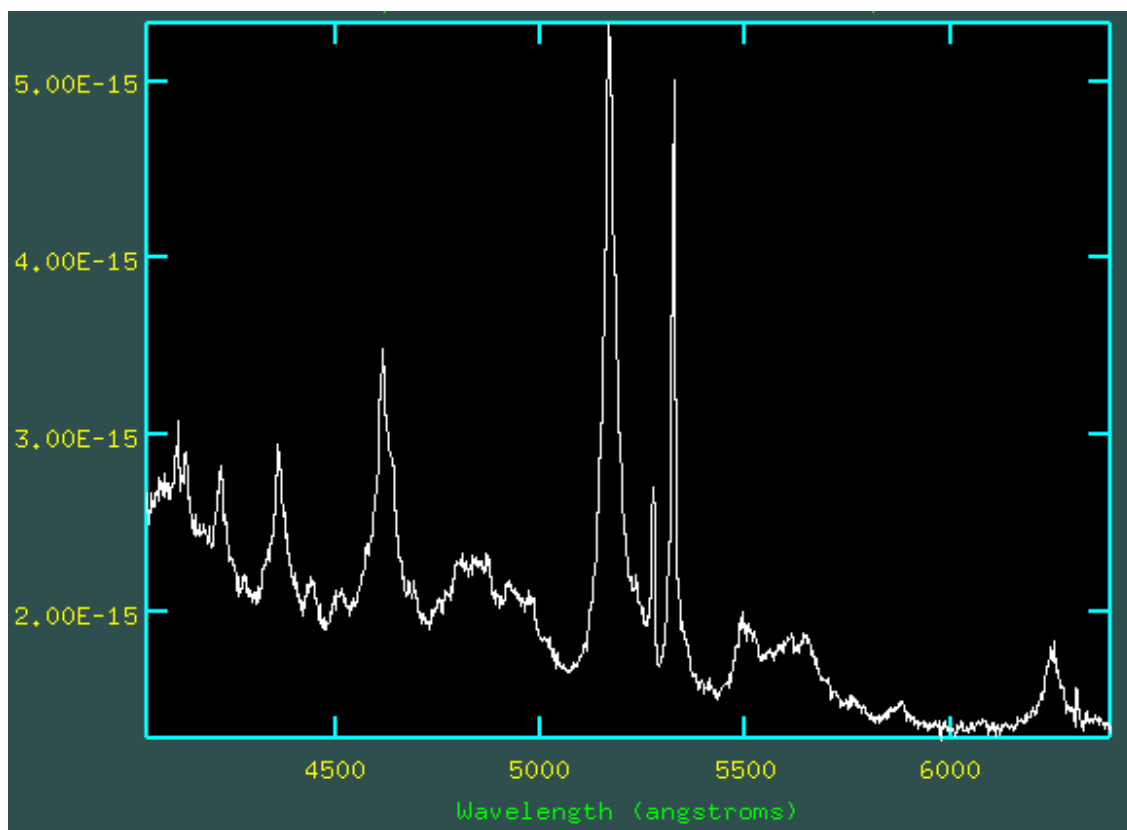


Spektrihavaintojen redusointi IRAFissa "Lyhyt oppimäärä"

Kari Nilsson
Suomen ESO-keskus
v1.02

13. joulukuuta 2013



1 Johdanto

Tämän ohjeen tarkoituksena on käydä läpi ns. longslit-spektrien redusointi IRAF-ohjelmistolla. Lukijalta oletetaan hieman kokemusta mm. IRAFin käynnistämisen ja peruskäytön suhteen (pakettien lataaminen, parametrien editointi,...). Lisäksi oletetaan, että lukija ymmärtää perusredusointivaiheiden (bias-vähennys ja tasoitus) merkityksen.

On hyvä huomata, että tämä ohje sisältää vain “lyhyen oppimäärän” eli minimimäärän operaatiota, joilla spektrien redusointi onnistuu. Suureen tarkkuuteen pyritäessä tarvitaan enemmän vaiheita ja kalibrointikuvia kuin mitä tässä ohjeessa on esitetty. Ohje on tarkoitettu lähinnä henkilöille, jotka käsittelevät spektrejä ensimmäistä kertaa ja havaintoihin, joiden signaali/kohinasuhde on korkea.

Esimerkkinä käytetään NOT-tiedekoulussa 2.11.2007 havaittua spektriä kohteesta II ZW 136, joka on punasiirtymällä $z = 0.0623$ sijaitseva Seyfert-galaksi. Näiden galaksien ytimessä havaitaan ylimääräistä aktiivisuutta supermassiiviseen mustaan aukkoon putoavan kaasun vaikutuksesta. Aktiivisuus näkyy mm. kirkkaina emisioviivoina, joiden avulla voidaan määrittää mm. kohteen punasiirtymä ja arvioida mustan aukon massa. Havainnot on tehty ALFOSC-spektrograafilla käyttäen grism n:o 7:ää.

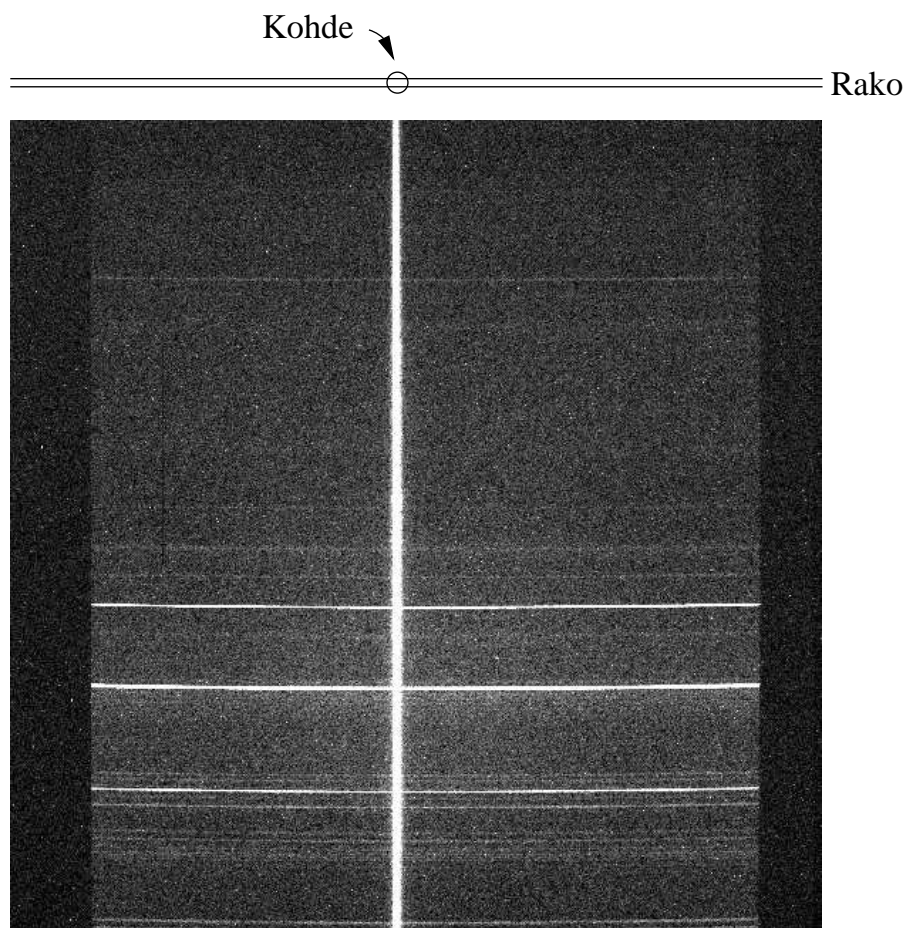
2 Havaintodata

Edellämainittujen suureiden määrittämiseksi on minimissään tehtävä seuraavat havainnot:

- kohteen spektri
- “viivalampun” spektri
- kontinuumilampun spektri
- kalibrointitähden spektri.

“Viivalampun” spektrillä tarkoitetaan spektriä, joka on otettu lampusta, joka lähettää säteilyä vain tarkalleen tunnetuilla aallonpituuksilla. Lamppu on valmiiksi asennettu ALFOSC-spektrograafiin. Viivalampun spektriä käytetään määrittämään aallonpituuskalibrointi, eli käytännössä muunnos pikselikoordinaateista aallonpituuksiin. Kontinuumilampun spektrillä korjataan CCD:n pikselien herkkyysvaihteluita, eli se vastaa tavallisen kuvauksen tasoituskuva. Kalibrointitähden spektrin avulla määritetään havaintolaitteen herkkyyden aallonpituuden funktiona, minkä avulla voidaan määrittää kohteen spektrin todellinen muoto.

On hyvä huomata, että ylläoleva joukko edustaa kuvien minimimäärää. Suureen tarkkuuteen pyritäessä tai kun spektri on hyvin heikko tarvitaan suurempi määrä kalibrointikuvia, mm. useita viiva- ja kontinuumihavaintoja.



Kuva 1: ALFOSC-spektrograafin “raakaspektri” kohteesta II Zw 136. Kuvan yläreunassa on havaintojärjestely esitetty kaavamaisesti.

3 Peruskäsitteitä

Kuvassa 1 on esitetty kohteen II Zw 136 spektri sellaisena kuin se tulee ALFOSC-spektrograafista. Kirkas pystysuora vana on kohteen spektri. Vaakasuorat viivat ovat taustataivaan spektriviivoja. Aallonpituus kasvaa kuljettaessa pystysuoraan ylhäältä alaspäin. Pystysuoraa akselia kutsutaan dispersioakseliksi (dispersion axis) tai dispersiosuunnaksi, eli suunnaksi, jossa dispersio (valon hajoaminen eri väreihin) tapahtuu.

Vaakasuoralle akselille ei ole hyvää suomenkielistä nimitystä, mutta englanninkielisestä termistä “spatial axis” voi vääntää termin spatiaalinen akseli (huono nimi, mutta parempaakaan ei ole). Termin ymmärtämiseksi on hyvä muistaa miten kuvan 1 spektri on havaittu. Spektrograafissa kohteen valo kulkee ensin kapean raon lävitse, mikä jälkeen se hajoitetaan eri aallonpituuksiin. CCD:lle tallentuu spektrograafin optiikan avulla *raon kuva eri aallonpituuksilla*.

Tämän periaatteen avulla on kuvan 1 muodostuminen mahdollista ymmärtää. Kuljettaessa vaakasuoraan, eli spatiaalista akselia pitkin, aallonpituus ei muutu, vaan kuljetaan pitkin rakoja, eli tarkastellaan miltä raossa esiintyvät kohteet näyt-

tävät tietyllä aallonpituudella. Kuljettaessa pystysuoraan pysytään samassa kohtaa rakoa, mutta aallonpituus muuttuu, eli tarkastellaan miltä ko. raon kohdan spektri näyttää. Jos ko. raon kohdassa sattuu olemaan jokin kohde, havaitaan pystysuorassa suunnassa kohteen + taustataivaan spektri. Jos taas ko. raon kohdassa on pelkästään taustataivasta, havaitaan pelkkä taustataivaan spektri. Koska taustataivas on aina läsnä, havaitaan raon joka kohdasta ainakin taustataivaan spektri, mikä tuottaa kirkkaan vaakasuoran viivan niillä aallonpituuksilla, joilla taustataivas on kirkas.

4 Reduktiovaiheet

Reduktion tarkoituksena on tuottaa kuvan 1 CCD-kuvasta kuvan 37 kaltainen valmis spektri eli kohteen kirkkaus aallonpituuden funktiona. Reduktio sisältää seuraavat vaiheet (suluissa on annettu tarvittavan IRAF-komennon nimi):

- 1) kuvien muuntaminen helpommin käsiteltävään muotoon (`imcopy`)
- 2) bias-vähennys kaikista kuvista (`ccdproc` tai `imarith`)
- 3) kontinuumikuvan suoristaminen (`response`)
- 4) kuvien tasoitus (`ccdproc`)
- 5) taustataivaan viivojen vähennys (`background`)
- 6) spektrin mittaaminen (`apall`)
- 7) spektrin mittaaminen viivalampun kuvasta (`apall`)
- 8) kalibroitaviivojen tunnistaminen (`identify`)
- 9) aallonpituuskalibrointi (`hedit`, `dispcor`)
- 10) vuostandardin käsittely (`background`, `apall`, `hedit`, `dispcor`)
- 11) spektrograafin herkkyyskäyrän määrittäminen (`standard`, `sensfunc`)
- 12) vuokalibrointi (`calibrate`).

Seuraavissa kappaleissa käydään jokainen vaihe läpi yksityiskohtaisesti.

4.1 Kuvien muuntaminen

NOT-teleskoopilla CCD-kuvat tallennetaan “Multi Extension Formaatussa” (MEF), mikä aiheuttaa hankaluuksia reduktion seuraavissa vaiheissa, ellei kuviin tehdä ensin pientä muunnosta. Tee ensin tekstitiedosto, jossa on lista kaikista kuvista esim. antamalla komento

```
ecl>!ls *.fits >muuta.cl
```

jolloin tiedostossa `muuta.cl` on lista kaikista hakemiston fits-kuvista. Huutomerkki komennon alussa tarkoittaa, että kyseessä ei ole IRAFin komento, vaan komento, joka välitetään Linux-käyttöjärjestelmälle. Esimerkin tapauksessa listassa olisi 4 kuvaa:

```
ALqk020036.fits (kohteen spektri)
ALqk020037.fits (viivalampun spektri)
ALqk020038.fits (kontinuumilampun spektri)
ALqk020075.fits (kalibrointitähden spektri)
```

(suluissa olevat osat eivät tietenkään ole tiedostossa, vain tiedostojen nimet). Editoi juuri luotua tiedostoa esim. komennolla

```
ecl>!emacs muuta.cl
```

mikä käynnistää emacs-editorin. Editoi tiedosto seuraavanlaiseksi:

```
imcopy ALqk020036[1] mALqk020036
imcopy ALqk020037[1] mALqk020037
imcopy ALqk020038[1] mALqk020038
imcopy ALqk020075[1] mALqk020075
```

ja tallenna. Olet nyt itse asiassa luonut yksinkertaisen IRAF-skriptin, jonka avulla kuvat muutetaan MEF-muodosta tavanomaisempaan FITS-muotoon. Skripti ajetaan komennolla

```
ecl>cat muuta.cl | cl
```

ja tuloksena on `mALqk`-alkuisia fits-kuvia, joita käytämme reduktioiden seuraavissa vaiheissa.

4.2 Bias-vähennys kaikista kuvista

Bias-vähennyksen tarkoituksena on vähentää CCD-kuvista niissä esiintyvä ns. *bias-taso*, eli signaalitaso, joka esiintyy kuvissa vaikka CCD:lle ei osu mitään valoa. Bias-vähennys voidaan tehdä kahdella tavalla tai käyttämällä molempien yhdistelmää. Tapa 1 on käyttää ns. *overscan-alueetta*. Tämä on CCD-kuvan reunalla esiintyvä alue, joka ei edusta CCD-sirun pikseleitä, vaan se on muodostettu lukemalla CCD-sirulta “tyhjiä” pikseleitä. Tuloksena on alue, jossa ei esiinny muuta signaalia kuin bias-taso. Tapa 2 on “valottaa” CCD-sirua 0 sekuntia ja lukea kuva ulos, jolloin kuvassa ei esiinny muuta signaalia kuin bias-taso.

Erittäin tarkassa työskentelyssä nämä kaksi tapaa yhdistetään, eli tavalla 1 määritetään bias-taso ja tavalla 2 sen riippuvuus paikasta CCD:llä. Tässä esimerkissä käytetään tapaa 1, sillä a) bias-kuvia ei ole otettu ja b) ALFOSCin bias-taso ei riipu paikasta CCD:llä kovinkaan voimakkaasti, joten tapaa 2 ei tarvita.

Se, miten “tapa 1” käytännössä toteutetaan riippuu siitä onko havaintojen aikana luettu koko CCD-siru (kuten tässä esimerkissä) vai vain osa sirua. Jälkimmäinen tapa on hyvin yleinen, sillä spektri yleensä sijaitsee kuvan keskellä, joten riittää lukea vain kuvan keskiosa. Jälkimmäisessä tapauksessa overscan-alue ei ole käytettävissä, joten bias-vähennys täytyy tehdä toisin.

Molemmissa tavoissa tehdään ensin tekstitiedosto, jossa on lista kaikista niistä kuvista, joille bias-vähennys tehdään:

```
ecl>!ls m*.fits >lista
```

Tarkista esim. emacsilla, että listassa ovat ainoastaan ne kuvat, joista bias-vähennys halutaan tehdä. Esimerkin tapauksessa tiedostossa `lista` olisi siis

```
mALqk020036.fits  
mALqk020037.fits  
mALqk020038.fits  
mALqk020075.fits
```

4.2.1 Bias-vähennys kun koko CCD-siru on luettu

Seuraavaksi ladataan tarvittavat paketit

```
ecl>noao  
ecl>imred  
ecl>ccdred
```

ja editoidaan `ccdproc`-komennon parametreja.

```
ecl>epar ccdproc
```

Kuvassa 2 on esitetty kaikki `ccdproc`-komennon parametrit. Tähdellä on merkitty ne parametrit, joita tarvitsee muuttaa bias-vähennystä varten. Huomaa, että parametrit

*images= @kuvat	List of CCD images to correct
(output =)	List of output CCD images
(*ccdtype =)	CCD image type to correct
(max_cache = 0)	Maximum image caching memory (in Mbytes)
(noproc = no)	List processing steps only?\n
(*fixpix = no)	Fix bad CCD lines and columns?
(overscan = yes)	Apply overscan strip correction?
(trim = yes)	Trim the image?
(*zerocor = no)	Apply zero level correction?
(*darkcor = no)	Apply dark count correction?
(*flatcor = no)	Apply flat field correction?
(illumcor = no)	Apply illumination correction?
(fringecor = no)	Apply fringe correction?
(readcor = no)	Convert zero level image to readout correction?
(scancor = no)	Convert flat field image to scan correction?\n
(readaxis = "line")	Read out axis (column line)
(fixfile = "")	File describing the bad lines and columns
(*biassec = [5:40,1:2052])	Overscan strip image section
(*trimsec = [290:1960,1:2052])	Trim data section
(zero = "")	Zero level calibration image
(dark = "")	Dark count calibration image
(flat = "")	Flat field images
(illum = "")	Illumination correction images
(fringe = "")	Fringe correction images
(minreplace = 1.)	Minimum flat field value
(scantype = "shortscan")	Scan type (shortscan longscan)
(nscan = 1)	Number of short scan lines\n
(interactive = no)	Fit overscan interactively?
(function = "legendre")	Fitting function
(*order = 2)	Number of polynomial terms or spline pieces
(sample = "")	Sample points to fit
(naverage = 1)	Number of sample points to combine
(niterate = 1)	Number of rejection iterations
(low_reject = 3.)	Low sigma rejection factor
(high_reject = 3.)	High sigma rejection factor
(grow = 0.)	Rejection growing radius
(mode = "ql")	

Kuva 2: ccdproc-komennon parametrit bias-vähennystä varten. Tähdellä on merkitty ne parametrit, joita tarvitsee muuttaa default-arvoista.

`biassec` (eli overscan-alueen sijainti) ja `trimsec` (eli mikä osa kuvasta säilytetään bias-vähennyksen jälkeen) riippuvat käytetystä CCD:stä, joten **niiden oikeat arvo kannattaa tarkistaa työn ohjaajalta**. Tässä esimerkissä `trimsec` on asetettu niin, että kuvan oikeasta ja vasemmasta reunasta leikkautuu pois se osa kuvasta, jolle ei näytä osuvan yhtään valoa.

Kun parametrit on asetettu, poistu `ctrl-d`:llä ja aja `ccdproc`:

```
ecl>ccdproc
```

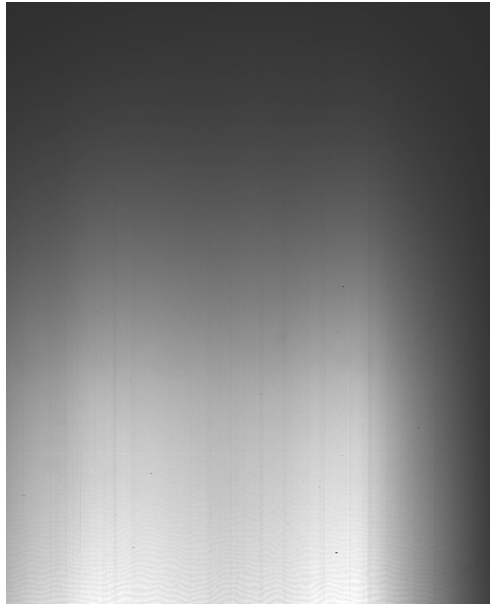
4.2.2 Bias-vähennys kun vain osa CCD-sirusta on luettu

Tee ensin lista bias-vähennettävistä kuvista kuten yllä on kuvattu. Valitse sitten jokin bias-kuva (tässä esimerkissä nimeltään `bias.fits`) ja anna komento:

```
ecl>imstat bias[x1:x2,y1:y2]
```

Hakasuluissa oleva osa määrittää kuvasta suorakulmaisen osa-alueen, jonka vasen alanurkka on pikselikoordinaateissa `x1`, `y1` ja oikea ylänurkka koordinaateissa `x2`, `y2`. Valitse noin 100×100 pix. kokoinen alue kuvan keskeltä. Komento tulostaa valitusta alueesta erilaisia tietoja, joista valitaan MEAN, mikä antaa ko. alueen keskimääräisen kirkkauden. Olkoon tämä arvo = `mean`. Bias-vähennys tehdään nyt komennolla

```
ecl>imarith @lista - mean @lista
```

Kuva 3: Kontinuumilampun kuva bias-vähennyksen jälkeen.

4.3 Kontinuumikuvan suoristaminen

Seuraavaksi on vuorossa kuvan *tasointus* eli pikselien herkkyysvaihteluiden korjaaminen. Tämä tehdään *tasointuskuvalla* (eng. *flat-field*), joka spektrihavaintojen tapauksessa on saatu aikaan ottamalla spektri ns. *kontinuumilampusta*, eli lampusta, joka lähettää jatkuvan spektrin. Kontinuumilampun kuva ei sellaisenaan vielä sovi tasointukseen, sillä sen intensiteetti muuttuu voimakkaasti pysty akselin suunnassa johtuen spektrograafin vaihtuvasta herkkyydestä aallonpituuden funktiona. Ennen tasointusta kuva täytyy “suoristaa”.

Kuvassa 3 on esitetty esimerkkihavaintojen tasointuskuva bias-vähennyksen jälkeen. Kuvasta näkyy selvästi, että sen kirkkaus muuttuu kuljettaessa pystysuoraan (eli kun aallonpituus muuttuu). Kirkkaus muuttuu myös hieman kuljettaessa vaakasuoraan (eli rakoa pitkin), mutta tätä epätasaisuutta emme korjaa näissä havainnoissa, koska se vaatisi ylimääräisiä kalibrointikuvia ja sen vaikutus tuloksiin on kohtalaisen pieni.

Ladataan ensin spektriredusoinneissa tarvittava paketti:

```
ecl>kpnoslit
```

Tarkistetaan, että *kpnoslit*-paketin asetukset ovat kunnossa:

```
ecl>epar kpnoslit
```

Tarkistetaan että *dispaxi* = “2”, eli kerrotaan *kpnoslit*:ille, että aallonpituus muuttuu pystysuoraan kuljettaessa. Tasointokuvan “suoristus” tapahtuu *response*-komennolla, siispä

*calibration = mALqk020038	Longslit calibration images
*normalizatio = mALqk020038	Normalization spectrum images
*response = resp	Response function images
(interactive = yes)	Fit normalization spectrum interactively?
(threshold = INDEF)	Response threshold
(sample = "")	Sample of points to use in fit
(naverage = 1)	Number of points in sample averaging
(function = "spline3")	Fitting function
(*order = 1)	Order of fitting function
(*low_reject = 3.)	Low rejection in sigma of fit
(*high_reject = 3.)	High rejection in sigma of fit
(*niterate = 5)	Number of rejection iterations
(*grow = 1.)	Rejection growing radius
(graphics = "stdgraph")	Graphics output device
(cursor = "")	Graphics cursor input
(mode = "ql")	

Kuva 4: response-komennon parametrit.

```
ecl>epar response
```

ja editoidaan parametrit kuvan 4 mukaisiksi. Tallenna ja aja response

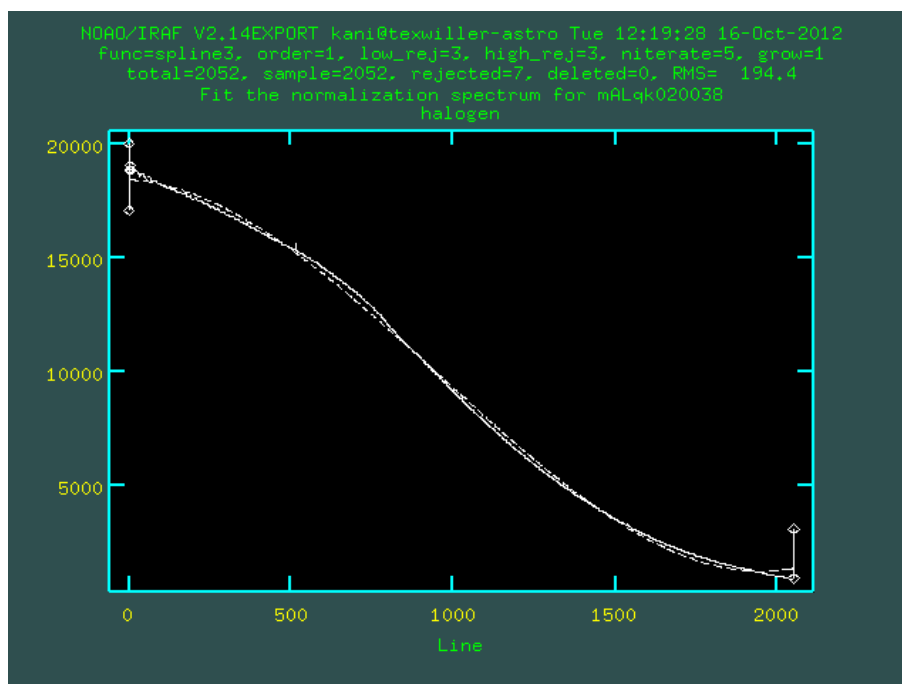
```
ecl>response
```

Koska asetimme parametrin `interactive` arvoon `yes`, avaa IRAF ikkunan, jossa on plotattuna kontinuumikuvan kirkkaus dispersioakselin suunnassa, eli tässä tapauksessa pystysuoran suunnan funktiona (kuva 5). Lisäksi näkyy kirkkauskäyrään sovitettu funktio katkoviivalla. Tavoitteena on saada sovitettu funktio vastaamaan havaittua dataa niin tarkasti kuin mahdollista, koska `response`-komento jakaa havaitun kontinuumilampun kuvan ko. funktiolla ja tallettaa lopputuloksen parametrin `response` ilmoittamaan kuvaan (tässä tapauksessa `resp.fits`), jolloin tuloksena on suoritettu kuva.

Kuvasta 5 nähdään, että sovitettu funktio (katkoviiva) ei ihan vastaa havaittua dataa (kiinteä viiva). Käytännössä olisi pyrittävä siihen, että nämä kaksi viivaa sulautuvat yhteen. Tämä saadaan aikaan nostamalla funktion kertalukua (parametri `order`). Siirretään siis kursori kuvan 5 ikkunan päälle (ja mahdollisesti klikataan ikkunan aktivoimiseksi) ja kirjoitetaan

```
:order 3
```

ja painetaan `<enter>` ja sen jälkeen painetaan `f` (fit). Nyt sovitettu funktio on lähempänä havaittua dataa. Kokeilemalla eri `order`-parametrin arvoja etsitään pienin arvo, joka "sulauttaa" kuvan 5 viivat yhteen. Kun ollaan tyytyväisiä, painetaan `q`.



Kuva 5: IRAFin avaama ikkuna `response`-komennon käynnistämisen jälkeen.

Esimerkin tapauksessa `order = 8` tuotti riittävän hyvän sovituksen. Tuloksena on kuva `resp.fits`, joka on esitetty kuvassa 6. Huomaa, kuinka kuvan “kaltevuus” pystysuorassa suunnassa on hävinnyt, mutta vaakasuorassa suunnassa on epätasaisuutta vielä jäljellä.

4.4 Kuvien tasoitus

Tasoituksen tarkoituksena on korjata CCD:n pikseleissä esiintyviä herkkyysvaihteluja, jotka johtuvat mm. valmistusvirheistä ja pölystä CCD:n edessä olevassa ikkunassa. Tasointa tehdään jakamalla havaitut kuvat tasoituskuvalle, joka luotiin edellisessä vaiheessa (`resp.fits`).

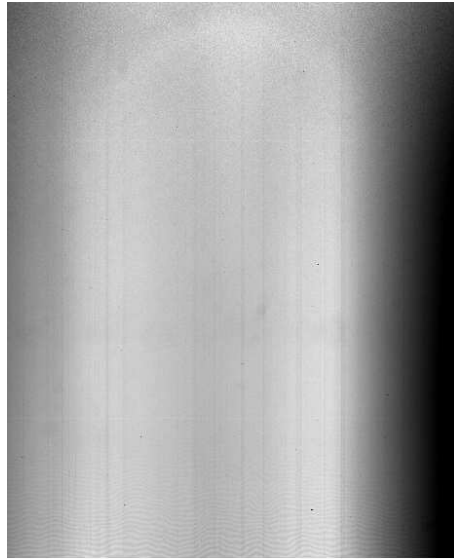
Editoidaan jälleen lista kuvista, joille tasoitus tehdään. Helpommin tämä käy kopiaamalla lista uudeksi tiedostoksi, esim. `tasoitettavat` ja poistamalla listasta `tasoitettavat` ne kuvat, joille tasoitusta ei tehdä. Esimerkin tapauksessa listalta poistuu kontinuumilampun spektri `ALqk020038` ja jäljelle jää 3 kuvaa. Kirjoita

```
ecl>epar ccdproc
```

ja aseta parametrit kuvan 7 mukaiseksi. Tämän jälkeen aja

```
ecl>ccdproc
```

ja kuvat on tasoitettu.



Kuva 6: Kontinuumilampun kuva `response`-komennon jälkeen.

4.5 Taustataivaan viivojen vähentäminen

Kuten jo aikaisemmin mainittiin, taustataivas lähettää viivasäteilyä eli säteilyä, joka on voimakasta tietyillä aallonpituuksilla. Tuloksena on vaakasuoria “viiruja” CCD-kuvassa (ks. kuva 1), joiden poistaminen spektristä on tämän vaiheen tavoitteena.

Ongelman luonteen ymmärtää ehkä parhaiten katsomalla kuvaa 8, jossa on zoomattu tasoitettuun kohteen spektriin. Kuva on tässä esitetty “negatiivina”, jotta himmeät yksityiskohdat näkyisivät paremmin. Kohteen spektri kulkee ylhäältä alaspäin kuvan keskellä. Neljä taustataivaan viivaa (yksi kirkas ja kolme himmeää) kulkee vaakasuoraan. Taustataivaan viivat kulkevat myös kohteen spektrin päältä ja aiheuttavat siihen ylimääräisiä emissioviivoja, jotka eivät ole peräisin kohteesta.

Taustavähennys tehdään mittaamalla taustaviivojen kirkkaus spektrin oikealla ja vasemmalla puolella (eli spatiaalisessa suunnassa) ja vähentämällä tulos koko kuvasta. Mittaus tapahtuu sovittamalla halutunasteinen polynomi taustataivaan pikseleihin spatiaalisessa suunnassa ja vähentämällä polynomi kaikista pikseleistä spatiaalisella akselilla. IRAFissa tämä tapahtuu `background`-komennolla. Kirjoita

```
ecl>epar background
```

ja editoi parametrit kuvan 9 mukaisiksi. Tärkeimmät parametrit ovat `axis`, `sample`, `naverage`, `function` ja `order`.

Parametri `axis` kertoo spatiaalisen akselin suunnan (1=vaakasuoraan, 2=pystysuoraan) eli mihin suuntaan polynomisovitus tehdään.

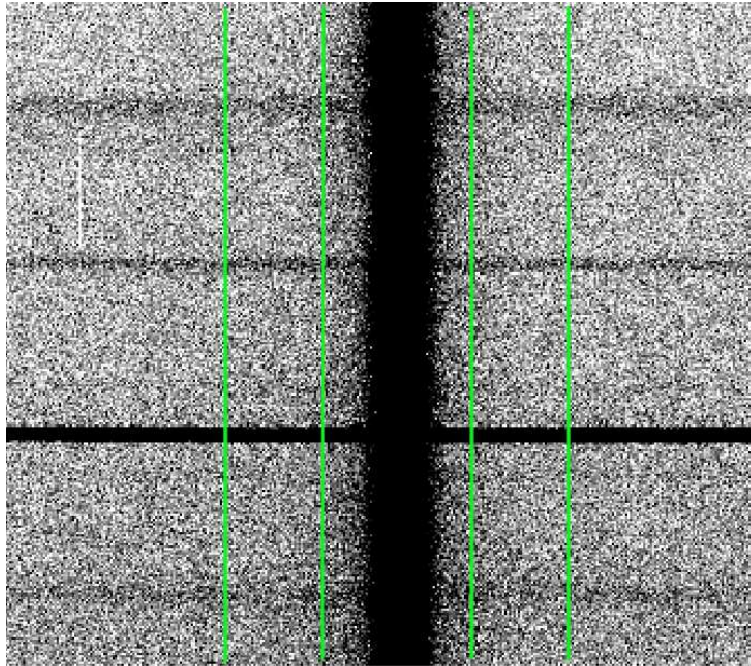
Parametri `sample` kertoo alueen, josta taustataivaan määrittäminen tehdään. Periaatteessa alueita voisi olla kuinka monta tahansa, mutta käytännössä riittää valita kaksi aluetta, yksi spektrin molemmilta puolilta. Alueet pitäisi valita mahdollisimman läheltä spektriä, mutta kuitenkin niin, että spektrin valoa ei havaittavasti osu niihin. Kuvassa 8 vihreät pystyviivat osoittavat spektrin molemmilta puolilta sopi-

```

    *images = @tasoitettavat List of CCD images to correct
  (output = )              List of output CCD images
  (*ccdtype = )           CCD image type to correct
(max_cache = 0)          Maximum image caching memory (in Mbytes)
  (noproc = no)          List processing steps only?\n
  (*fixpix = no)         Fix bad CCD lines and columns?
  (*overscan = no)      Apply overscan strip correction?
    (*trim = no)        Trim the image?
  (*zerocor = no)       Apply zero level correction?
  (*darkcor = no)       Apply dark count correction?
  (flatcor = yes)       Apply flat field correction?
  (illumcor = no)       Apply illumination correction?
  (fringecor = no)     Apply fringe correction?
  (readcor = no)       Convert zero level image to readout correction?
  (scancor = no)       Convert flat field image to scan correction?\n
  (readaxis = "line")  Read out axis (column|line)
  (fixfile = "")        File describing the bad lines and columns
  (*biassec = [5:40,1:2052] )  Overscan strip image section
  (*trimsec = [290:1960,1:2052])  Trim data section
    (zero = "")         Zero level calibration image
    (dark = "")         Dark count calibration image
    (*flat = resp)      Flat field images
    (illum = "")        Illumination correction images
    (fringe = "")       Fringe correction images
  (minreplace = 1.)    Minimum flat field value
  (scantype = "shortscan")  Scan type (shortscan|longscan)
    (nscan = 1)        Number of short scan lines\n
  (interactive = no)   Fit overscan interactively?
  (function = "legendre")  Fitting function
    (*order = 2)       Number of polynomial terms or spline pieces
    (sample = "*" )    Sample points to fit
  (naverage = 1)      Number of sample points to combine
  (niterate = 1)      Number of rejection iterations
  (low_reject = 3.)   Low sigma rejection factor
  (high_reject = 3.)  High sigma rejection factor
    (grow = 0.)        Rejection growing radius
    (mode = "ql")

```

Kuva 7: ccdproc-komennon parametrin tasoitusta varten.



Kuva 8: Zoomaus kohteen spektriin. Vihreät pystyviivat osoittavat alueen, josta taustataivaan määrittäminen kannattaa tehdä.

vat alueet taustataivaan määrittämiseen. Valitut alueet kerrotaan IRAFille muodossa $x1:x2,x3:x4$, missä esim. $x1$ on kuvan 8 vasemman alueen alaraja ja $x2$ sen yläraja vaakasuorassa suunnassa.

Parametrilla `naverage` voidaan ohjeistaa IRAFia käyttämään useamman pikselin keskiarvoa tai mediaania yksittäisten pikseliarvojen sijasta. Mediaani on yleensä tehokas suodattamaan pois esim. kosmisiä säteitä, jotka muuten tuottaisivat kohinaa tuloksiin. Asettamalla `sample` miinusmerkkiseksi kerrotaan IRAFille, että halutaan käyttää mediaania. Esimerkin tapauksessa siis lasketaan ensin viiden pikselin mediaani ja polynomisovitus tehdään näihin mediaaniarvoihin.

Parametri `function` kertoo sovitettavan funktion ja `order` kertoo polynomin kertaluvun jos funktioksi on asetettu `legendre`. Asettamalla `order = 1` sovitaan pelkkä vakioarvo, `order = 2` vastaa vakioarvoa + lineaarista muutosta, `order = 3` vakioarvoa + lineaarista muutosta + toisen asteen muutosta (paraabeli), jne. Mitä korkeampi `order` asetetaan, sitä "taipuisampi" polynomista tulee. Käytännössä `order = 2` antaa riittävän hyvän sovituksen taustataivaan pikseleihin, joskus `order = 3` voi tulla kyseeseen.

Kun parametrit on asetettu, aja

```
ecl>background
```

ja tuloksena on taustavähennetty kuva `backmALqk020036.fits`, joka on esitetty kuvassa 10. Huomaa, että kuvan reunoilla taustataivaan vähennys ei ole kovin hyvä, koska polynomisovitus tehtiin spektrin lähellä oleviin pikseleihin. Tämä ei

```

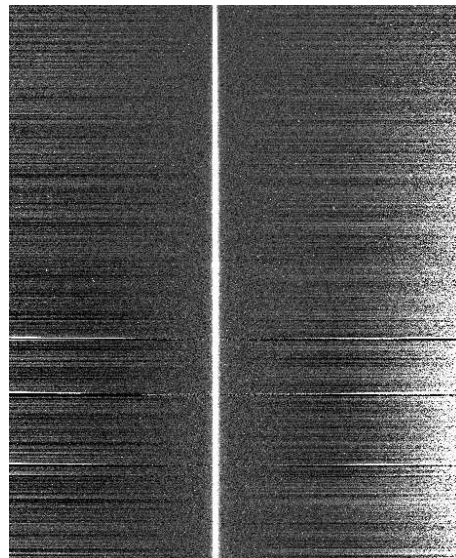


```

Kuva 9: background-komennon parametrit.

esimerkin tapauksessa haittaa, koska olemme kiinnostuneita vain kuvan keskellä olevasta alueesta.

Taustavähennyksen onnistuminen kannattaa vielä tarkistaa tutkimalla taustataivaan pikseliarvoja taustavähennyssä kuvassa. Pikseliarvojen tulisi vaihdella nollan molemmin puolin, sillä taustahan on nyt vähennetty, eli tausta = 0 ja jäljellä pitäisi olla vain kohinaa nollan molemmin puolin.



Kuva 10: Kohteen spektri taustaviivojen vähennyksen jälkeen.

4.6 Spektrin mittaaminen

Tässä vaiheessa kaksiulotteisesta CCD-kuvasta mitataan yksiulotteinen spektri. Mittaus tapahtuu komennolla `apall`, joka suorittaa seuraavat toimenpiteet:

- määritetään spektrin paikka karkeasti CCD-kuvassa
- määritetään “jälki” (trace), jota pitkin spektrin kirkkain kohta kulkee CCD:llä
- kuljetaan pitkin määritettyä “jälkeä” dispersiosuunnassa (tässä tapauksessa pystysuorassa) yhden pikselin askelein ja mitataan spektrin kirkkaus jokaisessa kohdassa. Koska spektri on levinnyt spatiaalisessa suunnassa (vaakasuorassa) ilmähän vaikutuksesta, täytyy kullakin askeleella summata yhteen kaikkien pikselien kirkkaus spatiaalisessa suunnassa.

Komennolla `apall`, on paljon säädettäviä parametreja, enemmän kuin millään muulla komennolla. Onneksi läheskään kaikkia parametreja ei tarvitse muuttaa vaan oletusarvo on suurimmassa osassa riittävä. Kirjoitetaan siis rohkeasti

```
ecl>epar apall
```

ja editoidaan parametrit kuvien 11, 12 ja 13 mukaiseksi. Parametrit, joiden arvoa täytyy muuttaa ovat:

`input` : Tähän tulee taustavähennetyn kuvan nimi.

`output` : Yksiulotteisen spektrin nimi.

`resize = no` : Default-arvo on, että spektrin yli kuljetettavan mittausapertuurin koko vaihtelee spektrin kirkkauden mukaan. Tämä on hyödyllistä tapauksissa, joissa signaali/kohinasuhde vaihtelee suuresti spektrin eri kohdissa. Tässä tapauksessa vakioapertuuri on kuitenkin riittävä, koska signaali/kohinasuhde on korkea.

`lower`, `upper` : Mittausapertuurin alempi ja ylempi raja spektrin keskikohdan suhteen. Tämä antaa siis mittausapertuurin koon spatiaalisessa suunnassa (tässä siis vaakasuorassa). Kaikki spektrin pikseliarvot, jotka osuvat apertuurin sisälle summataan spatiaalisessa suunnassa ja lopputulos tallennetaan spektrin kirkkaudeksi ko. kohdassa dispersioakselilla (eli käytännössä siis tallentuu spektrin kirkkaus pystysuoran pikselikoordinaatin funktiona). Apertuurin koko kannattaa valita siten, että se kattaa koko näkyvän osan spektristä.

`width`, `radius` : Keskitysapertuurin koko. Spektrin huippukohta määritetään apertuurissa, jonka koko annetaan tässä. Tämän apertuurin ei tarvitse kattaa koko spektrin näkyvää osaa, tyypillisesti noin yksi neljäsosa riittää. Huomaa, että `width` on apertuurin koko ja `radius = width / 2`.


```

    *input = backmALqk020036 List of input images
    (*output = 1dmALqk02003) List of output spectra
    (apertures = "") Apertures
    (format = "multispec") Extracted spectra format
    (references = "") List of aperture reference images
    (profiles = "") List of aperture profile images

(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?

    (line = INDEF) Dispersion line
    (nsum = 10) Number of dispersion lines to sum or median

#DEFAULT APERTURE PARAMETERS

    (*lower = -35.) Lower aperture limit relative to center
    (*upper = 35.) Upper aperture limit relative to center
    (apidtable = "") Aperture ID table (optional)

# DEFAULT BACKGROUND PARAMETERS

    (b_function = "chebyshev") Background function
    (b_order = 1) Background function order
    (b_sample = "-10:-6,6:10") Background sample regions
    (b_naverage = -3) 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

# APERTURE CENTERING PARAMETERS

    *(width = 14.) Profile centering width
    (*radius = 7.) Profile centering radius
    (threshold = 0.) Detection threshold for profile centering

```

Kuva 11: apall-komennon parametreja.

```

# AUTOMATIC FINDING AND ORDERING PARAMETERS

nfind = 1           Number of apertures to be found automatically
(minsep = 5.)      Minimum separation between spectra
(maxsep = 1000.)   Maximum separation between spectra
(order = "increasing") Order of apertures

# RECENTERING PARAMETERS

(aprecenter = "")  Apertures for recentering calculation
(npeaks = INDEF)   Select brightest peaks
(shift = yes)      Use average shift instead of recentering?

# RESIZING PARAMETERS

(llimit = INDEF)   Lower aperture limit relative to center
(ulimit = INDEF)   Upper aperture limit relative to center
(ylevel = 0.1)     Fraction of peak or intensity for automatic width
  (peak = yes)     Is ylevel a fraction of the peak?
  (bkg = yes)      Subtract background in automatic width?
(r_grow = 0.)      Grow limits by this factor
(avglimits = no)   Average limits over all apertures?

# TRACING PARAMETERS

(t_nsum = 10)      Number of dispersion lines to sum
(t_step = 10)      Tracing step
(t_nlost = 3)      Number of consecutive times profile is lost before
(t_function = "legendre") Trace fitting function
(*t_order = 3)     Trace fitting function order
(t_sample = "*")   Trace sample regions
(t_naverage = 1)   Trace average or median
(t_niterate = 0)   Trace rejection iterations
(t_low_reject = 3.) Trace lower rejection sigma
(t_high_rejec = 3.) Trace upper rejection sigma
(t_grow = 0.)      Trace rejection growing radius

```

Kuva 12: Lisää apall-komennon parametreja.

```

# EXTRACTION PARAMETERS

(background = "none")      Background to subtract
(skybox = 1)              Box car smoothing length for sky
(*weights = "variance")   Extraction weights (none|variance)
(pfitt = "fit1d")         Profile fitting type (fit1d|fit2d)
(clean = no)              Detect and replace bad pixels?
(saturation = INDEF)      Saturation level
(*readnoise = "5.3")      Read out noise sigma (photons)
(*gain = "0.736")         Photon gain (photons/data number)
(lsigma = 4.)             Lower rejection threshold
(usigma = 4.)             Upper rejection threshold
(nsubaps = 1)             Number of subapertures per aperture
(mode = "ql")

```

Kuva 13: Vielä lisää apall-komennon parametreja.

`t_order` : Kuten aikaisemmin jo mainittiin, `apall` seuraa spektrin huippukohtaa siirryttäessä pitkin dispersioakselia alhaalta ylöspäin, eli määrittää spektrin “jäljen”. Kuvasta 10 näyttäisi siltä, että spektrin paikka ei muutu kuljettaessa pystysuoraan, eli spektri näyttäisi pysyvän samoissa CCD-kuvan kolumneissa koko matkan alhaalta ylös. Todellisuudessa spektri on kuitenkin hieman kaareva johtuen spektrograafin optiikan aiheuttamista kuvausvirheistä. Ero on niin pieni, että sitä ei kuvasta 10 huomaa, eikä sillä tässä tapauksessa ole suurtakaan käytännön merkitystä, mutta koska se on helppo ottaa huomioon, tehdään se tässä.

IRAF tallentaa spektrin “jäljen” polynomina, jonka kertaluokka `t_order` kertoo kuinka “taipuisa” jälki halutaan tallentaa. Yleensä `t_order = 3` (paraabeli) antaa riittävän korkean kertaluokan.

`weights` : Kun tämä asetetaan arvoon `yes`, spektrin summaus tehdään painotetusti, eli kirkkaampia pikseleitä painotetaan enemmän kuin himmeitä pikseleitä, mikä on hyvä asia.

`readnoise`, `gain` : Tähän kirjoitetaan käytetyn CCD:n lukukohina ja vahvistuskerroin. **Huom! Tarkista nämä ohjaajalta, koska nämä riippuvat käytetystä CCD:stä!**

Kun kaikki parametrit on asetettu, aja

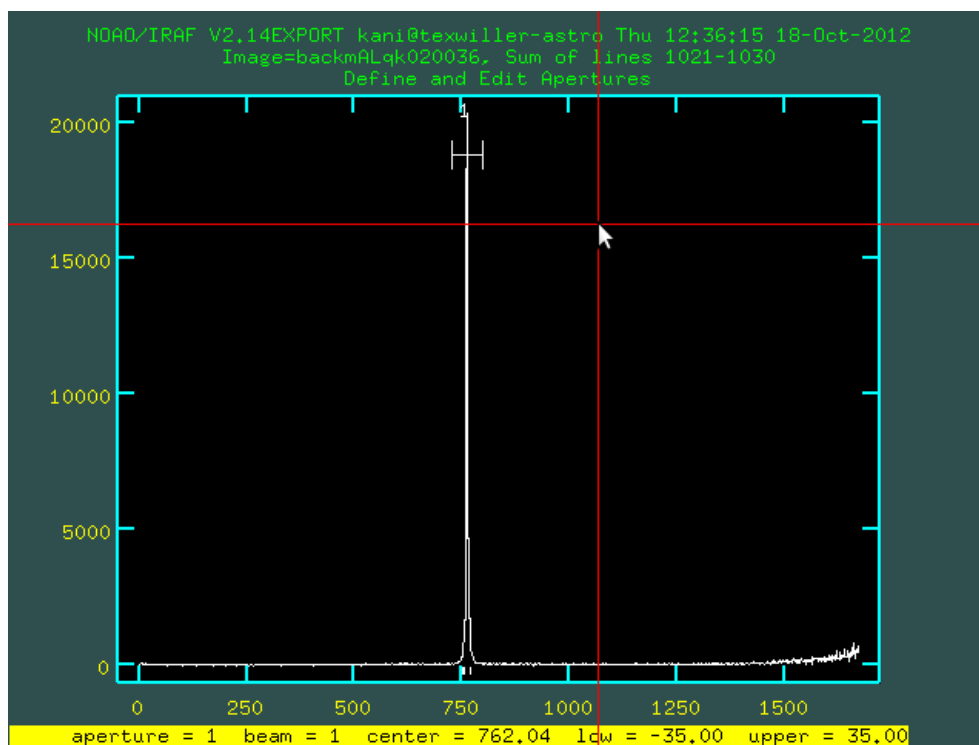
```
ecl>apall
```

jolloin `apall` kysyy ensin muutaman kysymyksen:

```

List of input images (backmALqk020036):
Find apertures for backmALqk020036? (yes):
Number of apertures to be found automatically (1):
Edit apertures for backmALqk020036? (yes):

```



Kuva 14: apall-komennon ensimmäiseksi avaama ikkuna.

joihin vastataan painamalla `<enter>` (eli hyväksytään suluissa olevat default-arvot). Tämän jälkeen `apall` avaa ensimmäisen ikkunan (kuva 14). Kuvassa on kuvan `backmALqk020036` poikkileikkaus vaakasuorassa suunnassa. Keskellä oleva korkea piikki on itse spektri. IRAF on automaattisesti etsinyt spektrin paikan ja merkinnyt mittausapertuurin paikan kuvaan vaakasuoralla palkilla ja numerolla 1. Mittaajan tehtävä tässä vaiheessa on tarkistaa, että apertuuri sijaitsee edes likipitäen spektrin kohdalla. Kuvassa näin näyttäisi olevan, eli voidaan painaa `q` (quit) ja jatkaa seuraavaan vaiheeseen.

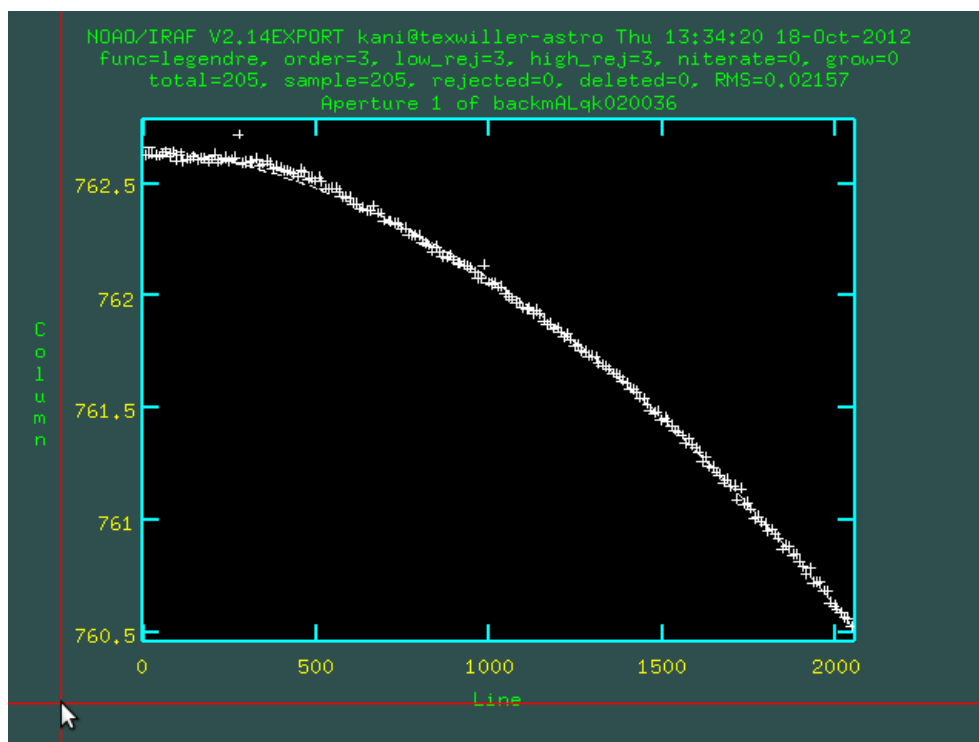
Mikäli jostain syystä IRAF ei onnistu löytämään spektrin paikkaa oikein, täytyy se tehdä manuaalisesti. Siirrä siis kursori lähelle palkkia ja paina `d` (delete) ja `r` (replot), jolloin väärin merkitty palkki katoaa. Siirrä nyt kursori spektriipiikin huipun kohdalle ja paina `m` (mark), jolloin apertuurin paikkaa ilmaisevan palkin pitäisi ilmestyä oikeaan kohtaan. Tämän jälkeen paina `q` (quit).

Nyt `apall` kysyy plottausikkunan alalaidassa

```
Trace apertures for backmALqk020036? (yes):
Fit traced positions for backmALqk020036 interactively? (yes):
Fit curve to aperture 1 of backmALqk020036 interactively? (yes):
```

joihin kaikkiin vastataan painamalla `<enter>` ja kuvan 15 kaltainen ikkuna aukeaa.

Kuvassa näkyy nyt spektrin huippukohdan sijainti kullakin CCD-kuvan rivillä (plus-merkit) ja sijainteihin tehty sovitus eli spektrin "jälki" (katkoviiva). Kuvasta nähdään, että spektrin huippu pysyy melko tarkkaan kolumnissa 761-762, eli se



Kuva 15: apall-komennon toiseksi avaama ikkuna.

muuttuu vain pari pikseliä kun kuljetaan kuvan alalaidasta ylälaitaan. Pieni kaarevuus on kuitenkin nähtävissä.

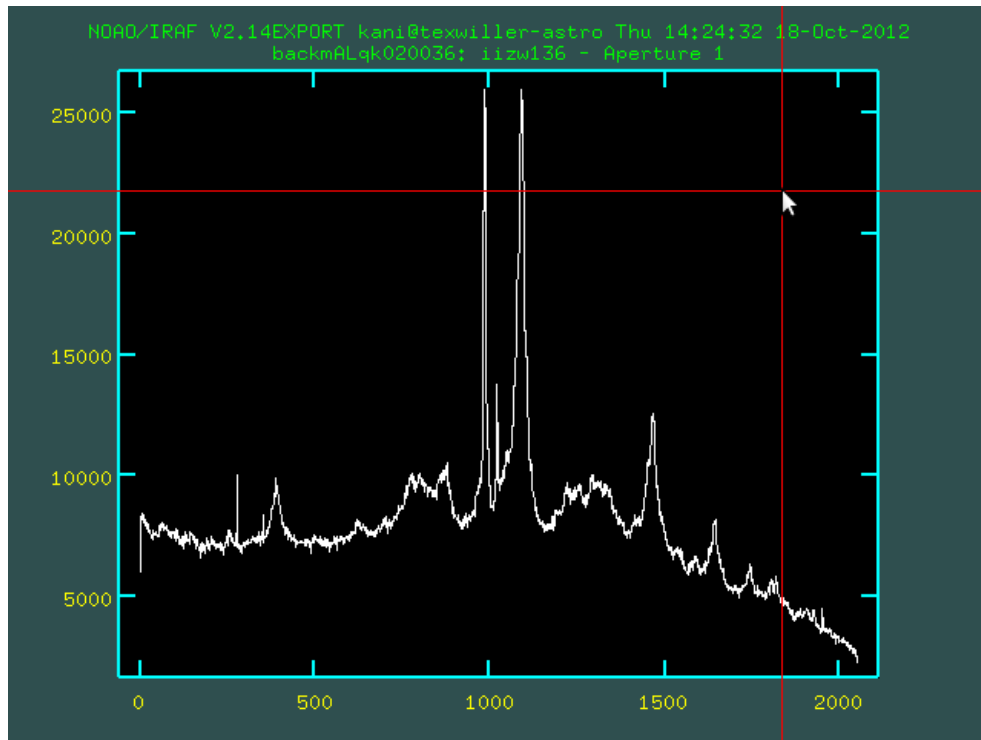
Mittaajan tehtävänä on nyt varmistaa että sovitettu “jälki” vastaa havaittuja sijainteja riittävän tarkasti. Mikäli näin ei ole, syynä saattaa olla esim. se, että valittu polynomifunktion kertaluokka (parametri t_order) on liian korkea/matala. Kertaluokkaa voi muuttaa kirjoittamalla plottausruudussa esim. `:order 1 <enter>` ja `f (fit)`. Kokeilemalla eri arvoja voidaan ehkä saavuttaa parempi sovitus. Joskus havaituissa sijainneissa esiintyy yksi muista kaukana oleva piste, joka aiheuttaa virhettä sovitukseen. Virheellinen piste voidaan poistaa asettamalla kursori sen päälle ja painamalla `d (delete)` ja `f (fit)`.

Kun ollaan tyytyväisiä sovitukseen painetaan `q (quit)`, jolloin `apall` kysyy plottausikkunan alalaidassa

```
Write apertures for backmALqk020036 to database? (yes):
Extract aperture spectra for backmALqk020036? (yes):
Review extracted spectra from backmALqk020036? (yes):
Review extracted spectrum for aperture 1 from backmALqk020036? (yes):
```

joihin kaikkiin vastataan taas painamalla `<enter>` ja kuvan 16 kaltainen ikkuna aukeaa.

Kuvassa on nyt `apall`-komennon lopputulos, eli kohteen 1-ulotteinen spektri. Esimerkin tapauksessa näkyy selvästi useita emissioviivoja, jotka syntyvät supermassiivista mustaa aukkoa ympäröivissä kaasupilvissä. Tämä ei ole kuitenkaan vielä valmis spektri, sillä vaaka-akselilla on pikselikoordinaatti eikä aallonpituus



Kuva 16: apall-komennon kolmanneksi avaama ikkuna.

ja pystyakselilla spektrin voimakkuus ADUissa eikä "oikeissa" fysikaalisissa yksiköissä.

Painetaan vielä q (quit) ja spektri tallentuu levyille, tässä tapauksessa nimellä `1dmALqk020036.fits`. Lisäksi hakemistoon database tallentuu spektrin "jälki".

4.6.1 Sivujuonne: `splot`

Tästä eteenpäin käsitellään yksiulotteisia spektrejä, joiden tarkasteleminen esim. IRAFin `display`-komennolla on toki mahdollista, mutta ei erityisen informatiivista. Yksiulotteisia spektrejä on helpointa visualisoida komennolla `splot`.

Tarkastellaan esimerkiksi juuri mitattua spektriä `1dmALqk020036.fits`. Kirjoitetaan

```
ecl>splot 1dmALqk020036
```

```
jolloin splot kysyy
```

```
Image band to plot (1:) (1):
```

johon vastataan painamalla `<enter>`. Tällöin avautuu kuvan 16 kaltainen ikkuna, jonka olemme nähneet jo aikaisemmin. `splot`-komennolla voi suorittaa monia operaatioita, joista alla tärkeimmät:

a-näppäimellä voidaan zoomata spektrin eri osiin. Siirrä kursori halutun zoomausalueen vasempaan laitaan ja paina **a**, ja toista sama halutun zoomausalueen oikeassa laidassa. Koko spektri saadaan taas näkyviin painamalla **c**.

q-näppäimellä lopetetaan `splot`.

k-näppäimellä voidaan mitata spektriviivojen paikkoja, leveyksiä ja voimakkuuksia. Aseta kursori mitattavan viivan vasemmalle puolelle kohtaan, jossa kontinuumi sijaitsee (kontinuumi on se osa spektristä, jossa ei näy spektriviivoja) ja paina **k**-näppäintä. Kursori asetetaan siis vaakasuorassa suunnassa mahdollisimman lähellä mitattavaa viivaa, mutta kuitenkin niin kauas ettei viivan säteily vaikuta tulokseen ja pystysuorassa suunnassa siihen kohtaan jossa spektri kulkee. Siirrä kursori viivan oikealle puolelle vastaavaan kohtaan ja paina uudestaan **k**.

IRAF näyttää nyt spektriviivaan sovitetun Gaussin funktion punaisella ja alalaidassa tietoja sovitetusta viivasta: viivan keskikohta (`center`), kirkkaus (`flux`) ja leveys (`g_fwhm`, ks. esim kuva 29). Vihreällä katkoviivalla on merkitty oletettu tausta, joka on lineaarinen ja jonka tason määrää kursorin sijainti **k**-näppäimen painamisen aikana.

```

        *input = "mALqk020037"   List of input images
        (*output = "1dmALqk020037") List of output spectra
        (apertures = "")          Apertures
        (format = "multispec")    Extracted spectra format
        (*references = "backmALqk020036") List of aperture reference images
        (profiles = "")          List of aperture profile images\n

(*interactive = no)             Run task interactively?
        (*find = no)             Find apertures?
        (*recenter = no)        Recenter apertures?
        (*resize = no)          Resize apertures?
        (*edit = no)            Edit apertures?
        (*trace = no)           Trace apertures?
        (*fittrace = no)        Fit the traced points interactively?
        (extract = yes)         Extract spectra?
        (extras = yes)          Extract sky, sigma, etc.?
        (review = yes)          Review extractions?\n

```

Kuva 17: apall-komennon parametrit viivaspektrin mittausta varten.

4.7 Spektrin mittaaminen viivalampun kuvasta

Seuraavana on vuorossa äsken saadun spektrin aallonpituuskalibrointi, ts. määritetään muunnos pikselikoordinaateista aallonpituuksiin. Tarkoitusta varten on havaittu viivalampun spektri, josta ensin mittaamme yksiulotteisen spektrin käyttäen jälleen apall-komentoa.

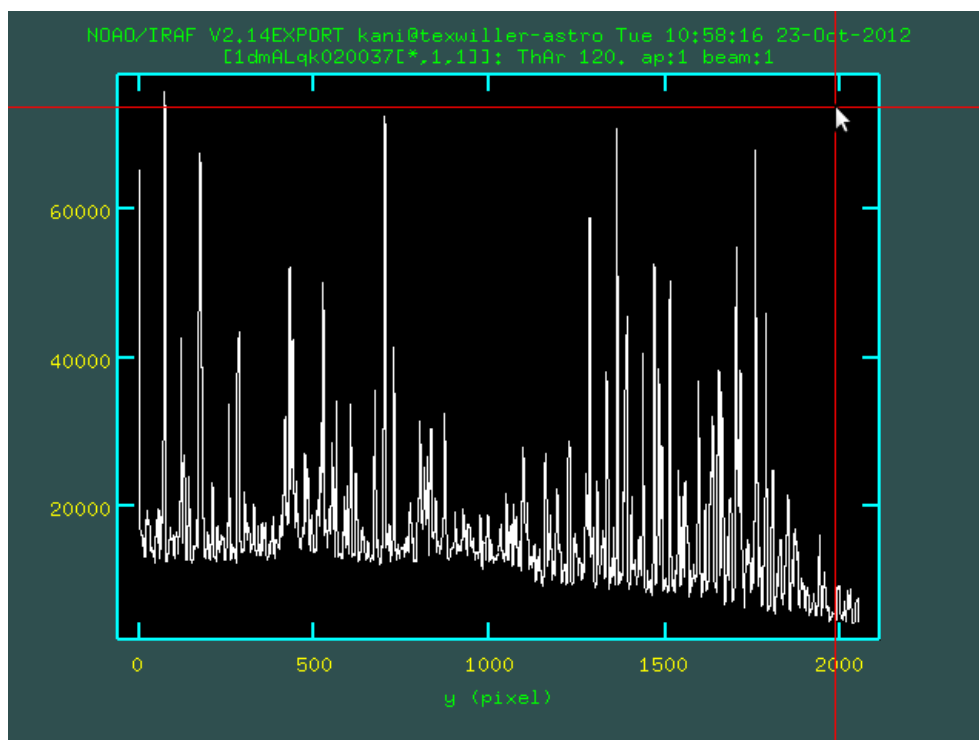
Viivalampun spektrin mittaaminen on onneksi huomattavasti helpompaa kuin kohteen spektrin. Joka kerta kun jokin spektri mitataan apall-komennolla, tallentuvat spektrin “jälki”, mittausapertuuri, yms. tiedot database-hakemistoon nimellä database/apXXXX, missä XXXX on sen kuvan nimi, johon apall ajettiin. Viivalampun spektri voidaan helposti mitata CCD-kuvasta käyttämällä jo aikaisemmin määritettyä “jälkeä”. Tämä on muutenkin erittäin suositeltavaa, sillä on tärkeää, että kalibrointispektri on mitattu täsmälleen samalla tavalla kuin kohteen spektri.

Editoidaan siis apall-komennon parametrit kuvan 17 mukaisiksi (muut parametrit jätetään ennalleen). Tärkeä ero edelliseen mittaukseen on parametri references, joka ei ole tällä kertaa tyhjä, vaan ilmoittaa referenssikuvan nimen (eli sen kuvan nimen, jonka “jälkeä” käytetään). Tässä tapauksessa käytetään referenssikuvana kohteen CCD-kuvaa backmALqk020036, jonka “jälki” mitattiin edellisessä vaiheessa. Tallennetaan parametrit ja ajetaan

```
ecl>apall
```

Tällä kertaa apall ei kysy mitään vaan mittaa viivaspektrin suoraan CCD-kuvasta käyttäen samaa “jälkeä” kuin kohteen spektrissä. Tulosta voi tarkastella komennolla

```
ecl>plot 1dmALqk020037
```

Kuva 18: Viivalampun spektri.

Tuloksena on kuvan 18 kaltainen spektri, joka sisältää runsaasti viivalampun lähettämiä emissioviivoja.


```

    *images = 1dmALqk020037  Images containing features to be identified
    (section = "middle line") Section to apply to two dimensional images
    (database = "database")   Database in which to record feature data
    (coordlist = "linelists$idheneardat") User coordinate list
    (units = "")              Coordinate units
    (nsum = "10")             Number of lines/columns/bands to sum in 2D imag
    (match = -3.)             Coordinate list matching limit
    (maxfeatures = 50)        Maximum number of features for automatic identi
    (zwidth = 100.)           Zoom graph width in user units
    (ftype = "emission")     Feature type
    (fwidth = 4.)             Feature width in pixels
    (cradius = 5.)           Centering radius in pixels
    (threshold = 0.)          Feature threshold for centering
    (minsep = 2.)             Minimum pixel separation
    (*function = "legendre")  Coordinate function
    (*order = 3)              Order of coordinate function
    (sample = "*")           Coordinate sample regions
    (niterate = 0)           Rejection iterations
    (low_reject = 3.)         Lower rejection sigma
    (high_reject = 3.)        Upper rejection sigma
    (grow = 0.)              Rejection growing radius
    (autowrite = no)          Automatically write to database
    (graphics = "stdgraph")  Graphics output device
    (cursor = "")            Graphics cursor input
    crval =                   Approximate coordinate (at reference pixel)
    cdelt =                   Approximate dispersion
    (aidpars = "")           Automatic identification algorithm parameters
    (mode = "ql")

```

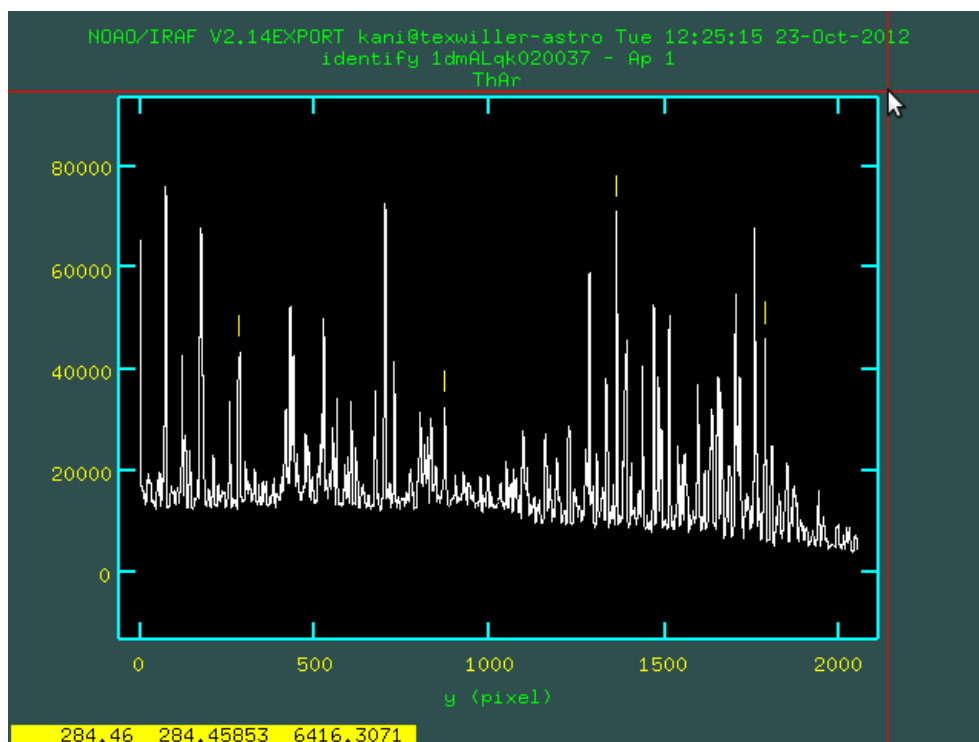
Kuva 20: identify-komennon parametrit.

```
ecl>identify
```

ja kuvan 21 kaltainen ikkuna aukeaa (tässä vaiheessa ei vielä näy kuvassa esiintyviä keltaisia pystyviivoja). Mittaajan tehtävänä on nyt tunnistaa kuvan 21 spektristä ne emissioviivat, joiden aallonpituus on merkitty kuvaan 19, merkitä ne ja kertoa IRAFille mitkä ovat merkittyjen viivojen aallonpituudet. Sen jälkeen IRAF laskee muunnoksen pikselikoordinaateista aallonpituuksiin käyttäen halutunlaista polynomia.

Tunnistaminen on joskus hieman hankalaa, sillä viivoja on paljon, niiden havaitut voimakkuudet eivät aina ihan täsmälleen vastaa kuvan 19 voimakkuuksia ja kuvassa 21 aallonpituus voi kasvaa vasemmalta oikealle tai oikealta vasemmalle (esimerkin tapauksessa aallonpituus kasvaa oikealta vasemmalle, eli havaittu spektri on kuvan 19 peilikuva).

Käytännössä viivojen merkitseminen tapahtuu niin, että siirretään kursori tun-



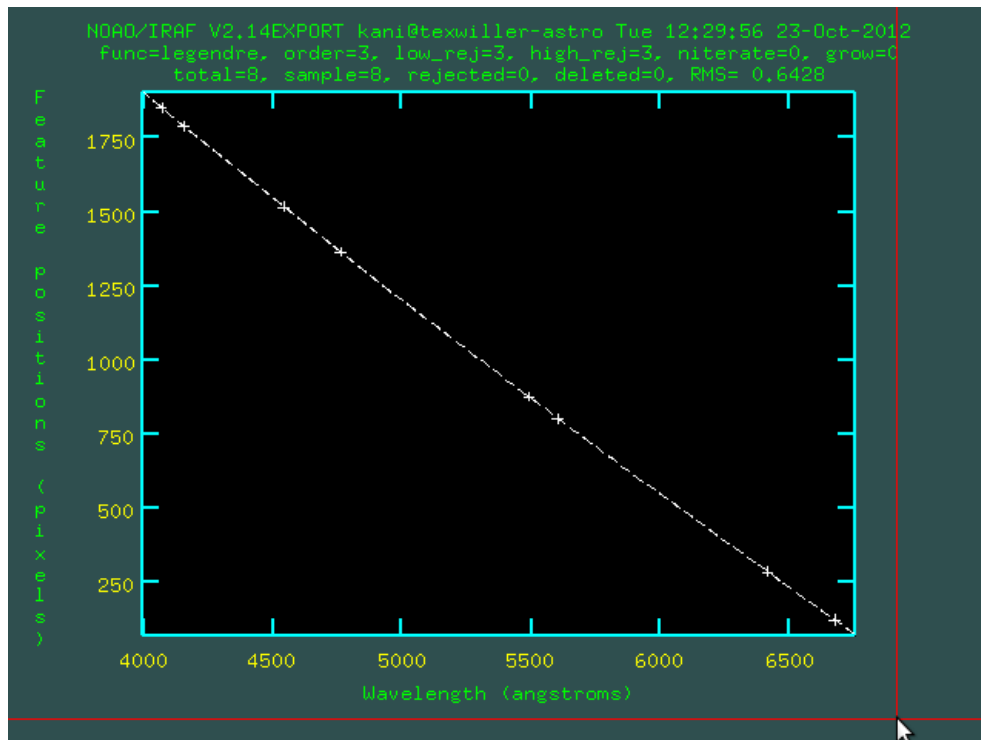
Kuva 21: `identify`-komennon avaama ikkuna. Neljä spektriviivaa on jo merkitty `m`-näppäimellä.

nistetun viivan huipun kohdalle ja painetaan `m`-näppäintä. Plottausruudun alalaitaan ilmestyy nyt joitakin numeroita ja kaksoispiste, eli IRAF odottaa juuri merkityn viivan aallonpituutta. Luetaan se kuvasta 19, kirjoitetaan plottausruutuun ja painetaan `<enter>`. Merkityn viivan kohdalla näkyy nyt keltainen pystyviiva. Toistetaan tämä neljälle eri viivalle, jotka on valittu koko spektrin alueelta (ks. kuva 21).

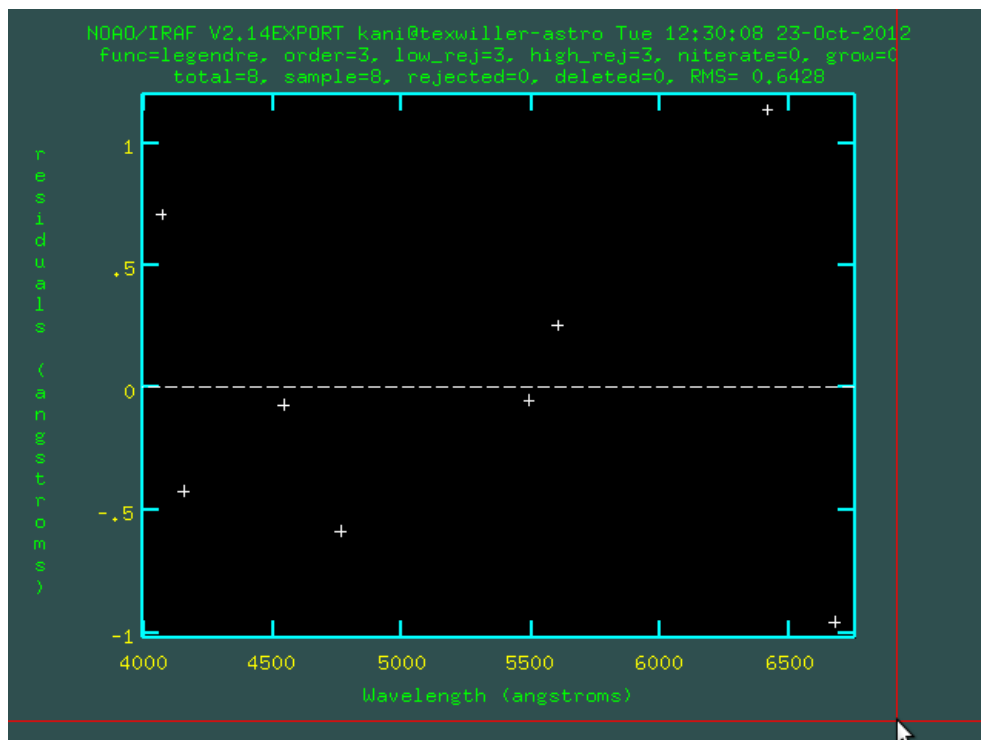
Jokaisen merkitsemisen jälkeen on hyvä tarkastaa, että keltainen merkki osuu juuri siihen viivaan kuin oli tarkoitus ja että se on viivan keskellä. Parhaiten tämä onnistuu painamalla `z` (zoom). Koko spektrin näkee taas painamalla `p` (pan). Mikäli merkki on väärässä paikassa, siirrä kursori ko. merkin päälle, paina `d` (delete), ja yritä uudestaan. Mikäli keskitys ei tunnu millään onnistuvan, voi kokeilla säätää `identify`-komennon parametreja `fwidth` ja `radius` hieman suuremmiksi.

Kun neljä viivaa on oikein merkitty, paina `f` (fit), jolloin IRAF laskee polynomin, joka parhaiten kuvaa muunnosta pikselikoordinaateista aallonpituuksiin. Tässä olemme valinneen polynomin asteeksi `order = 3` (paraabeli), mikä useimmin on riittävän “taipuisa” hyvän sovituksen aikaansaamiseksi. Laskun jälkeen IRAF esittää sovituksen tuloksen. Tulos esitetään kuvien 22, 23 ja 24 kaltaisina plottauksina, joissa tarkastellaan sovituksen tulosta eri tavoilla:

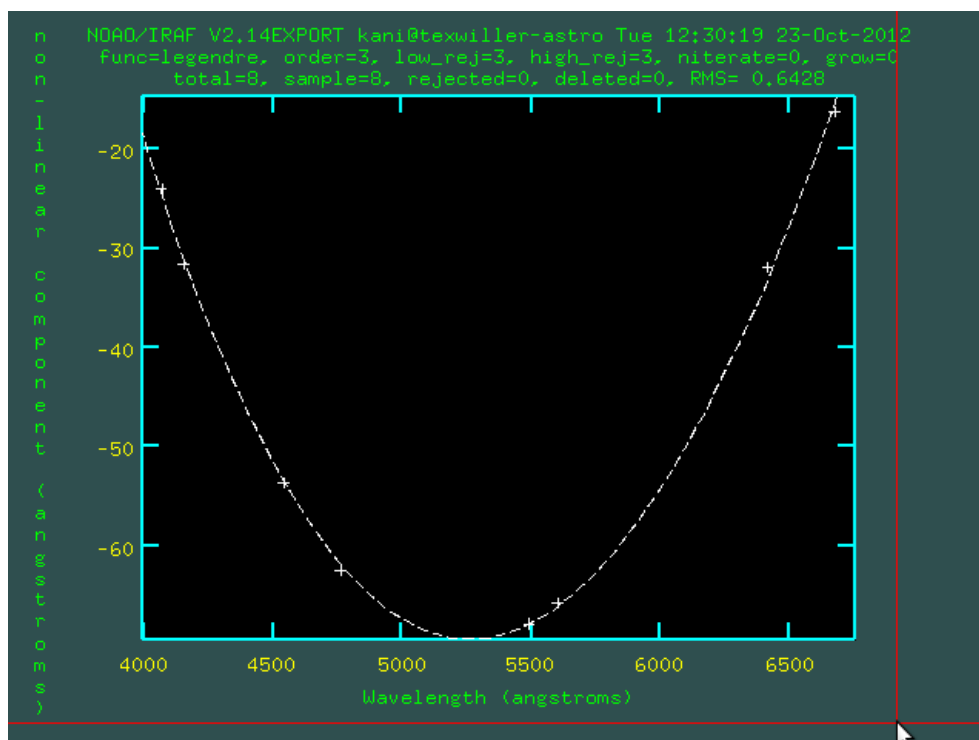
`h`-näppäimellä nähdään itse sovitus, ts. pikselikoordinaatit aallonpituuden funktiona (kuva 22). Luonnollisesti sovitetun funktion (katkoviiva) tulisi kulkea havaittujen pisteiden (plus-merkit) kautta.



Kuva 22: identify-komennon sovitussikkuna kun painetaan h-näppäintä.



Kuva 23: identify-komennon sovitussikkuna kun painetaan j-näppäintä.



Kuva 24: identify-komennon sovituskuna kun painetaan l-näppäintä.

j-näppäimellä nähdään residuaalit, eli havaittu miinus sovitettu (kuva 23). Residuaalien tulisi vaihdella satunnaisesti nollan molemmin puolin.

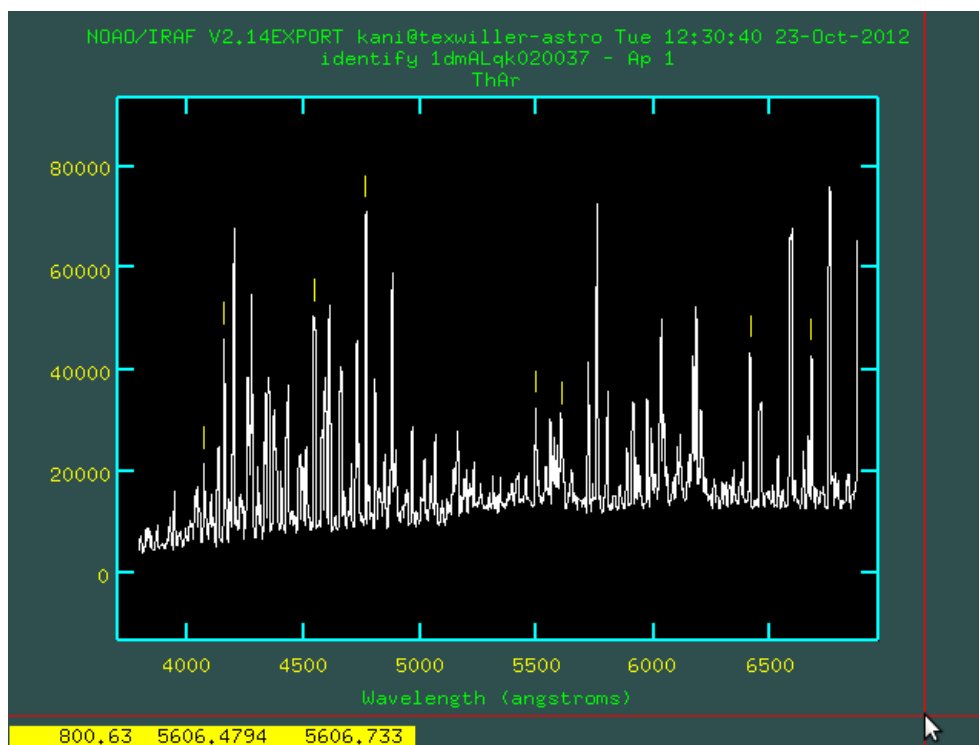
l-näppäimellä nähdään epälineaarinen osa sovituksesta, eli sovituksen “kaarevuus” (kuva 24). Tässäkin kuvassa sovitetun funktion tulisi kulkea havaittujen pisteiden kautta.

Mikäli viivat oli oikein tunnistettu, sovitus on hyvä ja painetaan q (quit), jolloin IRAF palaa kuvan 25 kaltaiseen näkymään. Suurin ero kuvaan 21 on se, että vaaka-akselilla on nyt aallonpituus eikä pikselikoordinaatti.

Sovitus perustuu tällä hetkellä neljään viivaan ja tuloksen parantamiseksi olisi hyvä tunnistaa ja merkitä lisää viivoja niin, että yhteensä niitä on noin 5-10 kpl. Tunnistamista helpottaa nyt se seikka, että vaaka-akselilla näkyy aallonpituus ja kuva ei enää ole peilikuva kuvan 19 spektrin suhteen. Merkitään siis lisää viivoja m-näppäimellä, tehdään sovitus f-näppäimellä, tarkastellaan sovitusta h, j ja l-näppäimillä ja palataan kuvan 25 näkymään q-näppäimellä. Kun riittävä määrä viivoja on merkitty, poistutaan q-näppäimellä, jolloin IRAF kysyy

Write feature data to the database (yes)?

johon vastaan yes. IRAF tallentaa sovituksen tuloksen database-hakemistoon idXXXX-



Kuva 25: identify-komennon ikkuna sen jälkeen kun sovitus on tehty.

nimisinä tiedostoina, missä XXXX on kuvan nimi. Seuraavan kerran kun identify käynnistetään samalle kuvalle, se osaa lukea aikaisemman sovituksen database-hakemistosta ja soveltaa sitä suoraan. Tätä ominaisuutta voi käyttää tallentamaan välituloksia viivojen merkitsemisen aikana.

Muutama sana “hyvästä” sovituksesta: sovituksen hyvyttä kannattaa tarkastella lähinnä j-näppäimellä (residuaalit) ja l-näppäimellä (epälineaarinen osa). Hyvässä sovituksessa epälineaarinen osa seuraa havaittua dataa tarkasti ja residuaalien rms-hajonta (keskihajonta) on pieni. Hajonnan näkee kuvan 23 plottauksen ylälaidasta, esimerkin tapauksessa lueimme sieltä RMS= 0.6428 eli keskihajonta havaintojen ja sovituksen välillä on 0.6428 Ångströmiä. Hajonta riippuu käytetystä spektrograafista ja resoluutiosta, mutta esimerkin tapaus antaa tarkkuuden, mihin tulisi pyrkiä ALFOSCin Grism #7:llä.

```

*images = "1dmALqk020036" images to be edited
*fields = "REFSPEC1"      fields to be edited
*value = "1dmALqk020037" value expression
  (*add = yes)            add rather than edit fields
(addonly = no)          add only if field does not exist
(delete = no)          delete rather than edit fields
(verify = yes)         verify each edit operation
  (show = yes)         print record of each edit operation
(update = yes)         enable updating of the image header
  (mode = "ql")

```

Kuva 26: `hedit`-komennon parametrit.

4.9 Aallonpituuskalibrointi

Tämän vaiheen tarkoituksena on soveltaa juuri määritettyä kalibrointia kohteen spektriin. Kalibrointi tapahtuu kahdessa vaiheessa: ensin kerrotaan IRAFille mitä kalibrointispektriä halutaan käyttää ja toisessa vaiheessa suoritetaan itse kalibrointi.

Vaihe 1: Meillä on siis kohteen spektri `1dmALqk020036.fits`, jonka haluamme aallonpituuskalibroida käyttäen viivaspektriin `1dmALqk020037.fits` tehtyä sovitusta. Vaiheessa 1 kirjoitamme kohteen spektrin FITS-headeriin käytettävän kalibrointispektrin nimen. Tämä tapahtuu `hedit`-komennolla. Editoidaan siis

```
ecl>epar hedit
```

ja asetetaan parametrit kuvan 26 mukaisiksi. Ajetaan

```
ecl>hedit
```

ja vastataan kaikkiin kysymyksiin painamalla `<enter>`. IRAF lisää kuvan `1dmALqk020036.fits` FITS-headeriin uuden kentän `REFSPEC1`, jonka arvoksi tulee `1dmALqk020037`. Tämä kenttä kertoo seuraavalle komennolle (`dispcor`) sen, että ko. kuvan `1dmALqk020036.fits` kalibroimiseen käytetään kuvan `1dmALqk020037.fits` sovitusta.

Vaihe 2: Kirjoitetaan

```
ecl>epar dispcor
```

ja editoidaan parametrit kuvan 27 mukaisiksi. Ajetaan

```
ecl>dispcor
```

ja tuloksena on spektri `lambdamALqk020036.fits`. Tarkastellaan tätä spektriä hie-


```

*input = "1dmALqk020036" List of input spectra
*output = "lambdamALqk020036" List of output spectra
(linearize = yes)           Linearize (interpolate) spectra?
(database = "database")    Dispersion solution database
(table = "")               Wavelength table for apertures
(w1 = INDEF)              Starting wavelength
(w2 = INDEF)              Ending wavelength
(dw = INDEF)              Wavelength interval per pixel
(nw = INDEF)              Number of output pixels
(log = no)                Logarithmic wavelength scale?
(flux = yes)              Conserve flux?
(blank = 0.)              Output value of points not in input
(samedisp = no)           Same dispersion in all apertures?
(global = no)             Apply global defaults?
(ignoreaps = no)          Ignore apertures?
(confirm = no)            Confirm dispersion coordinates?
(listonly = no)           List the dispersion coordinates only?
(verbose = yes)           Print linear dispersion assignments?
(logfile = "")            Log file
(mode = "ql")

```

Kuva 27: dispcor-komennon parametrit.

man tarkemmin. Kirjoita

```
ecl>plot lambdamALqk020036
```

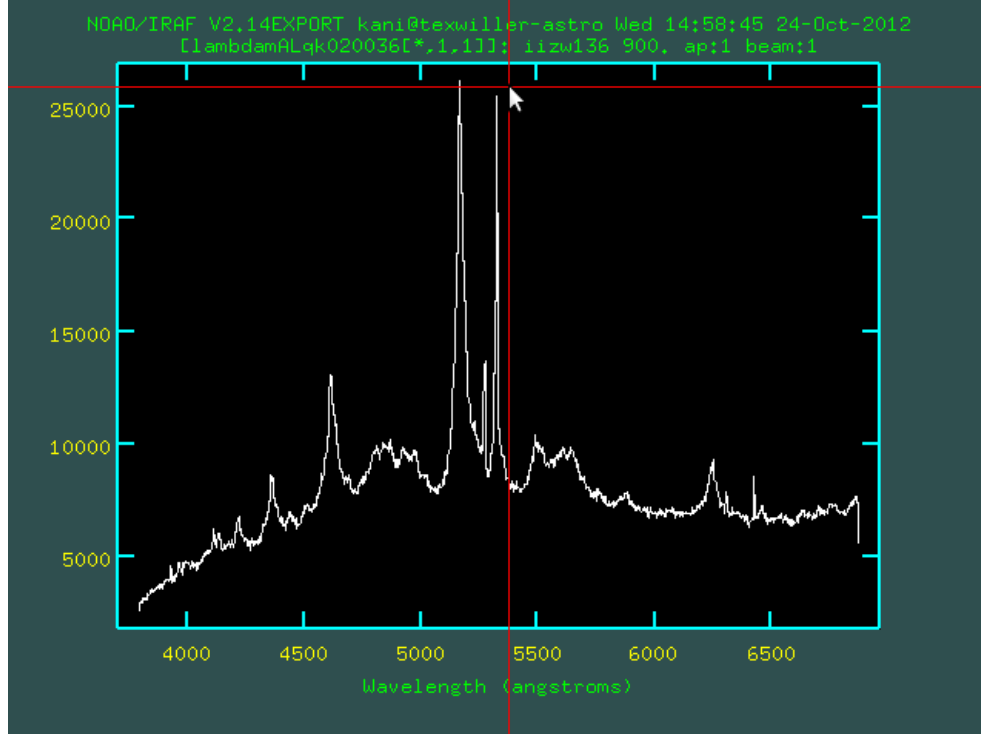
jolloin aukeaa kuvan 28 kaltainen kuva. Huomaa, että vaaka-akselilla on nyt aallonpituus. Tehdään pieni testi sen varmistamiseksi, että aallonpituuskalibrointi on tehty oikein: määritetään kohteen II ZW 136 punasiirtymä kuvasta 28 ja verrataan sitä kirjallisuuden arvoon $z = 0.0623$.

Maailmankaikkeuden laajenemisen vuoksi aallonpituudet “venyvät” matkalla kohteesta teleskooppiin, ts. kohteen valo siirtyy kohti spektrin punaista päätä. Mikäli kohde lähettää säteilyä jollakin aallonpituudella λ_0 , se havaitaan kaukana kohteesta pidempänä aallonpituutena λ . Punasiirtymä z määritellään

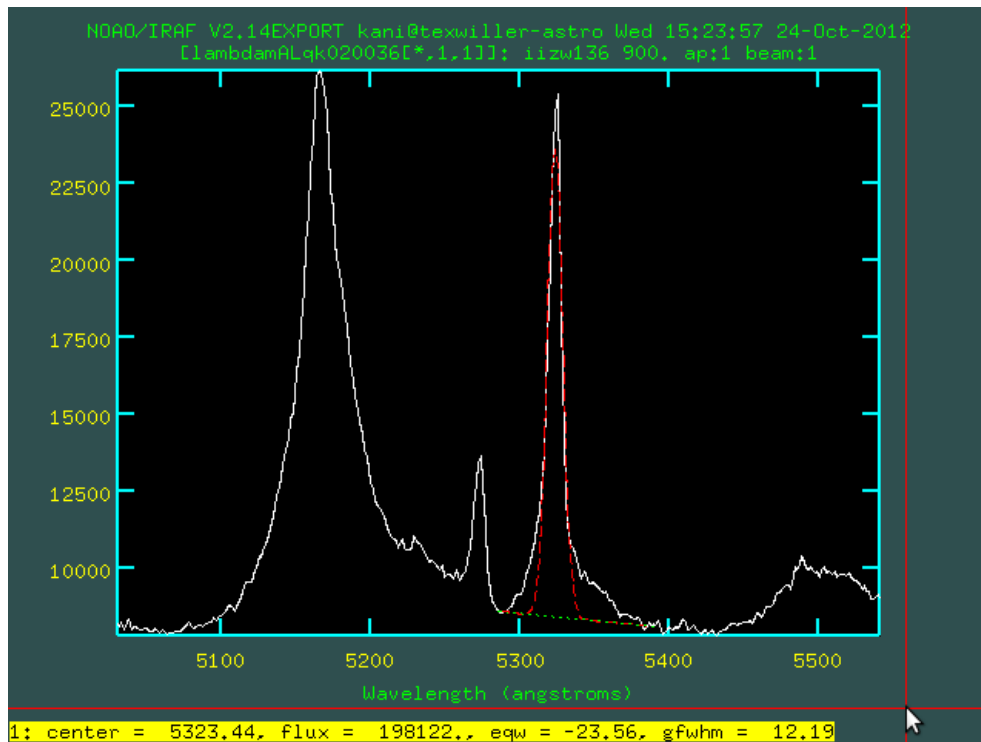
$$z = \frac{\lambda - \lambda_0}{\lambda_0} . \quad (1)$$

Tarkastellaan kuvan 28 spektrissä kapeaa emissioviivaa, jonka huippu sijaitsee lähimpänä punaista kursoria. Tämän viivan aiheuttaa kaksi kertaa ionisoitunut happi [OI-II] ja sen keskikohta sijaitsee aallonpituudella $\lambda_0 = 5007\text{\AA}$.

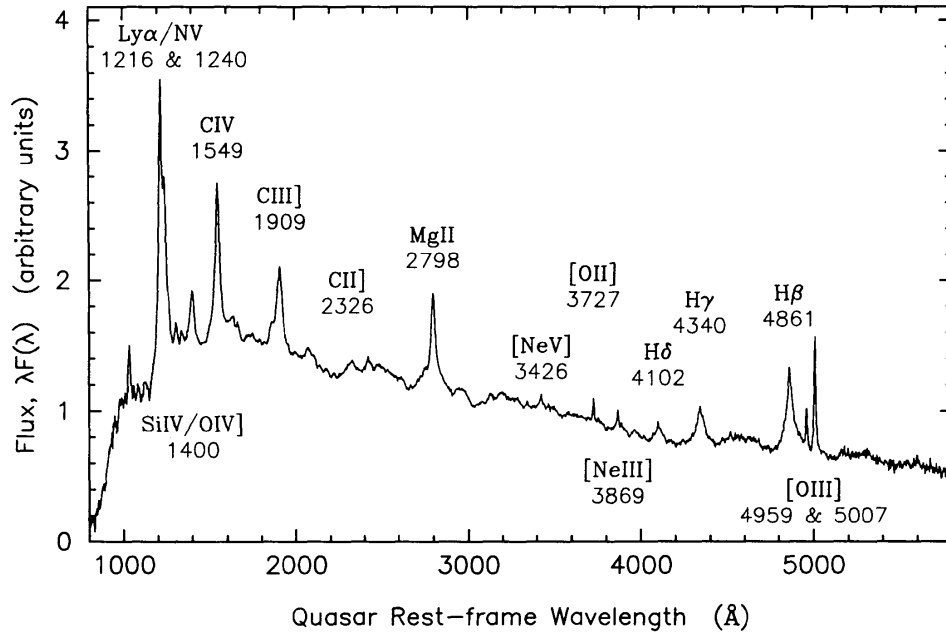
Zoomataan viivan lähelle kahdella a-näppäimen painalluksella ja mitataan viivan paikka kahdella k-näppäimen painalluksella (ks. tarkemmat ohjeet kappaleesta 4.6.1 ja kuvaa 29). Huomataan, että sovitettu viiva ei muodoltaan täysin vastaa havaittua viivaa, mutta tässä olemme lähinnä kiinnostuneita viivan paikasta, jon-



Kuva 28: Kohteen spektri aallonpituuskalibroinnin jälkeen.



Kuva 29: Emissioviivan keskikohdan määrittäminen.



Kuva 30: Kvasaareissa ja muissa aktiivisissa galaksiytimissä esiintyviä emissioviivoja (Francis et al. 1991, ApJ 373, 465).

ka sovitus näyttää määrittäneen hyvin. Luetaan kuvasta $\lambda = 5323\text{\AA}$, eli

$$z = \frac{5325 - 5007}{5007} = 0.0635, \quad (2)$$

mikä poikkeaa vain 2% kirjallisuusarvosta. Aallonpituuskalibrointi on ilmeisesti siis onnistunut melko hyvin. Tarkistuksen voi tehdä käyttämällä muitakin emissioviivoja, joita voi spektristä tunnistaa kuvan 30 avulla. Huomaa, että kuvassa 28 näkyy vain osa kuvan 30 viivoista.

4.10 Vuostandardin käsittely

Kuvan 28 spektri on nyt lähes valmis, enää puuttuu vuokalibroinnin tekeminen. Vuokalibroinnissa y-akselin yksiköt ADU muutetaan fysikaalisiksi yksiköiksi, yleensä $\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1}$ ($1 \text{ erg} = 10^{-7} \text{ J}$ ja kyllä, astrofysiikassa käytetään vielä cgs-yksiköitä). Muunnoksen mahdollistamiseksi on havaittu standarditähden HD 19445 spektri. Standarditähtien kirkkaus aallonpituuden funktiona tunnetaan hyvin tarkasti, mikä mahdollistaa spektrograafin kalibroinnin.

Ensiksi standarditähden spektri täytyy käsitellä samalla tavalla kuin kohteen spektri kunnes on saatu kuvan 28 kaltainen spektri. Tässä ei käydä vaiheita uudelleen yksityiskohtaisesti, muistutetaan vain käytettävät komennot siinä järjestyksessä kuin ne täytyy ajaa ja tärkeimmät kohdat:

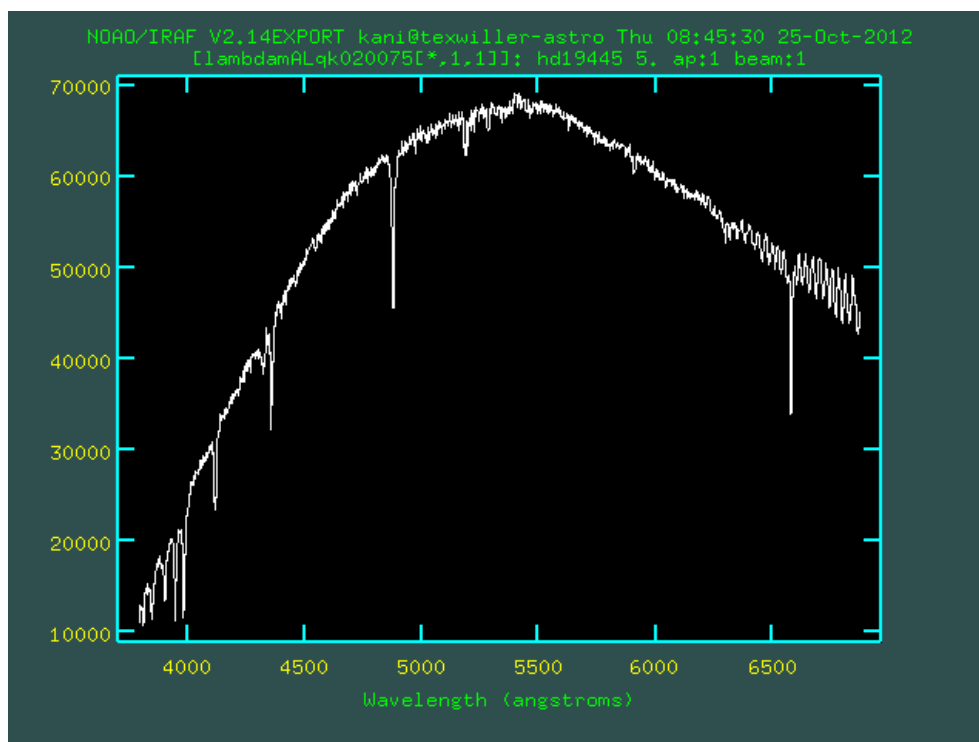
background: Tarkista parametrin `sample` asetukset. Standarditähden spektri ei välttämättä sijaitse samassa kohtaa CCD:llä kuin kohteen spektri. Taustataivaan viivat ovat yleensä heikompia kuin kohteen spektrissä, sillä valotusaika on lyhyempi ja käytetyn raon leveys suurempi.

apall: Aseta parametrit taas kuvan 11 mukaisiksi, mutta vastaamaan standarditähden spektriä. Mittaus tapahtuu samalla tavalla kuin kohteelle. Muista erityisesti asettaa parametri `references` tyhjäksi, jotta spektrin "jälki" määritetään oikein. Apertuurin kokoa (parametrit `lower` ja `upper`) täytyy myös todennäköisesti kasvattaa, sillä standarditähden spektri on usein hyvin kirkas.

hedt: Editoidaan standarditähden headeriin sama kalibrointispektri, eli asetetaan `REFSPEC1 = 1dmALqk020037`.

dispcor: Ajetaan samalla tavalla kuin kohteelle.

tuloksena on kuvan 31 kaltainen kuva `1dmALqk020075.fits`



Kuva 31: Standarditähtien spektri aallonpituuskalibroinnin jälkeen.

4.11 Spektrograafin herkkyyskäyrän määrittäminen

Tämän vaiheen tarkoituksena on määrittää spektrograafin herkkyys aallonpituuden funktiona. Määrittäminen tapahtuu komentojen `standard` ja `sensfunc` avulla.

Ennen näiden komentojen ajamista on selvitettävä löytyykö havaitun standarditähtien (tässä tapauksessa HD 19445) tietoja IRAFin kalibrointihakemistoista. Tämä on joskus hieman työläs tehtävä, sillä kalibrointihakemistoja on lukuisia ja sama tähti voi sijaita useassakin hakemistossa, joista jokaisessa on tietoja tähtestä hieman eri aallonpituusalueella. Oikean hakemiston löytämisessä voi kuitenkin käyttää apuna Linuxin komentoja, kuten alla selostetaan.

Kalibrointihakemistot sijaitsevat IRAFin asennushakemiston alla, joten niiden täsmällinen sijainti riippuu siitä, mihin kohtaan Linuxin hakemistorakenteessa IRAF on asennettu. Mikäli IRAFin “juurikansio” on `/iraf` (tämä on default arvo, joten todennäköisesti näin on), niin kalibrointihakemisto on

```
/iraf/iraf/noao/lib/onedstds
```

Tämän hakemiston alla on lisää hakemistoja, joissa tähtien kalibrointitiedot ovat `*.dat`-tiedostoissa, missä `*` on tähden nimi. Tämä nimi on usein valitettavasti kirjoitettu tavalla, jota on vaikea päätellä tähden “oikeasta” nimestä. Kalibrointitiedoston etsimisessä voi käyttää apuna Linuxin `locate`-komentoa. Esimerkin tapauksessa kirjoitetaan Linux-terminaalissa

```
$locate 19455
```

mikä käy läpi koko Linuxin hakemistorakenteen ja etsii jokaisen tiedoston, jossa esiintyy merkkisarja 19455. Tässä tapauksessa arvaamme siis, että todennäköisesti kalibrointitiedostossa esiintyy ainakin numerosarja 19455. Arvaus osui oikeaan, sillä tulos on

```
/iraf/iraf/noao/lib/onedstds/irscal/hd19445.dat  
/iraf/iraf/noao/lib/onedstds/redcal/hd19445.dat
```

Tähdestä löytyy siis kaksi kalibrointitiedostoa. Kumpi näistä on parempi? Mahdollisesti kummatkin ovat yhtä hyviä, yleensä kannattaa valita se tiedosto, joka kattaa spektrograafin aallonpituusalueen kokonaan. Kuvasta 28 nähdään, että aallonpituusalue on noin 3800-6800Å. Tutkitaan äsken löydettyjen tiedostojen aallonpituusaluetta esim. kirjoittamalla

```
$less /iraf/iraf/noao/lib/onedstds/irscal/hd19445.dat
```

ja tarkastelemalla vasemmanpuoleista kolumnia, joka kertoo aallonpituuden (<space> liikuttaa näkymää alas, b ylös ja q lopettaa). Esimerkin tapauksessa huomataan, että kalibrointitiedosto kattaa koko havaitun spektrin alueen ja runsaasti ylikin. Samoin voidaan todeta toisestakin tiedostosta. On siis makuasia kumpi valitaan, otetaan tässä tapauksessa ensimmäinen tiedosto.

Kerrotaan seuraavaksi IRAFille mistä kalibrointitiedosto löytyy. Kirjoitetaan

```
ecl>epar kpnoslit
```

ja editoidaan toiselle riville

```
caldir = onedstds$irscal/
```

missä \$onedstds on IRAFin sisäinen muuttuja, joka viittaa hakemistoon /iraf/iraf/noao/lib/. Parametrin caldir arvo on siis /iraf/iraf/noao/lib/irscal/ eli se osoittaa nyt hakemistoon, jossa kalibrointitiedosto tähdelle HD 19445 sijaitsee. Seuraavaksi kirjoitetaan

```
ecl>epar standard
```

ja editoidaan parametrin kuvan 32 mukaisiksi. Kohtaan star_name kirjoitetaan tähden nimi, eli äsken löydetyn kalibrointitiedoston nimi ilman .dat-päätettä. Kohtaan airmass kirjoitetaan havainnon ilmassa, mikä löytyy esim. kirjoittamalla

```
$less lambdamALqk020075.fits
```

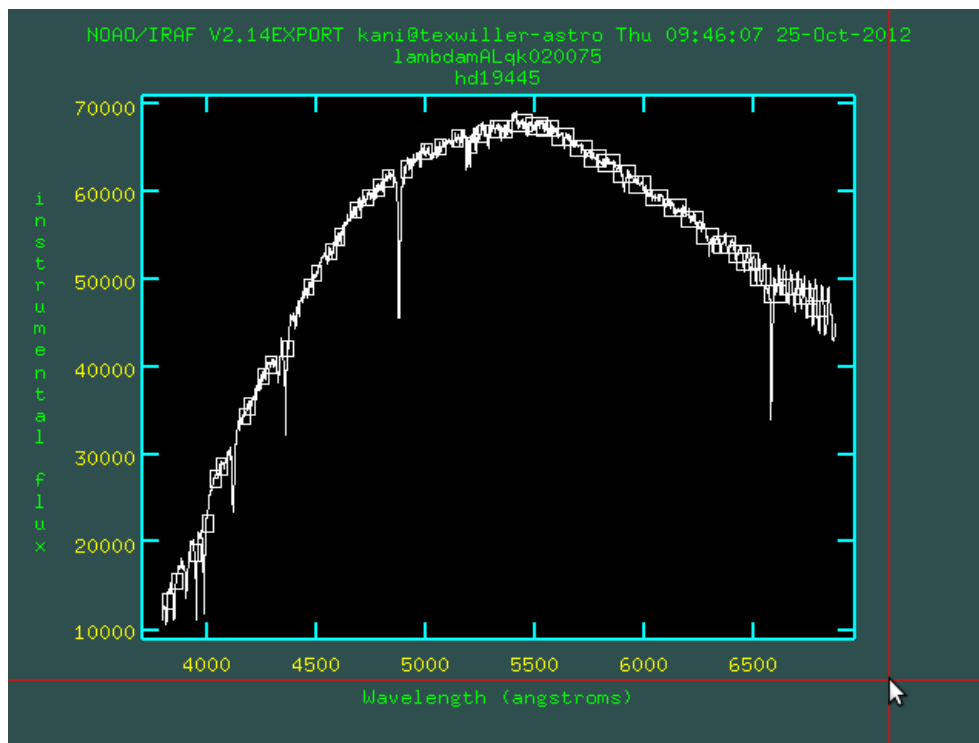
ja etsimällä FITS-headeristä kenttä AIRMASS. Parametrin exptime arvo löytyy myös samalla tavalla etsimällä kenttä EXPTIME. Kun parametrit on asetettu, ajetaan

```

    *input = "lambdamALqk020075" Input image file root name
    output = "std"           Output flux file (used by SENSFUNC)
    (samestar = yes)        Same star in all apertures?
    (beam_switch = no)      Beam switch spectra?
    (apertures = "")        Aperture selection list
    (bandwidth = INDEF)     Bandpass widths
    (bandsep = INDEF)       Bandpass separation
    (fnuzero = 3.680000000000000E-20) Absolute flux zero point
    (extinction = )_.extinction) Extinction file
    (caldir = )_.caldir)    Directory containing calibration data
    (observatory = )_.observatory) Observatory for data
    (interact = yes)        Graphic interaction to define new bandpasses
    (graphics = "stdgraph") Graphics output device
    (cursor = "")           Graphics cursor input
    *star_name = "hd19445"  Star name in calibration list
    *airmass = 1.06         Airmass
    *exptime = 5.           Exposure time (seconds)
    mag =                   Magnitude of star
    magband =               Magnitude type
    teff =                  Effective temperature or spectral type
    answer = "yes"          (no|yes|NO|YES|NO!|YES!)
    (mode = "ql")

```

Kuva 32: standard-komennon parametrin.



Kuva 33: standard-komennon avaama ikkuna.

```
ecl>standard
```

vastataan kaikkiin kysymyksiin painamalla `<enter>`, jolloin kuvan 33 mukainen ikkuna aukeaa. Kuvassa näkyy standarditähden spektri ja suorakulmaisia laatikoita, jotka ilmaisevat `standard`-komennon mittaukset tähden kirkkaudesta kapeilla aallonpituuskaistoilla (samoilla kaistoilla, jotka esiintyvät kalibroitiedostossa). Mittajan tehtävä on nyt varmistaa, että “laatikkoja” on tasaisesti yli koko spektrin (pitäisi olla, koska kalibroitiedosto kattoi koko havaitun spektrin) ja että ne sijaitsevat pystysuorassa suunnassa spektrin päällä. Mikäli jokin laatikoista on pahasti pielessä, sen voi poistaa siirtämällä kursori ko. laatikon päälle ja painamalla `d` (`delete`). Kun ollaan tyytyväisiä, painetaan `q` (`quit`). `standard` kirjoittaa nyt tiedoston `std`, johon äskeiset mittaukset tallentuvat.

Tämän jälkeen kirjoitetaan

```
ecl>epar sensfunc
```

ja asetetaan parametrin kuvan 34 mukaisiksi. Ajetaan

```
ecl>sensfunc
```

vastataan kysymyksiin painamalla `enter` ja kuvan 35 mukainen ikkuna aukeaa. Kuvan ylemmässä paneelissa näkyy `sensfunc`-komennon mittaama spektrograafin

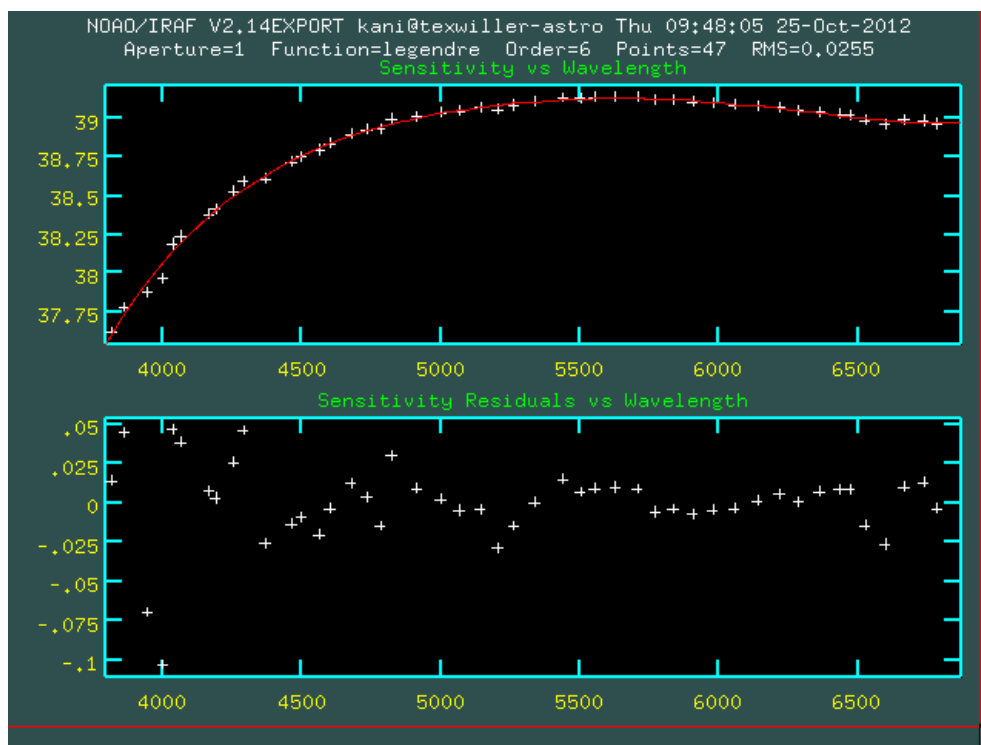
standards = "std"	Input standard star data file (from STANDARD)
sensitivity = "sens"	Output root sensitivity function imagename
(apertures = "")	Aperture selection list
(ignoreaps = yes)	Ignore apertures and make one sensitivity funct
(logfile = "logfile")	Output log for statistics information
(extinction =)_.extinction)	Extinction file
(newextinctio = "extinct.dat")	Output revised extinction file
(observatory =)_.observatory)	Observatory of data
(*function = "legendre")	Fitting function
*order = 6)	Order of fit
(interactive = yes)	Determine sensitivity function interactively?
(graphs = "sr")	Graphs per frame
(marks = "plus cross box")	Data mark types (marks deleted added)
(colors = "2 1 3 4")	Colors (lines marks deleted added)
(cursor = "")	Graphics cursor input
(device = "stdgraph")	Graphics output device
answer = "yes"	(no yes NO YES)
(mode = "ql")	

Kuva 34: sensfunc-komennon parametrit.

herkkyys aallonpituuden funktiona (plus-merkit) yksiköissä, joilla ei tässä yhteydessä ole merkitystä ja mittauksiin sovitettu käyrä (punainen viiva).

Tavoitteena on nyt löytää alin mahdollinen polynomin kertaluku `order`, joka vielä tuottaa hyvän sovituksen mittauksiin. Kertalukua voi muuttaa kirjoittamalla plottausikkunassa esim. `:order 3 <enter>` ja sen jälkeen `f` (fit) ja `r` (replot). Tässäkin voi esiintyä huonoja mittauspisteitä, eli esim. yksi plus-merkeistä poikkeava huomattavasti sovituksesta. Tällainen piste voidaan poistaa siirtämällä kursori ko. pisteen päälle ja painamalla `d` (delete).

Esimerkin tapauksessa `order = 6` riitti tuottamaan hyvän sovituksen (ks. kuva 35). Kun ollaan tyytyväisiä, painetaan `q` (quit) ja IRAF kirjoittaa tiedoston `sens.fits`, johon spektrograafin herkkyyskäyrä on tallennettu.



Kuva 35: sensfunc-komennon avaama ikkuna.

4.12 Vuokalibrointi

Viimeisenä vaiheena tehdään vuokalibrointi, eli spektrin voimakkuuden yksikkö muutetaan ADUista fyysikaalisiin yksiköihin $\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1}$. Kirjoitetaan

```
ecl>epar calibrate
```

ja editoidaan parametrin kuvan 36 mukaisiksi. Ilmamassa (airmass) ja valotusaika (exptime) saadaan selville samalla tavalla kuin edellä. Kun parametrit on asetettu, ajetaan

```
ecl>calibrate
```

vastataan kaikkiin kysymyksiin painamalla <enter> ja tuloksena on lopullinen spektri finalmALqk020036.fits. Spektriä voi tarkastella komennolla

```
ecl>splot finalmALqk020036
```

jolloin aukeaa kuvan 37 kaltainen kuva. Mikäli vuokalibrointi on onnistunut, pystyakselilla on nyt luokkaa 10^{-15} olevia lukuja.

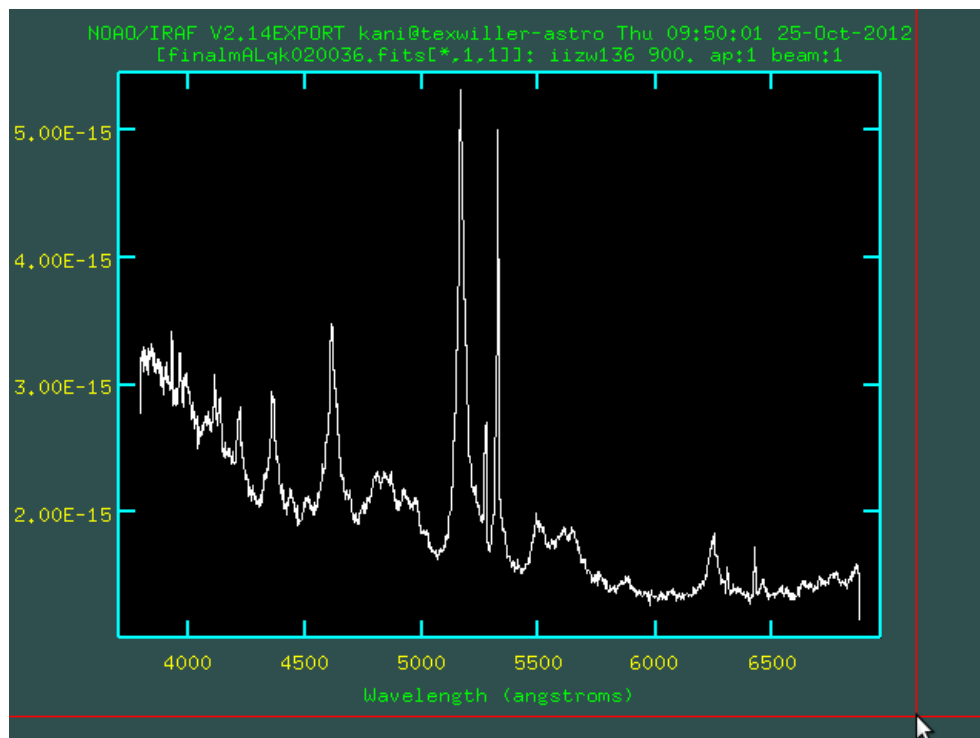
Lasketaan vielä huvin vuoksi kuinka suurella teholla kohteemme valaisee yhden neliömetrin kokoista alaa maapallolla. Kirjoitetaan

```

 *input = "lambdamALqk020036" Input spectra to calibrate
 *output = "finalmALqk020036 " Output calibrated spectra
(extinct = yes) Apply extinction correction?
(flux = yes) Apply flux calibration?
(extinction = )_.extinction) Extinction file
(observatory = )_.observatory) Observatory of observation
(ignoreaps = yes) Ignore aperture numbers in flux calibration?
(sensitivity = "sens") Image root name for sensitivity spectra
(fnu = no) Create spectra having units of FNU?
*airmass = 1.06 Airmass
*exptime = 900. Exposure time (seconds)
(mode = "ql")

```

Kuva 36: calibrate-komennon parametrit.



Kuva 37: Lopullinen vuokalibroitu spektri.

```
ecl>imstat finalmALqk020036[* ,1,1]
```

jolloin IRAF tulostaa

#	IMAGE	NPIX	MEAN	STDDEV	MIN	MAX
	finalmALqk020036[* ,1,1]	2052	1.963E-15	6.221E-16	1.138E-15	5.321E-15

Spektrin keskikirkkaus on siis

$$1.963 \times 10^{-15} \text{ erg s}^{-1} \text{ cm}^{-2} \text{ \AA}^{-1}.$$

Kun muistetaan, että $1 \text{ erg} = 10^{-7} \text{ J}$ saadaan keskikirkkaudeksi

$$1.963 \times 10^{-22} \text{ J s}^{-1} \text{ cm}^{-2} \text{ \AA}^{-1}.$$

Kun lisäksi huomataan, että spektri kattaa aallonpituusalueen $\approx 3800\text{\AA} - 6800\text{\AA}$ eli noin 3000\AA ja että $1 \text{ m}^2 = 10^4 \text{ cm}^2$, saadaan yhdelle neliömetrille koko spektrin alueelta energiaa

$$\begin{aligned} &1.963 \times 10^{-22} \text{ J s}^{-1} \text{ cm}^{-2} \text{ \AA}^{-1} \times 3000 \text{ \AA} \times 10000 \text{ cm}^2 = \\ &5.88 \times 10^{-15} \text{ J s}^{-1} = \\ &5.88 \times 10^{-15} \text{ W}. \end{aligned}$$

Ilmeisesti siis ei kannata aurinkopaneeleja ainakaan tähän kohteeseen suunnata! Tässä vaiheessa on hyvä huomauttaa, että laskettu teho vastaa kohteen aiheuttamaa "valaisua" ilmakehän ulkopuolella. Ilmakehä leikkaa tästä valosta noin 10%, joten maan pinnalla neliömetrille lankeava energiamäärä on hieman pienempi. On myös hyvä huomata, että laskimme kohteen säteilemän energian vain kuvan 37 spektrin kattamalla aallonpituusalueella. Kohde säteilee myös muilla aallonpituuksilla, joten siitä maahan lankeava kokonaisenergia on jonkin verran suurempi kuin äsken laskemamme.