

# Fit von benutzerdefinierten Funktionen

Die Befehle im Overlay **USERFITPRG** unterstützen den Fit von *benutzerdefinierten Funktionen* an gegebene Messkurven. Dazu werden in **INIT-FIT** oder **GET-FUNC** die Funktionsvorschriften in algebraischer C-Syntax in eine Zeile geschrieben und nach **<Ret>** unter dem Namen der Funktion, der dann Bezug auf die Funktion erlaubt, kompiliert und den **WORDS** in **FORTH** hinzugefügt. (Mit **SAVE-FUNC** kann die Funktion in eine Datei geschrieben und dann jederzeit mit **LOAD-FUNC** wieder eingeladen und kompiliert werden.)

Die Fit-Routine im Overlay **USERFITPRG** gestattet Ihnen die Parameter beliebiger Funktionen an einen vorgegebenen Kurvenverlauf optimal anzupassen. Insbesondere können Sie im Fit irgendwelche Nebenbedingungen, die die Parameter erfüllen sollen, berücksichtigen, indem Sie diese Bedingungen in die Funktionsvorschrift einbauen.

Sollte einmal eine Funktionsvorschrift mehr als eine Zeile Platz beanspruchen, gibt es, wie das folgende Beispiel zeigt, die Möglichkeit, die Funktion aus einer Quelldatei mit zu laden und zu kompilieren.

*Im folgenden Beispiel soll mit Hilfe des **Fits einer Modellfunktion** überprüft werden, wie gut die Modellfunktion das physikalische Experiment erklären kann.*

**Abb. 2** zeigt ein Xe-5p-Photoemissionsspektrum **PAX.UPS** von Xe adsorbiert auf Cu<sub>3</sub>Pt(111) bei 50 K nach dem Untergrundabzug. Das Experiment wird mit Hilfe einer **PAX<sup>1</sup>**-Funktion interpretiert. Wie gut die Modellfunktion das Experiment erklären kann zeigt dann die Güte des Fits der Funktionsparameter bzw. das Differenzspektrum zwischen dem Spektrum und dem Modell (**s. Abb. 2**).

Laden Sie zunächst das Xe-5p-Photoemissionsspektrum **PAX.UPS** in einen Datenspeicher. Es befindet sich im Ordner **PAX**.

```
PATH DATA/PAX <Ret>
LOAD PAX.UPS <Ret>
in buffer > {1}
```

Erweitern Sie mit **USERFITPRG** den Wortschatz des Programms **CSA** um alle für den **FIT** nötigen Befehle. Das entsprechende **OVERLAY** wird geladen.

Da die **PAX**-Funktion komplex ist, soll zunächst eine Quelldatei **PAX.SEQ** mit der Funktionsvorschrift für die **PAX<sup>1</sup>**-Funktion unter dem Funktionsnamen **PAX** erstellt werden.

Erzeugen Sie diese Quelldatei im aktuellen Datenordner und laden Sie die Datei in den Editor mit

**CREATE-FILE PAX.SEQ <Ret>**

Wenn die Datei bereits existiert, laden und editieren Sie die Datei mit

**EDIT PAX.SEQ <Ret>**

Schreiben Sie den Text in Abb. 1 in die Datei (Leerzeichen trennen Worte und dürfen nicht weggelassen werden).

```

: PAX ( -> y )
  A{   a[1]*(1+(2**(1/a[10])-1)*sq(2*(x-a[2])/a[3]))**(-a[10])
    +  a[4]*(1+(2**(1/a[10])-1)*sq(2*(x-a[5])/a[6]))**(-a[10])
    +  a[7]*(1+(2**(1/a[10])-1)*sq(2*(x-a[8])/a[9]))**(-a[10])
  }A
;

FIT-FUNC PAX OF 10 PARAMETERS

\      A[1]  A[2]  A[3]  A[4]  A[5]  A[6]  A[7]  A[8]  A[9]  A[10]
\      I0   X0   HW   I0   X0   HW   I0   X0   HW   GZ
A(    0.2   7.0   0.4   0.2   5.7   0.4   0.2   5.4   0.5   2.59 )

ls

```

**Abb. 1**      *Programmtext der Modellfunktion **PAX** in der Datei **PAX.SEQ**, der mit dem Befehl **GET-FUNC PAX.SEQ** kompiliert werden kann (s. Text).*

Mit dem Wort **:** schalten Sie den Kompiler ein. Danach wird der Name der Funktion erwartet, hier **PAX**, der später den Bezug auf den Funktions-Code erlaubt. In Klammern kann ein Kommentar angefügt werden. **( -> y )** ist ein sogenannter *Stack*-Kommentar und weist darauf hin, dass die Funktion **PAX** das Ergebnis der Funktionswertberechnung auf dem *Stack* (Stapelspeicher) übergibt, so dass im folgenden Text der Funktionsvorschrift keine Zuweisung **Wert:=** nötig ist. Die Funktionsvorschrift wird in algebraischer C-Syntax zwischen den Worten **A{** und **}A** formuliert. Die Fitparameter werden mit Hilfe eines Vektors **A(n)** mit der Dimension **n:=Zahl der Fitparameter** verwaltet und auf die einzelnen Parameter indiziert zugegriffen mit der Anweisung **A[k]** mit **k:=Index des Parameters**. Dem Kompiler bekannt sind die (globalen) Variablen **x** und **y**, in denen die Koordinaten des Punktes

<sup>1</sup> Die **PAX**-Modellfunktion ist eine Summe von 3 **Fraser-Suzuki-Funktionen** [FRA69], aufgrund der Spin-Bahn-Kopplung des Xe-5p-Photolochs eine für den Xe-5p<sub>1/2</sub>- und bedingt durch die elektronische Wechselwirkung im Xe-Adsorbat zwei für den Xe-5p<sub>3/2</sub>-Photoemissionszustand entsprechend den zwei elektronischen Zuständen Xe-5p<sub>3/2</sub>(m<sub>j</sub>=±1/2) und Xe-5p<sub>3/2</sub>(m<sub>j</sub>=±3/2). Eine **Fraser-Suzuki-Funktion** simuliert die Faltung einer **Lorentz-Funktion** mit einer **Gauß-Funktion** und berücksichtigt dadurch die Verbreiterung eines Photoemissionspeaks durch die endliche Auflösung des Analysators.

der Messkurve gespeichert sind, für den gerade der Funktionswert berechnet wird. Sie sehen also, dass keine weiteren Deklarationen zur Formulierung der Funktionsvorschrift nötig sind.

Mit der Zeile

### **FIT-FUNC PAX OF 10 PARAMETERS**

teilen Sie dem Programm mit, dass die Funktion **PAX** an eine Messkurve optimal angepasst werden soll und die Zahl der Parameter der Funktion **PAX** 10 beträgt. Den Parametervektor **A(10)** initialisieren Sie mit der Zeile,

```
A( 0.2 7.0 0.4 0.2 5.7 0.4 0.2 5.4 0.5 2.59 )
```

indem Sie einfach den Wert jedes Parameters in der richtigen Reihenfolge durch Leerzeichen getrennt angeben. Mit dem Wort \ leiten Sie einen Kommentar bis Zeilenende ein. Die Kommentarzeilen erläutern die Bedeutung der Fit-Parameter. Dabei bedeutet **I<sub>0</sub>**=Intensity, **X<sub>0</sub>**=Center und **HW**=Half-Width eines 5p-Emissionspeaks von adsorbiertem Xenon.

Der **10.** Parameter **A[10]** ist die sogenannte Gauß-Zahl. Die Gauß-Zahl (**GZ**) ist ein gerätespezifischer Parameter [FRA69], der die Formveränderung eines Lorentz-Emissionspeaks durch einen Analysator mit endlichem Auflösungsvermögen angibt. Dieser Parameter **GZ** wird nicht gefittet, sondern muss einmal für alle Messungen mit der gegebenen Apparatur bestimmt werden (z.B. durch Fit eines Xe-5p<sub>1/2</sub>; der Fit konvergiert für diesen Parameter im Exponent der Funktion erst nach vielen Iterationen).

Das Wort **\s** beendet den *Interpreter/Compiler*. Alle Zeilen dahinter werden ignoriert.

Verlassen Sie den Editor mit **<F10>**. In **GET-FUNC** können Sie jetzt das Programm veranlassen, die Datei **PAX.SEQ** zu laden und so die FIT-Funktion zu kompilieren.

### **GET-FUNC**

```
new function in a file <Y>n? <y>.  
filename > {PAX.SEQ}
```

...

Zur Initialisierung der **FIT**-Routine wird in diesem Befehl automatisch **SET-FUNC** ausgeführt, in dem Sie auch den Datenspeicher mit dem Spektrum für den Fit auswählen.

Starten Sie den Fit mit **FIT**.

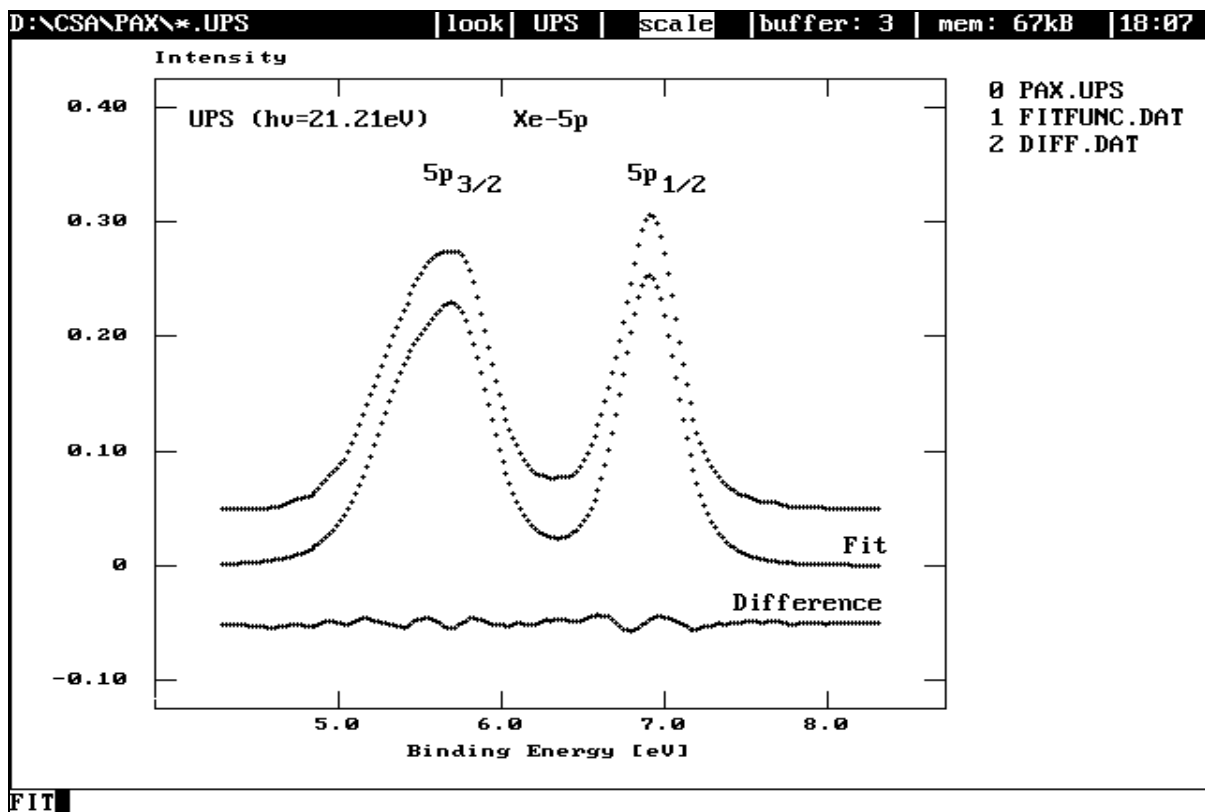
### **FIT <Ret>**

```
set limits of valid data range; from <*> <Ret> to <*> <Ret>  
parameter 1: {0.2} fit parameter <y>n? <Ret>
```

...

```
parameter 10: {2.59} fit parameter <y>n? <n>
```

Danach erhalten Sie das Ergebnis entsprechend **Abb. 2** und **Abb. 3**.



**Abb. 2** Xe-5p-Photoemissionsspektrum von Xe/Cu<sub>3</sub>Pt(111) bei 50 K (nach dem Untergrundabzug) und mit Hilfe des Fits optimal angepasster Modellfunktion **PAX** (s. Abb. 3) sowie die Differenz zwischen den beiden Kurven. (Zum Zweck einer besseren Darstellung sind die Kurven vertikal verschoben).

**Fit parameter of function PAX**

FLOAD \DATA\PAX\PAX.SEQ

a[1] = .25341E+00 ± 5 %  
 a[2] = .69055E+01 ± 0 %  
 a[3] = .40873E+00 ± 7 %  
 a[4] = .17692E+00 ± 41 %  
 a[5] = .57517E+01 ± 1 %  
 a[6] = .46117E+00 ± 23 %  
 a[7] = .13168E+00 ± 49 %  
 a[8] = .54112E+01 ± 2 %  
 a[9] = .52944E+00 ± 29 %  
 a[10] = .25900E+01 fixed

to spectrum of name PAX.UPS

**Abb. 3** Parameter der an das Spektrum in **Abb. 2** mit Hilfe des Fits optimal angepassten Modellfunktion **PAX** (**SEE-FIT** bzw. **WEDIT**).

## Ergänzung

Einfacher und schneller wären Sie in diesem Fall mit der Routine **FRASUZFIT** im *Overlay* **FITPRG**<sup>2</sup> zum Ziel gekommen, die den Fit einer Summe von **Fraser-Suzuki-Funktionen** [FRA69] an ein Spektrum erlaubt. Dort wird der Fit *spezieller* Funktionen sehr komfortabel unterstützt.

## Literatur

- [FRA69] Resolution of Overlapping Bands: Functions for Simulating Band Shapes  
R.D. Fraser and E. Suzuki, *Anal. Chem.* **41** (1969) 37.
- [NUM88] Numerical Recipes in C  
W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Cambridge University Press (1988).

---

<sup>2</sup> Die Fit-Routinen im *Overlay* **FITPRG** und im *Overlay* **USERFITPRG** basieren auf der Routine in [NUM88] nach Levenberg-Marquardt. Sie unterscheiden sich in der Bestimmung der Ableitungsterme der Fit-Funktion nach den Fit-Parametern. Im Fall der Routine im *Overlay* **FITPRG** werden die Ableitungsterme mit Hilfe der analytisch gewonnenen und kompilierten Ableitungsfunktionen berechnet, im Fall der Routine im *Overlay* **USERFITPRG** numerisch bestimmt. Dies kostet auf der einen Seite Zeit, auf der anderen Seite muss zum Fit nur die Funktionsvorschrift kompiliert werden.