Stefan Baunack
Institut für Festkörper- und Werkstofforschung
Postfach 270016, 01171 Dresden
Tel.: 0351-4659-387 Fax: 0351-4659-452
email: s.baunack@ifw-dresden.de

PCA_MEN

Version 1.1 Release Februar 1999

Allgemeines
PCA_MEN ist ein Programm zur Faktorenanalyse von spektralen Daten, das speziell für die Auswertung von AES- und XPS-Spektren entwickelt wurde. Die verwendete Mathematik basiert überwiegend auf „Factor Analysis in Chemistry“ von Malinowski und Howery [1], die Grundzüge sind im Theorieteil beschrieben.

1. Programmbeschreibung

1.1. Einleitung

Das Programm beruht auf MATLAB. MATLAB ist ein Interpreter für Matrizenrechnung, dessen Programme durch die Abarbeitung von MATLAB-Scripts (Filename *.m) ausgeführt werden. Die MATLAB-Scripts sind ASCII-Files, die vom Nutzer modifiziert werden können. PCA_MEN Version 1.1 wurde unter MATLAB, Version 4.2 realisiert, funktioniert weitgehend auch unter MATLAB 5.1 (außer Funktionen, die Mexfiles (*.DLL) und getbord.m nutzen). Eine MATLAB-Lizenz ist Voraussetzung zur Nutzung von PCA_MEN.
Das Programm verwendet an einigen Stellen auch Routinen, die in der Literatur bzw. den MATLAB Contributions (ftp://ftp.ask.uni-karlsruhe.de/pub/matlab, ftp://ftp.math.uni-hamburg.de/pub/soft/math/matlab, http://www. mathworks.com/ftp/) veröffentlicht wurden.
In der Beschreibung sind die MATLAB-Variablen und -Files als **`variablename`**, MATLAB-Funktionen (Befehle, Scripts, Functions) und Ausgaben ins MATLAB Command Window als `functionname` gekennzeichnet.

1.2. Installation

Das Programm PCA_MEN umfaßt in der Version 1.1 die in Tabelle 1 zusammengestellten Funktionen (MATLAB-Scripts, -Functions, Mexfiles (DLL)), die sich innerhalb des MATLAB Suchpfades befinden müssen (z.B. im Ordner \MATLAB\FAKTOR). Der Order muß durch Editieren von **`matlabrc.m`** in die Variable **`matlabpath`** eingetragen werden. Die DLLs wurden mit Microsoft Visual C++ 1.5 erzeugt und zusammen mit MATLAB, Version 4.2, unter Windows 3.x, Windows95 und Windows NT 4.0 getestet.

1.3. Einstellungen

Das Programm verwendet weitgehend die Standardeinstellungen von MATLAB. Die Farbreihenfolge kann mit dem Befehl `set(0,'DefaultAxesColorOrder')` geändert werden. Es wurde nur Grafik auf schwarzem Hintergrund verwendet, daher wird manchmal die Farbe weiß für Markierungen benutzt.

1) Hintergrundprozesse müssen aktiviert sein (Command Window: Options: enable background process) damit die Maus richtig arbeitet. Es kann Schwierigkeiten mit dem rechten Mausklick geben, wenn die Maustasten mit Befehlen belegt sind (z.B. Logitech-Maus).
2) Das Papierformat wird beim Start von `pca_men` auf A4 eingestellt, in **`startup.m`** sollte das Default-Format auf A4 Querformat gesetzt werden:

```
set(0,'DefaultFigurePaperType','a4letter','DefaultFigurePaperOrientation','landscape')
set(0,'DefaultFigurePaperUnits','centimeters','DefaultFigurePaperPosition',[1 1 28 20])
```

3) Ein Farbdrucker muß in **`printopt.m`** eingestellt werden.
4) Die Textausgabe wurde auf den Command Window Font = Courier New (nichtproportional) abgestimmt.

1.4. Programmablauf

Tabelle 2 enthält eine Zusammenstellung der vom Programm benutzten Variablen.
Das Programm zur Faktoranalyse wird am MATLAB-Prompt mit "**pca_men**" gestartet und beginnt mit dem zentralen Menü:

```
      -------     HAUPTKOMPONENTENANALYSE V.1.1 -------

       1)   Daten laden.
       2)   Daten aufbereiten.
       3)   Eigenwerte berechnen.
       4)   Hauptkomponenten auswählen.
       5)   Target-Rotation.
       6)   Target-Test.
       7)   Abstrakt-Rotation - automatisch.
       8)   Abstrakt-Rotation - manuell.
       9)   Ergebnisse aufbereiten.
      10)   KEYBOARD-Mode.
      11)   Sichern auf File.
      12)   alle sichern.

       0)   Ende.

Nr. der Funktion auswählen:
```

Die gewünschte Prozedur wird durch Eingabe der Zahl aufgerufen. Die Reihenfolge entspricht weitgehend dem Vorgehen bei Abarbeitung einer Faktorenanalyse. Die Routinen testen, ob die erforderlichen Variablen zur Verfügung stehen, so daß sie auch mehrmals durchlaufen werden können bzw. Änderungen von Daten im Keyboard-Mode möglich sind. Eine Prozedur, deren Eingangsdaten fehlen, führt zum Menü zurück.

1) Daten laden

Die Faktoranalyse erwartet die Datenmatrix in der MATLAB-Variablen **faktor1**, mit den Daten der **nkurv** Spektren (zu je **ndat** Punkten) in Spaltenform. Spalte 1 kann die unabhängigen Koordinaten (x-Werte, z.B. Energie) enthalten. Die Datenmatrix kann als ASCII-File oder als MATLAB-File eingelesen werden, voreingestellt ist das Einlesen eines MATLAB-Files **faktor1.mat** (mit **faktor1** und anderen Variablen) im MATLAB-Root-Ordner (Bereitstellung von PHI-Daten siehe Anhang 1). Über den Dateimanager können beliebige Ordner angesteuert werden.
Wenn der geladene File die Variable **faktor1** nicht enthält müssen die Daten umgespeichert werden:

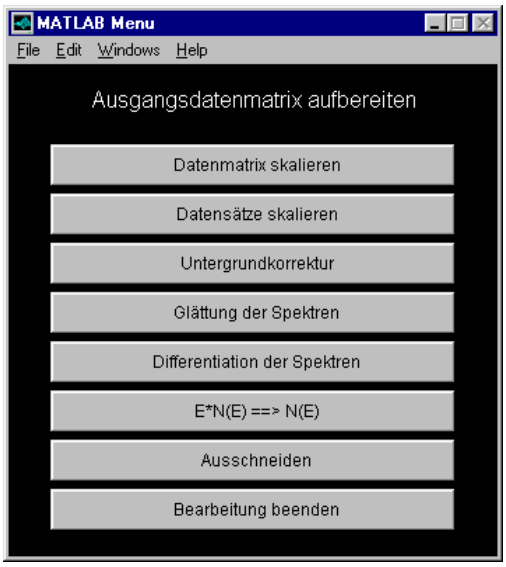
```
Umspeichern der Daten in faktor1:
faktor1 = Variable;
evtl. Variable löschen: clear filename
```

Vor dem Laden werde alle Daten gelöscht, nach dem Laden befindet sich MATLAB im Keyboard-Mode, damit sind noch Manipulation wie Umbenennen, Löschen von überflüssigen Variablen und Hinzuladen von Daten möglich. Die wichtigen Variablen können mit `keep` im MATLAB-Workspace gehalten werden:

```
keep faktor1  ...
```

Speichern der Eingangsdaten in **matlab.mat** ist empfehlenswert. Rückkehr zum Programm über die Eingabe von RETURN an der Tastatur.
Jetzt wird die Dimension der Datenmatrix bestimmt und es muß angegeben werden, ob die 1. Spalte die unabhängige Variable erhält (Energieskale). Falls ja, wird festgestellt ob die Datenpunkte äquidistant sind (wichtig für Glättung, Differentiation). Falls nein, wird eine äquidistante Skala erzeugt. Die Werte der x-Achse werden in **xdat,** die Rohdaten in **yh** gespeichert. Die eingelesenen Rohdaten **yh** werden dargestellt: 1. über der wahren Energieskale, 2. über der Nr. des Datenpunktes.

2) Daten aufbereiten
Auch wenn die Ausgangsdaten bereits aufbereitet wurden, kann dieser Programmpunkt nicht übersprungen werden (Verlassen mit Bearbeitung beenden).



Die Zahlen in der Datenmatrix sollten **skaliert** werden um numerische Probleme zu vermeiden. Das Programm bietet die Möglichkeit entweder die gesamte Datenmatrix oder jeden einzelnen Datensatz zu skalieren.
Arten der Skalierung sind:
1) (mittlerer) Betrag =1,
2) Maximum =1 (bes. für XPS)
3) Maximum-Minimum =1 (für AES)

Die Möglichkeiten der **Untergrundkorrektur** sind vor allem für AES-Surveys und XPS-Spektren gedacht.
Nach Auswahl eines Verfahrens können in folgenden Dialogen die Parameter festgelegt werden.

Zur **Spektrenglättung** stehen 3 Verfahren zur Verfügung. In der Literatur wird Glättung vor Faktorenanalyse nicht empfohlen, da dadurch Spektrenformen verschliffen werden können.
1) SAVITZKY-GOLAY [2] mit dynamischer Randpunktbehandlung, die Routinen basieren auf [3]. Grad des Polynoms, Intervallbreite und Zahl der Glättungszyklen können gewählt werden.
2) Spline-Interpolation auf Basis von MATLAB-Routinen.
3) Linear-Least-Square-Fit nach [3], der Glättungsparameter ist der Schätzwert des Rauschens.

Die **Differentiation** erfolgt ebenfalls nach SAVITZKY-GOLAY [2] mit dynamischer Randpunktbehandlung, die Routinen basieren auf [3]. Grad des Polynoms, Intervallbreite und Ordnung der Ableitung können gewählt werden. Die Prozedur führt zu einem konstanten Anstieg an den Intervallenden. Dieser Bereich sollte vor der Bestimmung der Eigenwerte abgeschnitten werden.

Umrechnung **E*N(E)** in **N(E)**, für Untergrundkorrektur

Ausschneiden eines Spektrenstückes (in der wahren Energieskale). Linke und rechte Grenze werden durch Klick mit linker bzw. rechter Maustaste festgelegt, dient besonders zum Abschneiden des konstanten Anstiegs nach der Differentiation.

Die bearbeitete Datenmatrix wird in der Variablen **yin** gespeichert und anschließend der Hauptkomponentenanalyse unterworfen. Die Skalierungsfaktoren stehen in **ysc**. Durch nochmaligen Aufruf der Aufbereitung kann **yin** weiter bearbeitet werden, **ysc** wird bei nochmaliger Skalierung überschrieben.

3) Eigenwerte berechnen

Es werden die Eigenwerte des Problems berechnet. Klassisch wird aus der Datenmatrix der Dimension (**ndat**, **nkurv**) die Kovarianzmatrix (**nkurv,nkurv**) gebildet und deren Eigenwerte berechnet. Im Programm wird zur Rechenzeitoptimierung nach [4] standardmäßig die kleinere Matrix (m,m) gebildet, mit m = min(**ndat**, **nkurv**). Die Rechnung mit der wahren Kovarianzmatrix ist möglich. Die Eigenwerte **yew** werden nach Größe sortiert und im halblogarithmischen Maßstab dargestellt.

Die Anzahl der Eigenwerte > 0 ist theoretisch ≤ min(**ndat**,**nkurv**). Wegen der begrenzten numerischen Genauigkeit können Eigenwerte sehr klein ($<10^{-15}$) oder < 0 sein. An dieser Stelle kann die Zahl der **Eigenwerte new** festgelegt werden (das ist **nicht** die Zahl der Hauptkomponenten **nfak**).
Der anschließende Dialog fordert zum Speichern der Ergebnisse auf (empfohlen **matlab.mat)** .

4) Hauptkomponenten auswählen

Es werden verschiedene Kriterien für die sinnvolle Zahl der Eigenwerte benutzt:

1) Matrixrekonstruktion: Das Programm rekonstruiert die Datenmatrix für eine wachsende Zahl k von Eigenwerten (1 £ k £ **new**) und stellt den R-Faktor (Anhang 2) jedes Spektrums als Maß der Differenzen zu den Meßwerten dar. Damit kann ein Eindruck von der Güte der Faktoranalyse gewonnen werden. Der R-Faktor ist 0....1. Ein großes R zeigt an, daß die Zahl der Eigenwerte nicht ausreicht, bzw. das Spektrum nur Rauschen enthält. Die Größe der Eigenwerte und der R-Faktor der gesamten Datenmatrix sind ebenfalls Kriterien für die Zahl der Hauptkomponenten.
2) Weiterhin berechnet das Programm verschiedene empirische Kriterien, wie die Faktor-Indikatorfunktion FIF(k) [1] (s. Theorie, Gl. 12a), die imbedded error function IEF(k) [1] (Gl. 12b), die Testfunktion nach [5] (Gl. 12c), die kumulative Varianz u.a. und stellt die Ergebnisse grafisch dar.

Nachdem ein Eindruck von der Zahl der möglichen Hauptkomponenten gewonnen wurde, wird im folgenden Prozeß die Zahl der sinnvollen Hauptkomponenten **nfak** festgelegt:

Es wird eine Zahl von Eigenwerten als Hauptkomponenten ausgewählt (am Beginn alle). Dargestellt werden:

Figure 1: Zahl der ausgewählten Eigenvektoren. oben: orthonormale Vektoren (d.h. Betrag = 1), unten: skaliert mit Eigenwert. Die orthonormierten Vektoren lassen aufgrund der vergleichbaren Länge die spektrale Struktur besser erkennen, die skalierten Vektoren geben einen Eindruck von der Größe des Spektrums.

Figure 2: R-Faktor aller Spektren für die gewählte Zahl von Hauptkomponenten als Maß für die Güte der Datenrekonstruktion.

Figure 1: mit der gewählten Zahl von Hauptkomponenten rekonstruierte Daten (**yrek**, sollen natürlich aussehen) und Residuen (**yres**: Differenz zwischen rekonstruierten Daten **yrek** und Eingangsdaten **yin** als Maß für die Güte der Datenrekonstruktion). Diese sollen keine spektralen Strukturen enthalten, sondern nur Rauschen.

Die Prozedur wird solange durchlaufen bis die Frage nach der Änderung mit „nein“ beantwortet wird. Nach dem ersten Aufruf von Punkt 4 kann manuell eine andere Zahl von Hauptkomponenten **nfak** eingestellt werden (ohne die Dialoge erneut zu durchlaufen).

Zur Darstellung des Ergebnisses wird der Bildschirm unterteilt. Unten werden die **nfak** gewählten Hauptkomponenten (Eigenvektoren, Basisspektren: **yfak**) ohne x-Achse, oben deren Anteile (Ladungen, loadings: **cfak**) am gemessenen Spektrum über dessen Nummer (Zyklus/Punkt) dargestellt.

5) Target-Rotation
Wird die Existenz reiner Komponenten im Datensatz vermutet, kann das System der Eigenvektoren so gedreht werden, daß bei einem bestimmten Zyklus/Punkt nur eine Komponente auftritt.
Die Möglichkeiten zur Target-Rotation sind:
* Auswahl des zu rotierenden Vektors erfolgt durch Mausklick (LM),
* manuelle Eingabe (durch Tastendruck "m" oder "M"),
* Abbruch (durch Tastendruck "q" oder "Q").

Beim Mausklick wird der dem angeklickten Punkt nächste Punkt des Datensatzes bestimmt und die Target-Rotation ausgeführt; Handeingabe fragt nach dem Eigenvektor und dem Zyklus für die Targetrotation.

6) Target-Test
Der Target-Test erfordert einen Testvektor (z.B. über Keyboard-Eingabe laden), der den Namen **`ytest`** erhalten muß. Das Programm erwartet einen einspaltigen Vektor oder eine Matrix [x y], von der y benutzt wird. Das Programm testet den Vektor nur auf die richtige Länge.
Der Vektor wird zusammen mit den Basisvektoren dargestellt (Symbol *) und nach dem zu testenden Vektor gefragt. Die Transformationsmatrix wird berechnet und eine Test-Transformation wird ausgeführt.
Kriterien für die Güte des Testvektors sind der Korrelationskoeffizient zwischen Testvektor und dem bei der Testtransformation erhaltenen Vektor und die SPOIL-Funktion nach [1, S. 93] mit den dort angegebenen Kriterien.

7) Abstrakt-Rotation - automatisch
Es ist nur die Varimax-Rotation realisiert. Um die Faktoren besser interpretieren zu können wird das ursprüngliche System der Eigenvektoren so gedreht, daß „einfache“ Faktoren entstehen. Ein „einfacher“ Faktor hat wenige große Ladungen und viele kleine. Mathematisch wird die totale Varianz der quadrierten Ladungen maximiert [1, S. 49]. Die Funktion entspricht VARIMAX.M in [6]. Das Verfahren basiert auf [7]. Nach unseren Erfahrungen sind die Ergebnisse nicht immer spektroskopisch sinnvoll, ein Versuch sollte unmittelbar nach Festlegung der Zahl der Hauptkomponenten, d.h. vor allen anderen Rotationen erfolgen.

8) Abstrakt-Rotation - manuell
Bei der Abstrakt-Rotation werden Spektren manuell gegeneinander verdreht, um unphysikalische Spektrenformen und negative Ladungen auszuschließen. Die manuelle Abstrakt-Rotation erfolgt in zwei Schritten.
Im ersten Schritt werden die zu bearbeitenden Hauptkomponenten ausgewählt (wird eine Kurve zweimal gewählt erfolgt die Abstrakt-Rotation der Kurve gegen sich selbst.). Die Möglichkeiten sind:
* Auswahl der zu rotierenden Komponente erfolgt durch Mausklick: beide Kurven werden durch Mausklick ermittelt und markiert (Kurve 1 mit "*", Kurve 2 mit "+")
* manuelle Eingabe (durch Tastendruck "m" oder "M"); die Nummern der Kurven und der Faktor α werden erfragt
* Abbruch (durch Tastendruck "q" oder "Q").

Im zweiten Schritt werden für Kurve 1 (*), Start- und Endpunkt der Abstrakt-Rotation durch Mausklicks markiert.

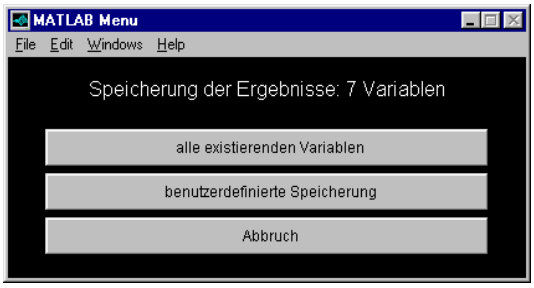
9) Ergebnisse aufbereiten
Hier werden die Daten zur Ausgabe aufbereitet.
1) Residuen **`yres`** und rekonstruierte Daten **`yrek`** neu berechnen. Diese Größen werden beim ersten Durchlaufen von Punkt 4 erzeugt, aber nicht bei den schnellen Änderungen der Zahl der Hauptkomponenten danach.
   Bei richtiger Wahl der Hauptkomponenten entspricht `yrek` rauschfreien Spektren!!
2) Skalierung der Ladungen **`cfak`** untereinander, dabei wird **`yfak`** mit geändert, aber **`yrek=yfak*cfak'`** nicht.
3) x-Achse an Vektoren anfügen: **`yout`** = [**`xdat yfak`**]
4) Reskalierung der Ladungen mit **`ysc`**, reskalierte Werte in **`cout`**
5) Summe der Ladungen auf 1 normieren: **`cnorm`**

10) KEYBOARD-Mode
Interpretermodus. Alle Variablen stehen zur Verfügung und können beeinflußt werden. Es ist auch möglich Ergebnisse mit den MATLAB-Befehlen darzustellen, zu speichern bzw. in die Zwischenablage zu kopieren.

11) Sichern auf File

Es werden die Namen der vorhandenen Ergebnisvariablen angezeigt. Alle Ergebnisvariablen oder ein Teil können in einen File gesichert werden. Die benutzerdefinierte Speicherung erfordert Angabe der zu speichernden Variablen.

```
Speichern der Ergebnisse der Faktoranalyse
Es existieren die Ergebnis-Variablen:
cfak   : Matrix der Anteile
cnorm  : normierte Matrix der Anteile
cout   : re-skalierte Matrix der Anteile
fif    : Vektor der Faktor-Indikator-Funktion
yfak   : Matrix der Eigenvektoren
yout   : Matrix der Eigenvektoren mit x-Achse
yrek   : rekonstruierte Daten
yres   : Residuen
ystat  : Statistik der Residuen
```



12) alle sichern

Der gesamte MATLAB Workspace wird in den File matlab.mat gesichert.

0) Beenden

Das Programm wird beendet, alle Variablen bleiben im MATLAB Workspace erhalten.

1.5 Literatur:

[1] E.R. Malinowski, D.G. Howery, Factor Analysis in Chemistry, J. Wiley & Sons, New York Chichester Brisbane Toronto 1980
E.R. Malinowski, Factor Analysis in Chemistry, 2nd Edition, John Wiley, 1991
[2] A. Savitzky and M. Golay, Anal. Chem. **36**, 1627 (1964)
[3] W. Gander, J. Hrebicek: Solving Problems in Scientific Computing using Maple and Matlab, 2nd Ed. (Chapter 9), Springer Heidelberg 1995
[4] Matlab Chemometrics Toolbox, The MathWorks, Inc.
[5] S. Alex, R. Savoie, Can. J. Spectrosc. **34**, 27 (1989)
[6] Leslie F. Marcus: „Program supplement to applied factor analysis in the natural sciences“, in: R. Reyment, K.G. Jöreskog: „Applied factor analysis in the natural sciences“, Cambridge University Press 1993.
[7] H.H. Harman: "Modern Factor Analysis", Chicago 1967, Chapter 14, section 4.

Tabelle 1: Zusammenstellung der Funktionen (Stand Februar 1999)

abs_clk	bestimmt Parameter der Abstraktrotation vom Bildschirm
abs_man	manuelle Eingabe für Parameter der Abstraktrotation
abs_rota	automatische Abstraktrotation (Varimax, Covarimin)
abs_rotm	manuelle Abstraktrotation
apph	APPH in mit der Maus festgelegten Fenstern, Emax<Emin ist berücksichtigt
bg_int1	Korrektur mit integralem Untergrund
bg_it1	iterative Bestimmung des integralen Untergrunds (Shirley)
bg_lin	Korrektur mit linearem oder konstanten Untergrund
bg_tug1	Korrektur mit integralem Untergrund (Tougaard)
bgtoug1	Bestimmung des Untergrunds nach TOUGAARD
contents	Inhaltsverzeichnis
dat_cut	Energiefenster ausschneiden
dat_dif	Differentiation nach Savitzky-Golay mit gleitender Randpunktbehandlung
dat_dif2	automatische Differentiation der Datenmatrix nach SAVITZKY-GOLAY
dat_e	Division durch x (E*N(E)==> N(E)
dat_in	Einlesen und Aufbereiten der Daten
dat_prep	Vorbehandlung der Daten
difc	Koeffizienten für Differentiation nach Savitzky-Golay
ew_calc	Eigenwerte der Matrix berechnen
ew_crit	Zahl der Hauptkomponenten festlegen
ew_crit2	new criteria for number of significant components in PCA
ewnr	Zahl der Eigenwerte der Matrix festlegen
fakind	Kriterien für Zahl der Hauptkomponenten berechnen
fakres	berechnet rekonstruierte Werte und Residuen
fak_plot	Darstellung der Anteile und Eigenvektoren mit Normierung
fak_rot	Erzeugung der Targetrotation (für Targetrotation und Targettest)
fak_scr	Darstellung von Basisvektoren und PCA-Anteilen in 2 Fenstern
fint1	Integration über y von unterer bis obere Grenze von x
getbord	getbord(fig) tracks the movement of a mouse-drag
janein	Abfrage mit Default
keep	KEEP is complementary to the "clear" command
key_mode	Übergang zum Keyboard-Mode
lfit1	Linear least square fit of data matrix yin to test data yt
lsq	Glättung mittels Least-Square-Fit
mormcol	berechnet die Norm jeder Spalte einer Datenmatrix (Vektor)
pca_men	Hauptmenü
pca_subs	Liste der Prozeduren und Funktionen (für PCA MEN)
phic2mat	conversion of MAT files created by phic(NT) in MATLAB format.
phispy	extrahiert Parameter aus einem PHI INFO-File
qtx	DLL-MEX-File für Least-Square-Fit (16 bit-DLL, nur Matlab 4.2)
relcrit	berechnet den Reliability-Faktor in Abhängigkeit von der Zahl der Eigenwerte
relfac	Reliability Factor zweier Datensätze
res_aufb	Ergebnisse aufbereiten
res_save	Ergebnisse speichern
res1	berechnet die mittlere Abweichung pro Datenpunkt in Abhängigkeit von der Zahl der Komponenten
rot_abs	führt Abstraktrotation aus
rot_tar	führt Targetrotation aus
rot_var	führt Varimaxrotation aus
sav_all	speichert Matlab Workspace
sc_mat	Datenmatrix skalieren
sc_vec	Vektoren skalieren
scorr1	Untergrundkorrektur nach SHIRLEY
setwin	legt Fenster mit Mausbewegungen fest
sg_difn	Differentiation von äquidistanten Datensätzen nach Savitzky-Golay
sg_smon	Glättung von äquidistanten Datensätzen nach Savitzky-Golay
smoc	Koeffizienten für Glättung nach Savitzky-Golay
smo_ls	Vorbereitung: Glättung durch Least-Square-Fit
smo_sg	Vorbereitung: Glättung nach Savitzky-Golay
smo_sp	Vorbereitung: Glättung durch Interpolation
spqr	DLL-MEX-File zur QR-Zerlegung (für Least-Square-Fit) (16 bit-DLL, nur Matlab4.2)
sp_smon	Glättung von äquidistanten Datensätzen durch Interpolation
strim	entfernt Leerzeichen am Anfang und Ende eines Zeichenkettenausdrucks
tcorr1	Matlab-Funktion, berechnet und korrigiert Untergrund nach TOUGAARD
tcorr2	DLL-MEX-File, berechnet und korrigiert Untergrund nach TOUGAARD, (16 bit-DLL, nur Matlab4.2)
tar_rot	Vorbereitung der Target-Rotation
tar_tst	Vorbereitung des Target-Test
varimax1	Varimax-Rotation
var_rot	Vorbereitung der Varimax-Rotation
vfunct1	berechnet Varianz für Varimax

Tabelle 2: Wichtige Variablen von pca_men (Stand Februar 1999)

art_sc	Art der Skalierung: 1: Matrix, 2: Vektoren
cev	Matrix der Ladungen aller Eigenvektoren
cfak	Ladungen der ausgewählten Hauptkomponenten
cmax	Faktor für Normierung der Ladungen
cnorm	Ladungen der ausgewählten Hauptkomponenten (auf Summe = 1 normiert)
cout	Ladungen der ausgewählten Hauptkomponenten (reskaliert)
crit	Vektor, enthält (1) Zahl der Eigenwerte, (2) Zahl der Eigenwerte > 0, Zahl der Eigenwerte > limit
dx	Schrittweite äquidistanter Spektren, sonst 1
ewcol	Spaltenvektor, enthält Indizes der Eigenwerte vor Sortieren
faktor1	Matrix der Eingangsdaten
fg	Variable für Papiergröße (für Figure-Handle)
fif	Vektor der Faktor-Indikator-Funktion
figh	Bezug (Handle) der objektorientierten Grafik (Bildschirmes 1)
figh2	Bezug (Handle) der objektorientierten Grafik (Bildschirm 2 (4 Kriterien))
filename	Name des geladenen Files (für uigetfile)
funks	Vektor, der den Namen der Routine enthält (Auswertung mit eval)
ie	Vektor der imbedded-error-Funktion
klcov	Schalter: = 1 für Benutzung der kleineren Kovarianzmatrix
lim	Vektor, der Grenzwert limit für alle Eigenwerte enthält
limit	Grenzwert, für den Eigenwerte > 0 gewertet werden
m	Dimension der Datenmatrix
minie	Minimum der imbedded-error-Funktion
minind	Minimum der Faktor-Indikator-Funktion
n	Dimension der Datenmatrix
ndat	Zahl der Punkte je Spektrum
new	Zahl der Eigenwerte (nach Kriterium)
nfak	Zahl der festgelegten Hauptkomponenten
nie	Lage (Feldindex) des Minimums der imbedded-error-Funktion
nind	Lage (Feldindex) des Minimums der Faktor-Indikator-Funktion
nkurv	Zahl der Kurven im Datensatz
pathname	Pfad des geladenen Files (für uigetfile)
ploth	Handel eines einzelnen Plots im Grafik-Bildschirm 1
ploth1	Handel des ersten Plots bei zwei Fenstern im Grafik-Bildschirm 1
ploth2	Handel des zweiten Plots bei zwei Fenstern im Grafik-Bildschirm 1
pos	Koordinaten der Textdarstellung
r	Rang der Kovarianzmatrix
re	"Experimental Error" nach Malinowski (Kriterium für Zahl der Hauptkomponenten)
Rmatrix	R-Faktor der Datenmatrix
Rvector	R-Faktor der einzelnen Spektren
scre	"SCREE-Test" (Kriterium für Zahl der Hauptkomponenten)
screen	Vektor der Bildschirmauflösung enthält
test	Variable für Entscheidungen
texthxy	Bezug (Handle) eines Textes der objektorientierten Grafik
var	Vektor der Varianz (Kriterium für Zahl der Hauptkomponenten)
varmin	Feldindex, bei dem Varianz < 1% wird
wahl	Schalter
wege	enthält Zahl der Unterprogramme im Menü
work	Hilfsvariable für Menü
x	Hilfsgröße
xdat	Vektor der x-Achse
xt	Vektor, der die Differenzen der x-Werte enthält (für Test auf Äquidistanz)
yev	Matrix der Eigenvektoren
yew	Vektor der Eigenwerte
yfak	Matrix, die die ausgewählten Hauptkomponenten (Basisspektren) enthält
yin	bearbeitete Datenmatrix vor Berechnung der Eigenwerte
yout	Matrix, der bearbeiteten Hauptkomponenten mit Energieskale
yrek	Matrix der rekonstruierten Daten
yres	Residuen: Differenz zwischen Eingangs- und rekonstruierten Daten
ysc	Skalierungsfaktor (Zahl für Matrixskalierung), Vektor für Skalierung der Datensätze

Anhang 1: Daten von PHI-Geräten bereitstellen

Die Hilfsprogramme phispy.exe, phic2mat.exe, info.exe unterstützen die Auswertung von Messungen mit PHI-ACCESS, werden aber zur Faktorenanalyse nicht benötigt und können in einem beliebigen Ordner stehen. Diese Programme wurden in GfA-Basic für Windows geschrieben und sind lauffähige 16-Bit-Windows-Applikationen. Die Programme laufen unter WfW3.x, Win95 und WinNT4.0, unterstützen aber keine langen Filenamen. Zur Ausführung der *.exe-Files ist die Datei gfarun10.dll (Laufzeitbibliothek für Intel-PCs 486 und höher) erforderlich, die im Windows-Ordner stehen muß. Der File phic2mat.ini enthält den Namen des Orders, in dem die Suche nach den Files beginnen soll, ist aber für die Funktion der Programme nicht erforderlich.

1. Datenstruktur

Unter WinNT4.0 erzeugt das mit den PHI-ACCESS-Versionen 7.0, 7.1, 7.2 gelieferte Konvertierungs-programmm **phic** mit der Option -t matlab Files, die von MATLAB 4.2 und 5.x nicht gelesen werden können. Ursache: die erste Zahl (type) des Headers ist falsch offenbar gesetzt. Das Programm **phic2mat** ersetzt den 4-Byte-Header durch 0000, die Daten können dann von MATLAB 4.2 und 5.x gelesen werden.

2. Speicherplatzbedarf

Alle Daten (auch Bilddaten von SE-Bildern) werden von phic als double precision (64bit, 8 Byte) erzeugt). Ein SE-Bild mit 512x512 pixel ist im *.mat-Format dann 2 MB groß, die Zahlenwerte sind aber ganzzahlig (0....255, 1 Byte). Matrizen werden von MATLAB V4.2 vor dem Speichern untersucht. Matrizen, die mehr als 10000 ausschließlich ganzzahlige Elemente enthalten, werden in Abhängigkeit vom Zahlenbereich der Elemente als integer (Elementbereich 0...255) bzw. long integer (Elementbereich 0...65535) gespeichert (in MATLAB 5 gibt es dafür das Format uint8). Durch Speichern von Bildfiles im *.mat-Format aus MATLAB 4.2 heraus:

```
>> clear
>> load bildfile
>> save bildfile
```

verringert sich dadurch der Speicherplatz für SE-Bilder von 2 MB auf 260 kB. Dieses Vorgehen kann auch den Speicherplatzbedarf von anderen großen Files (Window-Tiefenprofile) verringern, da die Originaldaten i. allg. die counts enthalten, d.h. integer sind. **Bei AES-Maps können abhängig vom Modus der Bilderfassung auch negative und gebrochene Zahlen auftreten!!**

3. Meßdaten umwandeln

1) Die Meßdaten werden mit phic konvertiert (wird phic nicht im Ordner der gemessene Files aufgerufen muß der Name den Pfad einschließen):

```
>c:\pfad_der_messfiles\phic –t matlab –g –o export messfiles*
```

-t: erzeugt Typ MATLAB (aber noch nicht lesbar)
-g: erzeugt inf-File (ASCII) mit verschiedenen Angaben,
-o export: (Unter)Ordner, in dem die umgewandelten Files gespeichert werden.
Es kann eine Liste von Filenamen aufgeführt werden, Wildcards sind erlaubt.
Für im Windows-Mode gemessene Tiefenprofile und Linescans werden damit Files erzeugt, die nur Peakintensitäten (APPH bei AES, Höhe oder Fläche bei XPS) und Sputterzeit bzw. Koordinate der Linie enthalten. Diese MATLAB-Files sollten umbenannt werden (z.B. mit der Erweiterung .AP für APPH), da sie sonst im nächsten Schritt überschrieben werden.
Tiefenprofile und Linescans, die spektrale Daten enthalten, werden nochmals mit der Option –S (groß!) konvertiert:

```
>c:\pfad_der_messfiles\phic –t matlab –g –o export -S messfiles*
```

Mit der Option –S können alle Files umgewandelt werden (für Surveys, Bilder, Multiplex gibt das Programm die Warnung „invalid option“ aus, wandelt aber korrekt um). Mit –S umgewandelte Tiefenprofile und Linescans enthalten keine Angaben über Sputterzeit bzw. Koordinate der Linie.

2) Die Files werden in das korrekte MATLAB-Format umgewandelt. Dazu gibt es 2 Programme:
- **phic2mat** als MATLAB-Script, dem eine Liste von Filenamen übergeben werden muß.
- **phic2mat.exe** als eigenständiges Programm. Ist ein File ausgewählt, kann entschieden werden, ob nur dieser File oder alle Files im Ordner umgewandelt werden sollen (dabei sind Files mit den Erweiterungen inf, tmp, doc und clp ausgeschlossen). Es wird ein Protokoll über Filetyp bzw. umgewandelte Variablen ausgegeben.

3) Die Daten der Tiefenprofile bzw. Linescans können in MATLAB zusammengefaßt werden, z.B.:

```
>> clear
>> load tiefenprofil.mat
>> load tiefenprofil.ap –mat
>> save tiefenprofil
```

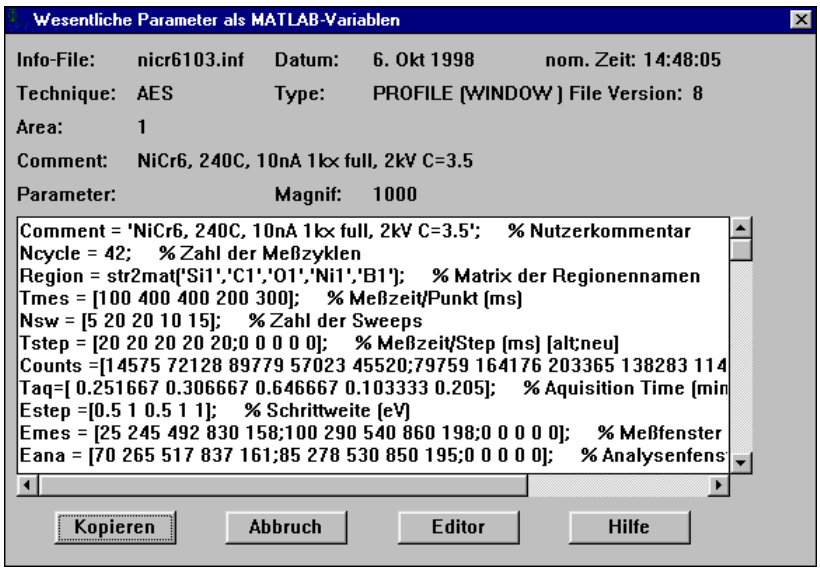
Dann sind im File `tiefenprofil.mat` alle Informationen der MATLAB-Files enthalten.

4) Die MATLAB-Files enthalten die Meßdaten in folgenden Variablen (Intensitäten meist in counts):

Technik	Intensitätwerte (meist counts)	Dimension	Energie
Survey	r1	Zahl der Datenpunkte, Zahl der area	xr1
Multiplex	r1: 1. spektrale Region r2: 2. spektrale Region	Zahl der Datenpunkte, Zahl der area Zahl der Datenpunkte, Zahl der area	xr1 xr2
Tiefenprofil (Window)	a1r1: 1. spektrale Region des 1. area a1r2: 2. spektrale Region des 1. area ..... a2r1: 1. spektrale Region des 2. area ..... a1pp: Peakint. des 1. area a2pp: Peakint. des 2. area ta1pp: zugehörige Sputterzeiten	Zahl der Datenpunkte, Zahl der Zyklen Zahl der Datenpunkte, Zahl der Zyklen Zahl der Datenpunkte, Zahl der Zyklen Zahl der Zyklen, Zahl der Regionen Zahl der Zyklen, Zahl der Regionen Zahl der Zyklen, Zahl der Regionen	xa1r1 xa1r2 xa2r1 ----- ----- -----
Linescan (Window)	l1r1: 1. spektrale Region der 1. line l1r2: 2. spektrale Region der 1. line ..... l2r1: 1. spektrale Region der 2. line ..... l1pp: Peakint. der 1. line l2pp: Peakint. der 2. line dl1pp: zugehörige Koordinate	Zahl der Datenpunkte, Zahl der Punkte Zahl der Datenpunkte, Zahl der Punkte Zahl der Datenpunkte, Zahl der Punkte Zahl der Punkte, Zahl der Regionen Zahl der Punkte, Zahl der Regionen Zahl der Punkte, Zahl der Regionen	xl1r1 xl1r2 xl2r1 ----- ----- -----
Map (AES)	a1r1: 1.Bild a1r2: 2.Bild ...	Zahl der Punkte, Zahl der Punkte Zahl der Punkte, Zahl der Punkte	keine

5) Angaben über die Meßzeit, die Namen der Regionen usw. sind nur über PHI-ACCESS und den *.inf-File zugänglich, die Koordinaten der areas und lines sind weder im *.mat-, noch im *.inf-File enthalten.

Der Inhalt des inf-Files läßt sich mit einem Texteditor ansehen. Mit dem Programm **phispy.exe** können die wesentlichen Informationen dargestellt und als MATLAB-Variablen bereitgestellt werden. Wichtige Größen werden im Fenster angezeigt und können über die Zwischenablage direkt in das MATLAB-Command-Window kopiert werden.



6) Mit dem Programm **info.exe** kann der Inhalt der Kommentarzeilen in den *.inf-Files (einzeln oder für ganze Ordner) betrachtet und in die Zwischenablage kopiert werden. Ab PHI-ACCESS Version V7.2 enthalten die *.inf-Files die Kommentarzeile in voller Länge (bei der Ausgabe von Directories mit LIST in den Texteditor wird der Text weiter abgeschnitten).

7) Zusammenfassen und Aufbereiten von Messungen

Oft besteht die Aufgabe, die unter gleichen Bedingungen gemessene Daten mehrerer Files (z.B. zusätzliche Messungen an Standards) für die Auswertung mittels Faktoranalyse in einem Datensatz zusammenzufassen. Das kann durch MATLAB-Scripts geschehen (siehe Beispiel). Dabei können auch häufig genutzte Verfahren der Datenvorbehandlung automatisch durchgeführt werden. Im Bsp. erfolgt eine Differentiation durch dat_dif2.m. Durch Bearbeiten mit einem komfortablen Texteditor können solche Files schnell für andere Meßaufgaben modifiziert werden.

```
% Aufbereitung Proben xxxxx
% 1. Messung vom 06.10.98
%nicr6103    Profile      #xxx, 240C, 10nA 1kx full, 2kV C=3.5
%nicr6118    Profile      #xxx, 120C
%nicr6127    Profile      #xxx, NiCr6, ungetempert

clear
LP=[pwd '\']
fils=str2mat('nicr6127','nicr6118','nicr6103');
[nfil, lfil]=size(fils);

Region = str2mat('Si1','C1','O1','Ni1','B1'); % Matrix der Regionennamen
Tmes = [100 400 400 200 300]/1000;            % Meßzeit/Punkt (ms)
Estep =[0.5 1 0.5 1 1];                       % Schrittweite (eV)
zstart=[1 1 1];
zstop=[28 27 25];
for L=1:nfil,
   clear a1r1 a1r2 a1r3 a1r4 a1r5 ta1pp
   load([LP fils(L,:)])
   Si1=[Si1 a1r1(:,zstart(L):zstop(L))];
    C1=[ C1 a1r2(:,zstart(L):zstop(L))];
    O1=[ O1 a1r3(:,zstart(L):zstop(L))];
   Ni1=[Ni1 a1r4(:,zstart(L):zstop(L))];
    B1=[ B1 a1r5(:,zstart(L):zstop(L))];
   tsp=[tsp;ta1pp(zstart(L):zstop(L),1)];
end

Si1=Si1/Tmes(1);
 C1= C1/Tmes(2);
 O1= O1/Tmes(3);
Ni1=Ni1/Tmes(4);
 B1= B1/Tmes(5);

ESi1=xa1r1;
 EC1=xa1r2;
 EO1=xa1r3;
ENi1=xa1r4;
 EB1=xa1r5;

Si1d=dat_dif2(Si1,Estep(1),2,9);
 C1d=dat_dif2( C1,Estep(2),2,5);
 O1d=dat_dif2( O1,Estep(3),2,9);
Ni1d=dat_dif2(Ni1,Estep(4),2,5);
 B1d=dat_dif2( B1,Estep(5),2,5);

save nc6w1.mat Si1 Si1d C1 C1d Ni1 Ni1d B1 B1d O1 O1d tsp ESi1 EC1 EO1 ENi1 EB1
```

Der MATLAB-Script genertp0.m erzeugt die MATLAB-Kommandos weitgehend automatisch. Der File sollte als Muster behandelt werden, d.h. für einzelne Meßprobleme Kopien anfertigen und im 1. Abschnitt des Files die entsprechenden Änderungen vornehmen.

Anhang 2: R-Faktor (Reliability Factor)

In der Faktorenanalyse wird die Zahl der signifikanten Peakformen an Hand empirischer Kriterien bestimmt [1, 2]. Entscheidend für spektroskopische Anwendungen ist, daß das Ergebnis die gemessenen Spektren reproduziert. Der „reliability factor“ [3] wird als Maß für die Güte der Spektrenreproduktion betrachtet (es besteht offenbar ein Unterschied zum in der Faktorenanalyse eingeführten Begriff des „reliability factor“ nach [1]).

Der R-Faktor wurde in [3] als Maß zum Vergleich theoretischer und experimenteller Kurven eingeführt, die positive und negative Werte haben müssen und ist auf das Intervall 0...2 normiert. Dann ist der R-Faktor zweier Kurven ($\mathbf{Y}_M$, $\mathbf{Y}_R$) mit N Punkten gegeben durch:

$$R = \frac{\sum_{i=1}^{N}\left(Y_{i,M} - Y_{i,R}\right)^2}{\sum_{i=1}^{N}\left(Y_{i,M}^2 + Y_{i,R}^2\right)} = \begin{cases} 0, \text{ wenn } \mathbf{Y}_M = \mathbf{Y}_R \\ 1, \text{ wenn } \mathbf{Y}_M, \mathbf{Y}_R \text{ unkorreliert} \\ 2, \text{ wenn } \mathbf{Y}_M, \mathbf{Y}_R \text{ antikorreliert} \end{cases} \tag{1}$$

Im folgenden soll die Güte eines durch k Hauptkomponenten rekonstruierten Spektrums $\mathbf{Y}_R$ mit den Meßwerten $\mathbf{Y}_M$ verglichen werden.

Fall 1: Durch die Faktorenanalyse wird die gemessene Spektrenformen gut reproduziert, die Residuen enthalten nur Rauschen: $\mathbf{Y}_R - \mathbf{Y}_M = \boldsymbol{\varepsilon}$. Da Rauschen und Signal nicht korreliert sind, ergibt sich für jedes Spektrum:

$$R = \frac{\sum_{i=1}^{N}\left(\varepsilon_i\right)^2}{\sum_{i=1}^{N}\left(2Y_{i,R}^2 + 2Y_{i,R}\varepsilon_i + \varepsilon_i^2\right)} \approx \frac{\sum_{i=1}^{N}\left(\varepsilon_i\right)^2}{\sum_{i=1}^{N}\left(2Y_{i,R}^2 + \varepsilon_i^2\right)} \approx \frac{\sum_{i=1}^{N}\varepsilon_i^2}{2\sum_{i=1}^{N}Y_{i,R}^2} = \frac{\mathrm{RMS}_{\mathrm{Noise}}^2}{2\cdot\mathrm{RMS}_{\mathrm{Signal}}^2} \tag{2}$$

Sind alle Hauptkomponenten gefunden, ist der R-Faktor nur durch das Verhältnis von Signal und Rauschen bestimmt und sollte sich wenig mit zunehmender Zahl der Eigenwerte ändern.

Fall 2: Die Zahl der Hauptkomponenten ist zu gering; die Residuen enthalten neben Rauschen deutliche Spektrenformen:

$$R = \frac{\sum_{i=1}^{N}\left(Y_{i,M} - Y_{i,R}\right)^2}{\sum_{i=1}^{N}\left(Y_{i,M}^2 + Y_{i,R}^2\right)} = \frac{\sum_{i=1}^{N}Y_{i,M}^2 + Y_{i,R}^2 - 2Y_{i,M}Y_{i,R}}{\sum_{i=1}^{N}\left(Y_{i,M}^2 + Y_{i,R}^2\right)} = 1 - \frac{2\sum_{i=1}^{N}Y_{i,M}Y_{i,R}}{\sum_{i=1}^{N}\left(Y_{i,M}^2 + Y_{i,R}^2\right)} < 1 \tag{3}$$

Der R-Faktor ist kleiner 1 und durch die Korrelation zwischen gemessenem Signal und Residuen bestimmt. Mit zunehmender Zahl der Hauptkomponenten verringern sich die Residuen und damit der R-Faktor.

Fall 3: Das gemessene Spektrum enthält nur Rauschen $\mathbf{Y}_R = 0$, $\mathbf{Y}_M = \boldsymbol{\varepsilon}$.

$$R = \frac{\sum_{i=1}^{N}\left(\varepsilon_i\right)^2}{\sum_{i=1}^{N}\varepsilon_i^2} = 1 \tag{3}$$

Enthält ein Spektrum kein Signal, ist R = 1.

Vorteile der Nutzung des R-Faktors sind:

1) Das Maß ist normiert auf 0...1 (da die Eigenvektoren nicht antikorreliert sein können) und von der Größe des Datensatzes unabhängig, der Minimalwert wird vom Rauschen bestimmt. Damit können Messungen verglichen und empirische Erfahrungen gewonnen werden.

2) Durch Darstellung des R-Faktors für jeden Sputterzyklen lassen sich Hinweise auf das Auftreten von Grenzflächenphasen u.ä. finden.

Literatur

[1] E.R. Malinowski, D.G. Howery, Factor Analysis in Chemistry, J. Wiley & Sons, New York Chichester Brisbane Toronto 1980

[2] S. Alex, R. Savoie, Can. J. Spectrosc. **34**, 27 (1989)

[3] J.B. Pendry, Reliability factors for LEED calculations, J. Phys. C.: Solid St. Phys. **13**, 937 (1980)