **END MODE**. Die Voreinstellung ist Ende-Modus (Flag 00 gelöscht).
- Denken Sie daran, daß der Zinssatz mit der Anzahl der Verzinsungsperioden vereinbar sein muß. (Ein *jährlicher* Prozentsatz ist nur verwendbar, wenn die Anzahl der Verzinsungsperioden mit der Anzahl von Jahren übereinstimmt.)
- Möglicherweise möchten Sie das Anzeigeformat auf zwei Stellen einstellen. ([FIX]2).

* Wenn Sie nur vier Variablen verwenden, muß die fünfte gleich Null gesetzt werden. Wenn Sie TVM starten oder Finanzdaten löschen (■[E]), werden alle Variablen gleich Null gesetzt. In diesen Fällen müssen Sie für die Variablen nicht Null eingeben.



Dieses Menü gibt Ihnen die Neubelegung der User-Tasten mit Funktionen aus TVM an. Sie können sich dieses Menü jederzeit durch Drücken der Taste **J** in der Anzeige ansehen. Der Programmablauf wird dadurch in keiner Weise gestört.

Durch Drücken der Taste **←** können Sie das Menü jederzeit löschen. Dann erscheint in der Anzeige der Inhalt des X-Registers. Das Programm wird dadurch nicht beendet. Sie können nun Berechnungen durchführen und dann das Menü durch Drücken von **J** wieder in die Anzeige holen. (Sie müssen die Anzeige des Programms *nicht* löschen, bevor Sie Berechnungen ausführen.)

Anweisungstabelle für TVM

		Umfang: 010
Anweisungen	Eingabe	Anzeige
1. Starten Sie das Programm TVM. Die Anzeige des Menüs gibt die neue Belegung in den beiden oberen Reihen des User-Tastenfelds mit <i>N, I, PV, PMT</i> und <i>FV</i> an.	XEQ TVM *	N, I, PV,PMT,FV
2. Überprüfen Sie den Zahlungsmodus, indem Sie in der Anzeige nach dem 0 -Indikator suchen. (0 bedeutet Beginn-Modus; keine 0 bedeutet Ende-Modus.) Ändern Sie den Modus ab, falls nötig.	■ D (Schaltet zwischen den Modi um) R/S oder J	END MODE oder BEGIN MODE N, I, PV,PMT,FV
3a. Geben Sie, falls <i>N</i> nicht die zu berechnende Variable ist, die Anzahl der Verzinsungsperioden <i>N</i> ein. Schritt 3b ist ein kürzerer Weg für das Berechnen der Monate aus einer gegebenen Anzahl von Jahren.	<i>N</i> A (N)	N= N†

Anweisungstabelle (Fortsetzung)

Anweisungen	Eingabe	Anzeige
3b. Alternativ zu 3a: Wenn Sie mit monatlichen Zahlungen oder mit monatlichen Verzinsungsperioden für eine bekannte Anzahl von Jahren arbeiten, berechnet Ihnen dieser Schritt automatisch N (zu $12 \times$ Jahre) und gibt diesen Wert ein. Geben Sie die Anzahl n der Jahre ein.	n <input type="checkbox"/> A	$N = 12 \times n$
4a. Geben Sie, falls Sie I nicht berechnen möchten, den periodischen Zinssatz I ein. Schritt 4b ist ein kürzerer Weg für das Berechnen des monatlichen Zinssatzes aus einem gegebenen jährlichen Zinssatz.	I <input type="checkbox"/> B (I)	$I = I$
4b. Alternativ zu 4a: Wenn Sie mit monatlichen Verzinsungsperioden arbeiten und der jährliche Zinssatz gegeben ist, berechnet dieser Schritt automatisch I (zu jährlicher Prozentsatz $\div 12$) und gibt diesen Wert ein. Geben Sie den jährlichen Prozentsatz APR (ann. perc. rate) ein. Beachten Sie, daß Sie hierbei einen nominalen—nicht effektiven—Zinssatz erhalten.	APR <input type="checkbox"/> B	$I = APR \div 12$
5. Geben Sie den Anfangswert PV ein, falls Sie ihn nicht berechnen möchten.	PV <input type="checkbox"/> C (PV)	$PV = \text{Eingabe}$
6. Geben Sie die period. Zahlung PMT ein, falls Sie sie nicht berechnen möchten.	PMT <input type="checkbox"/> D (PMT)	$PMT = \text{Eingabe}$
7. Geben Sie den Endwert FV ein, falls Sie ihn nicht berechnen möchten.	FV <input type="checkbox"/> E (FV)	$FV = \text{Eingabe}$
8. Berechnen Sie nun die verbleibende Variable, indem Sie die entsprechende Taste drücken.	<input type="checkbox"/> A oder <input type="checkbox"/> B oder <input type="checkbox"/> C oder <input type="checkbox"/> D oder <input type="checkbox"/> E	$N = \text{Ergebnis}$ oder $I = \text{Ergebnis}$ oder $PV = \text{Ergebnis}$ oder $PMT = \text{Ergebnis}$ oder $FV = \text{Ergebnis}$
9. Um den Wert jeder Variablen jederzeit abzufragen:	<input type="checkbox"/> RCL <input type="checkbox"/> A bis <input type="checkbox"/> E	Wert
10. Um das Hauptmenü (N, I, PV,PMT,FV) jederzeit in die Anzeige zu holen (ohne Ihre Eingaben und Berechnungen zu beeinflussen):	<input type="checkbox"/> J	N, I, PV,PMT,FV
11. Löschen Sie alte Finanzdaten, bevor Sie eine neue Aufgabe lösen.	<input type="checkbox"/> E	N, I, PV,PMT,FV

* Zur Ausführung eines Programms drücken Sie entweder **XEQ** **ALPHA** Alpha-Name **ALPHA** oder eine entsprechende benutzerspezifische Taste.

† Wenn Sie einen Drucker angeschlossen haben, wird das Hauptmenü (**N, I, PV,PMT,FV**) automatisch in die Anzeige geholt, nachdem die letzte Eingabe ausgedruckt wurde.

Bemerkungen

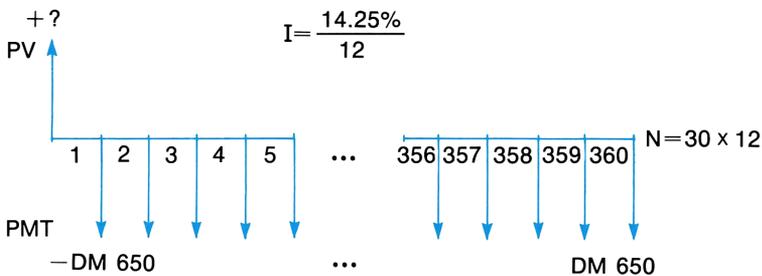
Dieses Programm verwendet lokale Alpha-Labels (im Benutzerhandbuch des HP-41 erklärt), die den Tasten \boxed{A} – \boxed{E} und den dazu umgeschalteten (außer \boxed{C}), sowie \boxed{J} zugeordnet sind. Diese lokalen Zuordnungen werden durch jede andere von Ihnen vorgenommene User-Tasten-Zuordnung überschrieben. Vergewissern Sie sich daher, daß Sie alle bestehenden Zuweisungen zu diesen User-Tasten gelöscht haben, bevor Sie das Programm starten. Vermeiden Sie ebenso, diese User-Tasten während des Programmablaufs neu zu definieren.

Mit für die Finanzvariablen zugeordneten Tasten können Werte nur abgespeichert werden, wenn Sie diese über das Tastenfeld eingeben. Wenn Sie beispielsweise einen Wert aus einem Register abrufen und dann eine Variablen-Taste drücken, wird der abgerufene Wert nicht abgespeichert, sondern die Variable berechnet. Wenn Sie Werte abspeichern wollen, die in die Anzeige (X-Register) nicht eingetippt wurden, drücken Sie vor dem Drücken der Variablen-Taste \boxed{STO} .

Beispiele

Ein Darlehensnehmer kann monatlich DM 650 für ein Darlehen mit 30 Jahren Laufzeit und $14\frac{1}{4}\%$ Zins abzahlen. Wie hoch kann das Darlehen maximal sein? Die erste Annuitätenzahlung erfolgt einen Monat nach Aufnahme des Darlehens. (Dies erfordert den Ende-Modus.)

Cash Flows, 1. Beispiel



Tastenfolge

FIX 2

XEQ **SIZE** 010

XEQ **TVM**

30 **A**

R/S

14.25 **B**

R/S

650 **CHS** **D** (PMT)

R/S

C (PV)

Anzeige

N, I, PV,PMT,FV

N=360.00

N, I, PV,PMT,FV

I=1.19

N, I, PV,PMT,FV

PMT=-650.00

N, I, PV,PMT,FV

PV=53955.92

Stellt das hier be-
nutzte Anzeigeformat
ein.

Optional—gibt die
Anzahl der vom Pro-
gramm benötigten
Speicherregister an.
Dies ist nicht notwen-
dig, wenn Ihre Ein-
stellung bereits
SIZE ≥ 010 ist.

Startet das Pro-
gramm. Alte Finanz-
daten werden ge-
löscht und der Ende-
Modus automatisch
eingestellt.

Anzahl der Perioden.
Abrufen des Menüs
(optional).

Monatlicher nomi-
naler Zinssatz.

Abrufen des Menüs
(optional).

Monatliche Zahlung.
Abrufen des Menüs
(optional).

Maximale Höhe des
Darlehens.

Wie hoch sind die monatlichen Zahlungen, wenn die Hypothek nur DM 53 500 beträgt? (Ändern Sie PV, behalten Sie die Werte der übrigen Variablen bei und berechnen Sie PMT.)

Tastenfolge

R/S

53500 **C** (PV)

R/S

D (PMT)

Anzeige

N, I, PV,PMT,FV

PV=53500.00

N, I, PV,PMT,FV

PMT=-644.51

Abrufen des Menüs
(optional).

Höhe des Darlehens
gegeben.

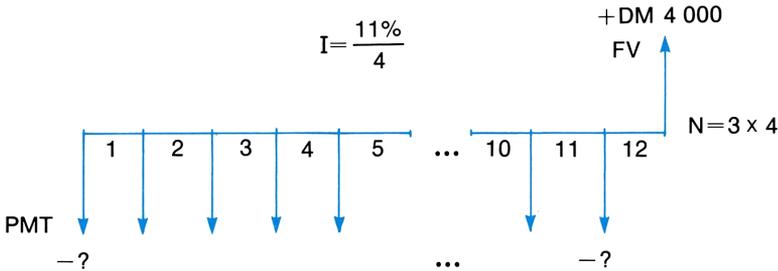
Abrufen des Menüs
(optional).

Monatliche Zahlung.

Beachten Sie, daß wenn Sie nach Eingabe eines Wertes eine Taste **A**–**E** drücken, dieser Wert der entsprechenden Variablen *zugewiesen* wird. Wenn Sie jedoch **A**–**E** *ohne* vorhergehende Eingabe drücken, wird die entsprechende Variable *berechnet*.

Wieviel muß vierteljährlich auf ein Sparkonto eingezahlt werden, wenn in drei Jahren DM 4 000 erspart werden sollen? Der jährliche Zinssatz beträgt 11%, und die Einzahlungen "beginnen" sofort.

Cash Flows, 2. Beispiel



Tastensequenz

E

D

R/S

3 ENTER↑ 4 x

A (N)

11 ENTER↑ 4 ÷

B (I)

4000 E (FV)

D (PMT)

Anzeige

N, I, PV,PMT,FV

BEGIN MODE

N, I, PV,PMT,FV

12.00

N=12.00

2.75

I=2.75

FV=4000.00

PMT=-278.22

Alte Finanzdaten werden gelöscht. (Es wird angenommen, daß Sie sich immer noch im Programm TVM befinden.)

Stellt Beginn-Modus ein. (In der Anzeige sollte der 0-Indikator erscheinen.)

Gesamte Anzahl der Einzahlungen.

Vierteljährlicher nominaler Zinssatz.

Angestrebter Betrag.

Erforderliche vierteljährl. Einzahlung.

Programmier-Informationen

Die nachfolgend aufgeführten Unterprogramme aus TVM können Sie in Ihre eigene Programme einbauen. Es kann die Anzahl der Verzinsungsperioden, Zinssatz, Anfangswert, Annuität oder Endwert berechnet werden, wenn die anderen vier Parameter gegeben sind.

Speicherminimum: SIZE 010.

Benutzte Flags: 00, 25.

Name des Unterprogramms	alter Registersinhalt	neuer Registerinhalt	zu Initialisierende Flags
N (Anzahl der Perioden)	$R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	X-Register = N $R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	SF 00 für Beginn-Modus CF 00 für Ende-Modus
*I (Zinssatz)	$R_{01} = N$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	$X = I$ $R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	SF 00 für Beginn-Modus CF 00 für Ende-Modus
PV (Barwert)	$R_{01} = N$ $R_{02} = I$ $R_{04} = PMT$ $R_{05} = FV$	$X = PV$ $R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	SF 00 für Beginn-Modus CF 00 für Ende-Modus
PMT (Zahlungsbetrag)	$R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{05} = FV$	$X = PMT$ $R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	SF 00 für Beginn-Modus CF 00 für Ende-Modus
FV (Endwert)	$R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$	$X = FV$ $R_{01} = N$ $R_{02} = I$ $R_{03} = PV$ $R_{04} = PMT$ $R_{05} = FV$	SF 00 für Beginn-Modus CF 00 für Ende-Modus

Anmerkung. Für das Benutzen dieser Unterprogramme initialisieren Sie die entsprechenden Register und setzen (bzw. löschen) Flag 00 für Beginn- (bzw. Ende-) Modus. Führen Sie dann das gewünschte Unterprogramm aus. Es gibt den gewünschten Wert ins X-Register (Anzeige) aus und speichert ihn im entsprechenden Register ab.

PROGRAMM-INDEX

Boolesche Logik	129, 130
CFIT	135
DIFEQ	89
INTEG	81
Komplexe Funktionen	95
Matrix-Operationen/-Funktionen	30, 58
MATRX	19
PLY	73
SOLVE	63
TR	119
TVM	145
Umwandlung zwischen Zahlensystemen	129, 130
VC	103



**Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.**

**European Headquarters
150, Route du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland**

**HP-United Kingdom
(Pinewood)
GB-Nine Mile Ride, Wokingham
Berkshire RG11 3LL**

**Bestellnummer
00041-90562**

Printed in Singapore 8/85