

Das Programmsystem

UNIRUN

in FORTRAN V

von Edwin DEUTSCH

Forschungsbericht Nr. 76

Juni 1973

INHALT

	Seite
1. Einleitung	4
2. Der inhaltliche Rahmen	5
2.1. Formeln	6
3. Systemmerkmale	9
3.1. Der modulare Aufbau	10
3.2. Commonbereiche	11
3.3. Algorithmen	13
4. Aufruf und Transformation von Daten	15
4.1. Vorlaufkarte	15
4.2. Variablenkarte und Transformationen	16
4.3. Programme des Segmentes TRANSF	19
5. Programmtypen und Kontrollsprache	21
5.1. Begriffe	21
5.2. Programme vom Typ 1 (UNICOM)	24
5.3. Programme vom Typ 2 (UNIMAX)	25
5.4. Programme vom Typ 3 (UNIOLS, UNILAG, UNIPAG)	26
5.5. Fehlernachrichten	27
5.6. Details der Programmstruktur	28
5.6.1. Flußdiagramm	29
5.6.2. Das Verzweigungsprogramm SWITCH	31
5.6.3. Die Variablenauswahl	32
5.6.4. Das Steuerprogramm OPT	34
6. Optionen	37
6.1. Syntax	37
6.2. Tafel der Optionen	39
6.3. Funktionsweise	40
6.4. Die Programme XYZSTV und XYZOPT	42
6.5. OPTKEN	44

7. Die Segmente von UNIRUN	46
7.1. UNICOM	46
7.2. UNIMAX	48
7.3. UNIOLS	50
7.4. UNILAG und UNIPAG	51
8. Anhang	55
8.1. Aufruf, Systemstandardwerte	55
8.2. Segmentierung	57
9. Literaturangaben	58

1. Einleitung

Seit dem Erscheinen des Forschungsberichtes Nr. 68: "Das Programmsystem UNIRUN, eine Darstellung seiner Methodik, von Edwin Deutsch, Juni 1972", hat der Autor das System in der Programmiersprache FORTRAN V neu verfaßt und vor allem in programmiertechnischer Hinsicht weiterentwickelt. In diesem Sinn bildet die vorliegende Arbeit eine direkte Fortsetzung des oben erwähnten Forschungsberichtes. Grob gesprochen enthält der erste Forschungsbericht eine Darstellung der Methoden, während sich der zweite mit den von nun an gültigen Programmsystemen befaßt. Die Programmstruktur ist ihrem Wesen nach zwar gleich geblieben, doch unterscheidet sich das neue System von der alten FORTRAN II-d Version überall dort, wo wirksamere Programmiermethoden zur Verfügung standen. Weiters wurden die Erfahrungen, die bei der Verwendung von UNIRUN bisher gemacht wurden, dazu benützt, um wesentliche Erleichterungen bei der Handhabung der Programme zu erzielen.

Das Programmsystem umfaßt zum Zeitpunkt der Veröffentlichung ein Programm, welches zum Einlesen und Transformieren von Daten dient, und fünf große Programme, welche die lineare Regressionstheorie zum Gegenstand haben: UNICOM, UNIMAX, UNIOLS, UNILAG, UNIPAG. In Zukunft soll das Programmsystem die verschiedensten Bereiche der Ökonometrie, im besonderen die Regressionstheorie und die Zeitreihenanalyse, umfassen.

Dieser Forschungsbericht stellt die wesentlichen Elemente des Systems UNIRUN dar. Für spezielle Details muß auf die Dokumentation des Systems verwiesen werden.

Folgende Herren möchte der Autor für das Zustandekommen der Arbeit danken: Herrn Erich Wang, mit dem der Autor im Vorstadium des Programmes zusammenarbeitete;

Herrn Gerhard Eichinger, der vor allem einen wichtigen Beitrag zum Entwurf der Kontrollsprache lieferte;

schließlich den Herren Ehtecham Darakhchan, Peter Hampapa, und Erhardt Heinrich, die bei der Erstellung der Dokumentation behilflich waren.

2. Der inhaltliche Rahmen

Wir betrachten einen allgemeinen stochastischen Prozeß zwischen Zeitreihen, den wir formal mit stochastischen Variablen y_i , nicht-stochastischen Variablen x_j , unbekanntem Parametern Θ , Zufallsgrößen ϵ und bekanntem funktionalen Zusammenhang F anschreiben:

$$F(y_i, x_j, \Theta, \epsilon) = 0 \quad (2.1.)$$

Das Ziel des Systems UNIRUN im weitesten Sinn liegt in der Schätzung der Parameter solcher Strukturen.

Wir betrachten jetzt eine Menge M von Zeitreihen $\{x_{it}\}$, $i=1, 2, \dots, m$ und $t=1, 2, \dots, T$. Unter diesen Zeitreihen wählen wir eine Teilmenge $\{x_{n(i), t}\}$ aus, mit $i=1, 2, \dots, k$. Wir suchen die Schätzwerte $\hat{\alpha}, \hat{\beta}_i$ des linearen Ansatzes

$$y_t = \alpha + \sum_{i=1}^k \beta_i x_{n(i), t} + \epsilon_t \quad (2.2.)$$

Die Programme von UNIRUN sollen nun den Benutzer in die Lage versetzen, optimale, d. h. ökonomisch interpretierbare und statistisch gesicherte Einzelgleichungen zu schätzen:

UNICOM und UNIMAX sind kombinatorische Programme, die aus der Menge M eine Auswahl vertretbarer Schätzgleichungen mit Variablen

$$\{y, x_{n(1)}, \dots, x_{n(k)}\} \text{ wiedergeben.}$$

UNIOLS berechnet einzelne Schätzgleichungen. Wie bei den vorigen Programmen ist eine Konstantenunterdrückung möglich.

UNILAG und UNIPAG untersuchen eine Regressionsgleichung nach vorhandenen Lags.

Die obigen Programme werden im folgenden auch als Segmente bezeichnet, weil sie eigentlich Unterprogramme des Systems darstellen. Der Begriff Segment hat zudem nach eine weitere Bedeutung im Sinn der Steuersprache EXEC - 8 (siehe 8.2.).

2.1. Formeln

Die folgenden Formeln geben die wichtigsten Resultate der Regressionstheorie wieder.

Gegeben seien $\{x_{n(i), t}\}$, die erklärenden Variablen,
 y_t , die endogene Variable, und der Modellansatz

$$y_t = \alpha + \sum_{i=1}^k \beta_i x_{n(i), t} + \varepsilon_t \quad (2.2.)$$

mit $i = 1, 2, \dots, k$ und $t = 1, 2, \dots, T$.

Wir bilden:

$$\bar{x}_{n(i)} = \frac{1}{T} \sum_{t=1}^k x_{n(i), t}, \quad \bar{y} = \frac{1}{T} \sum_{t=1}^T y_t \quad (2.1.1.)$$

Zeitreihen in Abweichungsform:

$$x'_{n(i), t} = x_{n(i), t} - \bar{x}_{n(i)}, \quad y'_t = y_t - \bar{y} \quad (2.1.2.)$$

Wir bilden die Matrix X und den Vektor Y :

$$X = \begin{bmatrix} x'_{n(1), 1} & \dots & x'_{n(k), 1} \\ \vdots & & \vdots \\ x'_{n(1), T} & \dots & x'_{n(k), T} \end{bmatrix} \quad Y = \begin{bmatrix} y'_1 \\ \vdots \\ y'_T \end{bmatrix} \quad (2.1.3.)$$

Dann wird berechnet:

$$\hat{\beta} = (X'X)^{-1} X'Y \quad (2.1.4.)$$

$$\hat{\alpha} = \bar{y} - \bar{X}'\hat{\beta} \quad (2.1.5.)$$

$$\text{und } \bar{X} = \begin{bmatrix} \bar{x}_{n(1)} \\ \vdots \\ \bar{x}_{n(k)} \end{bmatrix}$$

$$R^2 = \frac{X'Y \hat{\beta}}{Y'Y} \quad (2.1.6.)$$

$$e = Y - X \hat{\beta} \quad (2.1.7.)$$

$$s = \sqrt{\frac{e'e}{T-k-1}} \quad (2.1.8.)$$

$$\text{proza} = 100 \cdot \frac{s}{|\hat{\beta}|} \cdot \sqrt{\frac{1}{T} + \bar{X}'(X'X)^{-1}\bar{X}} \quad (2.1.9.)$$

$$\text{proz}_i = 100 \cdot \frac{s}{|\hat{\beta}_i|} \cdot \sqrt{(X'X)^{-1}_{ii}} \quad (2.1.10.)$$

$$dw = \frac{\sum_{t=2}^T (e_t - e_{t-1})^2}{\sum_{t=1}^T e_t^2} \quad (2.1.11.)$$

Unter proz_i werden die Prozentwerte der Standardabweichung der Variablen $x_{n(i)t}$ verstanden.

Wenn die Konstante unterdrückt wird, ändern sich die Formeln (2.1.3.) und (2.1.6.), die Formeln (2.1.5.) und (2.1.9.) entfallen.

Wir rechnen mit (2.1.12.) und (2.1.13.):

Die Matrix X und der Vektor y werden aus Originaldaten berechnet:

$$X = \begin{bmatrix} x_{n(1),1} & \dots & x_{n(k),1} \\ \vdots & & \vdots \\ x_{n(1),T} & \dots & x_{n(k),T} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \quad (2.1.12.)$$

$$R^2 = \frac{\left[\sum y_t \hat{y}_t - T \bar{y} \hat{\bar{y}} \right]^2}{\left[\sum y_t^2 - T \bar{y}^2 \right] \left[\sum \hat{y}_t^2 - T \hat{\bar{y}}^2 \right]} \quad (2.1.13.)$$

$$y_t - e_t.$$

Das obige s stellt die unverzerrte Residuenstreuung dar. Es wird auch eine "normale" Residuenstreuung definiert:

$$\tilde{s} = \sqrt{\frac{e' e}{T}} \quad \text{mit } \tilde{s} < s \quad (2.1.14.)$$

Bei den Lagprogrammen UNILAG und UNIPAG werden aus den ausgewählten Zeitreihen linear interpolierte bzw. splin-interpolierte Zeitreihen gebildet und auf diese dann die lineare Regression mit obigen Formeln angewendet. Dabei wird die Residuenlänge T automatisch um Eins verkürzt.

Geg. sei eine Zeitreihe $x_{n(i), t}$ und ein zugehöriger Lag λ_i . Die linear interpolierte bzw. gelagte Zeitreihe x_{it}^L lautet dann

$$x_{it}^L := x_{n(i), t} + 1 - \lambda_i := (1 - \lambda_i) x_{n(i), t+1} + \lambda_i x_{n(i), t} \quad (2.1.15.)$$

$$t = 1, 2, \dots, T-1$$

Bei der Spline-Interpolation lautet die ähnliche Formel

$$x_{n(i), t}^L = x_{n(i), t+1} - \lambda_i := sp_t \lambda^3 + sp_{t+1} (1-\lambda)^3 +$$

$$+ (x_{n(i), t+1} - sp_{t+1}) (1-\lambda) + (x_{n(i), t} - sp_t) \lambda. \quad t=1, 2, \dots, T-1$$

$$(2.1.16.)$$

Die genaue Ableitung des Splines sp_t der Zeitreihe $x_{n(i), t}$

findet man im Forschungsbericht Nr. 68 sowie in [2] (siehe 9. Literaturangaben).

3. Systemmerkmale

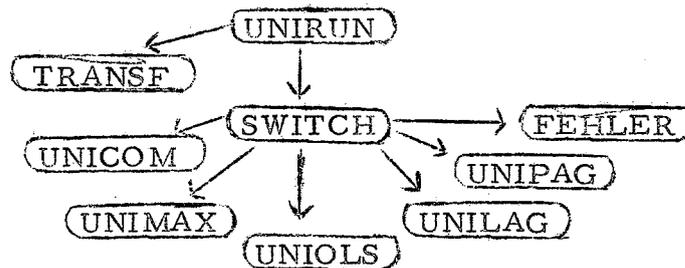
Die Einheitlichkeit des Systems UNIRUN erweist sich sowohl am inhaltlichen Rahmen, als auch in programmtechnischer Hinsicht. Die programmtechnische Einheit zeigt sich an folgenden Merkmalen:

- Der modulare Aufbau der Unterprogramme des Systems.
- ✓ Die Unterprogramme verkehren miteinander über Commonbereiche.
- Spezielle Unterprogramme für die Behandlung von Algorithmen.
- Flexibilität der Unterprogramme durch sorgfältige Wahl der Argumentenlisten.
- Gliederung der Segmente in Programmtypen, die den Ablauf der Kontrollsprache bedingen.
- Einheitliche Kontrollsprache mit Optionen.
- Optionen mit gleichen Namen erfüllen die gleiche Funktion.

3. 1. Der modulare Aufbau

Das System setzt sich aus hierarchisch gegliederten Unterprogrammen zusammen. Sie besitzen eine weitgehend systematisierte innere Struktur. Manche Unterprogramme werden in mehreren übergeordneten Programmen verwendet.

Das Hauptprogramm UNIRUN ist in Segmente gegliedert. Diese Segmente enthalten das gleichnamige Unterprogramm und dessen Unterprogramme.



Das Segment TRANSF stellt die Verbindung mit dem Datenträger her, der im allgemeinen Fall aus eingegebenen Datenkarten besteht.

Das Segment TRANSF transformiert die Daten und wird nur eingangs benützt (siehe 4.)

Das Segment SWITCH verbindet die eigentlichen Segmente des Systems (siehe 5. und 7.). Diese Segmente können mit Hilfe der Kontrollsprache beliebig oft hintereinander benützt werden.

Im Segment FEHLER werden Fehler behandelt (siehe 5. 5.).

3.2. Commonbereiche

Die Commonbereiche lassen sich in 2 Gruppen aufgliedern:
 Commonbereiche, die allen Segmenten gemeinsam sind, sowie
 Commonbereiche, die für die jeweiligen Segmente spezifisch sind.
 Es handelt sich stets um benannte Commonbereiche; der BLANK
 COMMON soll nirgends verwendet werden.

Die Commonbereiche sind samt zugehörigen Vereinbarungs-
 anweisungen, DATA-Segments und Parameterwerten in FORTRAN-
 Prozeduren mit gleichen Namen enthalten. Die Programme des
 Systems schließen die Prozeduren durch eine einfache Anweisung ein:

```
INCLUDE <Name>
```

Auf diese Weise können die Dimensionen der Bereiche und die
 Parameter leicht geändert werden. Die Programme nehmen stets
 auf die Parameterwerte bezug und sind von ihren konkreten Zu-
 weisungen daher unabhängig.

Zu den gemeinsamen Commonbereichen gehören:

/HAUPT/ enthält den Bereich der Zeitreihen sowie die zugehörigen
 Namen und Transformationsangaben.

/NEC/ enthält u. a. Anzahl und Länge der Zeitreihen.

/ERR/ enthält Lesebuffer und Printbuffer und Variable, welche
 die Ein- und Ausgabe regeln.

Welche Werte die Commonbereiche /HAUPT/ /NEC/ und /ERR/
 beim Verlassen von TRANSF enthalten, wird in 4.3. genau angeführt.

/REGREV/ Dieser Commonbereich enthält die Variablen zur Er-
 mittlung der Regressionsergebnisse sowie die Variablen
 IZL, die Bereiche NUMMER und IPL und den Parameterwert
 NERKLV (ihre Bedeutung siehe 3.3.).

Die segmentspezifischen Commonbereiche werden im Anklang an das zugehörige Segment benannt. Sie enthalten Optionsnamen und Optionsvariable der Segmente, sowie:

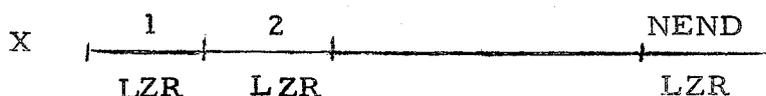
- /COM/ in UNICOM und UNIMAX, enthält die 2. Momente der Variablen.
- /MAX/ in UNIMAX, enthält die Informationen über das Kombinationsschema
- /OLS/ in UNIOLS, nur Optionsnamen und Variable.
- /LAG/ in UNILAG und UNIPAG, enthält die Informationen über die Lagstruktur.
- /PAG/ in UNIPAG, Bereich der Splines der Zeitreihen.

3.3. Algorithmen

Um Speicherplatz zu sparen, erwies sich die Speicherung der Zeitreihen auf einem eindimensionalen Bereich von Vorteil. Daraus resultiert jedoch eine bestimmte Gestalt der Algorithmen, die auf die lineare Speicherung Bezug nehmen muß.

Im folgenden werden die Variablen m , k und T aus (2.2.) entsprechend den programminternen Namen umbenannt: $m \hat{=} \text{NEND}$, $k \hat{=} \text{IZL}$, $T \hat{=} \text{LZR}$.

Die lineare Speicherung läßt sich dann so darstellen:



Das Problem lautet nun, gewisse Teilmengen E der Gesamtmenge M der Zeitreihen für die Algorithmen heranzuziehen.

Diese Teilmengen E werden entweder vom Benutzer selbst durch Eingabe von Variablenauswahlkarten bestimmt (siehe 5.6.3.) oder sie werden automatisch bei kombinatorischen Programmen hergestellt. Das Programm VARKOM (siehe ebenfalls 5.6.3.) bestimmt dabei die folgenden für die Algorithmen wesentlichen Variablen (Dabei ist hier die Auswahl im Sinn von erklärenden und endogenen Variablen verstanden):

IZL Anzahl der erklärenden Variablen

NUMMER..... Bereich der Nummern der erklärenden Variablen

NUMMER (i), $i = 1, \dots, \text{IZL}$

Die Nummern werden in TRANSF in der Reihenfolge des Einlesens aller Variablenkarten fest gelegt; der Bereich NUMMER bezieht sich dann auf diese Anordnung.

IPL Bereich der Indices auf X:

$$\text{IPL}(i) = (\text{NUMMER}(:) - 1) * \text{LZR}$$

Der Bereich IPL erleichtert den Zugriff auf X wesentlich, indem eine DO-Schleife stets mit den Indices $\text{IPL}(i) - j$, $j=1, \dots, \text{LZR}$ gezählt werden kann.

Die maximale Anzahl wählbarer (erklärender) Variabler wird durch den Systemstandardwert NERKLV festgelegt. Auf die endogene Variable wird dann stets mit den Werten $\text{NUMMER}(\text{NERKLV} + 1)$ und $\text{IPL}(\text{NERKLV} + 1)$ zugegriffen.

Die Variablen IZL, NUMMER und IPL sowie der Parameterwert NERKLV sind im Commonbereich /REGREV/ enthalten. Entscheidend ist nun, daß jene Unterprogramme, die Algorithmen verarbeiten, entweder keine Commonbereiche verwenden und somit über die Argumentlisten allgemein verwendbar sind, oder nur den Commonbereich /REGREV/ verwenden und somit unabhängig sind von der Reihenfolge der Speicherung der Zeitreihen.

Derzeit enthält das System UNIRUN Unterprogramme für eine Vielzahl von Algorithmen der linearen Algebra. Am Ausbau der Bibliothek wird ständig gearbeitet. Hinzu kommen vor allem auch Programme, die erweiterte Matrixoperationen gestatten.

4. Aufruf und Transformation von Daten

Der folgende Abschnitt schildert die Eingabe von Daten und deren Transformationsmöglichkeiten.

4.1. Vorlaufkarte

Der Aufruf des Systems geschieht mit

QXQT Q * FILE . UNIVEK (Siehe 8.1.)

Der Name UNIVEK bezieht sich auf die eindimensionale Speicherung von Zeitreihen. Auf der folgenden Vorlaufkarte werden Zeitreihen, Beginndatum der Zeitreihen, sowie Beginn und Enddatum der Residuen (d. h. der transformierten Zeitreihen) angegeben:

Anzahl der Zeitreihen		in der Spalte
		1-3
Beginnjahr] der eingelesenen Zeitreihen	5, 6
Periodizität		7
Beginnperiode		8, 9
Beginnjahr] der Residuen	10, 11
Periodizität		12
Beginnperiode		13, 14
Endjahr] der Residuen	15, 16
Periodizität		17
Endperiode		18, 19

Unter Periodizität wird verstanden:

J Jahresdaten
 S Semesterdaten
 Q Quartalsdaten
 M Monatsdaten

Periode ist eine zweistellige Zahl.

4.2. Variablenkarte und Transformationen

Auf der Variablenkarte werden der Name und der Transformationsstring der Zeitreihe angegeben:

	in der Spalte
Name der Zeitreihe	1-6
weitere Angaben nach Wunsch:	
Normierung	N 7
Logarithmus	L 8
Akzelerator	Z 9
Differenzenbildung.....	10
BLANKkeine Differenzen (Absolutwert)	
AAbsolute Differenz	
RRelative Differenz	
Ordnung der Differenz	11, 12
Lag (Verzögerung)	13, 14
Potenz	P 15
Wert der Potenz im Format F 5.2	
Klassencode	73-78

An die Variablenkarte werden unmittelbar eine Formalkarte und sovieler Datenkarten angefügt, wie in der Vorlaufkarte durch die Angaben 5-9 und 15-19 bestimmt wurden.

Der Klassencode stellt einen höheren Ordnungsbegriff der Zeitreihen dar. Er hat keinen Einfluß auf die Zuweisung von Nummern.

Die Transformationen erfolgen in dieser Reihenfolge:

1. $\left\{ \begin{array}{l} \text{Logarithmus} \\ \text{Potenz} \end{array} \right.$

2. $\left\{ \begin{array}{l} \text{absolute} \\ \text{relative} \end{array} \right.$ Differenz

3. Akzelerator

4. Lag

5. Normieren

T sei die Gesamtlänge der eingegebenen Zeitreihe.

Bei gegebener Zeitreihe x_t ($t = 1, \dots, T$) bewirken die Transformationen folgendes:

1. Logarithmieren: $x_t^* = \log_e x_t$ (4.2.1.)

Potenz $x_t^* = x_t^p$, $0.01 \leq p \leq 99.99$ (4.2.2.)

2. absolute Differenz k-ter Ordnung:

$$x_t^* = \frac{x_t - x_{t-k}}{x_{t-k}}, \quad t = k+1, \dots, T \quad (4.2.3.)$$

relative Differenzen k-ter Ordnung:

$$x_t^* = \frac{x_t - x_{t-k}}{x_{t-k}} \quad t = k+1, \dots, T \quad (4.2.4.)$$

3. Akzelerator derselben Ordnung wie in 2.

$$x_t^* = x_t - x_{t-k} \quad t = 2k+1, \dots, T \quad (4.2.5.)$$

Auf der rechten Seite steht das x_t aus 2.

4. Lag der Ordnung l :

$$x_t' = x_{t-l} \quad t = \Theta + l + 1, \dots, T \quad (4.2.6.)$$

mit einem Θ entsprechend 1., 2., 3:

0, k oder 2k.

Bezeichnen wir das Beginndatum der Residuen mit f (wobei $f = 1$ für das erste Datum der eingelesenen Zeitreihe gilt), so ergeben sich zwei Möglichkeiten:

- a) $f < \Theta + l + 1$: es gibt eine Fehlermeldung.
- b) $f \geq \Theta + l + 1$: Die transformierte Zeitreihe wird auf ihre endgültige Länge, die Residuenlänge gebracht.

Nach dem 4. Schritt werden also die Indices von x um $f - 1$ erniedrigt. Für die Residuenlänge gilt

$$\text{LZR} = T - f + 1$$

und man erhält die Zeitreihe x_t , $t = 1, \dots, \text{LZR}$.

5. Normieren:

$$x_t' = \frac{x_t - \bar{x}}{s} \quad (4.2.7.)$$

$$\text{wobei } \bar{x} = \frac{1}{\text{LZR}} \sum_{t=1}^{\text{LZR}} x_t$$

$$s^2 = \frac{1}{\text{LZR}-1} \sum_{t=1}^{\text{LZR}} (x_t - \bar{x})^2$$

4.3. Programme des Segmentes TRANSF

Die Programme des Segmentes TRANSF führen die unter 4.1. und 4.2. beschriebenen Aktivitäten durch. Das Segment enthält ein Programm namens XVEKFM, welches die transformierten Daten in Vektorform ausgibt. Folgende Variable sind nach einem Aufruf

```
CALL XVEKFM
```

bestimmt:

Commonbereich /HAUPT/

```
PREVK ..... Integer, Bereich der Klassencodes
NAMEVK ..... Bereich der Variablennamen
STRING ..... Bereich der Transformationsangaben
LAG ..... Bereich der Verzögerungen
X ..... Bereich der Zeitreihen
```

Commonbereich /NEC/

```
NEND ..... Anzahl der Variablen
LZR ..... Länge der Zeitreihen (Residuenlänge)
JAFORM ..... FALSE (d. h. in Originalform)
```

Commonbereich /ERR/

```
PREFIX .....FALSE . Klassencodes werden vorerst
                    nicht ausgedruckt.
ZEILE .....FALSE . Vorerst erscheint keine Hilfszeile
DEMAND .....TRUE . Demandbetrieb
                    .FALSE . Batchbetrieb
```

Nach der Durchführung eines Segmentes TRANSF bzw. nach dem Einlesen der Daten sind also die Zeitreihen für ihre weitere Verwendung in UNIRUN vollständig aufbereitet.

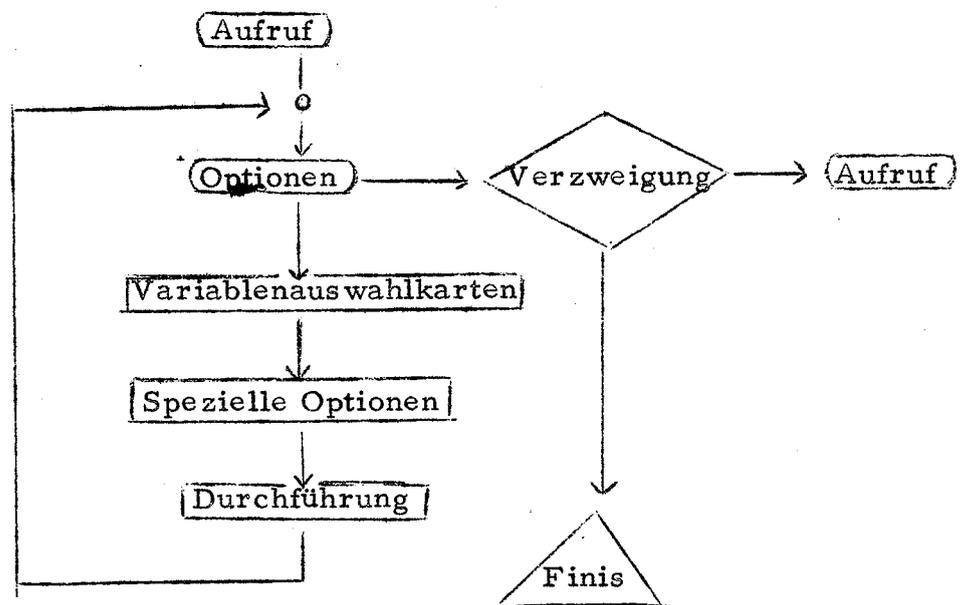
Der Vorteil der linearen Speicherung der Zeitreihen muß mit einem etwas größeren Aufwand an Programmstatements erkaufte werden. Außerdem ist die Anzahl der Zeitreihen begrenzt (siehe 5.6.3.). Deshalb wird derzeit ein ProgrammZMATFM entworfen, welches die gleiche Information liefert wie YVEKFM, nur daß statt des eindimensionalen Bereiches X ein zweidimensionaler Bereich Z auftritt. Der Nachteil dieser Speicherung liegt in einem weit größeren, im Normalfall nicht ausgenützten Speicherbedarf.

5. Programmtypen und Kontrollsprache

5.1. Begriffe

Alle Programme, deren Name mit den charakteristischen drei Buchstaben 'UNI' beginnt, besitzen eine einheitliche Struktur. Auf diese Weise wird dem Benutzer die Handhabung der Kontrollsprache wesentlich erleichtert. Unter Kontrollsprache werden im folgendem vor allem Optionen und Variablenauswahlkarten verstanden, aber auch im erweiterten Sinn beliebiger Kartenbilder bzw. Datenmengen, sogenannte "spezielle Optionen".

Die folgende Abbildung zeigt die verfügbaren Anweisungen und den Ablauf der Kontrollsprache, deren Begriffe sofort erläutert werden:



Aufruf

Darunter wird die Eingabe eines zugelassenen Programmnamens verstanden (z. B. UNICOM). Nach dem Aufruf werden Standardvereinbarungen für die Optionen getroffen.

Optionen

Optionen sind eine geschlossenen Folge von Kontrollanweisungen NAME = WERT (näheres siehe 6.) Optionen können gänzlich entfallen, an ihre Stelle wird dann die nächstfolgende Kontrollanweisung gesetzt. Wenn eine

Option entfällt, gilt der zuletzt im Segment gültige Wert, also frühestens der Standardwert.

Variablenauswahlkarte Diese Karte folgt auf die Optionen oder tritt an ihre Stelle. Sie besteht aus einer Kombination von Blanks und Einsen. Die Einsen stehen in jenen Spalten, deren Nummern den Nummern ausgewählter (erklärender) Variabler entsprechen. Die letzteren Nummern werden in TRANS den Variablen in der Reihenfolge ihres Einlesens zugeteilt.

Formal geschieht also eine Auswahl:

$$M \longrightarrow E \quad \text{bzw.} \quad \left[x_{it} \right] \quad \left[x_{n(j) t} \right]$$

mit $i = 1, 2, \dots, m$ und $j = 1, 2, \dots, k$.

Spezielle Optionen

folgen auf Optionen oder auf eine Variablenauswahlkarte oder treten an deren Stelle. Manche Segmente erfordern spezielle Kontrollanweisungen, für die sich keine allgemeinen Richtlinien angeben lassen. Auch sonstige Datenmengen fallen unter diesen Begriff.

Durchführung

An dieser Stelle beginnt das Programm mit der Durchführung. Nach der Ausgabe der Ergebnisse kehrt es an die Stelle zurück, wo von neuem Optionen eingegeben werden.

Verzweigung

Eine Karte, welche an die Stelle von Optionen tritt, bestimmt den weiteren Ablauf des Programmes:

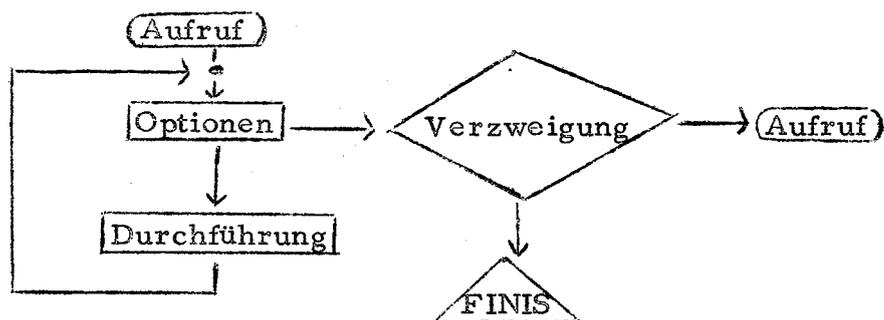
'FINIS' : Das Programm wird beendet.

'UNI...': Es wird zum Segment mit demselben Namen verzweigt (also auch zum eben benützten, wenn nötig).

Die Segmente von UNIRUN gebrauchen die gebotenen Möglichkeiten der Kontrollsprache in verschiedener Weise. Die folgende Einteilung nach drei Programmtypen stellt jedoch eine vollständige Untergliederung dieser Möglichkeiten dar. Die Programme in den Segmenten, welche die Kontrollsprache entziffern, berücksichtigen die Einteilung in drei Typen in entsprechender Weise.

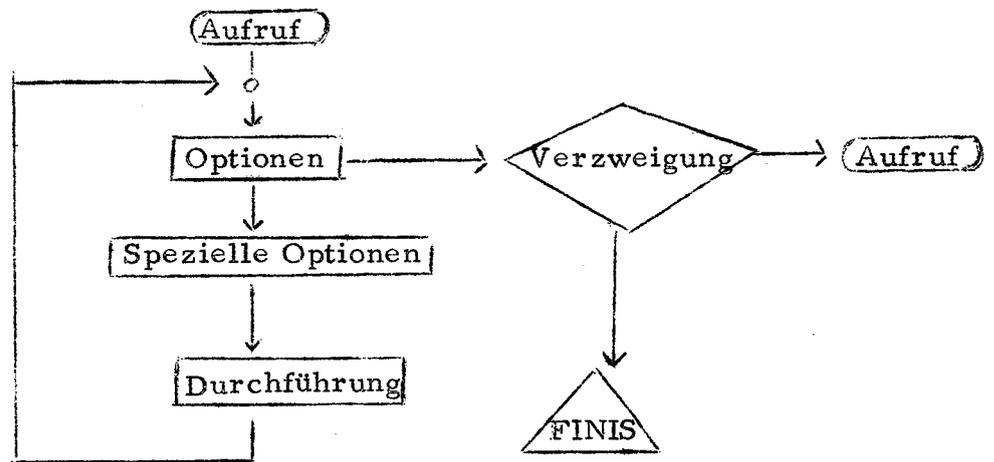
5.2. Programme von Typ 1 (UNICOM)

Unter diesen Typ fallen alle Segmente, die nur Optionen versehen:



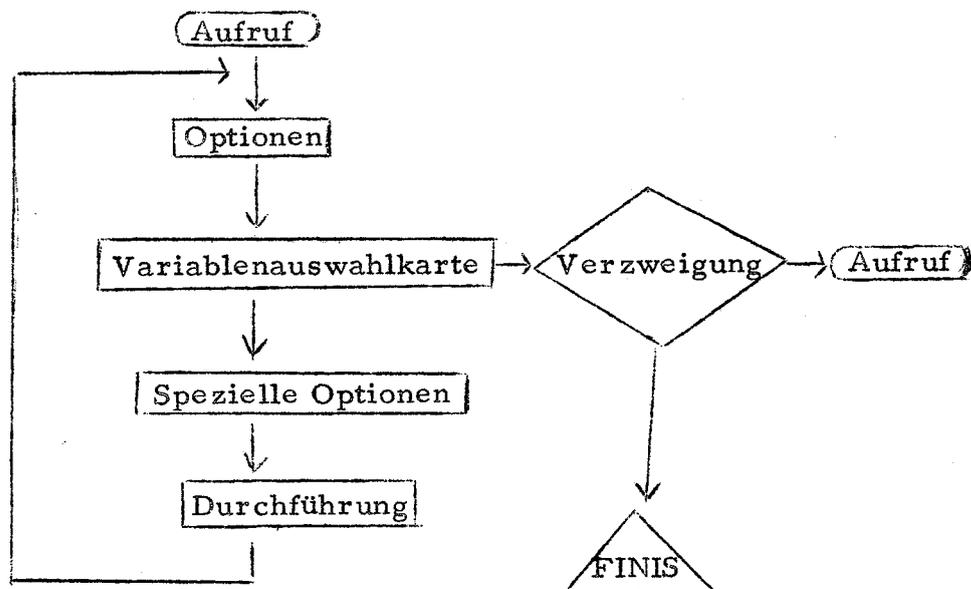
Nach den Optionen erfolgt unmittelbar eine Durchführung.
Schließt der Benützer an den Aufruf die nächste Verzweigung an,
wird eine Durchführung mit Standardvereinbarungen gerechnet.

5.3. Programme von Typ2 (UNIMAX)



Bei Programmen von Typ 2 muß nach den Optionen (oder, wenn keine vorhanden sind, nach dem Aufruf) eine Karte folgen, welche als erste Karte der speziellen Optionen kennzeichnend ist (z. B. 'FIX' bei UNIMAX). Spezielle Optionen haben normalerweise nicht die Gestalt der Optionen und sind verbindlich, d. h. sie dürfen nicht übersprungen werden.

5.4. Programme vom Typ 3 (UNIOLS, UNILAG, UNIPAG)



Die Variablenauswahlkarte nach den oder anstelle der Optionen ist verbindlich. Sind wie UNIOLS keine speziellen Optionen vorgesehen, dann können die Variablenauswahlkarten unmittelbar aufeinander folgen. In UNILAG und UNIPAG sind spezielle Optionen vorhanden und müssen berücksichtigt werden. Sie folgen auf jeweils eine Variablenauswahlkarte.

Der Ablauf der Kontrollsprache wird anhand konkreter Beispiele in 7. nochmals erläutert.

5.5. Fehlernachrichten

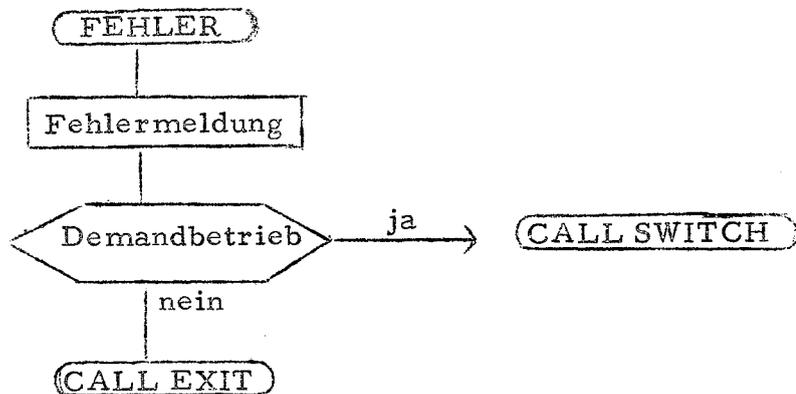
Für das Programmsystem ist eine ausführliche Fehldiagnose vorgesehen, deren Meldungen im Klartext erscheinen. Die letzte Eingabe, welche den Fehler verursacht hat, wird nach der Fehlermeldung ausgegeben.

Beim Batch-betrieb wird das Programm anschließend abgebrochen. Beim Demand-betrieb wird das zuletzt benützte Segment abgebrochen, und das Segment FEHLER aufgerufen. Nach der Fehlerdiagnose kann der Benutzer das Programm mit 'FINIS' beenden oder mit dem Aufruf eines Segmentes fortfahren.

Wenn bei der Eingabe der Kontrollsprache ein Fehler auftritt, wird der Integerwert RCODE im Commonbereich /ERR/mit einer Codezahl definiert, die in FEHLER verarbeitet wird. FEHLER wird einfach aufgerufen:

CALL FEHLER

Das Flußdiagramm von FEHLER läßt sich darstellen (siehe auch 5.6.2.):



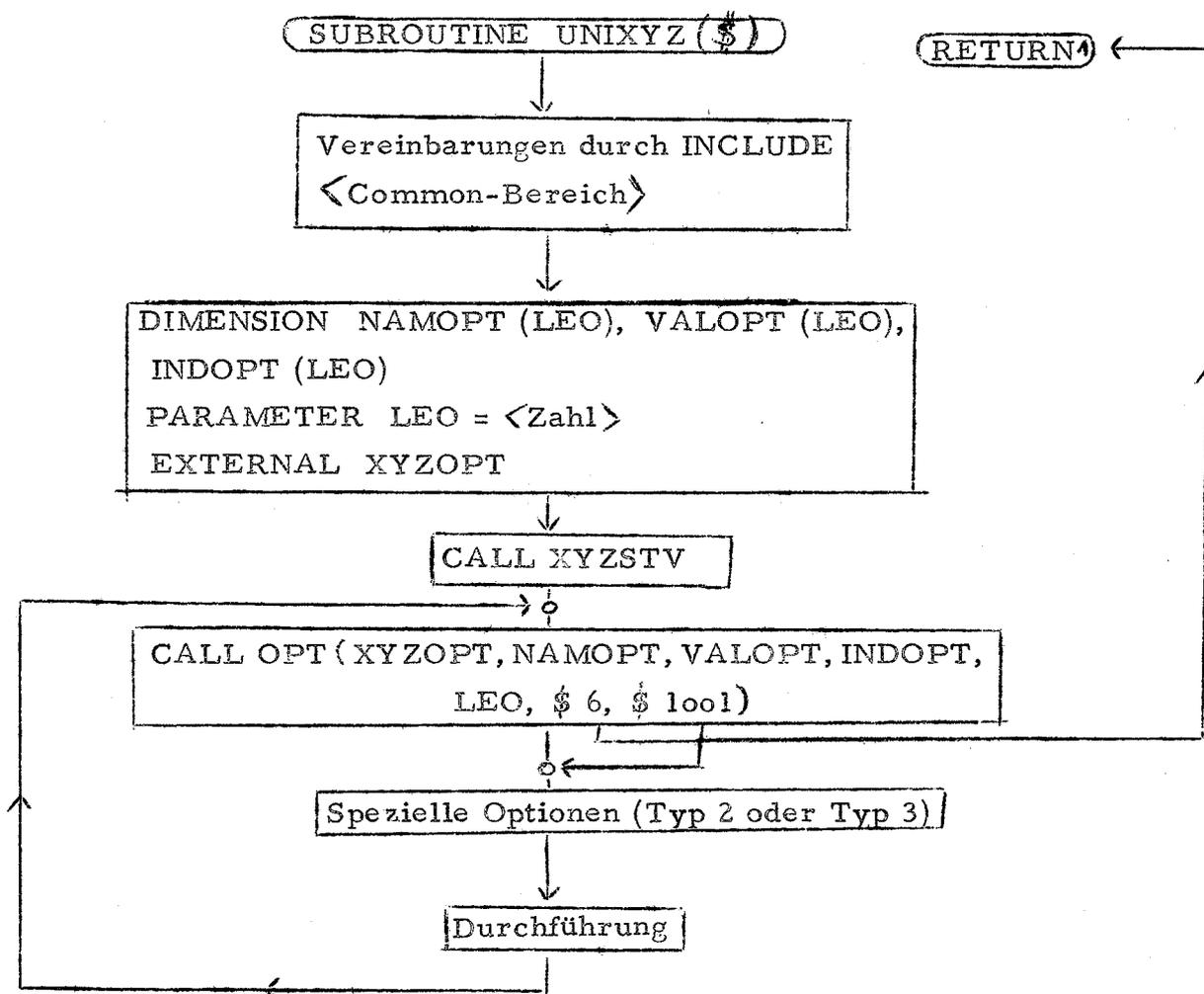
5.6. Details der Programmstruktur

Der folgende Abschnitt dient der Erläuterung programmtechnischer Details, soweit sie allen Segmenten gemeinsam sind. Die Darstellung konzentriert sich vor allem auf die Programme, die den Ablauf der Kontrollsprache steuern.

5.6.1. Flußdiagramm

Das Flußdiagramm in 5.1. "Begriffe" diene der Erläuterung der Abfolge von Anweisungen der Kontrollsprache. Es folgt nun ein ganz ähnliches Flußdiagramm, welches die Unterprogramme und Variablen darstellt, welche die Information der Kontrollsprache verarbeiten.

Das Flußdiagramm wird an hand eines formalen Segmentes 'UNIXYZ' dargestellt, wobei der Teilstring 'XYZ' im konkreten Fall durch eine gegebene Buchstabenfolge zu ersetzen ist, z. B. 'COM' für das Segment 'UNICOM'. Entsprechendes gilt für die Unterprogramme XYZSTV und XYZOPT. Wichtig ist dabei vor allem, daß die Struktur und die Argumentenliste der genannten Subroutinen in allen Segmenten gleich bleiben.



Erläuterungen

a) UNIXYZ wird in SWITCH durch

```
CALL UNIXYZ ($1)
```

aufgerufen (siehe 5.6.2.). Im Normalfall (außer bei Fehlern) wird UNIXYZ über diese Sprunganweisung wieder verlassen.

b) Durch das INCLUDE - Statement werden die benötigten Commonbereiche in UNIXYZ eingegliedert (siehe 3.2.).

c) NAMOPT, VALOPT und INDOPT sind Bereiche mit der Dimension LEO, die Optionsnamen, Optionswerte und Angaben über das Auftreten von Optionszuweisungen enthalten (siehe 6.4., 6.5.).

d) XYZSTV und XYZOPT sind Unterprogramme, die Standardvereinbarungen für Optionen treffen bzw. zur Verarbeitung von Optionslisten dienen (ebenda 6.4, 6.5).

Die Anweisung EXTERNAL XYZOPT kennzeichnet das Argument XYZOPT in OPT als Entry-Point eines Unterprogrammes.

e) OPT ist das zentrale Steuerprogramm jedes Segmentes (siehe 5.6.4.). Es liefert die Unterscheidung nach Programmtypen zusammen mit seinem jeweiligen Unterprogramm XYZOPT. Es wird also entschieden, ob nach Optionen unmittelbar die Durchführung anschließt (Typ 1), ob noch spezielle Optionen zu erwerten sind (Typ 2), oder ob eine Variablenauswahlkarte folgt, an welche sich eventuell, je nach Programm, noch spezielle Optionen anschließen (Typ 3).

f) Die Durchführung hängt vom jeweiligen Inhalt des Segmentes ab. Nach der Durchführung wird stets wieder OPT aufgerufen.

Um im Programmablauf die einwandfreie Verarbeitung der Kontrollsprache zu gewährleisten, wird jede Eingabe (außer bei speziellen Optionen) im A-Format auf einen Lesebuffer übertragen, der dann

nach verschiedenen Inhalten getestet wird. Eine mitgeführte logische Größe RR gibt jeweils an, ob der Inhalt des Lesebuffers noch entziffert werden muß oder ob ein neues Kartenbild in den Lesebuffer eingelesen werden soll.

5.6.2. Das Verzweigungsprogramm SWITCH

Das Verzweigungsprogramm SWITCH bildet den zentralen Kundenpunkt des Systems. Von hier aus werden die einzelnen Segmente des Systems aufgerufen. Der Aufruf von SWITCH erfolgt einfach mit

CALL SWITCH

Das Unterprogramm testet den Inhalt des Lesebuffers. Entspricht dieser Inhalt dem String¹FINIS¹ oder dem String eines vereinbarten Programmnamens, dann wird das Programm abgebrochen bzw. das entsprechende Segment aufgerufen:

CALL UNIXYZ (§ 1)

Von UNIXYZ führt im Normalfall über die Sprunganweisung § 1 der Weg wieder an jene Stelle in SWITCH zurück, wo von neuem der Inhalt des Lesebuffers abgefragt wird.

5.6.3. Die Variablenauswahl

Während der Durchführung von UNIRUN können aus der gegebenen Grundmenge M der Variablen beliebige Teilmengen E ausgewählt werden. Derzeit handelt es sich bei E um Mengen von erklärenden Variablen, die jeweils einer endogenen Variablen gegenübergestellt werden. Es ist jedoch dieselbe Art der Auswahl ohne die Einteilung in erklärende und endogene Variable möglich.

Die Auswahl der Variablen erfolgt mit der Variablenauswahlkarte, die in 5.1. bereits definiert wurde. Wenn die Grundmenge M der Variablen nicht zu groß ist, erscheint diese Art der Auswahl am vorteilhaftesten. Die maximale Anzahl der Variablen ist als Systemstandardwert festgelegt. Im Demandbetrieb bietet die Eingabe der Variablenauswahlkarte Schwierigkeiten, sobald mehr als etwa 10 Variable vorhanden sind. Mit Hilfe der Optionen ZEILE erscheint jedoch vor jeder Eingabe eine Hilfszeile auf dem Bildschirm, welche das Eintippen auf den richtigen Positionen mühelos gestattet.

Das Unterprogramm, welches eine Variablenauswahlkarte testet, heißt VAATST. Es wird aufgerufen mit

```
CALL VAATST
```

Wenn der Inhalt des Lesebuffers keine Kombination von Blanks und Einsern darstellt, wird die logische Größe VARAUS = . FALSE . gesetzt. Sonst gilt VARAUS = . TRUE . und die Kombination wird mit Nullen und Einsern auf den Variablenauswahlvektor IWAHL übertragen.

Die Bereiche NUMMER und IPL (siehe 3.3.) werden dann im Unterprogramm VARKOM je nach dem Inhalt von IWAHL bestimmt:

```
CALL VARKOM ( $, MIN, MAX, NONEND)
```

In VARKOM wird der Variablenauswahlvektor auf Zulässigkeit getestet. MIN und MAX sind Schranken für die minimale und maximale Anzahl ausgewählter Variabler bei kombinatorischen Programmen. Wenn diese Bedingungen verletzt sind, wird zur Sprunganweisung \$ verzweigt.

Bei kombinatorischen Programmen entfällt die Eingabe von Variablenauswahlkarten, es werden nur die Variablenauswahlvektoren kombiniert. Daher wird verständlich, weshalb die Variablenauswahl in zwei Schritten über VAATST und VARKOM erfolgt. Die Größe NONEND wird bei kombinatorischen Programmen 1, sonst \emptyset gesetzt (näheres siehe Dokumentation).

Entsprechend der Anzahl von Spalten pro Lochkarte dürfen in der Grundmenge M niemals mehr als 80 Variable enthalten sein, weil sonst die Methode der Variablenauswahl versagt. Für simultane Schätzverfahren und andere ökonomische Methoden müssen jedoch mehr als 80 Variable zulässig sein:

Für solche Verfahren werden die Zeitreihen in Matrixform gespeichert und die Variablenauswahlkarte enthält in Klammern die Nummern der ausgewählten Variablen, etwa

(1, 17, 19, 123)

Die angegebenen Nummern sind bereits die Werte des Bereichs NUMMER. Der Bereich IPL entfällt bei Matrixspeicherung. Ein Programm zur Interpretation der obigen Variablenauswahlkarte ist derzeit in Arbeit.

5.6.4. Das Steuerprogramm OPT

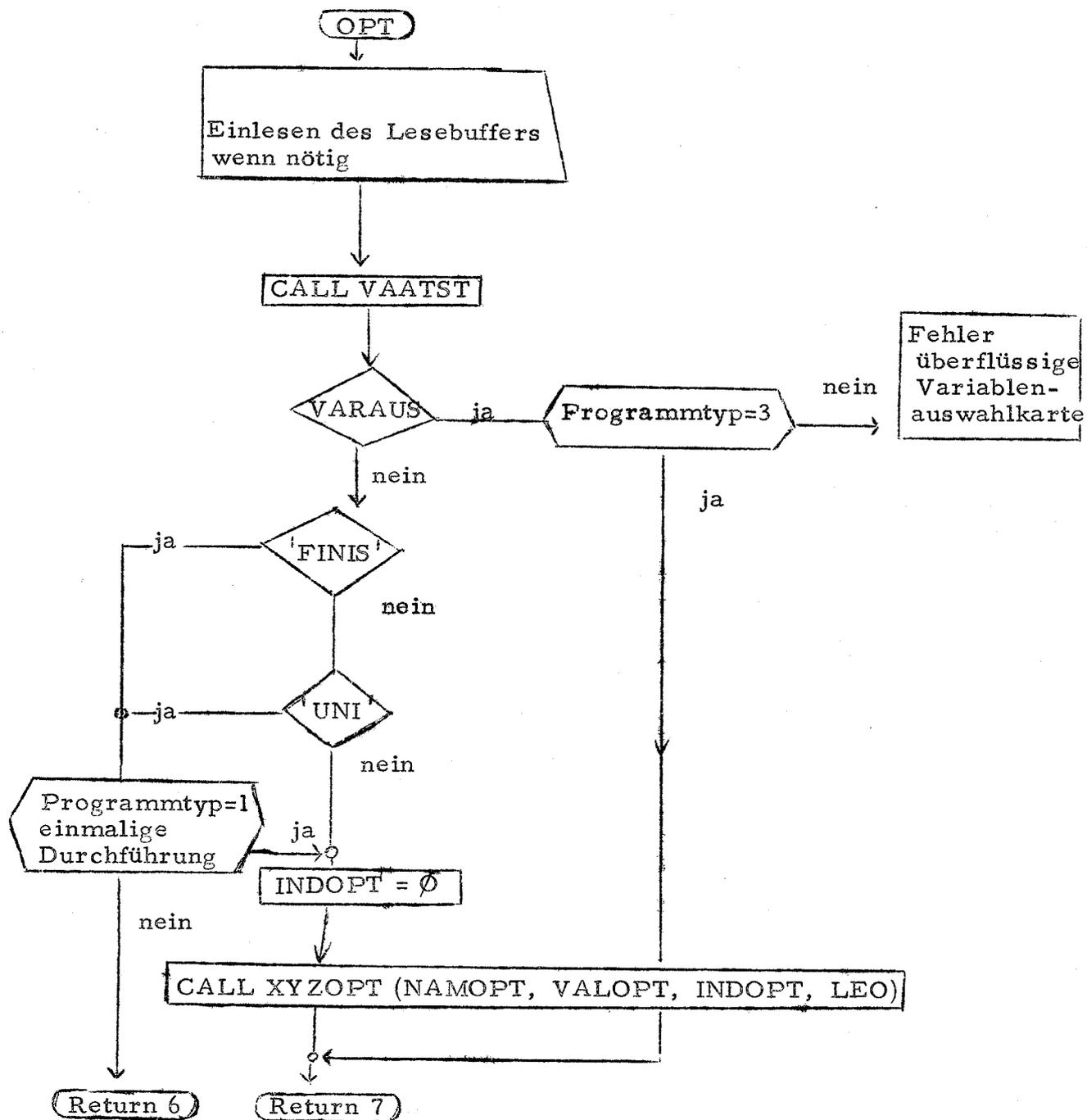
Das Unterprogramm OPT regelt den Ablauf innerhalb eines Segmentes von UNIRUN (Nebenbei sei hier noch einmal erwähnt, das Segment hier das Unterprogramm UNIXYZ mit seinen untergeordneten Programmen bedeutet). Somit enthält jedes UNIXYZ einen Aufruf der Gestalt

```
CALL OPT (XYZOPT, NAMOPT, VALOPT, INDOPT, LEO,
          § 6, § 1001)
```

XYZOPT ist das formale Argument einer external zu bestimmenden Subroutine (6.4.) LEO ist die Dimension der Bereiche NAMOPT, VALOPT und INDOPT. NAMOPT ist der Bereich der Optionsnamen. VALOPT ist ein transienter Bereich von Optionswerten, und INDOPT ist ein Ausgabebereich, der das Auftreten einer Optionszuweisung anzeigt (siehe 6.5.).

Die Sprunganweisung § 6 führt aus OPT über UNIXYZ wieder nach SWITCH zurück (siehe oben 5.6.1). Das Segment UNIXYZ wird also im Normalfall immer auf diese Weise verlassen, sobald eine Karte mit 'FINIS' oder 'UNI'....' eingegeben wurde.

Die Sprunganweisung § 1001 leitet zur Durchführung oder zum Einlesen spezieller Optionen über.



Das Steuerprogramm OPT liefert eine eindeutige Aufteilung in die unter 5.2, 5.3 und 5.4. beschriebenen Programmtypen:

Programmtyp 1: Auch wenn keine Optionen angegeben werden (die in XYZOPT interpretiert werden), ist für eine einmalige Durchführung des Programmes gesorgt (Die Rückkehr in XYZOPT erfolgt nur wegen des Ausdrucks einer Überschrift).

Programmtyp 2: In XYZOPT muß der Lesebuffer nach der ersten Anweisung spezieller Optionen abgefragt werden.

Programmtyp 3: Wie leicht zu sehen ist, können Variablenauswahlkarten beliebig aufeinander folgen. Wenn es auch spezielle Optionen gibt, muß im entsprechenden Unterprogramm für die Korrektheit der Eingaben Sorge getragen werden.

6. Optionen

6.1. Syntax der Optionen

Optionen dienen zur Steuerung der Durchführung von Programmen. Die Syntax der Optionen bleibt im ganzen System dieselbe und unterscheidet sich deshalb vom Aufbau der speziellen Optionen, die im jeweiligen Programm definiert werden.

Optionen bestehen aus einer Folge einzelner Optionsangaben. Eine Optionsangabe hat die Gestalt

$$\text{OPTNAME} = \text{WERT}$$

Als Werte sind erlaubt:

Typ i: Integergrößen, z. B.

$$\text{OPTNAME} = 2$$

Typ r: Realgrößen, z. B.

$$\text{OPTNAME} = 1.1$$

$$\text{OPTNAME} = -3\text{E}-4$$

Typ L: Logische Größen,

$$\text{OPTNAME} = \text{T} \text{ oder}$$

$$\text{OPTNAME} = \text{F.}$$

Eine Option vom Typ L wird widerrufen, indem man $\text{OPTNAME} = \text{F}$ setzt. Der Optionsname kann aus 12 alphanumerischen Zeichen bestehen, von denen jedoch nur die ersten 6 signifikant sind. Der Wert darf aus maximal 16 Stellen bestehen. Innerhalb einer Optionsangabe dürfen beliebig Blanks stehen.

Aufeinanderfolgende Optionsangaben gehören zur selben Folge von Optionen, wenn sie entweder durch Beistriche derselben Karte getrennt werden.

$$\text{OP } 1 = 14, \quad \text{OP } 2 = 6.3.$$

oder durch einen Strichpunkt auf verschiedenen Karten getrennt werden:

$$\text{OP } 1 = 14;$$

$$\text{OP } 2 = 6.3$$

Eine Folge von Optionen endet, wenn nach dem letzten Wert das Kartenende vorliegt oder ein Blank belassen wird. Daher müssen die Trennzeichen ' , ' und ' ; ' unmittelbar auf den letzten Wert folgen. Eine Optionsangabe der Form

$$OP\ 3 =$$

4

ist nicht erlaubt.

Die folgenden Steuerkarten stellen zwei verschiedene Optionsfolgen dar:

A = 3,	B = 6.2
C = 4;	
D = -5.E2	

Einzelne Optionen behalten ihren Wert solange bei, bis sie in einer Optionsfolge neue Werte erhalten. Ohne Wertzuweisung gilt für sie ein Standardwert. Im letzten Beispiel wird eine Durchführung mit $A = 3$, $B = 6.2$ und Standardwerten für C und D gerechnet, die nächste Durchführung erfolgt mit $A = 3$, $B = 6.2$, $C = 4$ und $D = -5.E2$.

Standardwerte werden jeweils nach dem Beginn eines Segmentes gesetzt. Mit zwei Ausnahmen (PREFIX und ZEILE, siehe 6.3) ist der Gültigkeitsbereich einer Option mit dem zugehörigen Segment identisch. Beim Aufruf eines neuen Segmentes werden auch bei gleichlautenden Optionen wieder Standardwerte gesetzt.

6.2. Tafel der Optionen

Name	Typ	Standardwert	UNICOM	UNIMAX	UNIOLS	UNILAG/PAG
PREFIX	L	F	x	x	x	x
ZEILE	L	F	x	x	x	x
ENDOGEN	i	nend	x	x	x	x
LISTE	L	F	x	x	x	x
NEUOPT	L	F	x	x	x	x
MIN	i	1	x	x		
MAX	i	nend - 1	x	x		
SRSQ	r	∅	x	x		x
SPROZ	r	9.E25	x	x		x
SA	r	9.E25	x	x		x
LIMITE	L	F	x	x		x
TEST	L	F	x	x		x
KONST	L	F	x	x	x	
RESIDUEN	L	F			x	x
FIX	S			x		
VAR	S			x		
END	S			x		
PERIODE	i					x
FIXLAG	S					x
VARLAG	S					x
ENDLAG	S					x

Mit "S" werden spezielle Optionen bezeichnet.

"nend" ist die Anzahl der Zeitreihen (mit dem m von 2. identisch).

6.3. Funktionsweise

PREFIX = T :

Die Klassencodes der Variablen werden mit ausgedruckt. Die Angabe PREFIX = T gilt über die Grenzen der Segmente hinweg.

ZEILE = T:

Hat nur für den Demandbetrieb Bedeutung. Es wird vor jeder Eingabe eine Hilfszeile ausgegeben:

5 0 5 0 5 0

Sie dient zur Orientierung und als Behelf bei der Eingabe der Variablenauswahlkarte. Auch ZEILE ist eine transiente Option wie PREFIX. Beide Optionen können allerdings erst unmittelbar nach dem ersten Aufruf eingegeben werden und Zeile ist erst bei der nächsten Eingabe wirksam.

ENDOGEN = n

Die Variable n ist die erklärende Variable. Im Normalfall gilt ENDOGEN = nend, d. h. die letzte eingegebene Zeitreihe dient als endogene Variable.

LISTE = T:

Nur für den Demandbetrieb sinnvoll und bewirkt eine einmalige Ausgabe der Liste der Variablennamen.

NEUOPT = T

Wenn diese Option gesetzt wird, wird die zugehörige Optionsliste interpretiert, und es ist die Angabe von weiteren Optionen noch vor der Durchführung möglich. NEUOPT wird automatisch sofort wieder F gesetzt und ist derzeit nur in Verbindung mit LISTE sinnvoll.

MIN = n

Minimale Anzahl erklärender Variabler bei kombinatorischen Programmen.

MAX = n

maximale Anzahl erklärender Variabler bei kombinatorischen Programmen. Falls nend ≥ 21 übersteigt, wird 20 als Standardwert genommen.

SRSQ = r	minimaler Regressionskoeffizient
SPROZ = r	maximaler Prozentwert der Standardabweichung
SA = r	maximaler Absolutbetrag der Konstanten.
	SRSQ, SPROZ und SA sind Schrankenbedingungen, die sinnlosen Ausdruck bei kombinatorischen Programmen verhindern.
LIMITE = T	Bewirkt die Annahme der Schranken SRSQ = .7, SPROZ = 8 ϕ , SA = 1 ϕ .
TEST = T	Bewirkt die Annahme der Schranken SRSQ = .1, SPROZ = 7 $\phi\phi$, SA = 1 ϕ .
KONST = T	Die Konstante wird unterdrückt.
RESIDUEN = T	Zusätzlicher Ausdruck von Residuen, beobachteten und errechneten Wert, sowie zwei Werten, für die Fehlerstreuung: unverz. Residuenstreuung = $\sqrt{\sum e_t^2 / T-k-1}$ normale Residuenstreuung = $\sqrt{\sum e_t^2 / T}$ (siehe (2.1.8.) und (2.1.14))
FIX, VAR und END	siehe UNIMAX
PERIODE = n	n ist die Schrittweite zur Bildung variabler Lags, normalerweise 1, 2, 4, 12 (entspricht Jahreslag, Semestern, Quartalen und Monaten).
FIXLAG, VARLAG, ENDLAG	siehe UNILAG.

6.4. Die Programme XYZSTV und XYZOPT

Wie bereits erwähnt, dienen die Programme XYZSTV und XYZOPT für Standardvereinbarungen bzw. für die Interpretation von Optionslisten.

In XYZSTV werden die Optionsnamen und die Standardwerte der Optionsvariablen bestimmt. Die Optionsnamen werden in bestimmter Reihenfolge auf dem Bereich NAMOPT gespeichert, und die zugehörigen Optionsvariablen erhalten einen Standardwert, z. B.

```
NAMOPT (1) = 'NEUOPT'  
NEUOPT = .FALSE.  
NAMOPT (2) = 'MIN'  
MIN = 1
```

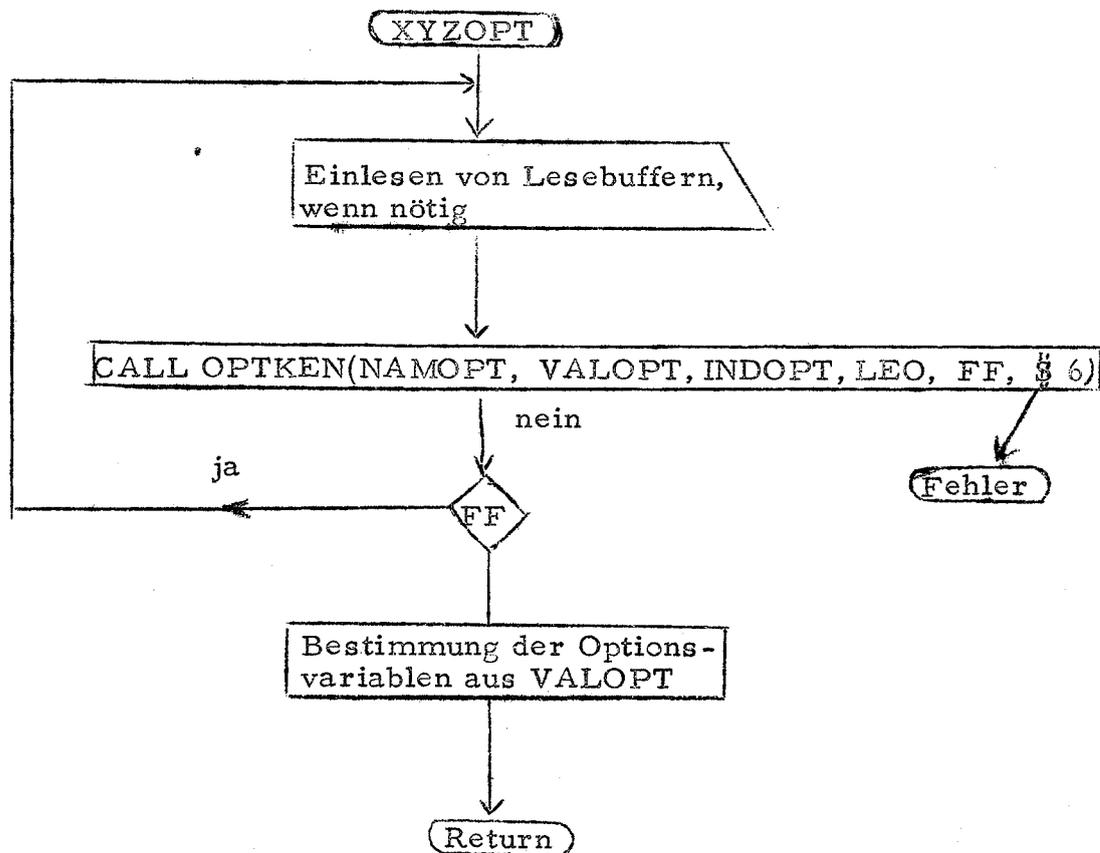
Nachher wird in derselben Reihenfolge der Bereich VALOPT mit Standardwerten definiert:

```
VALOPT (1) = Ø  
VALOPT (2) = MIN
```

Logische Größen werden auf VALOPT mit Ø und 1 gespeichert. Integer-Optionsvariable werden auf VALOPT zu Real-größen. Das Programm XYZSTV wird aufgerufen mit

```
CALL XYZSTV (NAMOPT, VALOPT, LEO)
```

In XYZOPT werden dann Optionslisten verarbeitet.



Die neu bestimmten Optionswerte werden in OPTKEN dem Bereich VALOPT zugeteilt. Nachher werden die Optionsvariablen nach den neuen Werten von VALOPT bestimmt.

XYZOPT wird aufgerufen mit

```
CALL XYZOPT (NAMOPT, VALOPT, INDOPT, LEO)
```

Das Unterprogramm OPTKEN wird im folgenden Abschnitt erläutert.

6.5. OPTKEN

Das Programm OPTKEN dient zur Interpretation einer Optionsliste. Eine solche Optionsliste entspricht ungefähr einer FORTRAN - V "NAMELIST" - Eingabe, mit dem Vorteil, daß die Eingabe durch OPTKEN wesentlich vereinfacht ist. Voraussetzung dafür ist allerdings der eindeutige Ablauf der Kontrollsprache, wie er in 5. beschrieben wurde.

Das Programm OPTKEN wird mir folgendem Statement aufgerufen:

```
CALL OPTKEN (NAMOPT, VALOPT, INDOPT, LEO, FF, § 6)
```

LEO ist die Dimension der Bereiche NAMOPT, VALOPT, und INDOPT. Auf NAMOPT sind die Namen der Optionen als Strings in einer bestimmten Reihenfolge eingetragen. Auf VALOPT befinden sich die zuletzt gültigen Optionswerte in derselben Anordnung. INDOPT kann beliebig Null und Eins definiert sein (es wird vorher in OPT Null gesetzt).

In OPTKEN wird der Bereich LESBUF nach Ausdrücken der Form

```
NAME = WERT
```

abgetestet. Wenn kein '=' auftritt, erfolgt ein RETURN mit der Annahme, daß es sich nicht um eine Optionsliste handelt. Auch wenn NAME oder WERT mehr als 16 Charakter aufweisen, geschieht derselbe RETURN. Zur Sprunganweisung RETURN 6 wird dann verzweigt, wenn die Optionsliste eindeutig fehlerhaft ist. Das ist immer dann der Fall, wenn NAME keinen vereinbarten String darstellt oder wenn auf '=' kein WERT folgt.

Die logischen Werte werden auf VALOPT mit 0 und 1 je nach F oder T übertragen. Die Zuweisung numerischer Werte erfolgt mit freiem Format. Integer-Werte werden auf VALOPT als Realgrößen zwischengespeichert, bis sie in XYZOPT wieder auf entsprechende Integer-Variable übertragen werden. Für Real-Größen ist nur einfache Genauigkeit gestattet.

Nach der Übertragung des Wertes wird der entsprechende Wert INDOPT
1 gesetzt.

FF ist eine logische Größe. Wenn FF=,TRUE., zeigt sie an, daß nach
dem Trennzeichen ';' noch eine Optionsliste zu erwarten ist. Es
wird dann OPTKEN mit einem weiteren Lesebuffer erneut aufgerufen
(siehe auch Flußdiagramm in 6.4.).

7. Die Segmente von UNIRUN

7.1. UNICOM

Programm vom Typ 1

Gegeben sei die Menge $M = \{x_{1t}, x_{2t}, \dots, x_{mt}\}$

von transformierten Zeitreihen. UNICOM berechnet alle Kombinationen von Regressionsgleichungen der Gestalt (2.2)

$$y_t = a + \sum_{i=1}^k \beta_i x_{i(t),t} + \varepsilon_t \quad (2.2.)$$

y_t ist im Normalfall die Zeitreihe x_{mt} , durch ENDOGEN = j bestimmt man dazu die Zeitreihe x_{jt} .

Die Konstante α wird mit KONST=T unterdrückt. Die Kombinationen erfolgen mit einem Medulo - 2 Zähler. Ihre Anzahl ist demnach $2^{m-1} - 1$. Diese bei hohem m exorbitante Zahl kann durch die Einschränkung auf Regresseionen gewisser Ordnung mit den Optionen MIN und MAX stark herabgedrückt werden. Ebenso ist es möglich, durch Angabe geeigneter Schranken für R^2 , proz und $\hat{\alpha}$ (siehe 2.1.) nur eine gewissen Auswahl an Regressionen ausdrucken lassen. Dazu dienen die Optionen SRSQ, SPROZ und SA.

Weitere Optionen siehe 6.2, 6.3.

Beispiel 1:

```

FINIS
LIMITE = T
KONST = T, SPROZ = 200, LIMITE = F
LIMITE = T
MIN=1, MAX=6, SPROZ = 200.
UNICOM
  
```

Es werden mit denselben Variablen 4 Durchgänge gerechnet, wobei sich die Durchgänge 1 und 2 von den folgenden nur um die Konstantenunterdrückung unterscheiden.

Beispiel 2:

FINIS
UNICOM

Ein Durchgang wird mit Standardwerten der Optionen gerechnet.

Beispiel 3 :

UNIMAX
ENDOGEN = 3
LISTE = T, NEUOPT = T
UNICOM

Zuerst wird eine Liste der Variablen zur Orientierung ausgegeben, dann werden Kombinationen mit der 3. Zeitreihe als endogener Variabler gerechnet und das Programm mit UNIMAX fortgesetzt.

7.2. UNIMAX

Programm vom Typ 2.

UNIMAX ist wie UNICOM ein kombinatorisches Regressionsprogramm, wobei jedoch die Auswahl der Kombinationen mit Hilfe der speziellen Optionen 'FIX' und 'VAR' gesteuert werden kann. Die Eingabe erfolgt durch geeignete Aufeinanderfolge von 'FIX' und 'VAR' mit anschließenden Variablenauswahlkarten.

Wenn 'FIX' auftritt, bildet es stets den Anfang der speziellen Optionen, die Karte 'END' bildet immer den Abschluß.

Die Auswahl der Kombinationen läßt sich gedanklich nach folgendem Schema darstellen:

FIX	VAR	VAR	VAR	VAR	
					1
					2
					3
					10

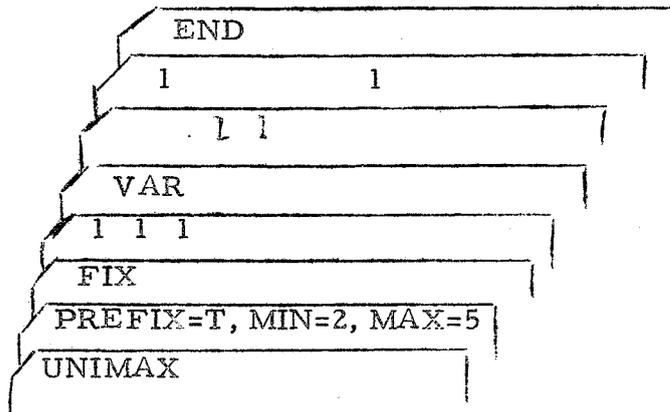
Die Spalten bilden Gruppen von Zeitreihen, die Kästchen bilden deren Untergruppen. Ein Kästchen ist mit der Variablenauswahlkarte gleichbedeutend und darf bis zu 10 Zeitreihen enthalten. Das Schema hat eine fixe und 9 variable Gruppen mit je 10 Untergruppen.

Es wird von links oben nach rechts unten angefüllt, wahlweise kann auf die fixe Gruppe oder die variablen Gruppen ganz verzichtet werden. Zeitreihen können auch mehrmals vorkommen.

Kombiniert werden bei UNIMAX nicht die Zeitreihen, sondern die Kästchen. Fixe Gruppe bedeutet in diesem Zusammenhang, daß jeweils eine ihrer Untergruppen mit den anderen Spalten kombiniert wird. Die variablen Gruppen werden horizontal kombiniert, und ihre Untergruppen werden vertikal substituiert. Wenn Zeitreihen, die in den Gleichungen nicht gleichzeitig auftauchen sollen, in verschiedenen Kästchen derselben Spalte angeordnet werden, beseitigt man auf diese Weise alle unerwünschten Kombinationen von Regressionsgleichungen.

Die normalen Optionen sind genau dieselben wie bei UNICOM.

Beispiel 1



Klassencodes werden ausgedruckt, berechnet werden nur Regressionsgleichungen mit mindestens 2 und höchstens fünf Variablen. Es werden 3 Gleichungen gerechnet nach folgender Formel:

Die Anzahl der angegebenen variablen Gruppen sei $nvar$, die Anzahl der Untergruppen in Spalte i sei $m(i)$. Dann lautet die Anzahl K der Kombinationen

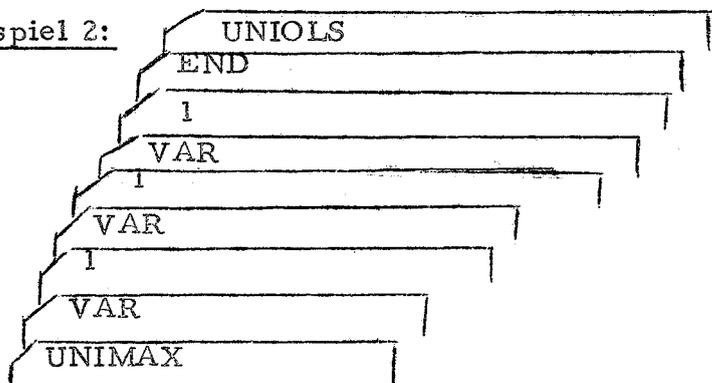
a) wenn keine fixe Gruppe vorhanden:

$$K = \prod_{i=1}^{nvar} (m(i)+1) - 1 \quad (7.2.1.)$$

b) wenn vorhanden

$$K = m(1) \cdot \prod_{i=1}^{nvar} (m(i) + 1) \quad (7.2.2.)$$

Beispiel 2:



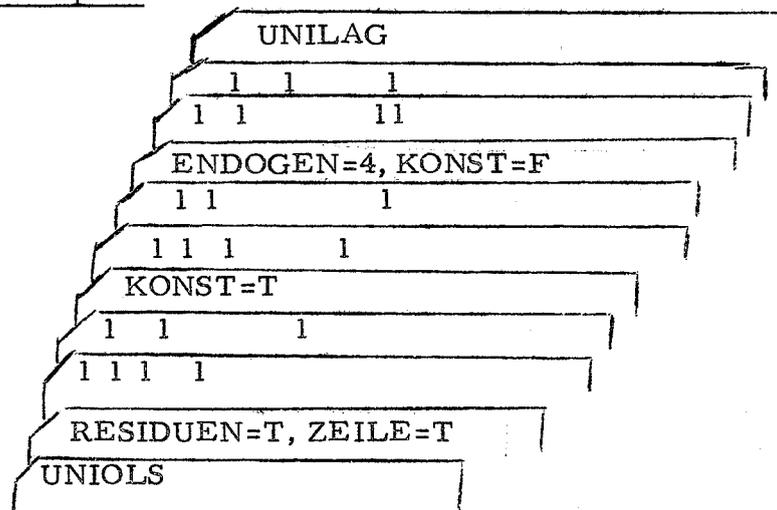
Schema mit 7 Kombinationen.

7.3. UNIOLS

Programm vom Typ 3.

UNIOLS berechnet einzelne Regressionsgleichungen. Konstantenunterdrückung, Ausdruck von Residuen und beliebige Wahl der endogenen Variablen ist möglich. Die Variablenauswahlkarten können unmittelbar aufeinander folgen.

Beispiel:



Residuen werden durchgehend ausgedruckt, und eine Hilfszeile erscheint vor jeder Eingabe auf dem Bildschirm. Später wird die Konstante unterdrückt, und dann wieder berechnet, sobald die Zeitreihe 4 endogen erklärt wird.

7.4. UNILAG und UNIPAG

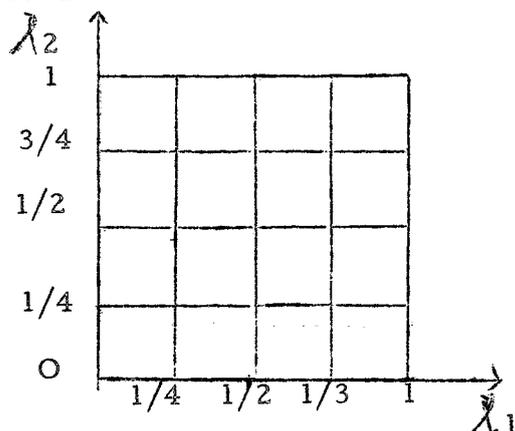
Programm vom Typ 3.

Die Lagprogramme dienen zur Bestimmung von optimalen Lags einer oder mehrerer Variabler, wobei die Lags mit linearer Interpolation (UNILAG) und mit Splineinterpolation (UNIPAG) gebildet werden. Formeln sind in 2.1. zusammengestellt. Hier wird die Methode beschrieben, auf welche Weise die Lagkombinationen hergestellt werden.

Gegeben sei eine Menge $\{x_{n(i)t}\}$ von erklärenden Variablen, $i = 1, 2, \dots, k$ und $t = 1, 2, \dots, T-1$, weil bei den Lagprogrammen die Residuenlänge automatisch um Eins verkürzt wird. Der Einfachheit halber setzen wir $n(i) = i$ und bezeichnen die Variablen aus dieser Menge, die nicht gelagt werden, mit $\{z_{j,t}\}$. Wir unterscheiden dann Variable mit fixem Lag $\mu_i: x_{i,t-\mu_i}$ und Variable mit variablem Lag $\lambda_i: x_{i,t-\lambda_i}$. Sind Variable mit variablem Lag vorhanden, dann suchen die Lagprogramme auf einem vorgebeschriebenen Gitter der λ_i alle Lagkombinationen, mit denen dann die OLS - Gleichungen gebildet werden:

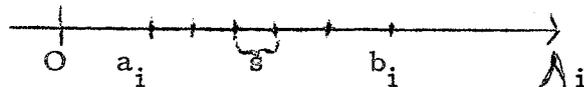
$$y_t = \sum_{i=1}^l \beta_i x_{i,t-\lambda_i} + \sum_{i=l+m}^{l+m} \beta_i x_{i,t-\mu_i} + \sum_{i=l+m+1}^k \beta_i z_{it} + \varepsilon_t \quad (7.4.1.)$$

Die Lagkombinationen erfolgen aus einem l -dimensionalen Gitter, für $l = 2$ also etwa



Das Laggitter kann auf zwei Arten hergestellt werden.

- a) Durch die Angabe PERIODE = p. Dann werden die Lags im Intervall $[0, 1]$ mit p Schritten gesucht. Die Kombinationen erfolgen für alle ausgewählten Variablen mit variablem Lag.
- b) Durch die Angabe der Schrittzahl p und der Intervalle für alle ausgewählten Variablen mit variablem Lag.



$$s = \frac{b_i - a_i}{p}, \quad \text{mit } \lambda_i \in [a_i, b_i]. \quad (7.4.2.)$$

Die Angabe der Lagstruktur erfolgt in Form der speziellen Optionen 'FIXLAG', 'VARLAG' und 'ENDLAG'. Sie folgen auf eine Variablenauswahlkarte, die welche die Gesamtheit der erklärenden Variablen bestimmt. (7.4.1.)

'FIXLAG':

Es werden fixe Lags gebildet. Auf 'FIXLAG' folgt genau eine Variablenauswahlkarte und dann folgen m Karten (siehe 7.4.1.), auf denen im freien Format die Lags λ_i , $i=1, \dots, m$ als Realgrößen angegeben werden.

'VARLAG':

Es werden variable Lags gebildet. Auf 'VARLAG' folgt genau eine Variablenauswahlkarte. Wenn PERIODE = p unter den Optionen aufscheint, ist keine weitere Angabe nötig und die Kombinationen werden automatisch gebildet. Sonst folgen k Karten (7.4.1.), auf denen in freiem Format Schrittzahl als Integergröße und Unter- und Obergrenze der Lags als Realgrößen eingetragen werden.

'ENDLAG':

Diese Eingabe schließt die Lagstruktur ab. 'FIXLAG' oder 'VARLAG' dürfen entfallen, 'ENDLAG' nicht.

Optionen von UNILAG und UNIPAG:

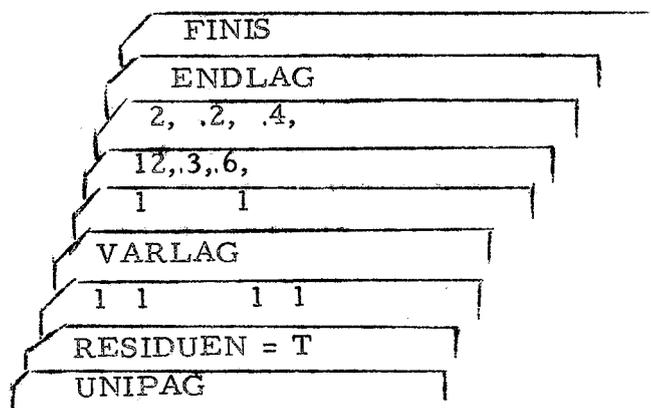
Die Option PERIODE ist schon mehrmals beschrieben worden. Sie wird widerrufen, indem man PERIODE = \emptyset setzt.

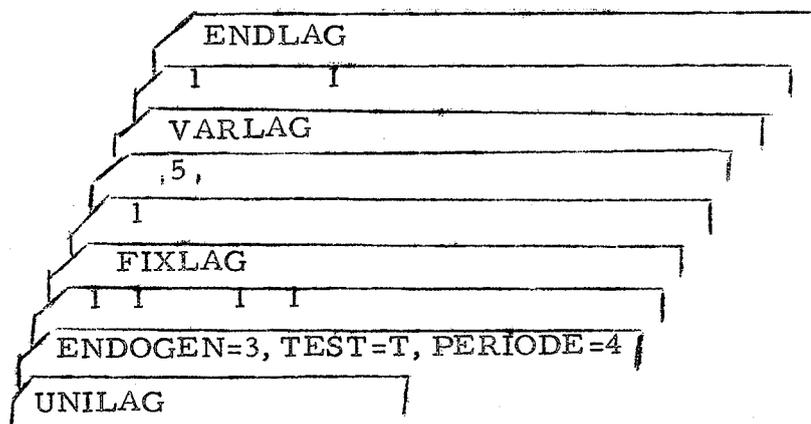
Der Ausdruck der Residuen ist möglich. Bei UNIPAG werden dann zusätzlich die Werte der interpolierten Zeitreihe ausgegeben.

Die Schranken SRSQ, SPROZ, SA, LIMITE und TEST dienen auch hier dazu, den Ausdruck einzuschränken. Besonders die Option TEST ist hier zu empfehlen.

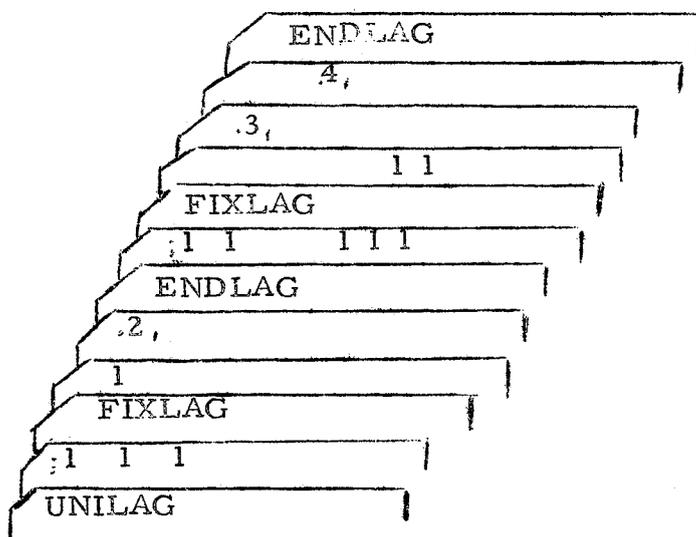
Weitere Optionen siehe wieder 6.2, 6.3.

Beispiel 1



Beispiel 2

Testschranken zur Vermeidung von unnützem Ausdruck, 2 Variablen werden nach optimalen Lags über Quartale getestet.

Beispiel

Aus diesem Beispiel geht hervor, warum den Programmen UNILAG und UNIPAG der Typ 3 zukommt.

8. Anhang

Im Anhang befinden sich Hinweise über Speicherung, Systemstandardwerte und Segmentierung des Programmsystems.

Die verwendete Steuersprache heisst ECL, und die folgende Darstellung bezieht sich daher auf die Verwendung des Systems auf einer UNIVAC-Grossrechenanlage.

Vorausgesetzt wird hier, dass sich das System auf einem File mit dem Namen

Q * FILE.

befindet. Im konkreten Fall sind dann Qualifier und Filename durchwegs durch die aktuellen Bezeichnungen zu ersetzen.

Der File Q*FILE sollte im read-only Status stehen.

8.1. Aufruf, Systemstandardwerte

Ein RUN, welcher nur einen Aufruf des Systems enthält, hat folgende Gestalt :

① RUN RUN - ID, ACCOUNTN, PROJEKT

② XQT Q * FILE. UNIVEK

[Vorlauf und Variablenkarten mit Formatkarte und Datenkarten]
[Steuerkarten der Kontrollsprache]

③ FIN

Vorlauf und Variablenkarten samt Formatkarte und Datenkarten werden unmittelbar an die ② XQT-Karte angefügt. Es ist jedoch ratsam, diese Karten vor dem Ablauf von UNIRUN in ein spezielles Element einzufügen. Ein solcher RUN hat dann etwa folgende Gestalt :

① RUN RUN- ID , ACCOUNTN, PROJEKT

② ASG, T PRIV.

③ ELT, I PRIV. VARIABLE

	[Vorlauf und Variablenkarten mit Formatkarte und Datenkarten]
④	XQT	Q * FILE. UNIVEK	
④	ADD	PRIV. VARIABLE	
		[Steuerkarten der Kontrollsprache
④	FIN		

Während der Durchführung müssen vom Benutzer einige Systemstandardwerte beachtet werden. Sie lauten :

MAXEND	maximale Anzahl von Zeitreihen
MAXLZR	maximale Zeitreihenlänge
MAXLX	Schranke für das Produkt (Anzahl der Zeitreihen) * (Zeitreihenlänge)
NERKLV	maximale Anzahl erklärender Variabler in einer Gleichung
MAXLXL	Schranken für das Produkt (Anzahl er- klärender Zeitreihen + 1) * (Zeitreihenlänge) in UNILAG und UNIPAG.
MSP	maximale Anzahl der Spalten im Schema von UNIMAX.
MK	maximale Anzahl von Untergruppen pro Spalte.
MVAR	maximale Anzahl von Variablen pro Unter- gruppe.

Die Systemstandardwerte müssen beim Implementieren des Systems als Parameterwerte in die Fortran-Prozeduren COMMONPROC und SEGMENTPROC eingegeben werden (näheres siehe Dokumentation).

8.2. Segmentierung

In diesem Abschnitt wird beschrieben, auf welche Weise das absolute (durchführbare) Element UNIVEK erzeugt wird.

Es sei der File $Q * FILE$. gegeben , wobei ab nun nur mehr FILE. geschrieben wird. Das MAP - Statement lautet zusammen mit dem erforderlichen RUN:

```

① RUN          RUN - ID, ACCOUNTN, Q
② ASGA        FILE.
③ PREP        FILE.
④ MAP,S      FILE . UNIMAP, .UNIVEK
⑤ FIN

```

Das symbolische MAP- Element UNIMAP lautet:

```

LIB          FILE.
SEG          UNIVEK
IN           FILE . UNIRUN
SEG          TRANSFORM,*(UNIVEK)
IN           FILE . XVEKFM
SEG          UNIRF *, (TRANSFORM7
IN           FILE . UNIRF
SEG          SWITCH *, (UNIVEK)
IN           FILE. SWITCH
SEG          UNICOM *, (SWITCH)
IN           FILE. UNICOM
SEG .        UNIMAX * , (SWITCH)
IN           FILE . UNIMAX
SEG          UNIOLS *, (SWITCH)
IN           FILE . UNIOLS
SEG          UNILAG *, (SWITCH)
IN           FILE . UNILAG
SEG          UNIPAG *, (SWITCH)
IN           FILE . UNIPAG
END

```

9 Literaturangaben

- [1] Deutsch Edwin, Das Programmsystem UNIRUN, eine Darstellung seiner Methodik. Forschungsbericht Nr. 68 des Instituts für Höhere Studien, Juni 1972
- [2] Hämmerlin, Numerische Mathematik I
B-I Hochschultaschenbuch Nr. 498a
- [3] Schönfeld Peter, Methoden der Ökonometrie Band I.
Verlag Franz Vahlen GmbH, 1969.