N1 - Sternspektren verschiedener Spektraltypen (BACHES)

Ziel des Versuches ist es eine Übersicht der Spektren verschiedener Spektraltypen zu gewinnen. Hierfür sollen anhand einer zur Verfügung gestellten Liste von Sternen (Koordinaten und scheinbare Magnituden) vier Sterne spektroskopiert werden, die während der Nacht gut zu beobachten sind. Die so gewonnenen Spektren gilt es dann anhand der Linien im Spektrum sowie der Form des Kontinuums einem Spektraltyp zuzuordnen. Aus der Verschiebung der Spektrallinien im Vergleich zu deren Ruhewellenlänge kann die Radialgeschwindigkeit der Sterne mittels des Dopplereffektes bestimmt werden.

Beobachtung

Der Versuch erfordert Nachtbeobachtungen am OST der Uni Potsdam mit dem Baches-Echellespektrographen. Der Versuchshintergrund und die Technik zur Aufnahme der Spektren werden im Rahmen von Vorträgen im Praktikumsseminar besprochen. Eine Liste mit geeigneten Objekten wird von den Betreuern zur Verfügung gestellt.

Beachte: Folgende Aufnahmen müssen für *jeden* Stern gemacht werden:

- das eigentliche Sternspektrum
- Darkframes für das Sternspektrum
- mindestens ein Kalibrationsspektrum mit der eingebauten ThAr-Kalibrationslampe

sowie einmalig:

- wenigstens drei Kalibrationsspektren mit der kontinuierlichen Kalibrationslampe
- Darkframes für das kontinuierliche Kalibrationsspektrum

Die QHY268M-Kamera **sollte** mit einem **2×2-Binning** betrieben werden.

Die Kalibrationsaufnahmen werden später benötigt, um von einer Pixelskala auf Wellenlängen umzurechnen bzw. um Geräteartefakte herauszurechnen.

Vorbereitungen: Auswahl und Prüfung der Daten

Die ersten zwei Schritte sind das Einloggen in den Praktikumspool und das Kopieren der Beobachtungsdaten (FITS-Files), inklusive Darkframe- und Kalibrations-Aufnahmen aus dem Verzeichnis `~/data/<datum>` ins eigene Verzeichnis `~/data_reduction/`. Zur Vereinfachung der Auswertung sollte für jeden Stern ein separates Verzeichnis erstellt werden indem man die Aufnahmen abgelegen kann. Des weiteren wird es nötig sein, die gemachten Aufnahmen nach Darks, Flatdarks, Flats (Aufnahmen der kontinuierlichen Kalibrationslampe), ThAr-Aufnahmen sowie Sternaufnahmen in Unterordner zu sortieren. Eine mögliche Ordnerstrucktur könnte z.B. so aussehen:

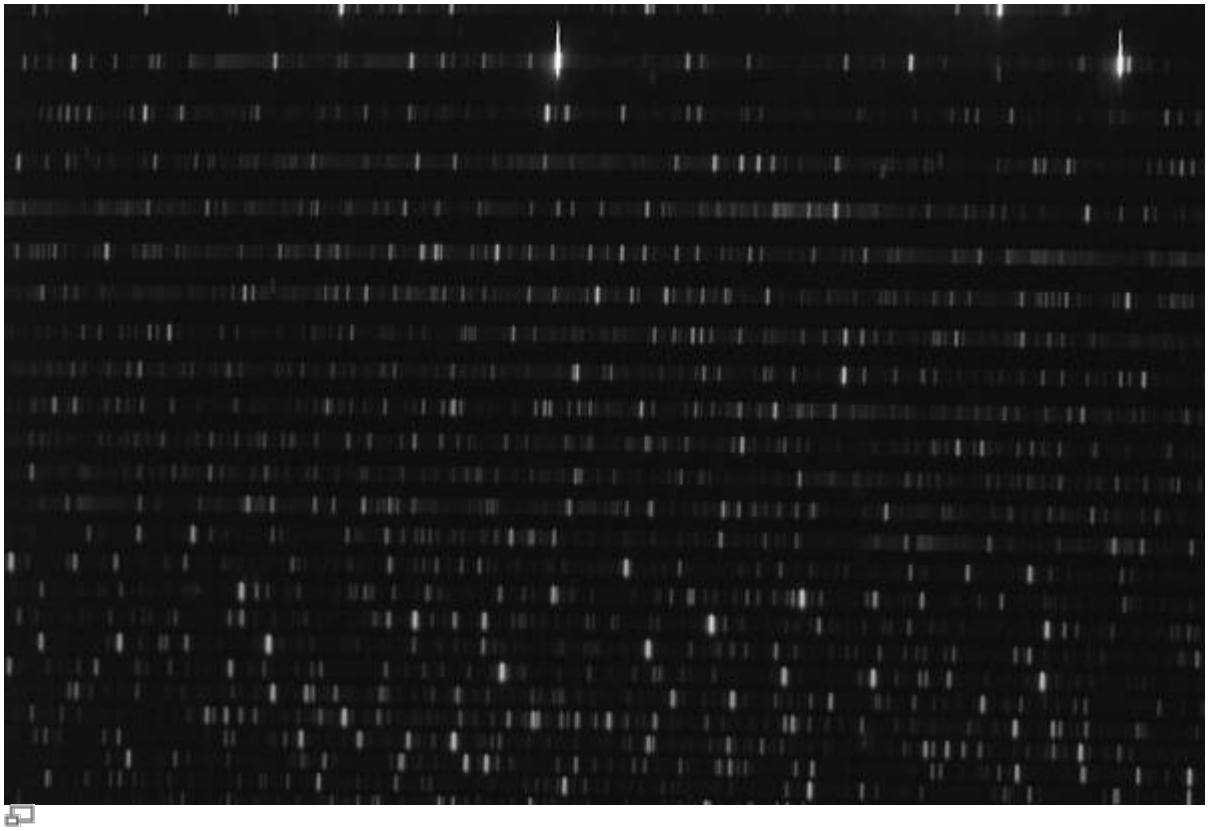
```
Stern_1/
```

```
    Darks/
    Flatdarks/
    Flats/
    ThAr/
    Stern_1/
 Stern_2/
    Darks/
    Flatdarks/
    ...
```

Es gibt verschiedene Tools, um die im FITS-Format abgelegten zweidimensionalen CCD-Bilder (Images) zu betrachten. Ein Beispiel ist *ds9*, dass im Terminal mittels

```
ds9 filename.fit
```

gestartet werden kann. Zuerst sollte die Ausrichtung der Aufnahmen überprüft werden. Hierfür bietet sich die Aufnahme der Thorium-Argon-Lampe (ThAr-Lampe) an, welche mit folgender ThAr-Aufnahme verglichen werden kann. Die korrekte Ausrichtung ist entscheidend für das spätere Ergebnis, da die Auswerteroutinen erwarten das die Wellenlänge des Spektrums von unten links nach oben rechts ansteigt. Sollte festgestellt werden, dass die Ausrichtung in einer der Aufnahmen nicht korrekt ist, muss diese in **allen** Aufnahmen, die für die Auswertung relevant sind korrigiert werden.



ThAr-Aufnahe mit der ST-8 so wie man sie in *ds9* und *MIDAS* sehen sollte.

Datenreduktion

Die Skripte für die Auswertung sind auf dem Praktikumsrechner im Ordner `~/scripts/n1_baches`

zu finden. Man kopiere sich das Skript `1_masterimages.py` in das lokale Verzeichnis, in dem man die Auswertung des jeweiligen Sterns vornehmen möchte (z.B. `Stern_1`). Mit diesem Skript kann man unter anderem das Masterdark und das Masterflat erstellen, welche in den folgen Schritten benötigt werden. Des Weiteren addiert (stacked) dieses Skript alle Sternaufnahmen, falls während der Beobachtungen mehr als eines aufgenommen wurden. Es bietet darüber hinaus die Möglichkeit die erstellten Aufnahmen im Nachhinein zu binnen, falls dies während der Beobachtung vergessen wurde. Auch schneidet es nicht benötigte Teile der erstellen Aufnahmen ab, die die weitere Analyse umständlicher machen würden. Zu allerletzt sollte in diesem Skript noch angeben werden, ob die Aufnahmen gespiegelt werden müssen (siehe vorheriger Schritt), sodass auch dies korrigiert werden kann. Der Konfigurationsbereich des Skripts sieht wie folgt aus:

```
##############################################################################
 ####           Configuration: modify the file in this section
####
##############################################################################

 ###
 #   Path to the directories with the images
 #
 #   Darks:
 path_darks = '?'

 #   Flat darks:
 path_flat_darks = '?'

 #   Flats:
 path_flats = '?'

 #   Thorium Argon exposures:
 path_thars = '?'

 #   Spectra:
 path_spectra = '?'

 #   Output directory for the reduced flats. The master files will be saved
in
 #   the current directory.
 out_path = 'output'

 ###
 #   Flip images? Possibilities: True and False
 #
 flip_bool = False

 ###
 #   Bin the images? Possibilities: True and False
 #
 bin_bool = False

 #   Binning factor
 bin_val = 2
```

```
 ###
 #   Trim images to remove non essential parts and thus simplify MIDAS
handling
 #   Possibilities: True and False; Default: True
 #
 trim_bool = True
```

In `path_darks`, `path_flat_darks`, `path_flats`, `path_thars` und `path_spectra` sind die entsprechenden Unterordner einzutragen, die oben im Abschnitt *Vorbereitungen* erstellt wurden. `out_path` ist das Verzeichnis, in dem das Skript Zwischenergebnisse ablegt. Die finalen Dateien (**master_dark.fit**, **master_flat.fit**, **master_thar.fit**, **master_spectrum.fit**) werden im lokalem Verzeichnis abgelegt, sodass sie für die folgenden Schritte einfacher zur Verfügung stehen.

Sollen die Aufnahmen gespiegelt werden muss noch `flip_bool` auf `True` gesetzt werden. Sollen die Aufnahmen auch gebinned werden ist ebenfalls `bin_bool` auf `True` zu setzen. Der Binningfaktor kann über die Variable `bin_val` bestimmt werden.

Datenauswertung

Grundlagen

Für die Auswertung steht die freie Datenreduktionssoftware *MIDAS* zur Verfügung welche von der ESO entwickelt wurde. Eine Kurzanleitung zu *MIDAS* kann hier gefunden werden. Ausführlichere Informationen kann man dieser Anleitung entnehmen.

Hinweis: *MIDAS* kann aus den Standardpaketquellen installiert werden, um die Datenreduktion am eigenen Rechner durchzuführen. Allerdings werden dazu zwei angepasste Routinen benötigt, andernfalls führt die Datenreduktion zu einer Fehlermeldung. Die angepassten Routinen können bei den Betreuern angefragt werden.

Von der Konsole aus lässt sich *MIDAS* mittels

```
inmidas
```

starten. Die für die Auswertung relevanten Befehle werden mittels

```
set/cont baches
```

geladen. Die Standardbefehle der zugrundeliegenden Bashshell (wie z.B. ls, cd, …) können innerhalb von *MIDAS* mittels eines vorgestellten $-Zeichen aufgerufen werden.

Auffinden der Ordnungen und Wellenlängenkalibration

Auffinden der Ordnungen

Das Auffinden der einzelnen Ordnungen, die Identifikation der Emissionslinien im ThAr-Spektrum sowie die entsprechende Zuordnung zwischen den Pixeln und der Wellenlänge wird durch das *MIDAS*-Skript `calibrate/baches` erledigt. Der Aufruf kann wie folgt aussehen:

```
calibrate/baches master_flat.fit master_thar.fit 26 26 24
```

Die Parameter haben die folgende Bedeutung:

1. `Flat field`: Name des Flatfields – hier muss der Name des Master-Flatfields angeben werden
2. `Calibration lamp`: Name des ThAr-Spektrums – hier muss der Name der Master-ThAr-Aufnahme angeben werden
3. `Num. of orders`: Anzahl der zu suchenden Ordnungen (empfohlene Wert: 26)
4. `Slit order width`: Breite der Ordnung in Pixel (empfohlene Wert: 26)
5. `Slit extraction width`: Breite des zu extrahierenden Bereichs in Pixel (empfohlene Wert: 24)

Die ersten zwei Parameter sind obligatorisch. Nach Aufruf des Skriptes werden die übergebenen Parameter (bzw. die Standardwerte) noch einmal im Terminal ausgegeben:

```
PARAMETERS FOR THIS CALIBRATION:
================================
Flat field  = master_flat.fit
Calibration lamp = master_thar.fit
Calibration table = thar.fit
Num. of orders  = 0026
Slit order width = 0026
Slit extraction width = 0024
Offset from center slit = 0.00000E+00
Tolerance on RMS = 1.00000E+00
Polynomial degree  = 0004
```

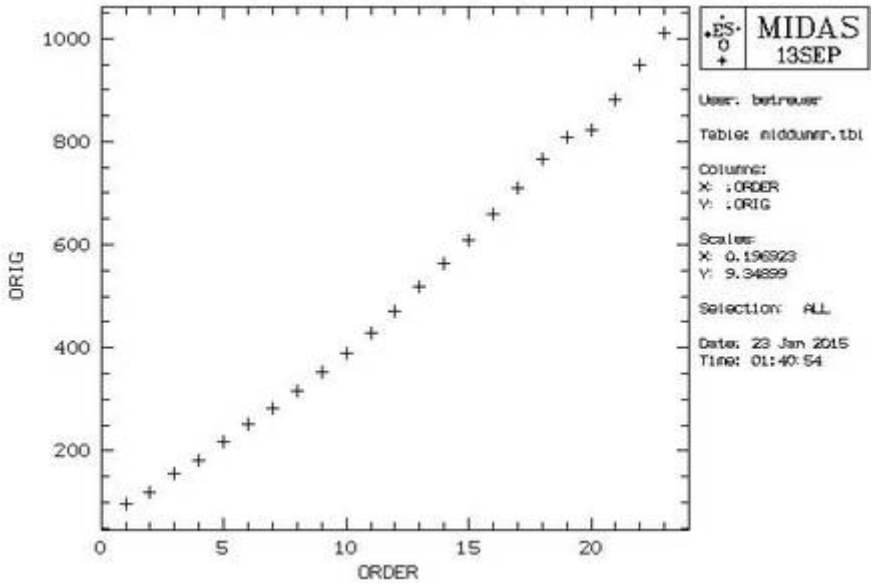
Anhand dieser Auflistung sollte nochmal kontrolliert werden ob alles korrekt ausgewählt wurde. Die weiteren oben nicht aufgezählten Parameter können in den meisten Fällen auf den Standardwerten belassen werden.

Nach einer Bestätigung mit y startet das Programm mit der Suche nach den einzelnen Ordnungen im Flatfield. Mit der ST-8 können maximal 24 Ordnungen aufgenommen werden, welche von dem Skript auch standardmäßig gesucht werden. Jedoch reicht der Fluss im Blauen oft nicht aus um eine automatische Identifikation der zwei kurzwelligsten Ordnungen zu gewährleisten. Falls es dem Skript nicht gelingt alle unter `Num. of orders` angegeben Ordnungen zu identifizieren wird in der Regel folgender Text im Terminal

```
|************************************************
 ****  Warning: Defaults in the order detection ***
 ****   Table middummr.tbl has been plotted     ***
 ************************************************
 *       Read the help of DEFINE/HOUGH,           *
 *       Section: In Case of Emergency            *
 ************************************************
 * If you modify interactively middummr.tbl,      *
 * restart later with DEFINE/HOUGH P4=FOLLOW      *
```

```
************************************************
* Now you can:                                 *
*  1 - Go on                                   *
*  2 - Stop here                               *
************************************************
```

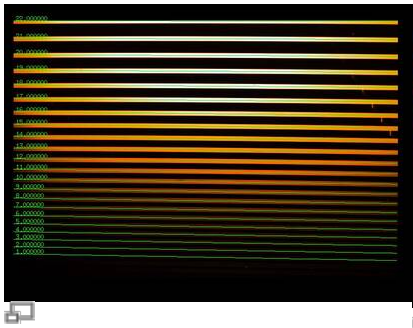
ausgegeben. Des Weiteren wird folgende Grafik in einem extra Fenster angezeigt:



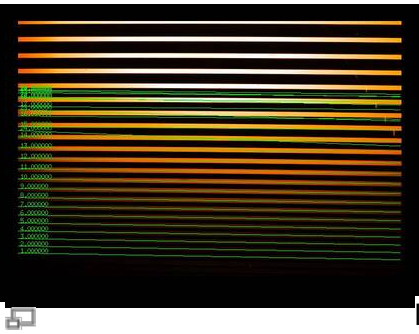
In diesem Fall sollte man das Skript mittels der Eingabe einer 2 beenden und mit einer reduzierten Zahl an Ordnungen erneut ausführen z.B.:

```
calibrate/baches flat.fit thar.fit 23 20 20
```

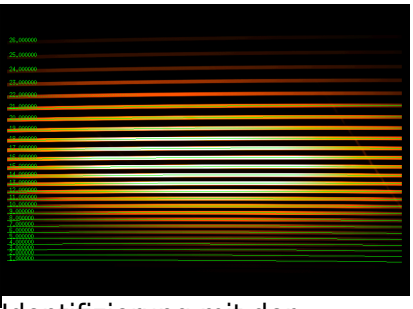
Es hat sich bewährt die Anzahl der Ordnungen schrittweise zu reduzieren bis ein erfolgreiches Auffinden möglich ist. Ist dies erfolgt wird das Flatfield sowie die identifizierten Ordnungen (siehe unten) in einem externen Fenster angezeigt. Hier sollte kontrolliert werden ob die Markierungen auch zentral auf den einzelnen Ordnungen liegen und ob diese horizontal verlaufen. Die linke Abbildung zeigt einen erfolgreichen Testlauf, wohingegen die mittlere Abbildungen eine fehlgeschlagene Identifizierung zeigt. Die rechte Abbildung zeigt wiederum eine Flatfield-Aufnahme mit der QHY268M, bei der es vorkommen kann, das *MIDAS* nicht alle identifizierten Ordnungen nicht fertig zeichnet. An den Ordnungsnummern, kann aber zumindest erahnt werden, dass alle Ordnungen erfolgreich identifiziert wurden.



Erfolgreiche Identifizierung



Fehlgeschlagene



Identifizierung mit der QHY268M

der Ordnungen Identifizierung der Ordnungen

Neben der Anzahl der Ordnungen sollten auch die Parameter für die Breite der Ordnungen (`Slit order width`) sowie die Breite des zu extrahierenden Bereichs innerhalb der einzelnen Ordnungen (`Slit extraction width`) spezifiziert werden. Die Defaultwerte von 20 Pixel sind nur eingeschränkt mit der BACHES-ST8-Kombination verträglich, daher müssen diese Werte von Fall zu Fall angepasst werden.

Troubleshooting: Bei der Datenreduktion kann folgende Fehlermeldung auftreten

```
(ERR) Problems in executing /usr/lib/eso-midas/1SSEP/stdred/exec/necdef.exe
```

Die Ursache dieses Fehlers konnte bisher nicht eindeutig identifiziert werden. Dieser Fehler könnte dadurch verursacht werden, dass die hellste Ordnung am Rand des Bildes liegt. Versucht diese abzuschneiden oder zusätzliche Pixel zur Aufnahme hinzuzufügen. Beachtet, dass Ihr auch alle anderen Bilder entsprechend zuschneiden bzw. erweitern müsst.

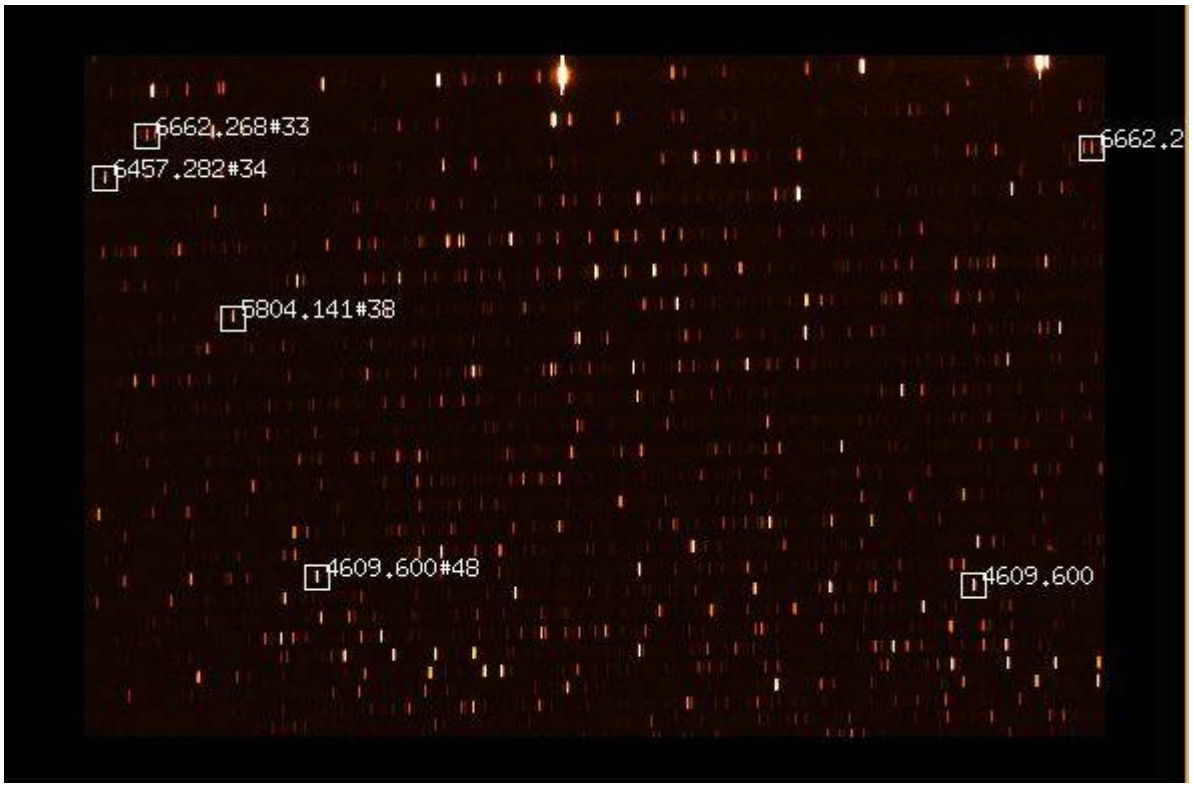
Wellenlängenkalibration

Nachdem das Identifizieren der Ordnungen abgeschlossen ist kann durch bestätigen mittels y die Wellenlängenkalibration gestartet werden.

Im ersten Schritt wird nach Emissionslinien gesucht die innerhalb der zuvor identifizierten Ordnungen liegen. Bei unserer Konfiguration sollte es dem Skript möglich sein deutlich über 1500 Linien zu finden. Sind es weniger Linien ist die Schärfe der ThAr-Aufnahme wahrscheinlich unzureichend. Ist das Identifizieren der Linien abgeschlossen sollte man ein Fenster vor sich auf dem Bildschirm sehen, das der folgenden Abbildung sehr ähnliche ist. Jedes weiße Kästchen ist eine von *MIDAS* erkannte Linie.

ThAr-Spektrum auf dem alle gefundenen Linien markiert sind.

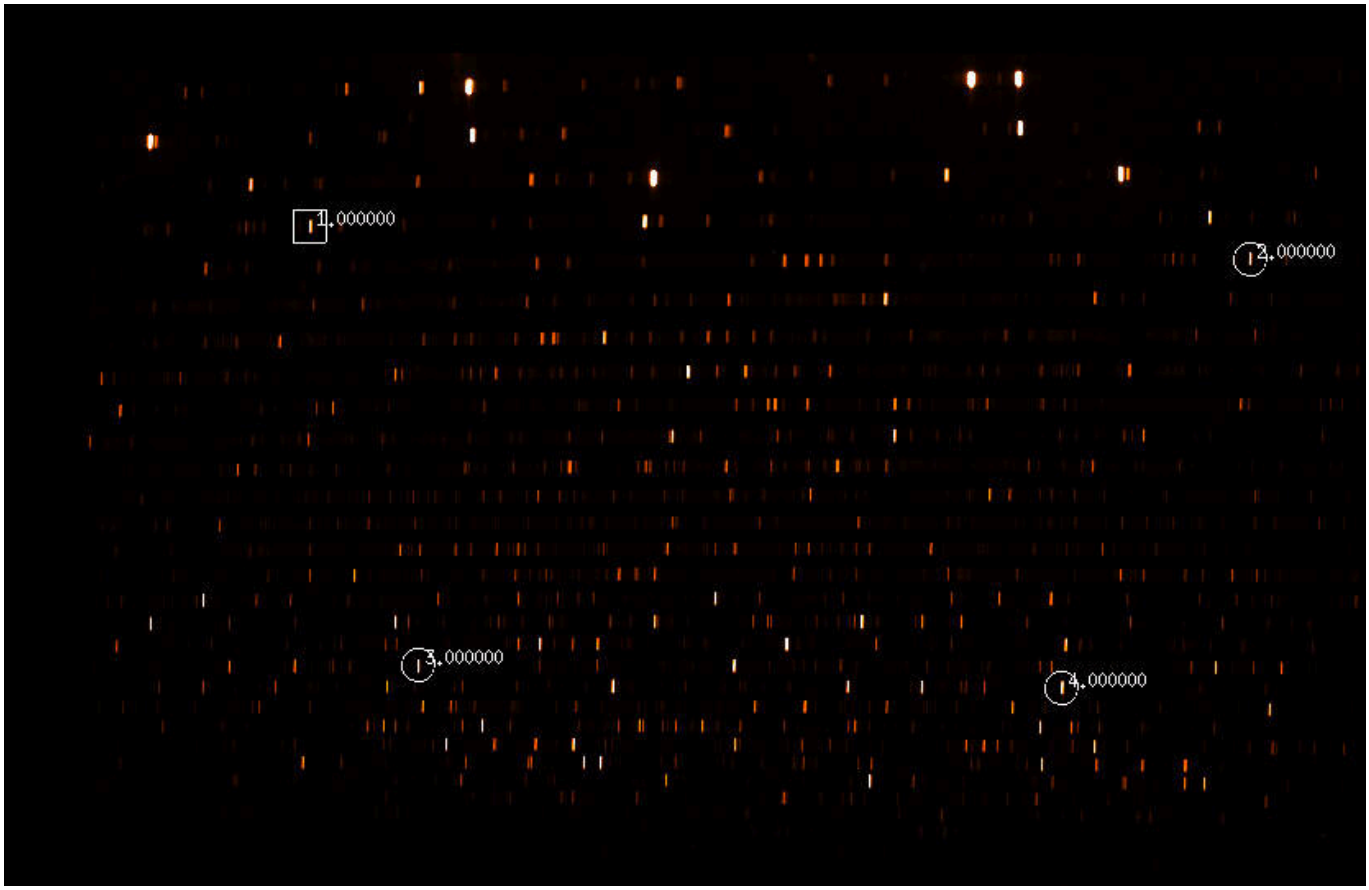
Nach einer weiteren Bestätigung mittels y müssen auf der nächsten Abbildung des ThAr-Spektrums zwei Linienpaare gefunden und per Maus markiert werden, damit das Skript alle Linien korrekt zuordnen und den Überlappungsbereich der Ordnungen feststellen kann. Ein Linienpaar sollte im roten Bereich und das andere im blauen Bereich des aufgenommenen Spektralbereichs liegen. Bewährt haben sich die Linien bei 6677.282Å und 4609.6Å. Alternativ zur ersten Linie kann aber auch die 6457.282Å oder die Linie 6662.268Å verwendet werden. Dies kann z.B. nötig sein, wenn */MIDAS* eine Linie nicht erkannt hat. In diesem Fall ist die entsprechende Linie in der Abbildung oben auch nicht durch eine weiße Box markiert. Um das Auffinden der Linien zu vereinfachen wird neben dem Fenster in dem die Linien markiert werden müssen auch die folgende Abbildung samt einigen Beispiellinien angezeigt:



ThAr-Spektrum mit einigen Beispielslinien

Im Vergleich zu diesem Beispiel sind auf unseren Aufnahmen in der Regel deutlich mehr Linien zu sehen, da unsere Kamerachips größer sind als die, die für die Erstellung dieses Beispiels verwendet wurden. Die 6677.282Å ist hier z.B. gar nicht eingezeichnet. Bei dieser Linie handelt es sich um die helle Linie rechts neben der 6662.268Å. Dies wird auf der folgenden Abbildung deutlich, auf dieser die beiden Standardlinienpaare markiert sind.

Auf einer entsprechenden Grafik in *MIDAS* müssen diese Linienpaare dann per Maus in der gegeben Reihenfolge markiert werden. Falls nötig kann die Größe der Auswahlbox für die einzelnen Linien über die Pfeiltasten auf der Tastatur geändert werden. Es ist entscheidend das die entsprechenden Linien in einer kontinuierlichen Sequenz markiert werden. Hat man alle 4 Linien markiert muss man die Auswahl mit einem Rechtsklick in das Fenster bestätigen.



ThAr-Spektrum mit den zwei markierten Linienpaaren

Anschließend ist die Nummer der Ordnung, die Wellenlänge des ersten Linienpaars sowie die Wellenlänge des zweiten Linienpaars in das Terminal einzugeben. Falls wie oben vorgeführt die Standardlinienpärchen ausgewählt wurden, liegt die erste markierte Linie in der Ordnung 33. Die Wellenlänge dieser Linien beträgt 6677.282Å. Falls die 6457.282Å verwendet wurde, dann muss als Ordnung 34 angegeben werden. Die Wellenlänge des zweiten Linienpaars beträgt 4609.6Å. Das Terminal müsste dann folgendermaßen aussehen:

```
Enter absolute order number of first pointed line (square mark) : 33
Sequence no. 0001, Order no. 0033. Enter wavelength : 6677.282
Sequence no. 0003, Order no. 0048. Enter wavelength : 4609.6
```

Es ist ausreichend die Ordnung für eine Linie anzugeben, da die Software sich die Ordnung für die anderen Linien leicht selber ausrechnen kann. Darauf folgend wird automatisch nach einer Lösung für die Wellenlängenkalibration gesucht.

Fehlerbehebung

Problem 1: Es kann sein, dass *MIDAS* bereits nach Eingabe einer der Wellenlängen mit einem Fehler ähnlich dem Folgenden abbricht:

```
   *** Sorry. Cannot identify feature 0002              ***
```

Dies liegt daran, dass eine der markierten Linien zuvor nicht von *MIDAS* als Linie erkannt wurde (in dem Beispiel hier ist es die 2. Linie) und daher z.B. oben in der Grafik mit den weißen Kästchen auch nicht markiert ist. In dem Fall muss ein alternatives Linienpaar genutzt werden.

Problem 2: Manchmal erscheint folgende Warnmeldung:

```
 *** Initial, coarse estimate ***
 Mean pixel width      : 9.99039E-02 wavel. units
 RMS error of relation : 1.37550E-01 wavel. units
 Maximum allowed error : 9.99039E-02 wavel. units
 This accuracy appears insufficient to attempt automatic line
identification.
 Please check your identifications.
  ************************************************************
  *             Wavelength Calibration Diagnosis             *
  ************************************************************
  *          Linear fit of : m * lambda = b * xr + a         *
  * A = 218935.70420 B=      3.89221 RMS =       29.363      *
  ************************************************************
  *    Parameter    *    Value    *    Optimum   *    RMS     *
  ************************************************************
  *    Order number *       33    *        33    *      29.36 *
  *  Rotation angle *    -1.95540 *     -2.97270 *      10.35 *
  *  Binning factor *     1.00000 *      1.23330 *      10.35 *
  *  Ident. 1 and 2 *  6677.28200 *   6682.09003 *       9.36 *
  *  Ident. 3 and 4 *  4609.60000 *   4606.28318 *       9.36 *
  ************************************************************
 However the accuracy may be still acceptable.
 Start anyway (y/n, default no) ?
```

Diese Meldung kann in der Regel ignoriert werden, da entgegen der Warnung die Wellenlängenkalibration dennoch genau ist. Um fortzufahren einfach y eingeben und mit ENTER bestätigen.

Anschließend kann noch die Flatfield-Korrektur vorbereitet werden. Hierfür die folgende Abfrage mit y beantworten:

```
 Do you want flat correction [yn] (y)?
```

Im weiteren Verlauf kann noch eine Kalibrierung des Flusses mittels eines Standardsterns durchgeführt werden. Diese Option benutzen wir im Praktikum in der Regel nicht. Daher kann die entsprechende Abfrage:

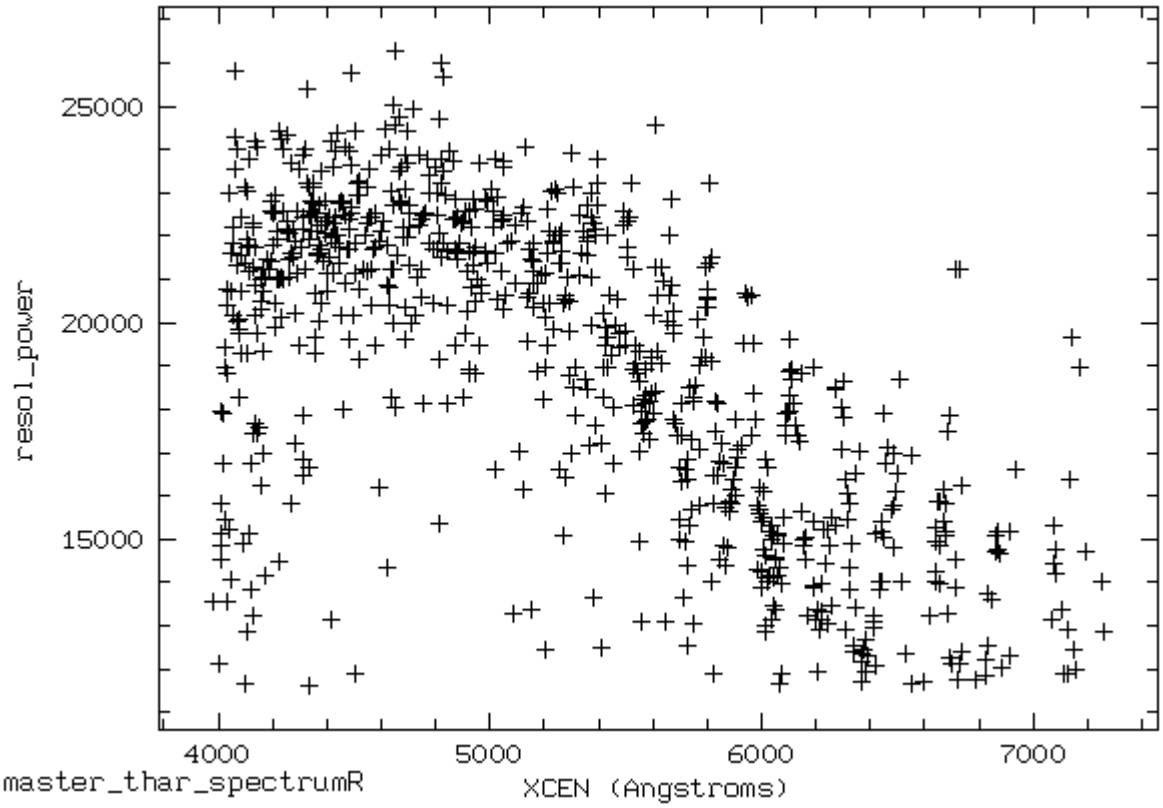
```
 baches_calib: Do you want to correct flux with a standard star [yn] (y)?
```

mit einem n beantwortet werden.

Die nächste Abfrage, bezüglich der Schwarzkörper-Korrektur, kann in der Regel wiederum mit y beantwortet werden:

```
 baches_calib: Do you want to correct flux with master_flat as a blackbody
[yn] (y)?
```

Im letzten Schritt der Kalibration muss man dann noch entscheiden, ob das mittlere Auflösungsvermögen errechnet sowie eine Grafik mit dem wellenlängenabhängigen Auflösungsvermögen ($R = \frac{\Delta \lambda}{\lambda}$) erstellt werden soll, welche z.B. so aussehen kann:



Wellenlängenabhängiges Auflösungsvermögen (resolving power)

Pipeline

Konvertierung von FITS im bdf (optional)

In einigen Fällen, z.B. bei Ordnerwechseln, kann es nötig sein, dass mache FITS-Dateien (z.B. das Masterflat) noch in bdf-Dateien umgewandelt werden müssen. Hierfür benötigen wir wieder *MIDAS*, welches mittels

```
inmidas
```

geöffnet werden kann. Mit dem Befehl

```
indisk/fits name.fit name.bdf
```

wir die jeweilige FITS-Datei in eine bdf-Datei konvertiert.

In aller Regel sollte dieser Schritt jedoch nicht nötig sein.

Falls *MIDAS* seit dem letzten Schritt in der Anleitung geschlossen wurde, muss dieses wieder mittels

```
inmidas
```

gestartet und die Einstellungen für den BACHES mittels dem Befehl

```
  set/cont baches
```

geladen werden. Mittels

```
 pipeline/baches master_spectrum.fit
```

kann anschließend die Pipeline gestartet werden, welche neben der Flatfieldkorrektur, die Extraktion der Spektren aus den einzelnen Ordnungen, deren Wellenlängenkalibration mittels den zuvor erstellten Kalibrationsdaten sowie die anschließende Vereinigung zu einem durchgehenden Spektrum übernimmt.

Troubleshooting

Wenn der folgende Fehler auftritt

```
(ERR) SCFOPN: starname.bdf - FRMNAC
(OSY) MID_ACCFRM:
Frame not accessible: Check the name of the frame you use in the given
context.
```

Überprüft die vorige Zeile `compute_fits.prg:…` *MIDAS* konvertiert das Spektrum automatisch in eine .bdf-Datei, hat aber ein Problem mit gewissen Sonderzeichen. Das erzeugte File heißt dann anders als *MIDAS* es intern erwartet und läuft daher in einen Fehler. Eine Umbenennung des Spektrums schafft Abhilfe. Problematisch ist u.A. das Minus (-).

Zur Überprüfung des Ergebnis lässt sich das kombinierte Spektrum plotten (**eine elegantere Darstellungsmöglichkeit ist unten beschrieben**). Hierzu sind folgende Befehle in das Terminal einzugeben:

```
 crea/grap
 plot/axes 4000,8000 0,10 0 "Wellenlänge" "Relative Intensität"
 overplot/row master_spectrum_wrm.fit
```

Der Plotbereich (zweite Zeile) muss unter Umständen angepasst werden. Dabei ist der Syntax wie folgt:

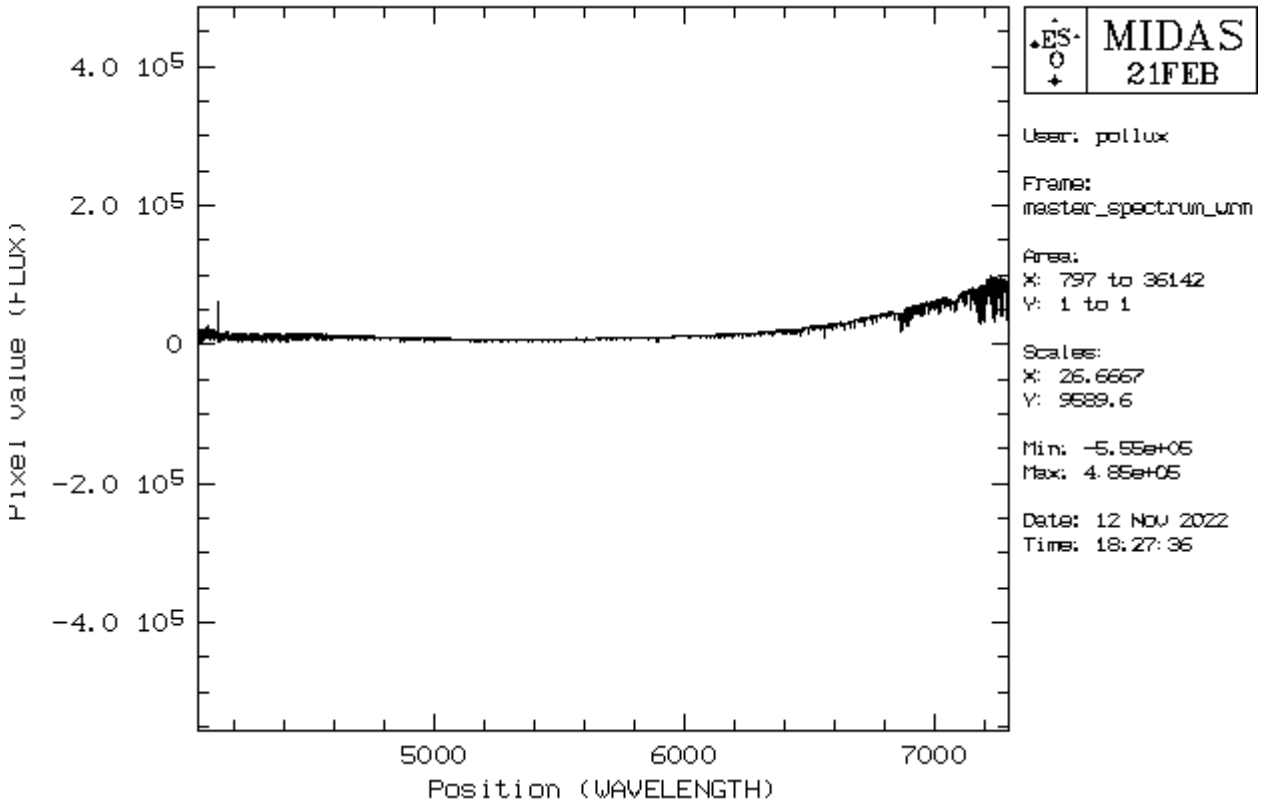
```
 plot/axes xmin, xmax, ymin, ymax, Beschriftung x-Achse, Beschriftung y-
Achse
```

4000 - 8000 Angstrom sind der typische Wellenlängenbereich des Spektographen. Werden kleinere Werte gesetzt kann auch in das Spektrum hineingezoomt werden.

Mittels

```
 copy/graph postscript
```

wird das Spektrum als Postscript abgespeichert werden. Die Datei muss anschließend noch umbenannt werden, da der obige Befehl nur Dateien namens `postscript` erzeugen kann. Auf diese Weise erhät man z.B. Abbildungen wie die folgende, welche ein “Mondspektrum” zeigt.



Spektrum von Mu Pegasi (G8 III)

All die kleinen Features sind kein Rauschen sondern reale Absorptionslinien!

Bestimmung der Radialgeschwindigkeit

Aufgrund des Dopplereffekt führt die Radialgeschwindigkeit der Sterne, die entweder auf uns zu oder von uns weg gerichtet ist, zu einer Verschiebung der Wellenlänge des emittierten Lichtes. Dies kann unter zur Hilfenahme von Spektrallinien (in der Regel Absorptionslinien) gemessen werden.

Falls *MIDAS* in der Zwischenzeit geschlossen wurde, muss dieses zuerst wieder geöffnet und das vollständig kalibrierte Spektrum geplottet werden:

```
crea/graph
plot master_spectrum_wrm.fit
```

Das folgende Kommando kann benutzt werden, um einen bestimmten Bereich in x-Richtung (xlow < x < xhigh) oder y-Richtung (ylow < y < yhigh) auszuwählen und so auf eine Spektrallinie zu zoomen:

```
set/gra xa=xlow,xhigh
set/gra ya=ylow,yhigh
plot master_spectrum_wrm.fit
```

Hierfür gilt es geeignete Spektrallinien auszuwählen, deren Ruhewellenlängen bekannt sind.

Tip: Falls hierbei Schwierigkeiten auftreten kann es hilfreich sein, sich für eine erste Abschätzung eine Wasserstofflinie anzuschauen. Da diese Linien oft sehr breit sind, wird die so bestimmte Radialgeschwindigkeit jedoch meist nicht sehr genau sein. Ein weiteres in vielen Fällen geeignetes Linienpaar würde man auch sehen, wenn man den Spektrografen auf Straßenlampen richtet, obwohl diese Art der Lampen zunehmend weniger werden. Alternativ kann man aber auch zum Abschnitt Erstellung der Plots springen und sich das Spektrum aufgeteilt in kleinere Abschnitte anzuschauen. Auf diese Weise bekommt man unter Umständen bereits einen besseren Eindruck des möglichen Spektraltyps und potentiell geeigneter Linien.

Für die ausgewählten Linien gilt es nun die zentrale Wellenlänge zu bestimmen. Um dies zu erreichen können die Linie mit einer Gaußfunktion gefitted werden:

```
center/gauss gcursor,2 ? absorption
```

Um den eigentlichen Fit durchzuführen muss mit der linken Maustaste einmal links und einmal rechts der Absorptionslinie geklickt werden. Um den Fit abzuschließen muss anschließend noch die rechte Maustaste betätigt werden. Im Terminal werden dann die Fitparameter ausgegeben. “CENTER” gibt die zentrale Wellenlänge an, während “FWHM” (Full Width at Half Maximum) die Breite der Gaußfunktion charakterisiert. Wiederhole die Messung mehrere Male um sicher zu sein, dass die Grenzen des Gaußfits gut gesetzt sind und um ein Gefühl für den Fehler dieser Methode zu erlangen. Dies sollte insgesamt für wenigstens fünf Absorptionslinien durchgeführt werden. Sind Unterschiede in der Genauigkeit der Messung zwischen verschiedenen Linien zu erkennen? Diskutiere die Gründe hierfür in deinem Protokoll. Mit Hilfe der Dopplerformel kann anschließend aus der Verschiebung der Linien im Vergleich zu deren Ruhewellenlänge die Relativgeschwindigkeit des Sterns berechnet werden. Die Ruhewellenlänge kann über die NIST-Datenbank ermittelt werden. Eine Anleitung hierfür haben wir natürlich auch.

Baryzentrische Korrektur

Die Radialgeschwindigkeit wird in der Regel in Bezug auf das Baryzentrum, den Massenmittelpunkt des Sonnensystems, angegeben, um den Effekt der Eigenbewegung der Erde herauszukorrigieren. Die hierfür nötige Korrektur kann mit Hilfe des Skripts `2_barycentric_velocity_correction.py` berechnet werden. Angegeben werden müssen noch die Koordinaten des Objektes, die Beobachtungszeit, die Zeitzone und die Koordinaten des Beobachtungsortes:

```
##########################################################################
######
 ########################### Script Parameters
#################################
##########################################################################
######

 ################   REPLACE EVERY "?" WITH ITS RESPECTIVE VALUE
################

 ##### Target coordinates #####
 # enter the object's ICRS coordinates
 right_ascension                    = '?h ?m ?s'            # as hour
```

```
angle: hour | minute | second
 declination                        = '+?deg ?m ?s'         # in deg |
arcmin |arcsec

 ##### Observation time #####
 observation_date                   = '????-??-?? ??:??:??' # enter the
observation date formatted as 'YYYY-MM-DD HH:MM:SS'.
                                                              Be careful,
to enter the correct day if observations were made
                                                              past
midnight!
 time_zone                          = ?                     # type +1 for
the central european time zone or +2 when time was
                                                              switched to
daylight saving

 ##### Observatory #####
 observatory_latitude               = ?                     # values in
degrees, north is positive
 observatory_longitude              = ?                     # values in
degrees, east is positive
 observatory_height_above_MSL       = ?                     # enter in
meter
```

Als Resultat bekommt man die baryzentrische Geschwindigkeitskorrektur in km/s.

Erstellung der Plots

Für die optische Aufbereitung der Spektren steht im Ordner `~/scripts/n1_baches/` das Skript `3_printmasterplot.py` zur Verfügung, welches sowohl das von *MIDAS* erstellte Gesamtspektrum als auch die einzelnen Spektren der unterschiedlichen Ordnungen aus den FITS-Dateien extrahieren und anschließend darstellen kann.

Die folgenden Optionen müssen in dem *Python*-Skript gesetzt werden:

Variable	**Beschreibung**
`File_orders`	Name der mit *MIDAS* erstellten FITS-Datei welche die einzelnen Ordnungen enthält (in der Regel master_spectrum_wr.fit)
`File_merged`	Name der mit *MIDAS* erstellten FITS-Datei welche das zusammengefügte Spektrum enthält (in der Regel master_spectrum_wrm.fit)
`objectname`	Name des Objektes
`radial_velocity`	Gemessene Radialgeschwindigkeit in km/s (ohne baryzentrische Korrektur)
`ions`	Ionen für die Linienidentifikation eingezeichnet werden sollen
`manual_lines`	Mittels dieser Variable können selbständig weitere Linienidentifikation hinzugefügt werden. Es muss hierbei jeweils ein Identifikationsstring wie z.B. “HeI”, die Wellenlänge und der Ausrichtungsparameter (“center”, “left” oder “right”) gesetzt werden.

Variable	Beschreibung
line_lower_continuum	Diese Variable bestimmt, wie tief Linien im Vergleich zum Kontinuum eingesenkt sein müssen, sodass Linienidentifikation für die entsprechenden Linien dargestellt werden. Je höher dieser Wert, desto weniger Linienidentifikation werden dargestellt, da dann nur um so stärke Linien dieses Kriterium erfüllen. **Hinweis:** Ist die Radialgeschwindigkeit falsch und hier ein Wert größer Null gesetzt, dann funktioniert die Linienidentifikation oft nicht. Daher sollte diese Variable auf Null gesetzt werden, falls die Radialgeschwindigkeit unbekannt ist.

```
##########################################################################
 ####         Configuration: modify the file in this section
####
##########################################################################

 #   Name of file with individual orders
 File_orders = "master_spectrum_wr.fit"

 #   Name of file with merged spectrum
 File_merged = "master_spectrum_wrm.fit"

 #   Name of the object
 objectname = "?"

 ###
 #   Radial velocity [km/s]
 #       The specification of the radial velocity is necessary for the
 #       line identification to work correctly (see below).
 radial_velocity = 0.


 ###
 #   Line identifications
 #
 #   Ions for which line markers are to be drawn.
 #   Example: ["HI", "FeI", ...]
 ions = []

 #   Add lines that are not in the default database
 #   Format: {"Element descriptor": [[wavelength, alignment parameter]]}
 #           alignment parameter possibilities: "center", "left", "right"
 manual_lines = {"Example Element": [[0.,"center"]], "Example Element 2":
[[0.,"left"]]}

 #   Percent the line flux must be lower than the continuum
 line_lower_continuum = 3.
```

Durch das Skript werden standardmäßig zwei PDF-Dateien erstellt: `spectrum_panels_MIDAS-merged_<objectname>.pdf` und `spectrum_total_MIDAS-merged_<objectname>.pdf`. Die Datei mit `total` im Namen zeigt das Spektrum in einem einzigen Plot. Da aufgrund des großen

Spektralbereichs in diesem Plot oft kaum Details zu sehen sind, wird in der zweiten Datei (mit `panels` im Namen) das jeweilige Spektrum verteilt über mehrerer “Panels” dargestellt, was wiederum die Identifizierung einzelner Linien deutlich vereinfacht.

Für die etwaige spätere Nachnutzung wird das extrahierte Spektrum als CSV-Datei in den Ordner `output` abgelegt. Dabei wird folgendes Format für den Dateinamen verwendet: `<object_name>_<merge_type>_<panel_number>.csv`.

Zwei Beispielplots für Arktur und die Sonne bzw. den Mond sind hier zu finden: Mu Pegasi (G8 III) & Xi Persei (O7.5 III).

Identifizieren der Spektrallinien

Die **Ionen**, die über die Linienidentifikationen identifiziert werden sollen, **müssen in der Variable “ions” angeben werden**. **Geschieht dies nicht werden keine Linienidentifikationen dargestellt!** Das Skript liest hierfür die Linieninformationen aus der Datei `atomic_lines.tsv`. **Da diese Datei aber nur eine Auswahl an Spektrallinien**, enthält, die in den vielen unterschiedlichen Sternen auftreten, **ist es nötig nach zusätzlichen Spektralllinien** und deren Übergängen z.B. in der NIST-Datenbank **zu suchen.** Wie bereits erwähnt haben wie für diese Datenbank auch eine Bedienungsanleitung. Die so bestimmten Linieninformationen müssen dann in die Variable `manual_lines` eingetragen werden.

Bestimmung des Spektraltyps

Anhand der im Spektrum identifizierten Linien und Ionen kann anschließend leicht der Spektraltyp ermittelt werden. Diese Grafik [1] kann hilfreich sein für eine erste Klassifizierung der Spektren. Des weiteren können die Spektren auch mit einem Spektralatlas verglichen werden.

[1] Struve, O. (1959): Elementary Astronomy (Oxford University Press, New York) p. 259

Protokoll

Ein üblicher Praktikumsbericht ist anzufertigen. Allgemeine Hinweise zum Schreiben von Praktkumsberichten zum Thema Struktur und Inhalt können hier eingesehen werden.

Die Übersicht über den theoretischen Hintergrund für diesen Versuch beinhaltet eine Beschreibung der Entstehung von Sternspektren, den verschiedenen Spektraltypen und deren Charakteristiken und das Konzept hinter der Messung von Radialgeschwindigkeiten.

In den Methoden sollen die Abläufe der Beobachtung beschrieben werden und die anschließende Reduktion der Daten. Dazu gehört eine allgemeine Beschreibung der unternommenen Schritte, möglichen Abweichungen von Standardprozedere und einer Auflistung alle gesetzten Parameter. Alle im Zuge der Datenreduktion anfallenden Grafiken sollen im Bericht vorhanden sein, können aber in den Appendix ausgelagert werden.

Der Resultateabschnitt des Reports präsentiert und beschreibt die reduzierten Spektren (ein paar signifikante Ordnungen im Text, der Rest kann in den Appendix). Dies beinhaltet auch die Messung

der Radialgeschwindigkeit für jeden Stern anhand von Spektrallinien. Genutzte Linien sollen zusammen mit ihren Wellenlängen und Verschiebungen gelistet werden.

Die Analyse beinhaltet die Bestimmung des Spektraltyps für die einzelnen Sterne basieren auf den Charakteristiken, die im theoretischen Hintergrund gelistet sind.

Abschließend, diskutiere die Ergebnisse und bringe sie in den größeren Zusammenhang. Dazu zählt z.B. ein Literaturvergleich wo möglich. Weiterhin soll eine Diskussion möglicher Fehlerquellen gemacht werden. Gibt es in euren Daten Inkonsistenzen oder Abweichungen vom Erwarteten? Oder gibt es Strukturen und Auffälligkeiten in den Spektren die ihr nicht erklären könnt? Beschreibe mögliche Lösungen und Erklärungen für die gefundenen Probleme.

Anmerkung: *Die Plots der individuellem Ordnungen der Spektren sind sehr speicherintensiv und gewöhnlich zu gross für einen Emailanhang. Ihr könnt den Report in den Universitaetseigenen Cloud Service (BoxUP) hochladen und uns den Link zusenden, oder aber den Dateipfad zu den Plots wenn ihr sie auf dem Praktikumsrechner gespeichert habt.*

Übersicht: Praktikum

From:
https://polaris.astro.physik.uni-potsdam.de/wiki/ - **OST Wiki**

Permanent link:
https://polaris.astro.physik.uni-potsdam.de/wiki/doku.php?id=de:praktikum:sternspektren_baches&rev=1689973790

Last update: **2023/07/21 21:09**