

Kurzanleitung für den Einstieg in Iraf

Vers. 2.5

Kurzanleitung für den Einstieg in Iraf

Vorwort

Ursprünglich hatte ich die Absicht, lediglich eine kurze Einführung in Iraf zu verfassen, sozusagen ein Kochrezept. Nun besteht aber im Internet kein Mangel an solchen Kurzanleitungen. Hinzu kommt, daß ich die Erfahrung gemacht habe, bei selteneren Aufgaben den entsprechenden Lösungsweg stets wieder erneut finden zu müssen. Daher bin ich mit zunehmender Iraf-Erfahrung zu der Überzeugung gekommen, daß es sinnvoller ist, den Rahmen für diese Einführung etwas weiter zu stecken. Man sollte aber auch weitere Literatur heranziehen, z. B. A Beginner's to Using Iraf von Jeannette Barnes oder A User's Guide to Reducing Slit Spectra with Iraf von Phil Massey, Frank Valdes und Jeannette Barnes. Eine Übersicht über Iraf Scripts bietet An Introductory User's Guide to IRAF Scripts von Ed Anderson revised by Bob Seaman.

Ich setze beim Leser ein grundsätzliches Verständnis für den spektroskopischen Reduktionsprozess voraus. Es sei hier insbesondere auf Spectroscopic Instrumentation von Thomas Eversberg und Klaus Vollmann, Kap. 12 verwiesen. Einen guten Einblick verschafft aber auch das Iraf Manual der Universität Bonn (<https://astro.uni-bonn.de/~lboldt/files/Iraf.pdf>). Wer schon mit anderer Software Erfahrung gemacht hat, wird hier schnell die parallele Vorgehensweise insbesondere zu Midas erkennen. Trotz der grundsätzlich gleichen Methodik bietet Iraf erhebliche Vorteile, da die Arbeitsweise mehr graphisch und interaktiv orientiert ist, was für den Benutzer erhebliche Vorteile bedeutet.

Bei eventuellen Problemen findet der Anwender sehr schnell ausreichende Hilfe im Internet. Insbesondere gibt es ein Iraf Forum. Unter der Internetadresse iraf.net kann man sich die neuesten Entwicklungen ansehen.

Die vorliegende Einführung richtet sich in erster Linie an Benutzer von Spaltspektrographen wie z. B. den Lhires III. **Ich empfehle als <http://www.twilightlandscapes.com/IRAFtutorial/index.html> ergänzende Lektüre. Dort werden auch wichtige Hinweise zum Gebrauch von Linux gegeben.**

Für Echelle-Spektrographen gibt es A User's Guide to Reducing Echelle Spectra with IRAF von Daryl Willmarth und Jeannette Barnes.

In den nachfolgenden Beispielen werden die Dateinamen nach jedem Bearbeitungsschritt durch ein Prefix ergänzt. Diese Systematik hat den Vorteil, daß im Browser alle Dateien eines Bearbeitungsschritts zusammenstehen. Das normalisierte Spektrum als Endergebnis hat als Prefix ein n, ist also im Browser am Ende zu finden, wenn wir von der heliozentrischen Korrektur absehen.

Kurzanleitung für den Einstieg in Iraf

Übersicht über die Standardprozedur

Wir öffnen ein Terminal, in das wir

```
cd IRAF  
./iraf
```

eingeben. Damit ist Iraf gestartet. Gleichzeitig steht damit auch GS9 zur Verfügung.

Ein weiteres Paket apextract laden mit

```
twodspec  
apextract
```

```
imarith @in.txt - Dark.fits d//@in.txt
```

```
apall b//@in.txt clean+ background="median"  
nbzw.  
apall object.fit
```

Aperture-Daten werden in ap-datei im Verzeichnis database gespeichert

```
apall FeAr_2400_1350mm_05_09_2013_1 out=FeAr_2400_1350mm_05_09_2013_1  
ref=dgA_2400_1350mm_30_10_2013_6 recen- trace- back- intera-
```

Aperture-Daten werden in ap-datei im Verzeichnis database gespeichert

```
identify FeAr_2400_1350mm_05_09_2013_1.ms.fits
```

Datei idFeAr_2400_1350mm_05_09_2013_1.ms wird erzeugt

```
hedit combgA_2400_1350mm_20_10_2013.ms.fits REFSPEC1  
"FeAr_2400_1350mm_30_10_2013_1.ms.fits" add+ ver- show+
```

```
dispcor combgA_2400_1350mm_20_10_2013.ms.fits
```

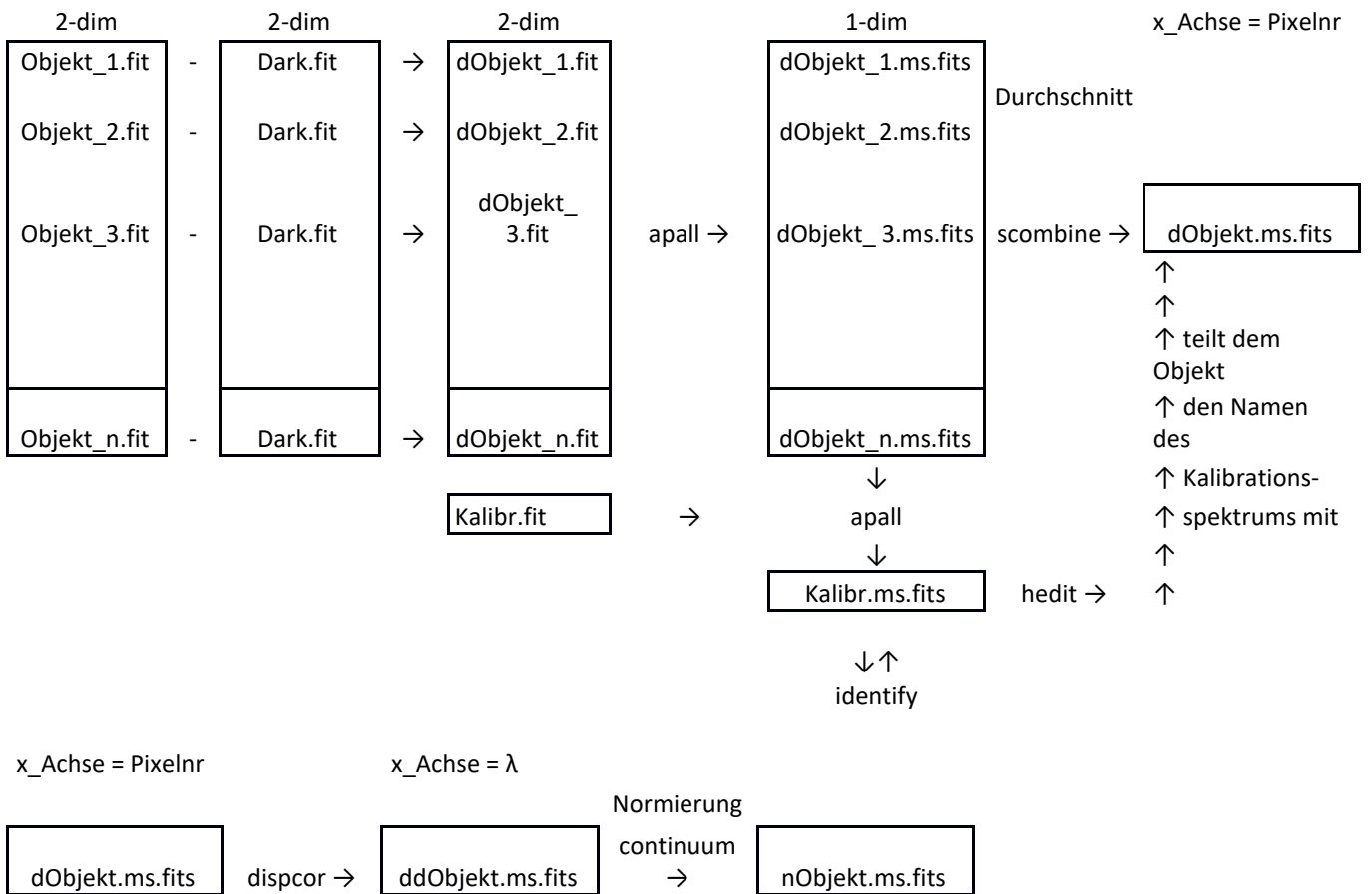
```
splot dcombgA_2400_1350mm_20_10_2013.ms.fits
```

```
continuum dcombgA_2400_1350mm_20_10_2013.ms.fits  
ngamAnd_2400_1350mm_30_09_2013.ms.fits
```

Kurzanleitung für den Einstieg in Iraf

Hat man von einem Objekt mehr als ein Spektrum, gibt es grundsätzlich zwei Wege zur Spektrenreduktion. Entweder wird jedes Einzelspektrum individuell reduziert und anschließend ein Mittelwert gebildet oder es werden gleich zu Beginn die Einzelspektren gemittelt und dann das gemittelte Spektrum reduziert. Letzteres bedeutet natürlich weniger Arbeitsaufwand, besonders wenn eine größere Anzahl von Einzelspektren vorliegt. Die erste Methode hat hingegen den Vorteil, daß man jedes einzelne Spektrum getrennt beurteilen und gegebenenfalls verwerfen kann. Im Zweifelsfall sollte man beide Methoden anwenden und die beiden Ergebnisse miteinander vergleichen.

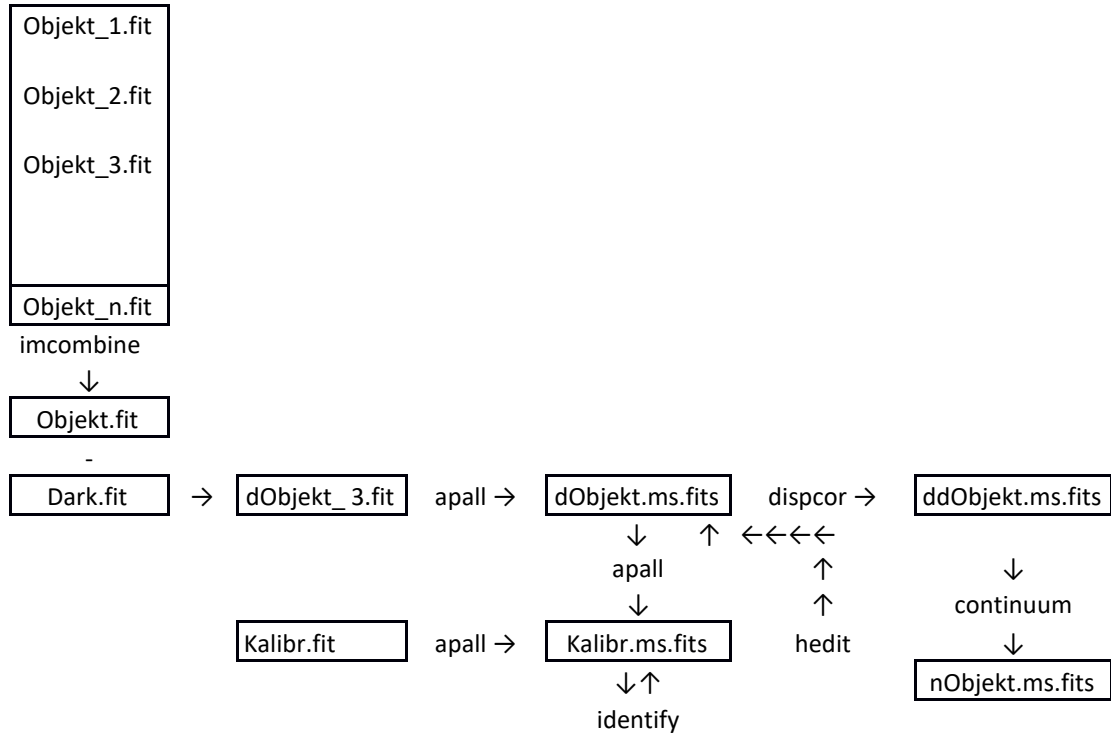
Methode 1



Kurzanleitung für den Einstieg in Iraf

Alternativ:

Methode 2 (schneller und kann im allgemeinen bevorzugt werden)



Kurzanleitung für den Einstieg in Iraf

Allgemeine Arbeitsweise

Diese Anleitung soll den Ein- oder Umstieg auf Iraf soweit es geht erleichtern. Sie soll kein umfassendes Handbuch ersetzen, sondern nur die wichtigsten Punkte in kurzer Form darlegen. Der Leser sollte zusätzlich andere Einführungen und vor allem die Hilfstexte der verwendeten und verwandter Funktionen zu Rate ziehen. Die folgende Beschreibung richtet sich nach Methode 1.

Hinweis: Für Echelles kann man im Internet eine gesonderte Anleitung finden.

Hier wird vorausgesetzt, daß Iraf einschließlich ds9 installiert ist. Unser Arbeitsverzeichnis soll IRAF heißen. Dort sollen alle weiteren Arbeitsschritte stattfinden.

Wir öffnen ein Terminal, in das wir

```
cd IRAF  
./iraf
```

eingeben. Damit ist Iraf gestartet. Gleichzeitig steht damit auch GS9 zur Verfügung.

Ein weiteres Paket apextract laden mit
twodspec
apextract

Wir haben damit mehrere Pakete in den Arbeitsspeicher geladen, die zahlreiche Funktionen, Umgebungsvariablen und auch Parameter enthalten. Die Werte der Umgebungsvariablen kann man sich mit dem Befehl

```
show
```

anzeigen lassen.

Die Werte der Paketvariablen (packages) werden mit *lparam*, also z. B.

```
lparam onedspec
```

angezeigt. Zum dauerhaften Verändern dient der Befehl **eparam**. S. auch weiter unten.

In unserem Beispiel liegen folgende Dateien vor, die mit einem Spaltspektrographen, in diesem Fall ein Lhires, aufgenommen wurden: (Gitter 2400 L/mm, Mikrometerschraube 13,50 mm)

Kurzanleitung für den Einstieg in Iraf

12 Objektdateien

```
gamAnd_2400_1350mm_30_10_2013_1.fit
gamAnd_2400_1350mm_30_10_2013_2.fit
...
gamAnd_2400_1350mm_30_10_2013_12.fit
```

1 Kalibrationsdatei

```
FeAr_2400_1350mm_30_10_2013_1.fit
```

3 Dark

```
Dark1.fit
```

```
Dark2.fit
```

```
Dark3.fit
```

Wir können zunächst die einzelnen Dateien mit ds9 inspizieren.

ds9 umfasst eine Vielzahl von Möglichkeiten, von der Farbdarstellung über Vergrößerung bis zum gleichzeitigen Betrachten von mehreren Bildern mit Blinken.

Zunächst wollen wir die drei Darks zu einer Datei zusammenlegen, wobei die den Durchschnitt bilden soll. Dazu können wir *imcombine* oder die Funktion *imsum* benutzen, die einige Parameter enthält, die wir mit

lparam imsum

lesen können. Wichtig ist hier der Parameter *option*, der als Standardwert „sum“ hat. Da wir aber den Durchschnitt der drei Darks bilden wollen, setzen wir ihn auf den Wert „average“. Dies kann wie bei jeder Iraf-Funktion auf zwei Arten geschehen:

1. lparam imsum

Die Eingabe wird mit :q geschlossen. Dann nimmt *option* dauerhaft den Wert *average* an. Mit

unlearn imsum

können die ursprünglichen Werte wieder hergestellt werden.

2. Wir fügen in die Befehlszeile noch

option=average

ein. In diesem Fall wird der Wert nur bei diesem Funktionsaufruf geändert. Viele Funktionen haben Boolesche Variablen, hier z. B. *verbose* die mit

Kurzanleitung für den Einstieg in Iraf

verbose=yes oder kurz mit **verbose+** auf ja bzw. mit **verbose=no** oder **verbose-** auf nein gesetzt werden können. Verbose heißt geschwätzig, es wird also bei „ja“ irgend ein Text auf unseren Terminal ausgegeben. Wir geben also den Befehl

imsum Dark*.fit Dark option=average

oder

imcombine Dark*.fit Dark combine=average

ein und erhalten mit Dark.fits das gewünschte Ergebnis, falls bei der Installation fits als Standardbildformat festgelegt wurde.

Kurzanleitung für den Einstieg in Iraf

Reduktion von Spektren

Im folgenden wird der Ausdruck „Aperture“ häufig gebraucht. Darunter verstehen wir bei einem Spaltspektrographen das Spektrum, bei einem Echelle das Spektrum einer bestimmten Ordnung. Aber auch ein Spaltspektrograph kann u. U. mehr als ein Objekt erfassen, z. B. bei theta Ori 1. Wir erhalten dann zwei Spektren oder auch mehr. Ein Aperture entspricht dann einem Spektrum.

Das weitere Vorgehen ist ähnlich wie bei anderen Softwarepaketen auch, was in der Natur der Sache begründet ist. Wir subtrahieren von allen Dateien unser neues Dark, um Dunkelstrom- und Ausleserauschen so gut es geht zu eliminieren. Wir können dies einzeln erledigen:

```
imarith gamAnd_2400_1350mm_30_10_2013_1.fit - Dark.fits gA1350_1
```

u.s.w.

Leider können wir nun nicht mit folgender Zeile arbeiten:

```
imarith gamAnd_2400_1350mm_30_10_2013_*.fit - Dark.fits gA1350_*
```

arbeiten, da jeder Input-Datei eine Outputdatei entsprechen muß.

Wir könnten nun alle 12 Inputdateien, durch Kommata getrennt, hintereinanderschreiben, ebenso alle Outputdateien. Eleganter ist jedoch folgender Weg:

Wir benutzen eine sogenannte @-Datei, indem wir die Namen der Inputdateien in einer Textdatei aufführen, die wir in.txt nennen wollen. Mit

```
files gamAnd*.fit > in.txt
```

schreiben wir alle 12 Objektdateien übereinander, was natürlich auch mit einem Editor möglich ist. Ebenso verfahren wir mit den Outputdateien, die

```
dgA_2400_1350mm_30_10_2013_1
```

...

```
dgA_2400_1350mm_30_10_2013_12
```

heißen sollen. Diese Datei kann durch Bearbeitung von in.txt erstellt werden. Wir speichern sie in unserem Verzeichnis unter out.txt ab.

Wir müssen nun lediglich

```
imarith @in - Dark.fits @out
```

eingeben und schon wird alles erledigt.

Kurzanleitung für den Einstieg in Iraf

Noch einfacher geht es, wenn wir folgenden Trick anwenden:

```
imarith @in.txt - Dark.fits d//@in.txt
```

Die Ergebnisdateien erhalten den gleichen Namen wie die Eingangsdateien, nur mit einem vorgesetzten d

Wir können nun wieder die neuen Dateien mit ds9 oder implot inspizieren:

```
implot dgA_2400_1350mm_30_10_2013_1
```

Der nächste Schritt ist die eigentliche Extraktion, die mit *apall* bewerkstelligt wird.

Allgemeine Beschreibung:

*Das eigentliche Spektrum nimmt auf der Objektdatei normalerweise einen quer laufenden schmalen Streifen ein, der etwas schräg verlaufen kann und häufig auch leicht gekrümmt ist. Die Helligkeit sollte in Spaltenrichtung in der Mitte ihr Maximum haben und zu den Rändern hin abfallen. Die Breite des Teilstreifens muss bestimmt werden, der durch Addition oder Mittelwertbildung zu einer Zeile zusammengefaßt wird. Außerdem müssen Bereiche definiert werden, in denen die Himmelshelligkeit bestimmt wird, die vom eigentlichen Spektrum subtrahiert werden kann. All diese Schritte sind unter der Funktion *apall* (apertures all) zusammengefaßt und können in einer graphischen Umgebung ausgeführt werden. Das Ergebnis ist das extrahierte eindimensionale, noch unkalibrierte Spektrum.*

*Parameter des Extraktionsprozesses werden im Unterverzeichnis database in einer gesonderten Datei gespeichert. Man sollte mittels *gedit* sich diese Datei anschauen, um einen Einblick in das „Innenleben“ von Iraf zu bekommen.*

Details:

Diese Funktion gehört zum Paket *apextract*, das wir zu Beginn geladen haben und das ebenfalls eine Reihe von Parametern besitzt, die für alle seine Funktionen gemeinsam sind. Darunter befindet sich die Dispersionsrichtung, die auf die y-Achse voreingestellt ist. Daher benötigen wir wieder *eparam*, um auf die x-Achse umzustellen:

eparam apextract

dispaxis= 1 setzen und mit *:q* schließen. Es handelt sich hier um eine Variable, die nicht einem Befehl, sondern dem Paket *apextract* zugeordnet ist.

Die Reduktion geschieht also mit *apall*. Vor dem ersten Gebrauch überprüfen wir die Parametereinstellungen von *apall* mit (bitte nicht erschrecken! Im Normalfall sind für uns nur wenige Parameter wichtig)

Kurzanleitung für den Einstieg in Iraf

```

lparam apall
    input = " "           List of input images
    nfind = 1            Number of Apertures to be found
automatically
    (output = "")       List of output spectra
    (Apertures = "")    Apertures
    (format = "multispec")  Extracted spectra format
    (references = "")  List of Aperture reference images
    (profiles = "")    List of Aperture profile images\n
    (interactive = yes) Run task interactively?
    (find = yes)       Find Apertures?
    (recenter = yes)  Recenter Apertures?
    (resize = no)     Resize Apertures?
    (edit = yes)      Edit Apertures?
    (trace = yes)     Trace Apertures?
    (fittrace = yes)  Fit the traced points interactively?
    (extract = yes)   Extract spectra?
    (extras = yes)    Extract sky, sigma, etc.?
    (review = yes)    Review extractions?\n
    (line = INDEF)    Dispersion line
    (nsum = 10)       Number of dispersion lines to sum or
median\n\n
    (lower = -5.)     Lower Aperture limit relative to
center
    (upper = 5.)     Upper Aperture limit relative to
center
    (apidtable = "")  Aperture ID table (optional)\n\n#
DEFAULT BACKG
    (b_function = "chebyshev")  Background function
    (b_order = 1)              Background function order
    (b_sample = "-20:-8,8:20")  Background sample regions
    (b_naverage = -100)        Background average or median
    (b_niterate = 0)           Background rejection iterations
    (b_low_reject = 3.)        Background lower rejection sigma
    (b_high_rejec = 3.)        Background upper rejection sigma
    (b_grow = 0.)              Background rejection growing
radius\n\n# APERTU
    (width = 10.)             Profile centering width
    (radius = 10.)           Profile centering radius
    (threshold = 0.)         Detection threshold for profile
centering\n\n#
    (minsep = 5.)            Minimum separation between spectra
    (maxsep = 1000.)         Maximum separation between spectra
    (order = "increasing")   Order of Apertures\n\n# RECENTERING
PARAMETERS\n
    (aprecenter = "")        Apertures for recentering calculation
    (npeaks = INDEF)         Select brightest peaks
    (shift = yes)            Use average shift instead of
recentering?\n\n#
    (llimit = INDEF)         Lower Aperture limit relative to
center

```

Kurzanleitung für den Einstieg in Iraf

(ulimit = INDEF) center	Upper Aperture limit relative to
(ylevel = 0.1) automatic wid	Fraction of peak or intensity for
(peak = yes) (bkg = yes) width?	Is ylevel a fraction of the peak? Subtract background in automatic
(r_grow = 0.) (avglimits = no)	Grow limits by this factor Average limits over all
Apertures?\n\n# TRACING	Number of dispersion lines to sum
(t_nsum = 10)	Tracing step
(t_step = 10)	Number of consecutive times profile
(t_nlost = 3) is lost bef	Trace fitting function
(t_function = "legendre")	Trace fitting function order
(t_order = 2)	Trace sample regions
(t_sample = "*")	Trace average or median
(t_naverage = 1)	Trace rejection iterations
(t_niterate = 1)	Trace lower rejection sigma
(t_low_reject = 3.)	Trace upper rejection sigma
(t_high_rejec = 3.)	Trace rejection growing radius\n\n#
(t_grow = 0.) EXTRACTION	Background to subtract
(background = "median")	Box car smoothing length for sky
(skybox = 1)	Extraction weights (none variance)
(weights = "variance")	Profile fitting type (fit1d fit2d)
(pfit = "fit1d")	Detect and replace bad pixels?
(clean = no)	Saturation level
(saturation = INDEF)	Read out noise sigma (photons)
(readnoise = "0.")	Photon gain (photons/data number)
(gain = "1.")	Lower rejection threshold
(lsigma = 4.)	Upper rejection threshold
(usigma = 4.)	Number of subApertures per Aperture
(nsubaps = 1)	
(mode = "ql")	

Kurzanleitung für den Einstieg in Iraf

Bei der derzeit aktuellen Version v2.16 ist `background` defaultmäßig auf „none“ gestellt, es fällt somit keine Subtraktion des Himmels-hintergrundes statt. Wie wir bereits wissen, kann man das mit

`eparam apall`

dauerhaft ändern. Bei der Reduktion wird unser Spektrum automatisch gesucht. Als Voreinstellung wird angenommen, daß unser 2D-Spektrum nur ein Spektrum enthält (Variable `nfind=1`). Es können mit `apall` aber auch 2D-Spektren mit mehr als einem Spektrum in einem Zug verarbeitet werde. In diesem Fall wird im Dialog `nfind` auf den entsprechenden Wert gesetzt. Der Bildausschnitt, der für die Reduktion zum 1D-Spektrum verwendet wird, heißt „Aperture“. Normalerweise enthält ein 2D-Spektrum nur eine Aperture, daher die Voreinstellung `nfind=1`. Die Apertures werden übrigens standardmäßig von unten angefangen durchnummeriert.

In jedem Fall werden alle Apertures eines 2D-Spektrums in einer Datei erfaßt. Da wir 12 Belichtungen haben, benötigen wir auch 12 unterschiedliche Dateien zur Speicherung der Apertures, die in unserem Fall

```
dgA_2400_1350mm_30_10_2013_1.ms.fits
...
dgA_2400_1350mm_30_10_2013_12.ms.fits
```

heißen. `ms.fits` (Multispec format) ist ein Dateiformat, das speziell für Iraf geschaffen wurde.

Mit der Reduktion kommen wir zu den interaktiven Funktionen, bei denen ein Fenster geöffnet wird. Grundsätzlich muß das Fenster mit einem Klick mit der linken Maustaste aktiviert werden. Durch Eingabe von „?“ werden im Terminal alle Befehle dieser Funktion angezeigt. Es ist jedoch ratsam, durch

```
help funktion > funktion.txt, also zum Beispiel
```

`help apall > apall.txt`

den ganzen Hilfetext dauerhaft in eine Textdatei zu speichern.

Wir haben nun auch wieder mehrere Möglichkeiten:

```
1. apall dgA_2400_1350mm_30_10_2013_*.fits clean+
background="median"
```

oder

```
2. apall @out clean+ background="median"
```

```
3. apall b//@in.txt clean+ background="median"
```

Kurzanleitung für den Einstieg in Iraf

Erst wird man gefragt, ob man Apertures finden will (yes), und anschließend wie viele. Die Voreinstellung ist yes bzw. 1, so daß man nur zweimal die Entertaste drücken muß. Gibt man einen anderen Wert ein, wird der beim nächsten Funktionsaufruf als Voreinstellung verwendet. Danach geht ein Fenster auf, indem der Intensitätsverlauf auf der mittleren Pixelspalte als Kurve wiedergegeben wird.

Festlegen der Breite des nutzbaren Spektrums:

Als erstes muß mit dem Cursor auf der Bildfläche die linke Maustaste (LM) gedrückt werden, um die Graphik zu aktivieren. Mit den Tasten w LM e LM e kann ein Ausschnitt aus der gesamten Graphik gewählt werden, wobei vor den beiden Maustasten der Cursor erst auf die linke untere Ecke und nach Eingabe von e auf die rechte obere Ecke des gewünschten Ausschnitts positioniert wird. Man sieht über jedem Spektrum einen Balken, der anzeigt, welcher Bereich der mittleren Pixelspalte für die Reduktion genutzt wird. Unten erscheint eine Anzeige mit Mittelpunkt, unterer und oberer Grenze dieses Bereichs. Auf Wunsch kann z. B. mit

```
:center 570
:low -8
:upper 8
```

dieser Bereich beliebig verändert werden. Bei mehreren Spektren kann ich mit '+' und '-' zwischen den einzelnen Spektren hin und her wechseln.

Festlegen des Bereichs für die Himmelshelligkeit:

Nach Eingabe von b erscheint ein enger Ausschnitt mit zwei Balken rechts und links unten. Der Ausschnitt wird durch die Variable

```
b_sample = "-10:-6,6:10"
```

festgelegt. Mir war er zu eng und ich habe mit eparam apall einen größeren Ausschnitt definiert. Dabei sind die Minuszeichen vor den beiden ersten Zahlenwerten wichtig.

Dies kann auch durch die Eingabe von w,a (ganzes Bild zeigen) und anschließend w,e,e (Bildausschnitt festlegen) während der Reduktion gemäß dem aktuellen Bedarf geschehen.

Die vorhandenen Balken können mit 't' gelöscht werden und auf jeder Seite mit jeweils zweifachen LM s und entsprechender Cursorposition neu definiert werden. Mit 'q' kehren wir in das erste Bild zurück.

Fitten des nutzbaren Spektrums:

Ein weiteres 'q' führt zum nächsten Schritt. In der unteren Zeile erscheinen drei Fragen, die wir alle mit 'yes' beantworten. Es erscheint eine dicke, durch '+' markierte Linie, die den Verlauf der Mittellinie unseres Spektrums wiedergibt. Sie wird im allgemeinen etwas gekrümmt sein, was durch Unregelmäßigkeiten des Gitters und

Kurzanleitung für den Einstieg in Iraf

durch die Optik bedingt ist. Weiter sieht man eine dünne gestrichelte Linie, die meistens etwas von der dicken Linie abweicht und einen linearen Fit darstellt, da `t_order` defaultmäßig auf 2 eingestellt ist. `t_order` gibt die Anzahl der Terme eines Fittingpolynoms an. Wenn `t_order = 2` ist das Fittingpolynom also eine lineare Funktion. Bei Bedarf geben wir `:funct spline3` ein und erzeugen mit `'f'` ein neues Fit. Bei Splines ändert `t_order` seine Bedeutung. Es gibt jetzt die Anzahl der Splinstücke, also die Anzahl der Polynome an. `Spline3` ist ein Spline dritter Ordnung, der zunächst nur aus zwei Polynomen besteht, da die Variable `t_order` auf den Wert 2 eingestellt ist. Sind wir mit unserem neuen Fit nicht zufrieden, können wir mit

```
:order 3
```

die Anzahl der Polynome auf drei erhöhen. Mit `'f'` wird wieder ein neuer Fit erzeugt. Wir fahren solange fort mit der Erhöhung von `t_order` fort, bis wir mit dem Ergebnis zufrieden sind. In der Praxis sollte man aber Werte ≤ 3 nehmen. Dies muß Aperture für Aperture und durchgeführt werden, wobei wir uns durch die Fragen in der unteren Zeile leiten lassen. Nach jedem Aperture drücken wir die `'q'`-Taste, um zum nächsten zu gelangen. Zum Schluß werden wir gefragt, ob die oder das Aperture in die Datenbank im Unterverzeichnis `database` eingetragen werden soll. Wir antworten mit `'yes'`, ebenso auf die Frage, ob die Extraktion durchgeführt werden soll. Zum Schluß haben wir mit einem weiteren `'yes'` die Möglichkeit, uns das oder die auf einem Bild vorhandenen Spektren anzuschauen. Dieser Prozess muß Bild für Bild durchgeführt werden.

In unserem Fall haben wir am Ende die oben genannten zwölf Dateien

```
dgA_2400_1350mm_30_10_2013_1.ms.fits
```

```
...
```

```
dgA_2400_1350mm_30_10_2013_12.ms.fits
```

Im Falle nur eines Bildes wird der nächste Arbeitsgang übersprungen. Um die 1D-Spektren in den zwölf Dateien zu einer einzigen Datei zusammenzufassen, bemühen wir die Funktion `imcombine`. Sie hat nur eine einzige Datei als Output, die wir

```
combgA_2400_1350mm_20_10_2013.ms.fits
```

nennen. Wieder ist die Endung `ms.fits` wichtig. Unsere Eingabe lautet also

```
scombine dgA_2400_1350mm_30_10_2013_*.ms.fits  
dgA_2400_1350mm_30_10_2013.ms.fits group=apertures combine=average  
reject=none weight=exposure
```

Bei mehr als einem Aperture pro Datei werden die einzelnen Apertures entsprechend ihrer Nummer getrennt übereinandergelegt. Wir brauchen uns also keine Sorge zu machen, daß dabei etwas durcheinander gerät.

Kurzanleitung für den Einstieg in Iraf

Wir wollen eine nach Belichtungszeit gewichtete Durchschnittsbildung, weshalb die letzten beiden Variablenzuweisungen hinzugefügt werden.

Bearbeitung des Kalibrationsspektrum:

Das Kalibrationsspektrum ist ebenfalls zweidimensional. Wir müssen exakt den gleichen Streifen extrahieren, den wir auch beim Objektspektrum ausgesucht haben. Dies wird über die oben erwähnte Datei im Verzeichnis database gewährleistet. Auch hier kann man sich den Inhalt der Datei mittels gedit ansehen.

Das Kalibrationsspektrums bearbeiten wir entsprechend mit

```
apall FeAr_2400_1350mm_05_09_2013_1 out=FeAr_2400_1350mm_05_09_2013_1  
ref=dgA_2400_1350mm_30_10_2013_6 recen- trace- back- intera-
```

Die ersten zwei Parameter dürften wohl inzwischen klar sein. Die Variable references, kurz ref verweist aus den oben genannten Gründen auf das Objektspektrum. Wenn die beiden ersten Parameter gleich sind, erhalten wir automatisch ein ms-Spektrum. Wir wählen ein Objektspektrum aus der Mitte unserer Aufnahmeserie. Die Namen von Funktionen und Variablen können abgekürzt werden, solange dadurch keine Mehrdeutigkeiten entstehen. Es folgen vier weitere Variable. recen- besagt, daß das Kalibrationsspektrum nicht erneut zentriert wird, trace-, daß kein erneuter Fit erzeugt wird, back-, daß kein Himmelshintergrund subtrahiert wird und schließlich intera-, daß dieser Task nicht interaktiv laufen soll.

Wir haben nun wieder eine neue Datei

```
FeAr_2400_1350mm_05_09_2013_1.ms.fit
```

erzeugt. Die Linienidentifikation erfolgt mit

```
identify FeAr_2400_1350mm_05_09_2013_1.ms.fits
```

Allgemeines:

Die Parameter werden in einer speziellen Datei im Verzeichnis database gespeichert. Dies garantiert, daß man zu einem späteren Zeitpunkt die Kalibrierung verbessern kann. Dazu muß allerdings die entsprechende Datei auch im Verzeichnis database vorhanden sein.

Dazu müssen wir zunächst einen Linienkatalog festlegen, etwa coordlist="linelists\$fear.dat\$", falls man eine Eisen-Argon HKL benutzt. *linelists* ist eine Umgebungsvariable und hat den Wert iraf/iraf/noao/lib/linelists, der sich aber je nach Installation hiervon unterscheiden kann. In diesem Verzeichnis befinden sich einige Linienkataloge für gebräuchliche Kalibrationslampen, unter anderem Thorium-Argon, Eisen-Argon.

Kurzanleitung für den Einstieg in Iraf

Es erscheint das 1D-Kalibrationsspektrum. Wir wählen zwei oder drei bekannte Linien mit jeweils einem `l`m und nachfolgenden `'m'` aus, in dem wir in der Fußzeile die jeweilige Wellenlänge näherungsweise eingeben. `identify` sucht sich dann aus dem Katalog die nächstliegende Linie selber aus. Wir geben `'f'` ein und sehen die linearen Abweichungen. Bei nur drei oder vier Linien sind die natürlich sehr gering. Mit `'q'` kehren wir zum Spektrum zurück und geben `'l'` ein. Nun werden alle identifizierten Linien angezeigt und wir können erneut mit `'f'` die linearen Abweichungen sowie das RMS (root mean square) ablesen. Sollten wir uns bei einer Linie in der Wellenlänge geirrt haben, kann die mit `l`m und `'d'` gelöscht werden. Wenn wir mit unserer Kalibration zufrieden sind geben wir wieder `'q'` ein, um zum Spektrum zurückzukehren. Bei mehr als einer Aperture ist diese Prozedur entsprechend oft zu wiederholen. Ein erneutes `'q'` beendet den Vorgang und wir werden auf dem Terminal gefragt, ob das Ergebnis im Ordner `database` abgespeichert werden soll. Wir antworten mit `yes` und erzeugen damit die o.g. Textdatei. Da unsere Eingaben in einer getrennten Datei `database/idFeAr_2400_1350mm_30_10_2013_1.ms` gespeichert werden, können wir sogar zu einem späteren Zeitpunkt die Kalibration verbessern. Wenn wir danach in einem weiteren Versuch die Kalibration verbessern wollen, können wir an dieser Stelle entscheiden, ob wir beim alten Ergebnis bleiben oder es durch die neue Kalibration ersetzen wollen.

Mit `k` kommen wir zum nächsten Aperture, mit `j` zum vorherigen Aperture.

Soll die Kalibration durch Wasserdampflinien erfolgen, wird an Stelle von `FeAr_2400_1350mm_05_09_2013_1.ms.fit` einer der obigen 1D-Objektspektren genommen. Dann muß `ftype=absorption` gesetzt werden!

Zusammenführen von Objekt- und Kalibrationsspektrum:

Die wichtigsten und arbeitsaufwendigsten Schritte sind hiermit erledigt. Bevor wir jedoch das Objektspektrum kalibrieren können, ist noch etwas Verwaltungsarbeit nötig. Es muß die Verknüpfung zwischen Objekt- und Kalibrationsspektrum über die Variable `REFSPEC1` im Header von `combgA_2400_1350mm_20_10_2013.ms.fits` hergestellt werden. Diese Variable ist aber im Header nicht vorhanden und wird durch

```
hedit combgA_2400_1350mm_20_10_2013.ms.fits REFSPEC1  
"FeAr_2400_1350mm_30_10_2013_1.ms.fits" add+ ver- show+
```

neu erzeugt und als Wert wird der Namen des Kalibrationsspektrums eingetragen. `add+` bedeutet, daß die Variable zum Header hinzugefügt wird, `ver-`, daß keine Verifikation der Datenbankoperation stattfindet und `show+`, daß der Erfolg angezeigt wird.

Schließlich wird das Objektspektrum mit (dispersion correction)

```
dispcor combgA_2400_1350mm_20_10_2013.ms.fits
```

Kurzanleitung für den Einstieg in Iraf

kalibriert. Auf die Frage nach dem Namen des kalibrierten Spektrums geben wir

```
dcombgA_2400_1350mm_20_10_2013.ms.fits
```

ein. Es ist aber auch möglich den Namen des kalibrierten Spektrums als zweite Variable einzugeben.

```
dispcor  combgA_2400_1350mm_20_10_2013.ms.fits  
dcombgA_2400_1350mm_20_10_2013.ms.fits
```

Mittels der Parameter w1, w2, dw und nw kann das 1-d Spektrum noch nachträglich gebinnt werden. Anschauen und bearbeiten geschieht mit

```
splot  dcombgA_2400_1350mm_20_10_2013.ms.fits
```

Mit `,o`` (=overlay) und anschließenden `,g`` können wir eine zweite Kurve im selben Fenster abbilden. Es öffnet sich ein Eingabefenster, in der Name der zusätzlichen Kurve eingegeben werden muß. Mit

```
:/color N  (0<=N<=9)
```

wird die Farbe festgelegt. = ist schwarz, 1 weiß, so daß normalerweise ein Wert > 1 gewählt wird.

Normalisierung:

Mit `'t'` können wir das Spektrum schon hier normalisieren. Dies kann aber auch mit

```
continuum dcombgA_2400_1350mm_20_10_2013.ms.fits  
ngamAnd_2400_1350mm_30_09_2013.ms.fits
```

geschehen, wobei der zweite Parameter wieder den Namen des Resultats wiedergibt. Auf die Frage nach interaktiver Bearbeitung geben wir `yes` ein. Wieder wird die Graphik erst mit der linken Maustaste angeklickt. Anschließend können wir wie bei `apall` oder `identify` mit `:func spline3, :order 3, 'f'` usw. die Approximation verbessern

Den Mittelwert, RMS, S/N usw. erhalten wir mit:

```
splot , 2x m.
```

Mit `'w'` und anschließend `2x'e'` können wir einen Teil heranzoomen. Mittels `'i'` und anschließender Namenseingabe wird der Bildteil als neuer Bild gespeichert.

Kurzanleitung für den Einstieg in Iraf

Heliozentrische Korrektur

Für die heliozentrische Korrektur wird

rvcorrect

benötigt. Die Variable `observatory` wird auf `obspars` gesetzt:

observatory set obspars

Man gibt entsprechend dem Hilfetext die Beobachtungszeit und die Objektkoordinaten in eine Textdatei ein. Hier das Beispiel aus dem Hilfetext:

```
cl> type rv.obs
1987 10 21 11:00:24 3:36:15 0:22:04
1987 10 21 11:08:00 8:19:35 -0:51:35
1987 10 21 11:15:47 8:35:12 6:40:29
1987 10 21 12:12:10 9:13:20 61:28:49
1987 10 21 12:16:03 9:27:48 9:07:08
1987 10 21 12:20:43 9:50:45 -6:06:58
1979 3 25 11:22:59 16:07:28 -23:37:49
```

Dies kann auch mittels **hselect** geschehen:

```
hselect spec*.fit $DATE_OBS > date.dat
```

Dann müssen nur noch hinter jedem Datum die Koordinaten eingegeben werden.

```
cl> rvcorrect f=rv.obs > rv.dat

cl> type rv.dat
##      HJD          VOBS    VHELIO     VLSR    VDIURNAL    VLUNAR
VANNUAL  VSOLAR
 2447089.96358      0.00     11.07     -2.74     -0.189     0.008
11.246  -13.808
 2447089.96296      0.00     28.05     13.56      0.253     0.010
27.790  -14.498
 2447089.96813      0.00     29.04     16.64      0.262     0.011
28.770  -12.401
 2447090.00834      0.00     22.06     25.26      0.114     0.010
21.940    3.200
 2447090.00884      0.00     27.70     18.55      0.250     0.009
27.438   -9.152
 2447090.01129      0.00     23.99     13.50      0.275     0.007
23.704  -10.484
 2443957.97716    -67.50    -41.37    -31.48      0.002     0.012
26.117    9.884
```

Kurzanleitung für den Einstieg in Iraf

Um die Wellenlängenscala zu korrigieren, nimmt man VHELIO mit umgekehrten Vorzeichen:

```
cl>dopcor ngamAnd_2400_1350mm_30_09_2013.ms.fits  
hcgamAnd_2400_1350mm_30_09_2013.ms.fits 41.37 isvel+
```

Numerischer Werte in Spektrum verwandeln und umgekehrt

Dies kann nötig werden, wenn wir die Responsefunktion unserer Ausrüstung benötigen. Ich beschreibe hier einen vereinfachten Weg, der allerdings wahrscheinlich für eine Flußkalibration nicht ausreichen wird. Wir benutzen ein Flat, das mit einer Halogenlampe o. ä. Erzeugt wurde, deren Temperatur wir abschätzen können. In der Regel beträgt diese zwischen 2800 und 3200 K. Das Flat muß mit dem gewünschten Spektralbereich erzeugt werden. Handelt es sich um eine Datei im Multispecformat, muß sie erst in eine normale fits-Datei umgewandelt werden:

scopy flat.ms.fits flat.fits

Als nächstes wird das Spektrum von einer Bilddatei in eine Textdatei verwandelt:

wspectext flat.fits flat.txt header+

Den Header brauchen wir noch und kopieren ihn in eine weitere leere Textdatei. Anschließend wird er in ngamAnd.txt gelöscht, um diese Datei leichter in Excel oder LibreOffice calc einlesen zu können. Wir haben dann zwei Spalten: Die linke gibt die Wellenlängen an, die rechte die Intensitäten. Die rechte Spalte ersetzen wir im Spreadsheet durch die Werte unserer Planckkurve. Je nach Einstellung müssen in der Textdatei vorher die Dezimalpunkte gegen Dezimalkommata ausgetauscht werden. Nach Abspeichern des fertigen Spreadsheet als Planck.txt müssen natürlich die Kommata wieder durch Punkte ersetzt werden. Der Header wird wieder am Beginn einkopiert und das ganze wird mit

rspectext Planck.txt Planck.fits

in eine Fit-Datei transformiert. Um wieder eine Multispecdatei zu erhalten, benötigen wir noch einen letzten Schritt:

mkms Planck.fits Planck.ms.fits

Um die Responsekurve zu erhalten, führen wir die Division

sarith Planck.ms.fits / flat.ms.fits korr.ms.fits

aus. Wenn wir unsere nichtnormierten, aber kalibrierten Spektren damit multiplizieren, erhalten wir ein Spektrum, das dem "wahren" Spektrum zumindest nahe kommt:

**sarith ngamAnd_2400_1350mm_30_09_2013.ms.fits * Korr.ms.fits
kgamAnd_2400_1350mm_30_09_2013.ms.fits**

Nachträgliches Binnen

Will man eine Aufnahme nachträglich binnen, so kann man die Funktion

blkavg imagein imageout 2 2

verwenden. Bei diesem Beispiel wird ein 2x2 Binning durchgeführt. *1-dimensionale Spektren können mit `dispcor` unter Verwendung des Parameters `nw` gebinnt werden.*

Trocknen u. SpektroTools

Unter Iraf kann man mit dem Befehl `telluric` Spektren „trocknen“, d. h. die Wasserdampflinien werden in gewissen Wellenlängenbereichen so gut es geht eliminiert. Trotzdem ist es wünschenswert, auch SpektroTools nutzen zu können. SpektroTools kann Multispec-Dateien verarbeiten, jedoch nur eingeschränkt. Solange man gemäß Methode 1 reduziert, treten keinerlei Probleme auf. Geht man jedoch nach Methode 2 vor, erhält man eine mehrdimensionale `ms`-Datei. Jedoch kann man mit `scopy` eine Kopie mit einem Aperture erstellen, die von SpectroTool unterstützt wird. Hier ein Beispiel für eine 3-dimensionale Datei:

```
scopy ngamCas_2400_2075_24_11_2014.ms.fits  
ngamCas_2400_2075_24_11_2014.fits aperture=1 band=1
```

Wenn man nicht sicher ist, welche Dimension das Spektrum hat, schaut man es sich mit `splot` an. In der Kopfzeile kann man die Dimension erkennen.

Mehrere Spektren in einer Graphik zeigen

Bei zwei Spektren zeigt man eines mit `splot` an. Nach anklickender Graphik gibt man zuerst `o` und dann `g` ein. Danach wird man in der Fußzeile aufgefordert, den Namen des zweiten Spektrums einzugeben. Die Farbe des zweiten Spektrums wird miteinander

```
:/color N 1 <= N <= 9
```

bestimmt.

Bei mehr als zwei Spektren sollte die Funktion `specplot` gewählt werden. Diese ermöglicht die Eingabe der Spektrennamen mittels `@`-Datei, die durch den Befehl `files` erstellt werden kann.

Kurzanleitung für den Einstieg in Iraf

```
files ngamCas*.ms.fits > in.txt  
specplot @in.txt sysid- yscale+
```

In diesem Plot kann man an jeder Stelle Labels eintragen. Mit

```
:.snap eps
```

kann der Plot als encapsulated PostScript Datei gespeichert werden. Das funktioniert auch mit den Befehlen

```
implot
```

und

```
splot (s.o.)
```

Kalibrationsspektrum ohne Objektspektrum untersuchen (Spektren ohne Teleskop)

Die nachfolgenden Bemerkungen gelten für alle Spektren, die direkt ohne Verwendung eines Teleskops gewonnen wurden, z. B. auch Sonnenspektren. Normalerweise wird das Kalibrationsspektrum mit dem Befehl *apall* erzeugt, wobei mit dem Parameter *ref* auf das Apertur eines Objektspektrums verwiesen wird. Möchte man sich jedoch einfach ein eindimensionales Kalibrations erzeugen, scheidet dieser Weg aus. Man kann dann durch

```
scopy
```

beliebige Zeilen aus dem 2-dimensionalen Kalibrationsspektrum extrahieren, z. B.

```
nsum=10  
scopy FeAr_2400_1.fit FeAr_2400_1.ms.fits aperture=1260
```

Dann werden symmetrisch um die Zeile 1260 standardmäßig 10 Zeilen addiert und als eindimensionales Spektrum unter dem Namen *FeAr_2400_1.ms.fits* gespeichert. *nsum* ist eine Paketvariable von *onedspec*.