

N1 - Sternspektren verschiedener Spektraltypen (DADOS)



Die vorliegende Anleitung wird im Zuge der Umstellung auf eine erweiterte Auswertungssoftware zur Zeit überarbeitet. Die Anleitung ist jedoch derzeit vollständig. Sie kann ohne weiteres für die Auswertung der Daten verwendet werden. Nur wenn irgendwo „transition“ oder „transition_version“ auftaucht, bitte nicht wundern.

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 besprochen. Gegenwärtig wird der *DADOS* mit einem Gitter von 900 Linien/mm in Kombination mit dem *QHY 268M* eingesetzt. 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

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 Aufnahmen 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.

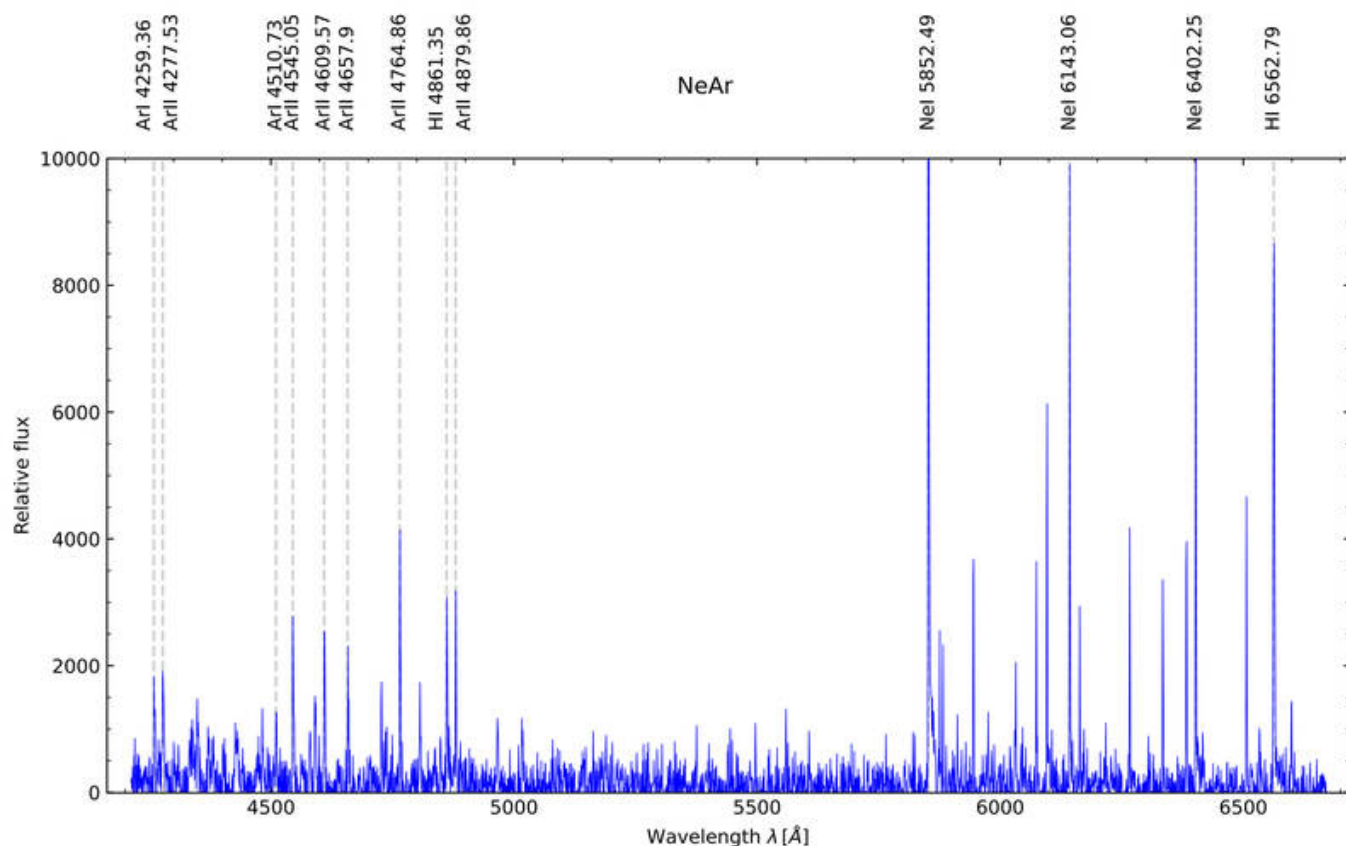
Parameter

Das zugehörige Skript heißt **2_findcaliblines.py**. In dieser Datei muss man mit einem Editor seiner Wahl (z.B. *Kate*) nur den Zeilenbereich editieren, in dem das Kalibrationsspektrum zu finden ist (specRegionStart und specRegionEnd; kann derselbe sein, aus dem man das Sternspektrum extrahieren will) und einen Zeilenbereich, der **außerhalb der Spalte** liegt (bgRegionStart, bgRegionEnd):

```
# Region (rows on the image) containing the calibration spectrum
specRegionStart = 476
specRegionEnd = 487

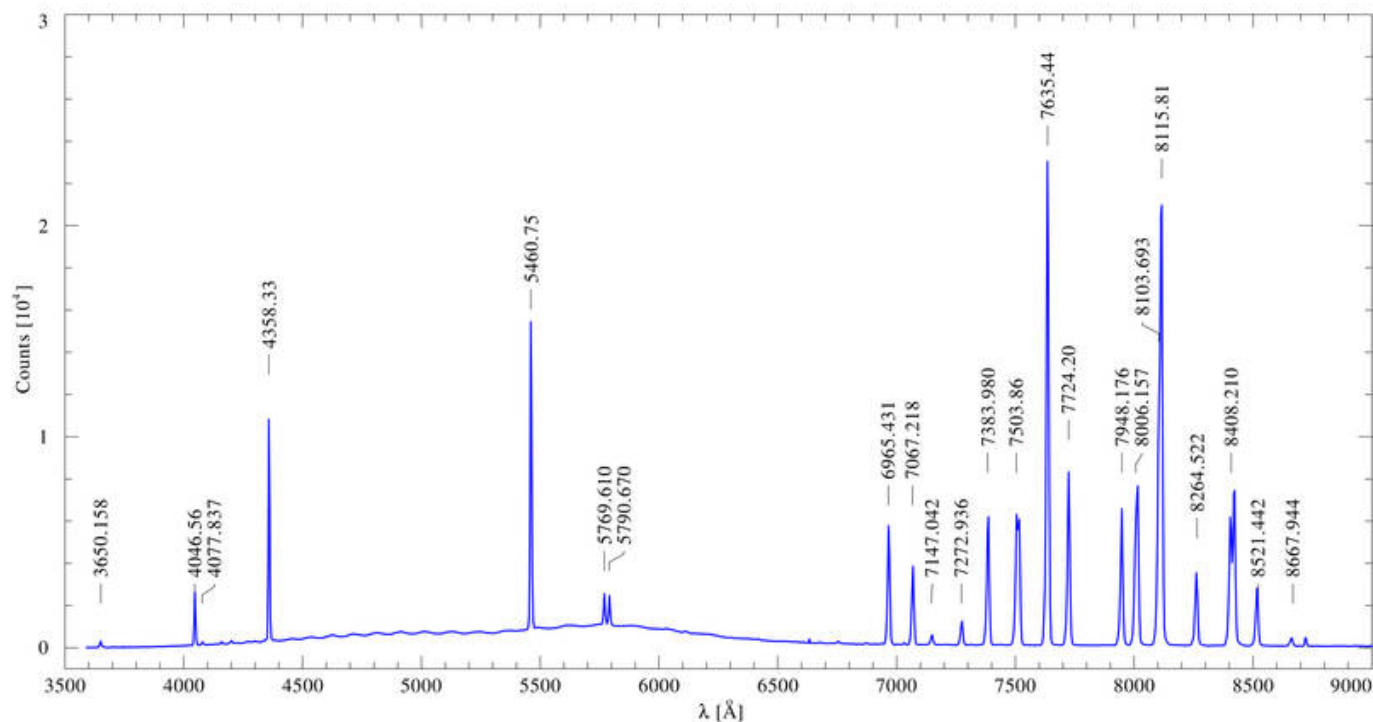
# Background region (rows on the image) -> need to be outside of the slits
bgRegionStart = 815
bgRegionEnd = 870
```

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



Neon- und Argonemissionslinienspektrum mit entsprechenden Linienidentifikation

Alte Hg- & Ar-Kalibration



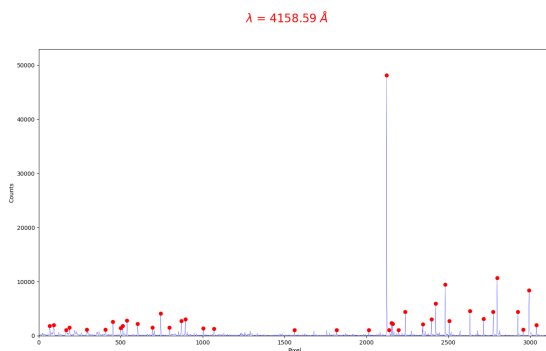
Quecksilber- und Argonemissionslinienspektrum mit entsprechenden Linienidentifikation

Ausführen des Skriptes

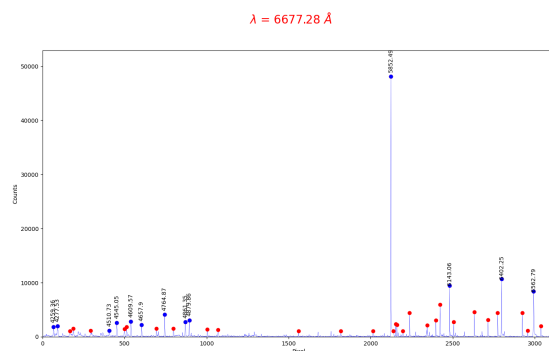
Nun kann das Skript ausgeführt werden:

```
python 2_findcaliblines.py
```

Nach dem Ausführen des Skriptes erscheint folgendes Fenster, in dem das Emissionslinienspektrum von Neon und Argon dargestellt ist und alle durch das Skript identifizierten Linien mit einem roten Kreis markiert sind. In diesem Fenster müssen nun alle Emissionslinien markiert werden, deren Wellenlänge bekannt ist. Dabei ist das obere Beispielspektrum mit den identifizierten Linien sehr hilfreich. Das Skript gibt die möglichen Linien vor, so dass jeweils nur die entsprechenden roten Kreise durch einen Linksklick markiert werden müssen. Die Wellenlänge der aktuellen Linie wird oben im jeweiligen Fenster rot angezeigt. Sollte die Wellenlänge einer Linie angezeigt werden, die nicht mit einem roten Kreis markiert ist, kann diese mit einem Rechtsklick übersprungen werden. Erfolgreich markierte Linien erscheinen dann blau und die entsprechende Wellenlänge wird neben den Emissionspeak geschrieben (siehe unten). Für eine erfolgreiche Wellenlängenkalibrierung müssen mindestens vier Linien markiert werden. Wenn alle möglichen Linien markiert sind, kann der Vorgang durch Drücken der „Q“-Taste auf der Tastatur abgeschlossen werden.

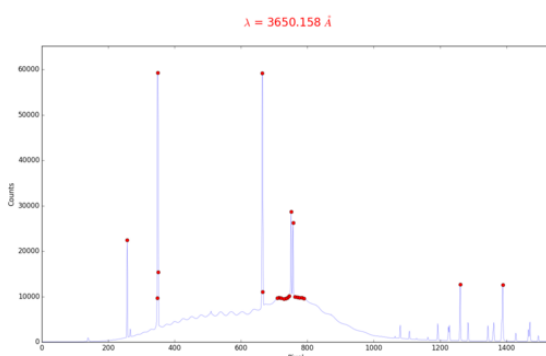


Kalibrationsplot ohne markierte Linien

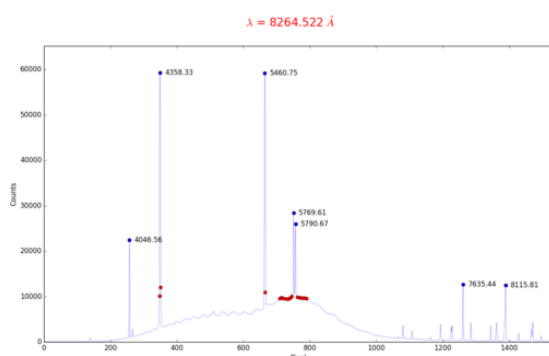


Kalibrationsplot mit sieben markierten Linien

Alte Hg- & Ar-Kalibration



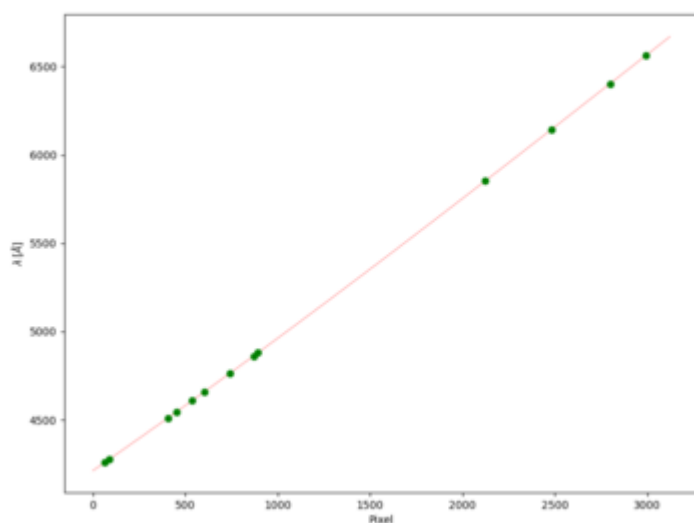
Kalibrationsplot ohne markierte Linien



Kalibrationsplot mit sieben markierten Linien

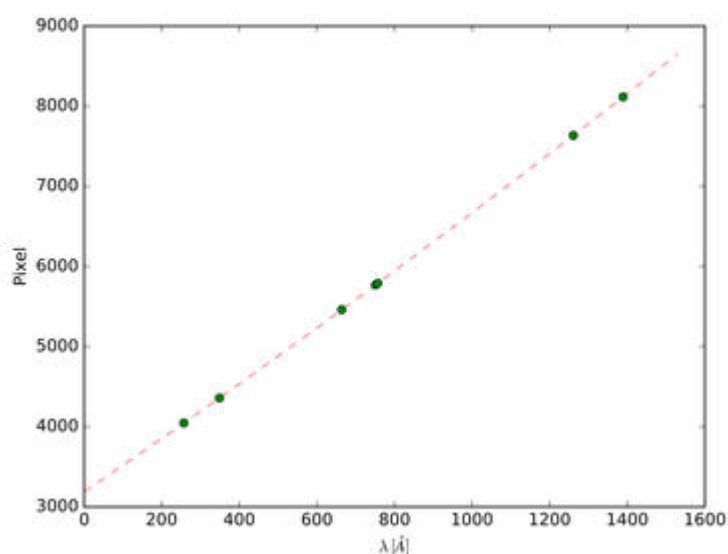
Anschließend wird von dem Skript die Kalibrationskurve geplottet (siehe unten). Eine Erfolgreiche Kalibration erkennt man daran, dass die Kalibrationskurve nahezu linear ist.

$$\lambda = -2.80\text{E-}09 \times \text{pxl}^3 + 2.91\text{E-}05 \times \text{pxl}^2 + 7.23\text{E-}01 \times \text{pxl} + 4212.31$$



Kalibrationskurve für den DADOS-Spektrographen

Alte Hg- & Ar-Kalibration



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 Wellenlängen-Pixel-Zuordnung kann nun das eigentliche Sternspektrum ausgewertet werden. Standardmäßig wird das Spektrum durch das Flatfield geteilt und anschließend die Wellenlängenkalibrierung durchgeführt. Zusätzlich besteht die Möglichkeit, das Spektrum zu normieren und die im Spektrum identifizierten Spektrallinien zu markieren.

Parameter

Das zugehörige Skript heißt `3_extractspectrum.py`. In dieser Datei werden einige Parameter ganz analog zu den obigen Skript gesetzt. Die Parametersektion kann beispielsweise so aussehen:

```
# Name of the object
object_name: str = "star"

###
# Extraction regions
#
# Region containing the science spectrum
spec_region_start: int = 759
spec_region_end: int = 772

# Sky background region (inside the slit)
background_sky_start: int = 710
background_sky_end: int = 730

###
# Plot range
#
# Set the variables to '?' for an automatic resizing
lambda_min: str | float = '?'
lambda_max: str | float = '?'
flux_min: str | float = '?'
flux_max: str | float = '?'

###
# Normalization ?
```

```
# Possibilities: True or False
#
normalize: bool = False
```

Unter `object_name` kann der Name des beobachteten Sterns angegeben werden. Die Variablen `spec_region_start` und `spec_region_end` definieren die Zeilen des Kamerachips, die das Spektrum enthalten und ausgelesen werden sollen. Vom Sternspektrum muss noch der Himmelshintergrund subtrahiert werden. Dazu muss eine Region ausgewählt werden, die **innerhalb des Spalts** liegt, aber **kein Spektrum** enthält. Dieser Bereich wird durch die Variablen `background_sky_start` und `background_sky_end` definiert. Mit den Optionen `lambda_min` und `lambda_max` sowie `flux_min` und `flux_max` kann der Plotbereich auf der X- bzw. Y-Achse eingeschränkt werden. Wird in diese Variablen `?` geschrieben, so wird der Plotbereich automatisch festgelegt.

Ausführen des Skriptes

Analog zu oben wird das Skript mittels:

```
python 3_extractspectrum.py
```

ausgeführt. Standardmäßig werden folgende Dateien erzeugt:

- `spectrum_panels_{object_name}.pdf` - mit dem Spektrum in mehreren Panels (Zoom-Version)
- `spectrum_total_{object_name}.pdf` - mit dem Spektrum dargestellt in einem einzigen Panel
- `{object_name}_spectrum_total.dat` - mit dem Spektrum in Tabellenform
- `{object_name}_spectrum_total.csv` - mit dem tabellierten Spektrum im CSV-Format

Identifizieren der Spektrallinien

Es besteht die Möglichkeit im Spektrum gefundenen Linien im Spektrumsplot einzuzeichnen. Die **Ion**en, die über die Linienidentifikationen identifiziert werden sollen, **müssen in der Variable „ions“ angegeben 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 Spektrallinien** und deren Übergängen z.B. in der [NIST-Datenbank](#) zu suchen. Wie bereits erwähnt haben wir für diese Datenbank auch eine [Bedienungsanleitung](#). Die so bestimmten Linieninformationen müssen dann in die Variable `manual_lines` eingetragen werden.

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

Variable	Beschreibung
<code>radial_velocity</code>	Gemessene Radialgeschwindigkeit in km/s (ohne baryzentrische Korrektur)

Variable	Beschreibung
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.
percentage_line_flux_must_be_below_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 starke 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.

Protokoll

Es ist ein üblicher Praktikumsbericht anzufertigen. Allgemeine Hinweise zum Aufbau und Inhalt von Praktikumsberichten können [hier](#) eingesehen werden.

Der Überblick über den theoretischen Hintergrund des Praktikums beinhaltet eine Beschreibung der Entstehung von Sternspektren, der verschiedenen Spektraltypen und deren Eigenschaften.

Die Methoden beschreiben den Beobachtungsablauf und die anschließende Datenreduktion. Dies beinhaltet eine allgemeine Beschreibung der durchgeführten Schritte, eventuelle Abweichungen vom Standardverfahren und eine Auflistung aller eingestellten Parameter. Alle im Zuge der Datenreduktion erstellten Grafiken sollten im Bericht enthalten sein, können aber auch in den Anhang ausgelagert werden.

Im Ergebnisteil des Berichts werden die reduzierten Spektren dargestellt und beschrieben.

Die Analyse beinhaltet die Bestimmung des Spektraltyps für jeden Stern, basierend auf den Eigenschaften, die im theoretischen Hintergrund aufgelistet sind.

Abschließend werden die Ergebnisse diskutiert und in einen größeren Zusammenhang gestellt. Dazu gehört z.B. ein Literaturvergleich, wo dies möglich ist. Es sollte auch eine Diskussion über die möglichen Fehlerquellen geführt werden. Gibt es Inkonsistenzen in den Daten oder Abweichungen vom Erwarteten? Oder gibt es Strukturen und Auffälligkeiten in den Spektren, die sich nicht erklären lassen? Beschreibt mögliche Lösungen und Erklärungen für die gefundenen Probleme.

Anmerkung: Diese [Grafik](#) [1] kann hilfreich sein für eine erste Klassifizierung der Spektren. Des weiteren können 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 individuellen 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.

[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=1733478750>

Last update: **2024/12/06 09:52**

