

# N1 - Sternspektren verschiedener Spektraltypen (DADOS)



Diese Anleitung wird aufgrund der Umstellung auf das Gitter mit 900 Linien/mm und der Umstellung auf eine erweiterte Auswertesoftware überarbeitet. DADOS ist jedoch voll funktionsfähig und einsatzbereit. Bei der Auswertung werden die Betreuer bei einigen Schritten behilflich sein, die sich geringfügig von der aktuellen Version dieser Anleitung unterscheiden.

## Aufgabe

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.

## Beobachtung

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

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

- Sternspektren
- Kalibrationsspektren mit einer diskreten Lichtquelle
- Kalibrationsspektren mit einer kontinuierlichen Lichtquelle
- Darkframes für die Sternspektren und die kontinuierlichen Kalibrationsspektren

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

## Datenauswertung

Die Skripte für die Auswertung sind auf dem [Praktikumsrechner](#) im Ordner `~/scripts/n1_dados/transition_version/` zu finden.

## Datenreduktion

### Vorbereitungen

Die ersten zwei Schritte sind das Einloggen im [Praktikumspool](#) und das Kopieren der Beobachtungsdaten (FITS-Files), inklusive Darkframe- und Kalibrations-Aufnahmen aus dem Verzeichnis ~/data/<datum> ins eigene Verzeichnis ~/data\_reduction/.

### Datenreduktion

Für die Datenreduktion steht das Skript „1\_masterimages.py“ zur Verfügung. Dieses Skript kombiniert die einzelnen Aufnahmen zu entsprechenden „Master“-Dateien. So werden z.B. die einzelnen Darkaufnahmen entsprechend der Belichtungszeit zu **Masterdarks** kombiniert, indem für jedes Pixel der Median über alle Aufnahmen gebildet wird.

In dem Skript müssen folgende Variablen gesetzt werden:

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

# Flat darks:
path_flat_darks: str = '?'

# Flats:
path_flats: str = '?'

# Darks for wavelength calibration exposures:
path_wavelength_darks: str = '?'

# Wavelength calibration exposures:
path_wavelength: str = '?'

# Spectra:
path_spectra: str = '?'
```

path\_darks ist der Pfad zu den Darkaufnahmen mit der gleichen Belichtungszeit wie die Aufnahmen für die Spektren. Der Pfad zu letzteren muss in path\_spectra angegeben werden. Die Flats sind unter path\_flats und die entsprechenden Darks unter path\_flat\_darks anzugeben. Gleiches gilt für die Aufnahmen zur Wellenlängenkalibrierung, die unter path\_wavelength und path\_wavelength\_darks anzugeben sind.

Zusätzlich sollte noch die Variable trim\_image auf **False** gesetzt werden.

## Spektrumauswahl

Es gibt verschiedene Tools, um die im FITS-Format abgelegten zweidimensionalen CCD-Bilder (Images) zu betrachten. Beispielsweise *ds9*, dass im Terminal gestartet werden kann

```
ds9 filename.fit
```

- Man notiere sich den Zeilenbereich, der das Sternspektrum enthält (*master\_spectrum.fit*).
- Man notiere sich einen Zeilenbereich, den man als Hintergrund verwenden kann. Dieser Bereich muss also außerhalb des Sternspektrums liegen, jedoch noch innerhalb des benutzten Spektrographenschlitzes. Falls man letzteren nicht auf dem Bild mit dem Stern-Spektrum sehen kann, orientiere man sich auf den Bild des Lampenspektrums (*master\_wave.fit*).

## Wellenlängenkalibration

### Funktionsprinzip

Das Skript sucht die Maxima im Kalibrations-Spektrum, markiert sie und notiert die Pixelnummer, bei der sich die Intensitätsmaxima befinden. Schließlich wird diesen Werten jeweils eine Wellenlänge zugeordnet. So entsteht eine Pixel-Wellenlängen-Zuordnung, die für die Auswertung des Sternspektrums verwendet wird.

### Parameter

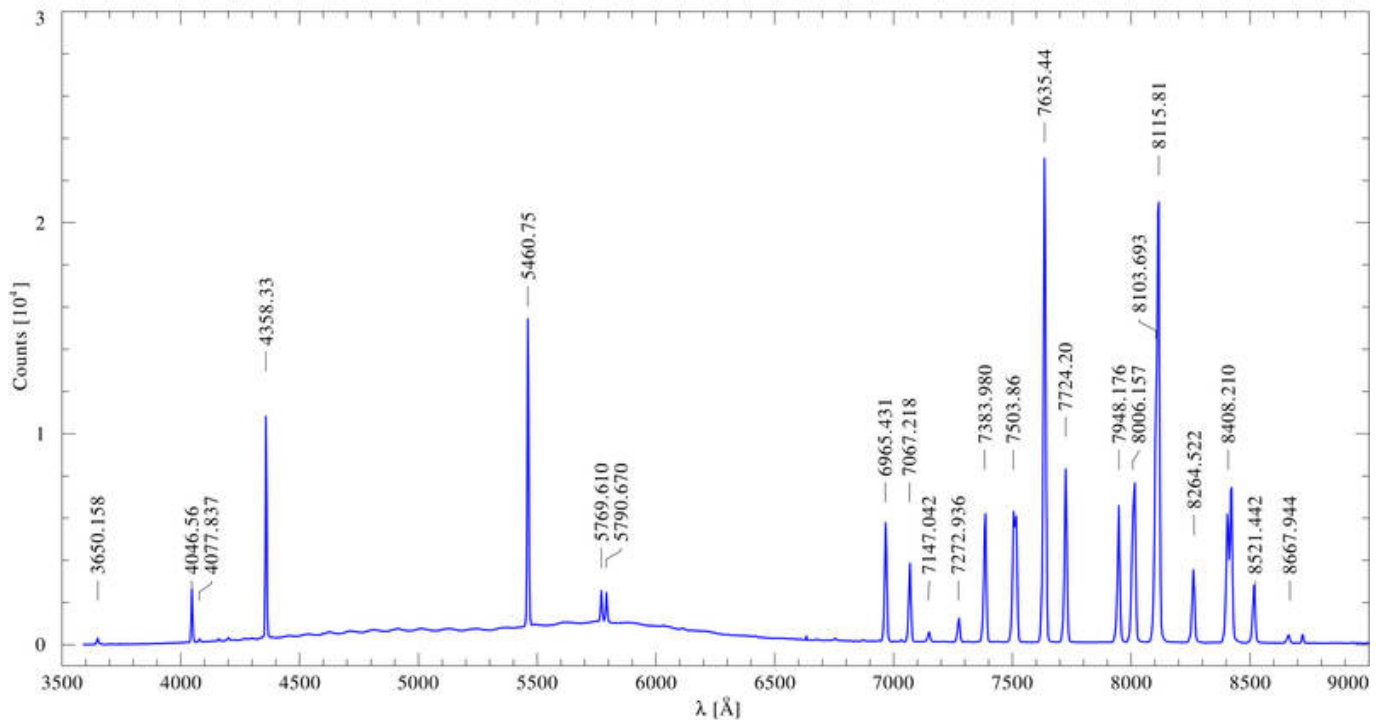
Das zugehörige Skript heißt *1\_findcaliblines.py*. In dieser Datei editiert man mit einem Editor seiner Wahl (z.B. *Kate*) den Pfad und den Namen der Datei mit dem Spektrum der Energiesparlampe, i.e. die schwarze Lampe mit dem Linienspektrum, sowie den Zeilenbereich, in dem man auch das Sternspektrum extrahieren will und einen Zeilenbereich, der **außerhalb des Spaltes** liegt. Beispielsweise:

```
# name of the file with the wavelength calibration spectrum
calibFileName    = "calib_wave.FIT"

# region (rows on the image) containing the calibration spectrum
specRegionStart = 495
specRegionEnd   = 600

# background region (rows on the image), which needs to be outside of the
slits
bgRegionStart    = 0
bgRegionEnd      = 200
```

Die Kalibration beruht darauf, dass Quecksilber- und Argonlinien identifiziert werden. Die stärksten Linien, welche zu erwarten sind, sind in der folgenden Grafik identifiziert.



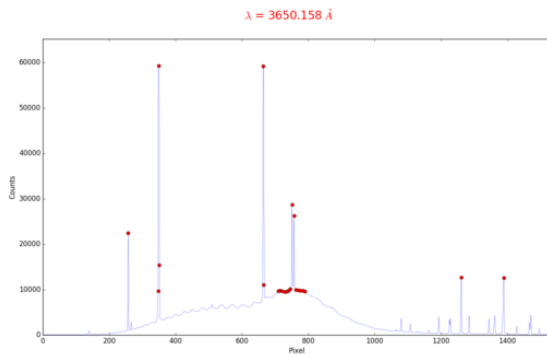
Quecksilber- und Argonemissionslinienspektrum mit entsprechenden Linienidentifikation

## Ausführen des Skriptes

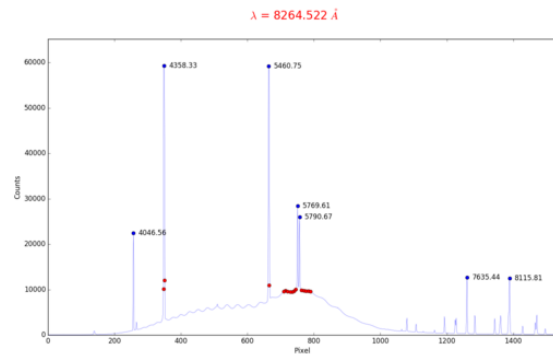
Nun kann das Skript ausgeführt werden:

```
python 1_findcaliblines.py
```

Nach dem Ausführen des Skriptes erscheint folgendes Fenster, auf dem das Quecksilber- und Argonemissionslinienspektrum dargestellt ist und alle Linien, die durch das Skript identifiziert wurden mit einem roten Kreis markiert sind. Nun müssen in diesem Fenster alle Emissionslinien markiert werden, für die die Wellenlänge bekannt ist. Hierfür ist das obere Beispielspektrum mit den identifizierten Linien sehr hilfreich. Das Skript gibt die möglichen Linien vor, sodass jeweils nur die entsprechenden roten Kreise durch einen Linksklick markiert werden müssen. Die Wellenlänge der aktuellen Linie ist in rot oben in dem jeweiligen Fenster zusehen. Sollte die Wellenlänge einer Linie angezeigt werden, die nicht mit einem roten Kreis markiert ist, kann diese durch einen Rechtsklick übersprungen werden. Erfolgreich markierte Linien erscheinen dann blau und die entsprechende Wellenlänge wird neben den Emissionspeak geschrieben (siehe unten). Es müssen wenigstens vier Linien markiert werden um eine erfolgreiche Wellenlängenkalibration zu ermöglichen. Wenn alle möglichen Linien markiert sind kann dieser Prozess mittels drücken der Q-Taste auf der Tastatur abgeschlossen werden.



Kalibrationsplot ohne markierte Linien



Kalibrationsplot mit sieben markierten Linien

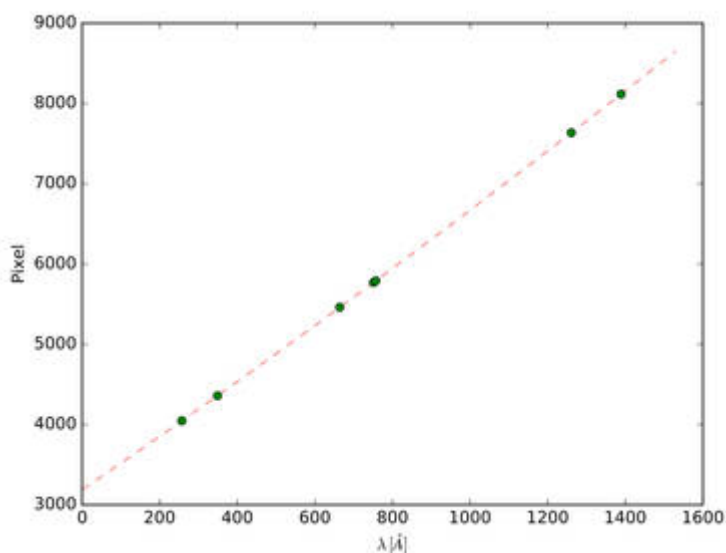
## Fehleroutput

Es kann passieren, dass das Skript folgenden Fehleroutput produziert:

```
can't invoke "event" command: application has been destroyed
while executing
"event generate $w <<ThemeChanged>>"
(procedure "ttk::ThemeChanged" line 6)
invoked from within
"ttk::ThemeChanged"
```

Dieser kann getrost ignoriert werden, da er keinen Einfluss auf das Kalibrationsresultat hat.

Anschließend wird von dem Skript die Kalibrationskurve geplottet (siehe unten). Eine Erfolgreiche Kalibration erkennt man daran, dass die Kalibrationskurve nahezu linear ist.



Kalibrationskurve für den DADOS-Spektrographen

Von dem Skript werden standardmäßig folgende Dateien erstellt:

- `calibration_spectrum.dat` - darin wird jedem Pixel eine Wellenlänge in Angstrom zugeordnet

- `calibration_selection.pdf` - Plot mit den ausgewählten Linien für die Wellenlängenkalibration (bitte an das Protokoll anhängen)
- `calibration_fit.pdf` - hier ist die Wellenlängenkalibration, also `wcalibration_spectrum.dat`, geplottet (bitte an das Protokoll anhängen)

## Fehlerbehebung

Die Plots sollten dann auf mögliche Fehler inspiziert werden. Sollte der Kalibrationsplot nicht linear aussehen, muss das Skript korrigiert und neu ausgeführt werden. Fehlerquellen können sein:

- falsch markierte Linien
- ein anderer Wellenlängenbereich als normalerweise wird beobachtet oder eine andere Kalibrationslampe wird benutzt. Man kann z.B. in der [NIST Datenbank](#) nachsehen und andere Linien ausfindig machen, die dann in der Variable `linelist` im Skript eingetragen werden müssen.

## Das Sternspektrum

### Funktionsprinzip

Nach der Bestimmung der Kalibrationskurve kann nun das eigentliche Sternspektrum ausgewertet werden. Zunächst wird das Darkframe des Sternspektrums abgezogen. Dann wird das Spektrum durch das Flatfield geteilt und schließlich die Wellenlängenkalibration durchgeführt. Außerdem besteht die Möglichkeit im Spektrum identifizierte Spektrallinien zu kennzeichnen.

### Parameter

Das zugehöriges Skript heißt `2_extractspectrum.py`. In dieser Datei werden viele Parameter ganz analog zu den obigen Skript gesetzt. Die Parametersektion kann beispielsweise so aussehen:

```
### science spectrum file ###
# file with stellar spectrum
science          = 'star.FIT'
# directory of the darkframe for the stellar spectrum
darkframe_dir    = 'darks/??s/'
# flatfield directory
flatfield_dir     = 'flats/'
# directory of the darkframe for the flats
flatdark_dir     = 'darks/??s/'

### Data that should be extracted ###
# region containing the science spectrum
specRegionStart  = 495
specRegionEnd    = 590
```

```
# sky background region (inside the slit)
bgSkyStart   = 96
bgSkyEnd     = 104

### Plot range ###
# set the variables to '?' for an automatic resizing
lambdamin = '?'
lambdamax = '?'
#lambdamin = 3500.
#lambdamax = 5000.
```

Zu diesen Parametern sei angemerkt:

- Über die Variablen `specRegionStart` und `specRegionEnd` werden die CCD-Zeilen definiert, die das Spektrum enthalten und die ausgelesen werden.
- Der Himmelshintergrund soll noch abgezogen werden. Dafür muss ein Bereich ausgewählt, der **innerhalb vom Spalt** liegt, aber **kein Spektrum** enthält. Wenn möglich, sollen die gleiche Zeilenanzahl wie beim Spektrum verwendet werden. Also  $(\text{specRegionStart} - \text{specRegionEnd}) = (\text{bgSkyStart} - \text{bgSkyEnd})$ .
- Mittels der Option `lambdamin` und `lambdamax` kann der Plotbereich eingeschränkt werden. Werden in diese Variablen `?` geschrieben, dann wird der Plotbereich automatisch festgelegt.

### Zusätzliche Parameter

**Auswertungsmethode:** Die Variable `mode` steuert, die Auswertungsmethode. Bei der Methode *mean* wird über alle extrahierten Zeilen aus dem CCD-Bild gemittelt. Bei *median* wird der Median gebildet. Diese Methode ist sollte standardmäßig ausgewählt werden.

```
### Image reduction mode ###
#mode = 'mean'
mode = 'median'
```

**Linienidentifikation:** Die Linienidentifikation kann über den Parameter `plotident` an oder abgeschaltet werden:

```
### Idents ###
# plot idents yes or no
plotident = 'yes'
#plotident = 'no'
```

Die Datei mit den gefundenen Spektrallinien, welche standardmäßig einfach `absorption_lines` heißt, wird zum Ausführen des Skriptes nicht zwingend gebraucht. In diesem Fall wird der Variable `lineFile` einfach nur ein leerer String zugewiesen oder alternativ ein Pfad zu einer leeren Datei übergeben:

```
#File containing line identifications
lineFile = ""
# oder
lineFile = "Ordner/leere_datei.dat"
```

Siehe unten für eine Erklärung wie die Linienidentifikation für die einzelnen Sterne individuell angepasst werden kann.

**Dateinamen:** Des Weiteren besteht die Möglichkeit die Namen der Outputdateien sowie des für die Wellenlängenkalibrierung benötigten Kalibrationsfiles aus dem ersten Schritt der Datenreduktion individuell anzupassen.

## Ausführen des Skriptes

Analog zu oben wird das Skript mittels:

```
python 2_extractspectrum.py
```

ausgeführt. Dabei werden in den Standardeinstellungen folgende Dateien erstellt:

- `stern_spectrum.dat` - mit dem tabellierten Spektrum
- `stern_spectrum.pdf` - mit dem geplotetem Spektrum
- `flatfield.pdf` - mit dem geplotetem Flatfield (bitte an das Protokoll anhängen)

## Identifizieren der Spektrallinien

Es besteht die Möglichkeit im Spektrum gefundenen Linien im Spektrumsplot einzuzeichnen. In der Datei `absorption_lines` im Skriptverzeichnis sind einige wichtige Linien aufgelistet. Allerdings **enthält diese Datei nicht alle Spektrallinien**, die in den vielen unterschiedlichen Sternen auftreten, die wir beobachten. **Daher ist es nötig nach zusätzlichen Spektrallinien und deren Übergängen** z.B. in der [NIST-Datenbank zu suchen](#). Eine [Anleitung](#) hierfür haben wir natürlich auch. Darüber hinaus ist es empfohlen, für jeden Stern eine individuelle Datei mit Linienidentifikationen zu erstellen. Um Linien im Plot zu kennzeichnen, überträgt man die relevanten Wellenlängen und Elemente/Ionen in die separate Datei, deren Inhalt dann wie folgt aussehen könnte (Wellenlänge in Å | Ident):

```
3888.052    HI
3970.075    HI
4861.38     HI
6562.88     HI
5801.33     5811.98    CIV
```

Wie im letzten Eintrag zu sehen ist, können auch Ionen mit Multipletübergängen dargestellt werden. **Linien, die nicht im Stern wieder gefunden werden, sind aus der entsprechenden Datei zu entfernen.**

Hat man diese Datei für ein Spektrum erstellt, kann der dazugehörige Pfad nun in der Parametersektion gesetzt werden. Beispielsweise

```
#File containing line identifications
lineFile = "directory/line_list_for_starname.dat"
```



Dann kann das Skript erneut ausgeführt werden und im Plot sollten die Linien benannt sein.

## Protokoll

Ein ueblicher Praktikumsbericht ist anzufertigen. Allemeine Hinweise zum Schreiben von Praktikumsberichten zum Thema Struktur und Inhalt koennen [hier](#) eingesehen werden.

Die Uebersicht ueber den theoretischen Hintergrund fuer 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 Ablaeufe der Beobachtung beschrieben werden und die anschliessende Reduktion der Daten. Dazu gehoert eine allgemeine Beschreibung der unternommenen Schritte, moeglichen Abweichungen von Standardprozedere und einer Auflistung alle gesetzten Parameter. Alle im Zuge der Datenreduktion anfallenden Grafiken sollen im Bericht vorhanden sein, koennen aber in den Appendix ausgelagert werden.

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

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

Abschliessend, diskutiere die Ergebnisse und bringe sie in den groesseren Zusammenhang. Dazu zaehlt z.B. ein Literaturvergleich wo moeglich. Weiterhin soll eine Diskussion moeglicher Fehlerquellen gemacht werden. Gibt es in euren Daten Inkonsistenzen oder Abweichungen vom Erwarteten? Oder gibt es Strukturen und Auffaelligkeiten in den Spektren die ihr nicht erklaren koennt? Beschreibe moegliche Loesungen und Erklaerungen fuer die gefundenen Probleme.

**Anmerkung:** Diese [Grafik](#) [1] kann hilfreich sein fuer eine erste Klassifizierung der Spektren. Desweiteren koennen die Spektren auch mit einem [Spektralatlas](#) verglichen werden. Die [NIST web page](#) erlaubt es explizit nach Spektrallinien zu suchen. Auf dieser [Website](#) gibt es eine gute Anleitung zum Klassifizieren von Sternspektren.

**Anmerkung:** Die Plots der individuellem Ordnungen der Spektren sind sehr speicherintensiv und gewoehlich zu gross fuer einen Emailanhang. Ihr koennt 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.

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

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&rev=1733406661>

Last update: 2024/12/05 13:51

