

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
СПЕЦИАЛЬНАЯ АСТРОФИЗИЧЕСКАЯ ОБСЕРВАТОРИЯ РОССИЙСКОЙ АКАДЕМИИ НАУК

Методическое пособие по обработке эшелле-спектров с использованием
IRAF
(Image Reduction and Analysis Facility)

Граужанина А.О.

Направление подготовки: 03.06.01 ФИЗИКА И АСТРОНОМИЯ
Профиль подготовки: 01.03.02 АСТРОФИЗИКА И ЗВЕЗДНАЯ АСТРОНОМИЯ

п. Нижний Архыз, 2020

Данное методическое пособие является дополнительным материалом к практическому занятию «Введение в систему редукиции и анализа данных астрофизических наблюдений IRAF» курса «Оптические наблюдательные методы в астрофизике» проводимого на базе САО РАН. Материал изложен доступным языком и может быть использован как аспирантами, так и студентами, желающими овладеть навыками обработки эшелле-спектрокопии. Подробно рассмотрены все необходимые этапы редукиции. По итогу выполнения всех пунктов данного методического пособия из двумерного ПЗС-изображения спектра будет получен одномерный эшелле-спектр откалиброванный по длине волны.

Содержание

1	Описание	4
2	Начало работы с IRAF	6
2.1	Установка	6
2.2	Запуск	7
2.3	Отображение данных	9
2.4	Кросс-разрез изображения	11
3	Подготовка рабочего материала	13
3.1	Разбиение кадров на части	13
3.2	Обрезка оверскана	13
3.3	Поворот кадра	14
4	Основные корректировки	14
4.1	Учет bias	14
4.2	Темновой ток	16
4.3	Маскировка плохих пикселей	16
4.4	Исправление следов космических частиц	18
5	Первая экстракция спектра	19
6	Рассеянный свет	25
7	Плоское поле	28
8	Экстракция одномерных спектров	32
9	Калибровка длин волн	33
9.1	Идентификация спектральных линий спектра сравнения	33
9.2	Идентификация длин волн линий и масштабирование длин волн	34
9.3	Калибровка спектров по длине волны	36
9.4	Нормализация континуума	37

1 Описание

Цель данного методического пособия - получение из двумерного ПЗС-изображения (рис. 1) одномерного спектра, откалиброванного по длине волны (рис. 2). Процедура включает калибровку ПЗС-данных, извлечение спектральных порядков, калибровку по длине волны и нормализацию континуума.

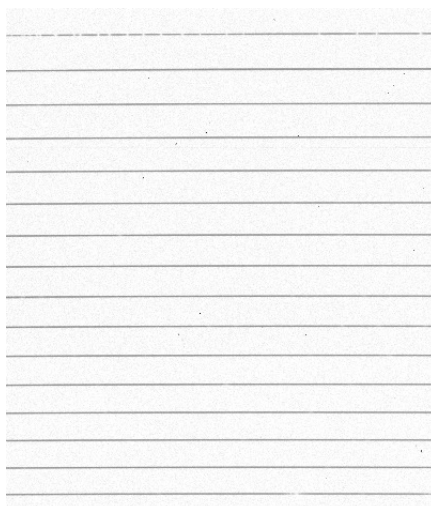


Рис. 1: Двумерное изображение объекта. Горизонтальная и вертикальная оси соответствуют направлениям вдоль дисперсии и вдоль щели соответственно.

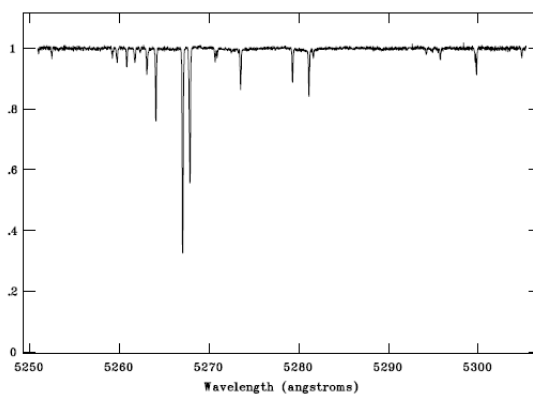


Рис. 2: Участок одномерного спектра откалиброванного по длине волны.

Перед началом обработки эшелле-спектров (как и любых других) необходимо убедиться в наличии калибровочных кадров таких как:

- **bias** (байес) – изображение подложки, полученное при нулевой экспозиции. Являются составляющей всех кадров без исключения.
- **dark** (дарк) – кадр с изображением темнового тока. Получается при съемке с закрытым затвором. При охлаждении чипа до низких температур вклад темнового тока становится незначительным и дарк не снимают.

- **flat** (флэт) – изображение плоского поля. Необходимо для учета неоднородностей чувствительности пикселей матрицы.
- Спектр сравнения (н-р, **ThAr**). Он необходим для привязки исследуемого спектра по длинам волн.

Без калибровочных кадров спектры самого объекта будут бесполезным материалом.

Перед началом работы необходимо выполнить резервное копирование всех исходных файлов. Также необходимо проверить в какой-либо программе (например, **ds9**) все кадры на предмет сильных отклонений. Если кадры плохие, их необходимо удалить из обработки.

2 Начало работы с IRAF

2.1 Установка

Необходимо выполнить команду **mkiraf**, а затем ответить на несколько вопросов об инициализации файлов с параметрами и настройке терминала:

```
> mkiraf
```

```
Initialize uparm? (y|n): y
```

```
- initializing uparm
```

```
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
```

```
Enter terminal type: xgterm
```

```
A new LOGIN.CL file has been created in the current directory.
```

```
You may wish to review and edit this file to change the defaults.
```

После этого в каталоге, где мы работаем, появится файл **login.cl**. Это установочный файл для IRAF, который можно изменить вручную. В файле **login.cl** можно найти строки, которые выглядят так:

```
set home = "/home/username/IRAF _ directory/"
```

```
...
```

```
set uparm = "home $ uparm/"
```

Первая строка - это каталог, в котором находится **login.cl**, а другая определяет местоположение каталога **uparm**, в котором хранятся текущие параметры команд. При работе с данными с разных инструментов, рекомендуется создать отдельные стартовые каталоги (отдельные **login.cl** и **uparm**) для каждого инструмента. По умолчанию в IRAF установлен формат файлов ***.imh** и ***.pix**. Удобно установить формат ***.fits**. Для этого необходимо в файле **login.cl** изменить строку:

```
set imtype = "imh"
```

```
на
```

```
set imtype = "fits"
```

Также можно указать пакеты и команды, которые будут автоматически подгружаться при запуске IRAF. Для этого необходимо указать их названия в конце файла **login.cl**:

```
imred
```

```
ccdred
echelle
...
task [name] = [path]/[filename].cl
```

В последней строке показано, как подключать дополнительные задачи, которые хранятся вне каталога IRAF (обычно файлы с расширением **.cl**).

2.2 Запуск

Перед тем как начать работу непосредственно с самими спектрами, необходимо отсортировать все файлы по датам и по времени получения. Лучше разбить кадры одной ночи на столько папок, сколько было снято полных серий калибровок. И не забываем про резервные копии!

Если файлы имеют неподходящее расширение (IRAF работает с файлами формата **fits**), то поменять расширение мы можем с помощью команды:

```
> rename *.fts fits field=extn
```

***.fts** – говорит о том, что операция будет применена ко всем fits-файлам в данной папке (своеобразная маска для файлов, с которыми программе надо работать).

Чтобы создать список всех fits-файлов в данной директории, выполняем команду:

```
> ls *.fits > all_files.dat
```

На выходе получим файл **all_files.dat** со списком всех fits-изображений, которые содержатся в данной директории.

Запускаем терминал Linux и запускаем **xgterm**. Запуск **xgterm** необходим для работы графического терминала IRAF:

```
> xgterm&
```

Тег **&** позволяет закрывать терминал, не прерывая работы **xgterm**.

После запуска **xgterm** мы окажемся в домашней папке, а запуск IRAF необходимо производить из каталога, где храниться файл **login.cl**. Для просмотра содержимого папки, в который мы находимся, выполняем:

```
>ls
```

Переход вниз по древу между директориями осуществляется командой:

```
>cd ./folder_name
```

Команда `cd ../` позволяет подняться вверх по древу на одну директорию.

После перехода в необходимую папку, мы запускаем сам IRAF (рис. 3):

```
>ecl
```

```
NOAO/IRAF PC-IRAF Revision 2.16.1 EXPORT Mon Oct 14 21:40:13 MST 2013
This is the EXPORT version of IRAF V2.16 supporting PC systems.

Welcome to IRAF. To list the available commands, type ? or ??. To get
detailed information about a command, type 'help <command>'. To run a
command or load a package, type its name. Type 'bye' to exit a
package, or 'logout' to get out of the CL. Type 'news' to find out
what is new in the version of the system you are using.

Visit http://iraf.net if you have questions or to report problems.

cannot access host 'iraf.noao.edu:80'
The following commands or packages are currently defined:

      ctio.      fitsutil.  lists.      obsolete.   softtools.  tables.
      dataio.   guiapps.  mscred.    plot.       stecf.      utilities.
      dbms.     images.  mxtools.   proto.      stsdas.     vo.
      esowfi.   language. noao.      rvsao.     system.     xdimsum.

ecl> █
```

Рис. 3: Приветственное окно и список основных пакетов IRAF.

`ecl>` в начале строки теперь говорит о том, что мы используем IRAF. После запуска IRAF перемещаемся в рабочую папку используя команду `cd`. Для завершения работы IRAF используем команду `log out`.

Все процедуры выполняются в интерактивном режиме с использованием команд IRAF. В зависимости от роли, все команды сгруппированы в «пакеты». Чтобы выполнить ту или иную команду необходимо сначала открыть пакет, включающий эту задачу, набрав имя пакета. Например, задача **apall** находится в пакете **echelle**, который находится в пакете **imred**. Поэтому, сначала вводим имена пакетов **imred** и **echelle**:

Чтобы вернуться к предыдущему пакету, используем команду `bye`. Имя пакета, которое включает команду, которую вы хотите применить, можно узнать в справке. Например, в случае задачи **apall** (рис. 5):

```
ecl> help apall
```



```

ecl> imred
      argus.      crutil.      echelle.      iids.      kpnocoude.  specred.
      bias.       ctioslit.   generic.     irred.     kpnoslit.   vtel.
      ccdred.     dtoi.      hydra.      irs.      quadred.

imred> echelle
      apall      aprecenter  demos      refspectra  sflip
      apdefault@ apresize    deredden   sapertures  slist
      apedit     apscatter  dispcor   sarith      specplot
      apfind     apsum     doecslit  scombine    specshift
      apfit      atrace    dofoe     scopy       splot
      apflatten  bplot     dopcor    sensfunc    standard
      apmask     calibrate  ecidentify setairmass
      apnormalize continuum  ecreidentify setjd

echelle> █

```

Рис. 4: Необходимые пакеты для вызова команды **apall**.

```

APALL (Sep96)      noao.twodspec.apextract      APALL (Sep96)

NAME
  apall -- Extract one dimensional sums across the apertures

USAGE
  apall input

PARAMETERS

  input
  List of input images.

  output = ""
  List of output root names for extracted spectra. If the null
  string is given or the end of the output list is reached before
  the end of the input list then the input image name is used as
  the output root name. This will not conflict with the input
  image since an aperture number extension is added for onedspec
  [q=quit,d=downhalf,f|sp=downfull,j|cr=downline,N=next.]

```

Рис. 5: Результат вызова `help` для команды `apall` (имя пакета указано в заголовке).

При выполнении задач IRAF, наряду с входными и выходными файлами необходимо настраивать некоторые параметры самих задач. Список параметров вызывается командой **eparam** (**epar**), после которой идет имя нужной команды. Для того, чтобы сохранить измененные параметры, при выходе одновременно нажимаем **Ctrl d**, а выход без сохранения изменений - **Ctrl c**. Примеры будут показаны далее.

2.3 Отображение данных

ПЗС-изображение отображается в **ds9** (SAOIMAGE). Программа **ds9** запускается через терминал с указанием имени файла:

```
> ds9 filename.file
```

Изображение из среды IRAF отображается с помощью функции **display**. Ниже приведен пример для файла с именем **X.fits**:

```
ecl> display X.fits 1 z1=1420. z2=1967.862
```

Диапазон отображения настраивается параметрами задачи **display**:

ecl> epar display

появляется следующий список параметров:

I R A F

Image Reduction and Analysis Facility

PACKAGE = tv

TASK = display

image = X.fits image to be displayed

frame = 1 frame to be written into

(bpmask = BPM) bad pixel mask

(bpdysl= none) bad pixel display (none|overlay|interpolate)

(bpcolor= red) bad pixel colors

(overlay=) overlay mask

(ocolors= green) overlay colors

(erase = yes) erase frame

(border= no) erase unfilled area of window

(select= yes) display frame being loaded

(repeat = no) repeat previous display parameters

(fill = yes) scale image to fit display window

(zscale = yes) display range of greylevels near median

(contras= 0.25) contrast adjustment for zscale algorithm

(zrange = yes) display full image intensity range

(zmask =) sample mask

(nsample= 1000) maximum number of sample pixels to use

(xcenter= 0.5) display window horizontal center

(ycenter= 0.5) display window vertical center

(xsize = 1.) display window horizontal size

(ysize = 1.) display window vertical size

(xmag = 1.) display window horizontal magnification

(ymag = 1.) display window vertical magnification

(order = 0) spatial interpolator order (0=replicate, 1=linea

(z1 =) minimum greylevel to be displayed

(z2 =) maximum greylevel to be displayed

(ztrans = linear) greylevel transformation (linear|log|none|user)

(lutfile =) file containing user defined look up table

(mode = ql)

Диапазоны отображения регулируются значениями параметров **xmag**, **ymag** и т.д. Чтобы завершить настройку параметра, нажмите **Ctrl d** или введите символ двоеточия **:**, а затем **q**. Если ввести двоеточие **:** и **go**, то задача (в данном случае **display**) начнет свою работу.

2.4 Кросс-разрез изображения

Кросс-разрез изображения полезен для оценки спектральных данных. Он получается с помощью задачи **implot**. Здесь приведены данные для плоского поля в качестве примера:

```
ecl>implot flat.fits
```

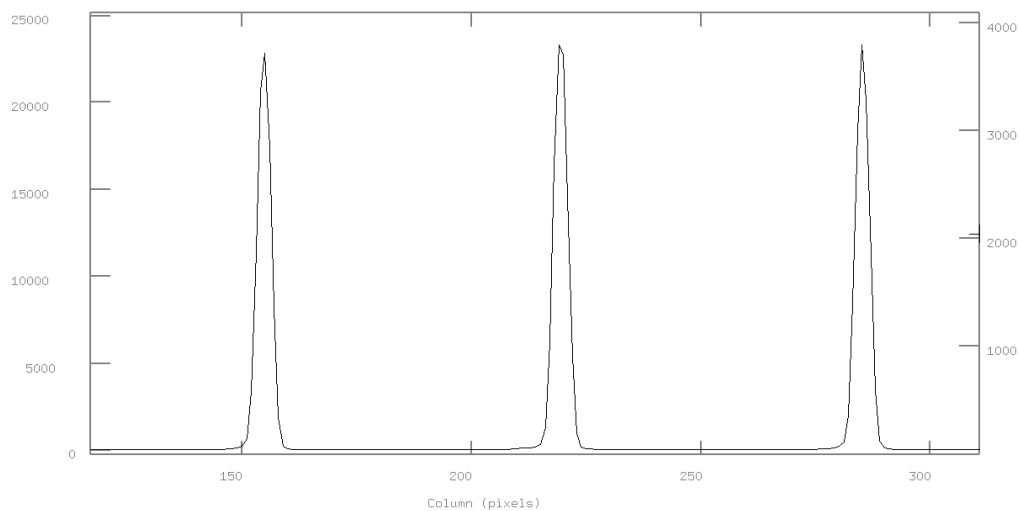


Рис. 6: Кросс-разрез изображения плоского поля. Здесь показан случай для 2046 линии (т.е. центр ПЗС-изображения) вдоль направления дисперсии.

Появляется новое окно, в котором показан кросс-разрез изображения вдоль щели (рис. 6). Диапазон отображения изменяется путем ввода двоеточия **:** и чисел пикселей, таких как **x 100 300**, в результате чего на диаграмме будут показаны данные между заданными значениями пикселей. Для получения более подробной информации см. справку по задаче **implot**. Если один раз нажать **c** в этом окне, то будет показан кросс-разрез вдоль дисперсии (направление перпендикулярное направлению щели) (**c=column**). С нажатием клавиши **l** снова будет показан кросс-разрез вдоль направления щели (**l=line**). Для зуминга используется нажатие клавиши **e** на краях необходимой области. Для возврата к показу полного изображения, нажимаем **a** дважды. Если нужно построить кросс-разрез нескольких изображений, то их названия вводятся в **implot** через запятую без пробелов. Переключение

между ними будет осуществляться кнопками **m** и **n**. Выход из процедуры **implot** осуществляется нажатием клавиши **q**.

3 Подготовка рабочего материала

3.1 Разбиение кадров на части

Эта процедура необходима, если вы, например, работаете с данными со спектрографа HARPS. У этого спектрографа в каждом файле содержится сразу два изображения, поэтому в начале работы нам необходимо разделить эти снимки. Для разбиения кадров на части в IRAF применяем команду:

```
ecl>imcopy name.fits[1,inherit] a_name.fits  
ecl>imcopy name.fits[2,inherit] b_name.fits
```

цифра **1** означает, что копируется только первая часть изображения; **inherit** – используются обе fits-шапки; **a_name.fits** – новое имя файла с первой частью изображения будет начинаться с **a** (голубая область спектра), а со второй частью изображения с **b** (красная область). Таким образом, после выполнения данных команд мы получим два набора файлов, которые соответствуют синей и красной областям.

3.2 Обрезка оверскана

ВАЖНО! Необходимо чтобы все изображения спектра и всех калибровочных кадров были ориентированы покраснением вверх и вправо (одинаково). Обрезку оверскана мы можем совершить той же командой **imcopy** только с другими параметрами:

```
ecl> imcopy name[1:2048,1:2048] new_name
```

или

```
ecl>imcopy *.fits[1:2048,1:2048] *.fits
```

копируется часть изображения с размерами **[1:2048,1:2048]** пикселей и создается новое изображение уже без оверскана. Для определения области оверскана удобно пользоваться программой **ds9**. Для ее запуска в IRAF необходимо ввести команду:

```
ecl>!ds9
```

(запуск команд Linux при работе в IRAF осуществляется с помощью написания перед командой знака "!").

3.3 Поворот кадра

Если необходимо повернуть изображение, то нужно использовать команду **rotate**:

```
ecl>rotate name new_name 180
```

число в конце команды говорит об угле поворота изображения в градусах. Указание положительного угла осуществит поворот изображения против часовой стрелки по оси **X**. Если необходимо отразить изображение по какой-либо оси (**X** или **Y**), то используем команду:

```
ecl>imcopy name[*,-*] new_name
```

в данном случае отражение по оси **Y**. При совпадении названия файлов **name** и **new_name** изображения перезаписываются, а если они отличаются, то создается новый файл.

4 Основные корректировки

В этой главе кратко изложены первые шаги редукции данных, которые следует произвести при работе с любыми данными, не только спектроскопическими. Некоторые из них должны применяться всегда, некоторые - в зависимости от инструмента.

4.1 Учет **bias**

Учет **bias** делается с использованием кадров с нулевой экспозицией. Чтобы повысить качество данных, необходимо объединить несколько кадров **bias**, приняв медианное значение для каждого пикселя. Для этой цели используем задачу **imcombine**. При выполнении **eparam imcombine** появится следующее окно параметров:

I R A F

Image Reduction and Analysis Facility

PACKAGE = immatch

TASK = imcombine

input = List of images to combine

output = List of output images
 (headers=) List of header files (optional)
 (bpmasks=) List of bad pixel masks (optional)
 (rejmask=) List of rejection masks (optional)
 (nrejmas=) List of number rejected masks (optional)
 (expmask=) List of exposure masks (optional)
 (sigmas =) List of sigma images (optional)
 (logfile STDOUT) Log file
 (combine= median) Type of combine operation
 (reject = none) Type of rejection
 (project= no) Project highest dimension of input images?
 (outtype= real) Output image pixel datatype
 (outlimi=) Output limits (x1 x2 y1 y2 ...)
 (offsets= none) Input image offsets
 (masktyp= none) Mask type
 (maskval= 0.) Mask value
 (blank = 0.) Value if there are no pixels
 (scale = none) Image scaling
 (zero = none) Image zero point offset
 (weight = none) Image weights
 (statsec=) Image section for computing statistics
 (expname=) Image header exposure time keyword
 (lthresh= INDEF) Lower threshold
 (hthresh= INDEF) Upper threshold
 (nlow = 1) minmax: Number of low pixels to reject
 (nhigh = 1) minmax: Number of high pixels to reject
 (nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
 (mclip = yes) Use median in sigma clipping algorithms?
 (lsigma = 3.) Lower sigma clipping factor
 (hsigma = 3.) Upper sigma clipping factor
 (rdnoise= 0.) ccdclip: CCD readout noise (electrons)
 (gain = 1.) ccdclip: CCD gain (electrons/DN)
 (snoise = 0.) ccdclip: Sensitivity noise (fraction)
 (sigscal= 0.1) Tolerance for sigma clipping scaling correction
 (pclip = -0.5) pclip: Percentile clipping parameter
 (grow = 0.) Radius (pixels) for neighbor rejection
 (mode = ql)

Имена входных файлов указываются в строке параметра **input** (**bias1,bias2,bias3...**) через запятую (**.fits** - можно опустить). В стро-

ке параметра **output** указывается имя выходного файла (н-р, **sbias**). Как альтернатива, имена входных файлов могут быть представлены в виде списка (файла, каждая строка которого содержит отдельное имя файла). Например, входной файл **bias_all**, содержащий названия файлов **bias**, будет указан как **@bias_all**. Параметр **combine** должен быть указан как **median**. После выполнения команды будет создан файл **sbias.fits**. Далее полученный файл **sbias** должен быть вычтен из всех остальных файлов калибровки и из файлов самого объекта.

Для вычитания **bias** используется задача **imarith**:

```
ecl> imarith name - sbias.fits new_name
```

или для списка файлов

```
ecl> imarith @all_files - sbias.fits @all_files2
```

, где файл **all_files2** содержит новые имена файлов после вычитания **sbias**.

4.2 Темновой ток

Даже когда на детектор не попадает свет, на матрице могут появляться некоторые отсчеты. Это называется темновым током и является результатом теплового движения электронов в чипе и электронике. Темновой ток оценивается по кадрам без открытия затвора (**dark**). Его уровень зависит от времени, поэтому лучше использовать кадры **dark** с тем же временем экспозиции что и научные кадры. У **dark** нужно убрать оверскан и вычесть **bias**.

Подготовка **dark** может быть выполнена с помощью задачи **darkcombine**, которая использует **ccdproc**. Больше информации можно найти в **help**. На хороших приемниках темновой ток незначителен, поэтому этот шаг может быть пропущен.

4.3 Маскировка плохих пикселей

На чипах приемников часто встречаются "плохие" пиксели или колонки. Они могут плохо повлиять на некоторые процессы редукции. Если плохих пикселей и столбцов не так много, то можно подготовить простой текстовый файл. Каждая строка этого файла должна представлять **xbegin xend ybegin yend** для прямоугольной области и **x y** для изолированных пикселей. Например:

1671 1671 1 4095 //в случае плохого столбца

1780 560 //в случае плохого пикселя

так будет выглядеть содержимое нашего файла с плохими пикселями, назовем его например **badpix**. Определение координат плохих пикселей осуществляем в программе **ds9**. Маска (файл **badpix**) может быть получена путем выделения на кадрах **bias** пикселей с ненормальными отсчетами. Для исправления плохих пикселей используем команду **fixpix** (рис. 7):

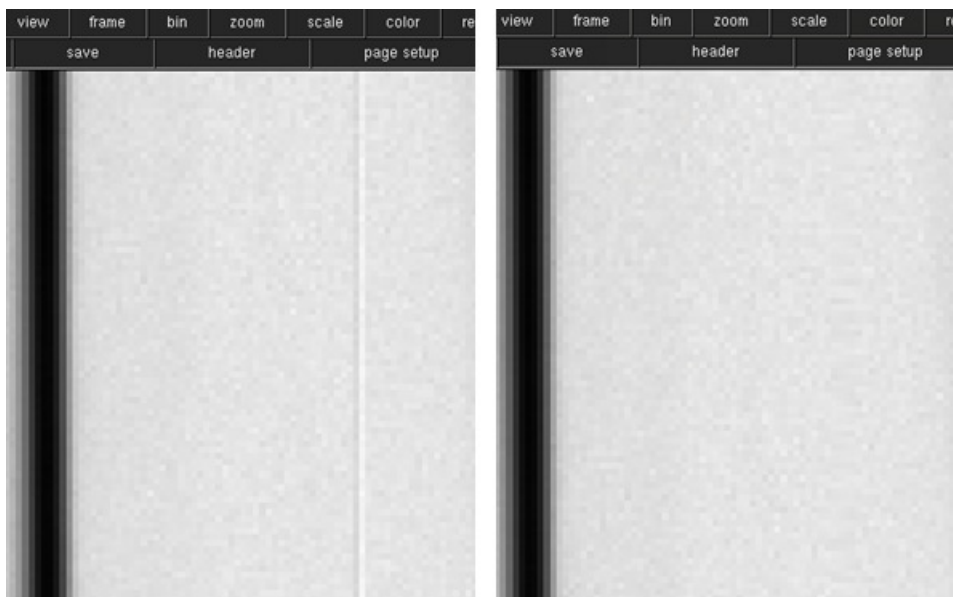


Рис. 7: Часть изображения до и после маскировки плохого столбца.

I R A F

Image Reduction and Analysis Facility

PACKAGE = proto

TASK = fixpix

images = @all_files List of images to be fixed

masks = badpix.pl List of bad pixel masks

(linterp= INDEF) Mask values for line interpolation

(cinterp= INDEF) Mask values for column interpolation

(verbose= no) Verbose output?

(pixels = no) List pixels?

(mode = ql)

или

```
ecl> fixpix @all_files badpix
```

или

```
ecl>fixpix *.fits badpix
```

4.4 Исправление следов космических частиц

На пзс-матрицу постоянно попадают высокоэнергетичные космические частицы, создавая очень большие отсчеты в пикселях, на которые они попали. Следы космических частиц легко исправляются когда изображения (**bias**, **flat**, множественные экспозиции одного и того же объекта) объединяются, например, процедурой **imcombine**. Для единичных же изображений используется задача **lacos_spec/lacos_im** в пакете **stsdas**. Здесь скачиваем скрипт: <http://www.astro.yale.edu/dokkum/lacosmic/download.html> Этот скрипт копируем в рабочую папку и сохраняем как **locos_im.cl**. Запускаем IRAF, заходим в пакет **stsdas**. Далее:

```
stsdas>task lacos_im = path/lacos_im.cl
```

где **path/** - путь к папке, в которой лежит скрипт.

Редактируем параметры:

```
stsdas>epar lacos_im
```

жмём **Ctrl d**, для сохранения изменений. Запускаем скрипт:

```
stsdas>lacos_im input output outmask
```

где **input** – имя входного файла, **output** – имя файла на выходе, а **outmask** – файл-маска с положением следов космических частиц. По ходу работы данного скрипта отвечаем на некоторые вопросы, после чего наше изображение будет очищено от космических частиц полностью.

5 Первая экстракция спектра

Для дальнейшей калибровки спектроскопических данных (например, деление на плоское поле, учет рассеянного света) удобно сначала экстрагировать все спектральные порядки из одного изображения, используя команду **apall**. Это даст эталонные данные для дальнейшей процедуры калибровки. Перед запуском определяем направление дисперсии (горизонтально или вертикально). В параметрах пакета **echelle** (**epar echelle**) устанавливаем значение **1** для параметра **dispaxi** в случае горизонтальной и **2** в случае вертикальной ориентации порядков.

Как уже говорилось ранее, для запуска пакета **apall** необходимо сначала перейти в пакет **imred** и **echelle**. Используем следующие параметры **apall**:

```
echelle> epar apall
```

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = apall

input = Y List of input images

(output = Y_ec) List of output spectra

(apertur=) Apertures

(format = echelle) Extracted spectra format

(referen=) List of aperture reference images

(profile=) List of aperture profile images

(interac= yes) Run task interactively?

(find = yes) Find apertures?

(recente= yes) Recenter apertures?

(resize = yes) Resize apertures?

(edit = yes) Edit apertures?

(trace = yes) Trace apertures?

(fittrac= yes) Fit the traced points interactively?

(extract= yes) Extract spectra?

(extras = no) Extract sky, sigma, etc.?

(review = yes) Review extractions?

(line = INDEF) Dispersion line

(nsum = 100) Number of dispersion lines to sum or median

DEFAULT APERTURE PARAMETERS

```

(lower = -5.) Lower aperture limit relative to center
(upper = 5.) Upper aperture limit relative to center
(apidtab= ) Aperture ID table (optional)
# DEFAULT BACKGROUND PARAMETERS
(b_funct= chebyshev) Background function
(b_order= 1) Background function order
(b_sampl -10:-6,6:10) Background sample regions
(b_naver= -3) Background average or median
(b_niter= 0) Background rejection iterations
(b_low_r= 3.) Background lower rejection sigma
(b_high_ = 3.) Background upper rejection sigma
(b_grow = 0.) Background rejection growing radius
# APERTURE CENTERING PARAMETERS
(width = 5.) Profile centering width
(radius = 10.) Profile centering radius
(thresho= 0.) Detection threshold for profile centering
# AUTOMATIC FINDING AND ORDERING PARAMETERS
nfind = 22 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
(aprecen= ) Apertures for recentering calculation
(npeaks = INDEF) Select brightest peaks
(shift = yes) Use average shift instead of recentering?
# RESIZING PARAMETERS
(llimit = -10.) Lower aperture limit relative to center
(ulimit = 10.) Upper aperture limit relative to center
(ylevel = 0.1) Fraction of peak or intensity for automatic wid
(peak = yes) Is ylevel a fraction of the peak?
(bkg = no) Subtract background in automatic width?
(r_grow = 0.) Grow limits by this factor
(avglimi= yes) 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 bef
(t_funct= legendre) Trace fitting function
(t_order= 3) Trace fitting function order
(t_sampl= *) Trace sample regions

```

```

(t_naver= 1) Trace average or median
(t_niter= 0) Trace rejection iterations
(t_low_r= 3.) Trace lower rejection sigma
(t_high_= 3.) Trace upper rejection sigma
(t_grow = 0.) Trace rejection growing radius
# EXTRACTION PARAMETERS
(backgro= none) Background to subtract
(skybox = 1) Box car smoothing length for sky
(weights= none) Extraction weights (none|variance)
(pfit = fit1d) Profile fitting type (fit1d|fit2d)
(clean = no) Detect and replace bad pixels?
(saturat= INDEF) Saturation level
(readnoi= 8) Read out noise sigma (photons)
(gain = 1.) Photon gain (photons/data number)
(lsigma = 4.) Lower rejection threshold
(usigma = 4.) Upper rejection threshold
(nsubaps= 1) Number of subapertures per aperture
(mode = ql)

```

Имена входного и выходного файлов указаны в первых двух строках. В значении параметра **nfind** нужно указать верное число порядков для данного инструмента.

Задача **apall** вначале ищет отдельные эшелле-порядки в кросс-разрезе изображения. Затем определяет положение и размер апертуры и предоставляет возможность ручной корректировки. Для выполнения этих задач необходимо указать параметры **find**, **recente**, **resize** и **edit** как **yes**. Для определения размера и положения апертуры, вычленяются отдельные порядки и отсчеты в них суммируются вдоль направления щели. Такие параметры как **trace**, **fittrac** и **extract** должны быть установлены как **yes**. Другие важные параметры:

```
# AUTOMATIC FINDING ... (nfind): дает количество извлеченных спектров
```

```
# RESIZING PARAMETERS (peak and ylevel): если параметр peak установлен как yes, то размер апертуры определяется как ширина кросс-разреза изображения отдельного спектра на высоте пика ylevel. Например, если ylevel составляет 0.3, то размер апертуры принимается равным 30% высоты пика кросс-разреза.
```

На первом этапе параметры **extras** и **bkg** должны быть установлены как

no, в то время как **avglimi - yes**. После запуска задачи, появится диаграмма как на рисунке 8 (окно **irafterm**). Для изменения параметров процедуры без прекращения ее работы в интерактивном режиме достаточно набрать в активном окне **irafterm : параметр значение**. Например: **: function spline3**.

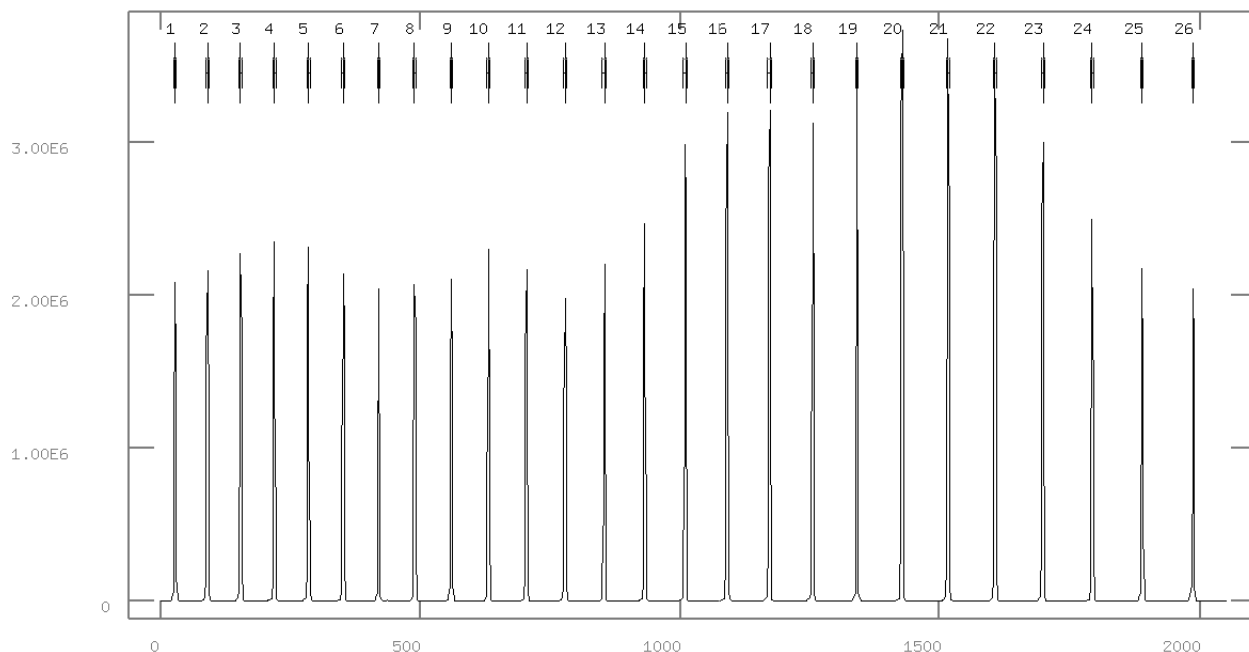


Рис. 8: Кросс-разрез изображения. Результатом первого шага команды **apall** является определение позиции и номера порядков.

Рисунок 8 - это результат автоматического поиска положения и размера апертуры. Положение и размер апертуры могут корректироваться при необходимости. Для определения центра апертуры наводим курсор на центр и нажимаем **n**. Верхняя и нижняя границы апертуры задаются кнопками **u** и **l** соответственно. Наведите курсор на апертуры и нажмите **d** для ее удаления, нажмите **m** чтобы вручную отметить апертуру, которую вы хотите извлечь. Для получения дополнительных команд, смотри справку (введите **?**). Масштаб окна можно менять (порядков бывает много, и в маленьком окне они сливаются). Для этого нажимаем **w** и затем клавишей **e** отмечаем левый нижний и правый верхний углы области окна, которую нужно увеличить. Вернуть прежний масштаб можно, нажав **w** и **a**. Особое внимание обратите на крайние (синие и красные) порядки. Во-первых, в зависимости от спектрального класса объекта сигнал в синей или красной части может быть слабым. Во-вторых, синие порядки расположены очень близко друг к другу, а в красной части между порядками возникают интерференционные полосы, которые часто тоже распознаются как порядки. Если поиск апертуры вас не удовлетворяет, то можно остановить задачу нажав **q** и ответив **no** на все последующие вопросы, и снова запустив задачу

с другими параметрами. Особое внимание стоит уделить номерам порядков. Числа должны быть расположены слева направо без пробелов и дублирования. Чтобы отсортировать апертуры, подведите курсор близко к одной, нажмите **o**, и введите номер, который вы хотите присвоить данной апертуре в этой последовательности. Если положение, размер апертуры и нумерация порядков вас устраивает, то введите **q** и ответьте **yes** на все последующие вопросы. Появится диаграмма похожая на рисунок 9.

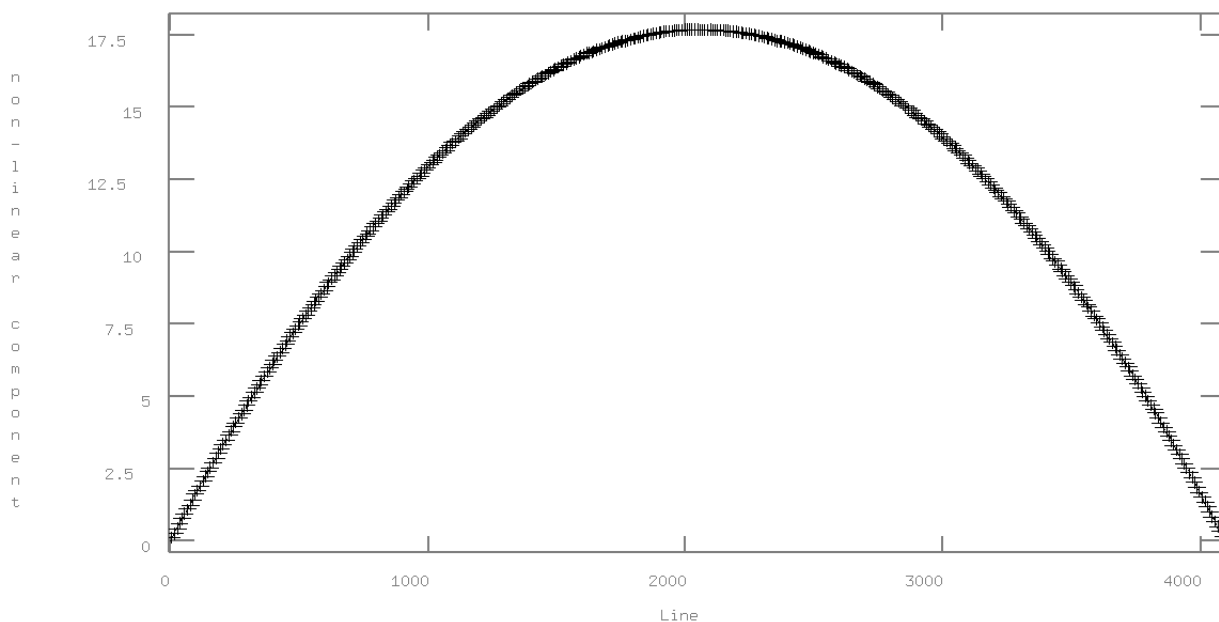


Рис. 9: Трассировка порядков спектра. Горизонтальная ось - направление дисперсии, а вертикальная ось - направление щели (единицы измерения - число пикселей на ПЗС)

На этой диаграмме показан предварительный результат трассировки порядка, где горизонтальная и вертикальная оси указывают номер пикселей в направлении дисперсии и щели соответственно. Обнаруженный пик спектра показан символом **+**, в то время как результат фитирования обнаруженных пиков показан пунктиром. Точки, автоматически исключенные из фитирования отмечаются зеленым.

Можно увеличить порядок фитирующей функции и/или изменить диапазон фитирования для горизонтальной оси. Порядок функции фитирования можно изменить, например, на **3**, набрав на дисплее **:order 3**. Диапазон фитирования изменяется путем нажатия клавиши **s** на две позиции на дисплее (диапазон фитирования инициализируется нажатием **t**). Фитирование с новыми параметрами запускается клавишей **f**. Отдельные точки могут быть удалены нажатием клавиши **d** (когда курсор находится рядом с ней). Такие точки помечаются символом **x**. Чтобы добавить их обратно нажимаем **u**. Чтобы оценить соответствие, можно проверить невязки нажав **j**, или отношение места положения обнаруженных пиков к результату фитирования клавишей **k**.

Нажав **h**, мы вернемся в исходное окно. Описанные выше команды работают практически для каждой интерактивной процедуры фитирования в IRAF (например, в команде **continuum**). Если результат фитирования хороший, то введите **q** и примените аналогичную процедуру к следующему порядку. После фитирования всех спектров содержащихся на изображении, появится экстрагированный спектр первого эшелле-порядка (рис. 10).

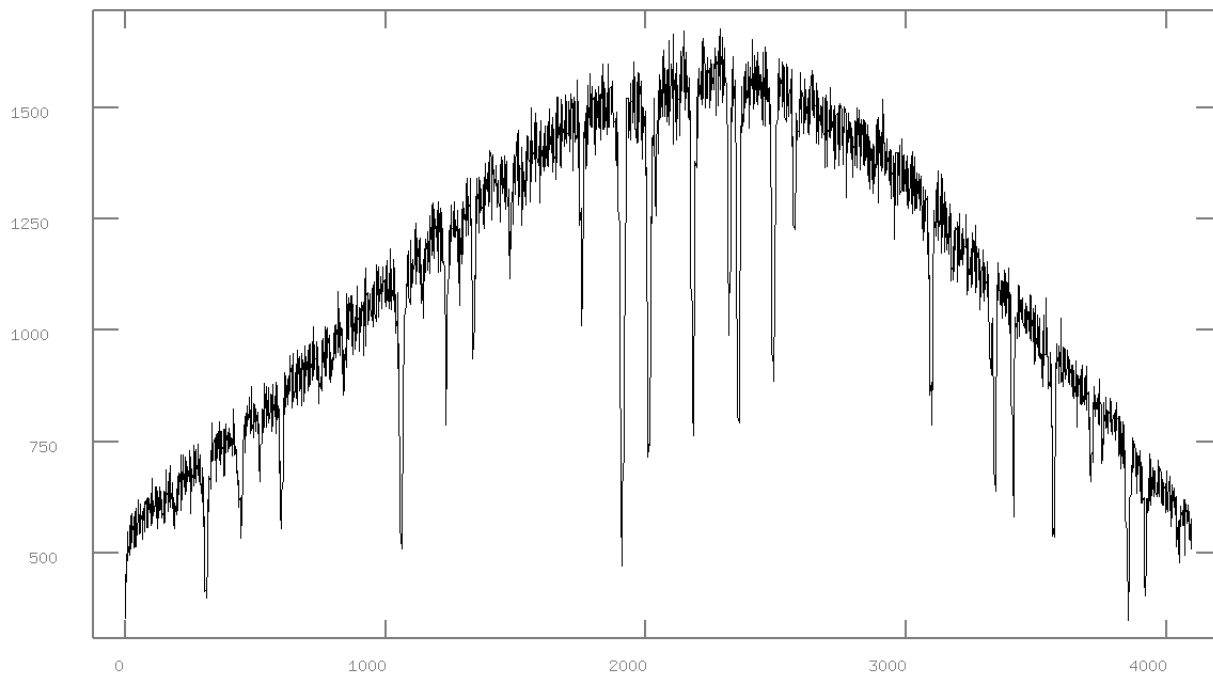


Рис. 10: Спектр, полученный задачей **apall**. Единицей измерения горизонтальной оси на данном этапе является номер пикселя ПЗС.

Для визуализации спектра удобно использовать команду **splot**:

```
echelle> splot Y_ec 1
```

, где **1** — это номер порядка, который мы хотим отобразить.

6 Рассеянный свет

Как можно видеть на изображении кросс-разреза объекта (рис. 11), отсчеты фона между двумя порядками не равны нулю из-за рассеянного света внутри спектрографа. Оценка фона делается путем маскировки апертур спектра и применения поверхностного фитирования других регионов. Здесь вычитание фона делается только для кадров объекта (также вычитание фона должно быть сделано для **flat** перед делением на него). Для этой цели используется задача **apscatter**. Пример настройки параметров этой задачи приведен ниже. Они аналогичны параметрам команды **apall**, только на этот раз изображение объекта, к которому экстракция уже применялась, может быть использовано как опорное изображение (параметр **referen**). Параметры **find**, **trace** и **fittrac** устанавливаются как **no**, а **recente**, **resize**, **edit** как **yes**.

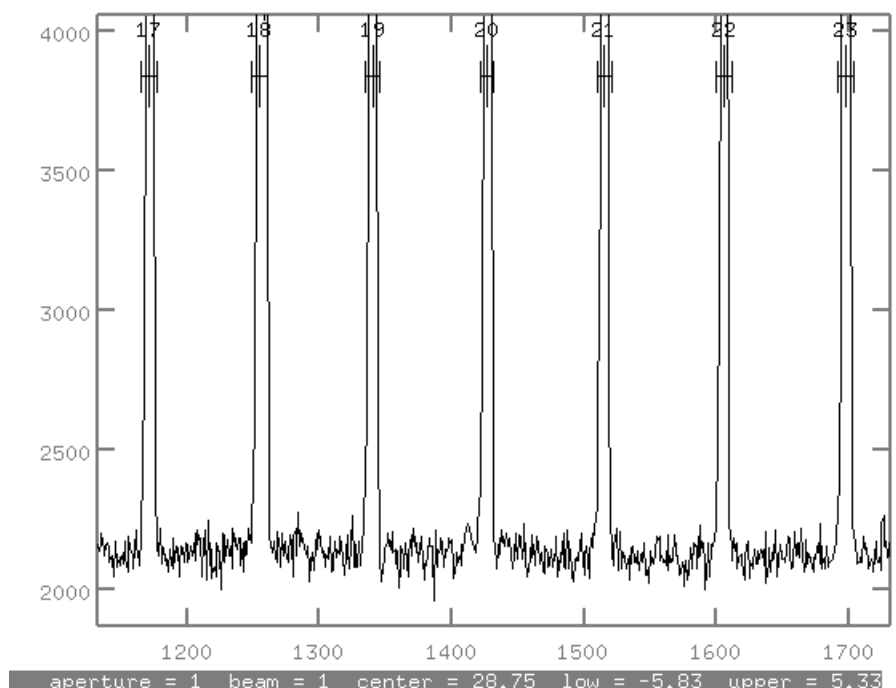


Рис. 11: Кросс-разрез изображения объекта в ходе работы процедуры **apscatter**.

Автоматически определяются порядки, результат фитирования в направлении щели отображается как на рисунке 12. Можно корректировать функцию фитирования и ее порядок. Клавиша **q** завершает процесс фитирования для этого кросс-разреза. После чего можно увидеть результат фитирования для других строк, введя, например, **line 200**. Если фитирование в направлении щели выглядит приемлемо, то нажимаем **q** или **quit**, затем будет показано фитирование в направлении дисперсии как на рисунке 13 (это может занять некоторое время). Можно увидеть кросс-разрез других колонок, например, **column 2500** и т.д. Если подгонка удовлетворительна, то вводим **q** или **quit**, тогда получается изображение с вычтенным рассеянным светом.

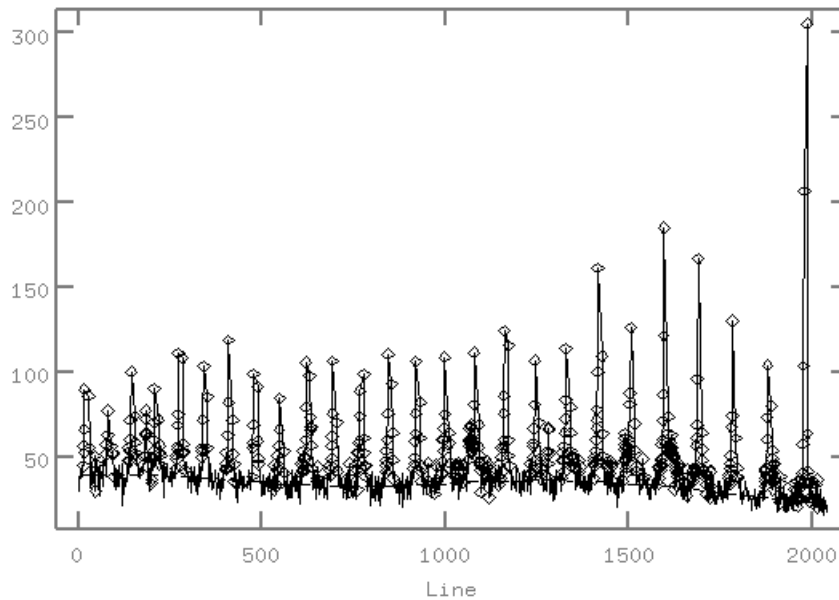


Рис. 12: Кросс-разрез изображения с замаскированными апертурами. Результат фитирования фона показан пунктиром.

I R A F

Image Reduction and Analysis Facility

PACKAGE = echelle

TASK = apscatter

input = X List of input images to subtract scattered lig

output = Xs List of output corrected images

(apertur=) Apertures

(scatter= Y) List of scattered light images (optional)

(referen=) List of aperture reference images

(interac= yes) Run task interactively?

(find = no) Find apertures?

(recente= yes) Recenter apertures?

(resize = yes) Resize apertures?

(edit = yes) Edit apertures?

(trace = no) Trace apertures?

(fittrac= no) Fit the traced points interactively?

(subtrac= yes) Subtract scattered light?

(smooth = yes) Smooth scattered light along the dispersion?

(fitscat= yes) Fit scattered light interactively?

(fitsmoo= yes) Smooth the scattered light interactively?

(line = INDEF) Dispersion line

(nsum = 100) Number of dispersion lines to sum or median

(buffer = 1.) Buffer distance from apertures

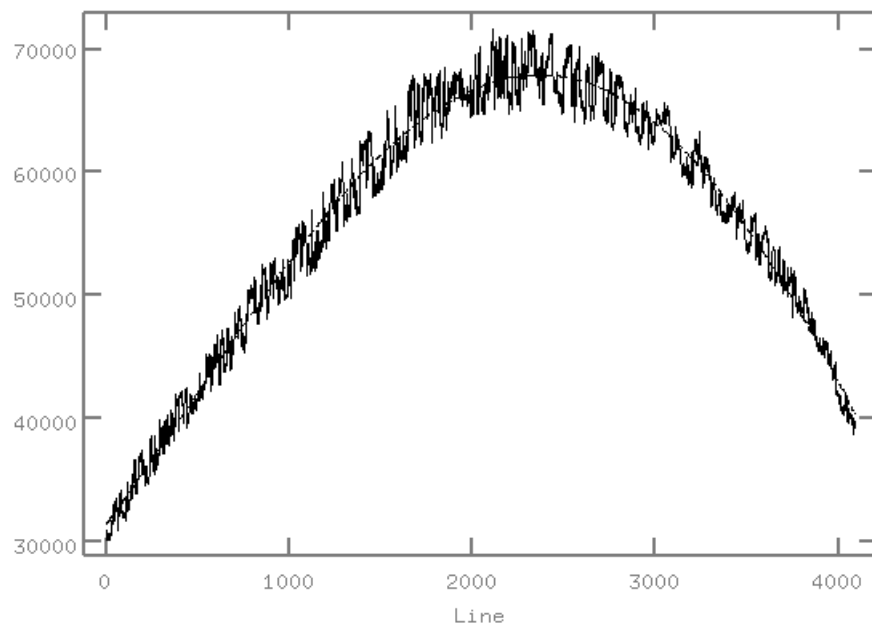


Рис. 13: То же изображение, что и на рисунке 12, только в направлении дисперсии.

(apscat1=) Fitting parameters across the dispersion
(apscat2=) Fitting parameters along the dispersion
(mode = ql)

7 Плоское поле

Чтобы исправить неоднородность чувствительности детектора от пикселя к пикселю, получают изображения лампы белого цвета (**flat**) с теми же настройками спектрографа, что и для научных спектров (рис. 6). Эти данные дают оценку чувствительности спектрографа, включая зависимость от длины волны.

Обычно изображения плоского поля получаются из нескольких кадров путем медианного усреднения всех **flat**, как и в случае **bias** с помощью команды **imcombine**. Учет плоского поля происходит путем деления изображения объекта на **flat**. Однако перед этим обычно **flat** нормализуется. Нормализация **flat** выполняется командой **apnormalize** (или **apflatten**). Параметры **apnormalize** следующие:

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = echelle
TASK = apnormalize
input = flat List of images to normalize
output = flatn List of output normalized images
(apertur= ) Apertures
(referen= Y ) List of reference images
(interac= yes) Run task interactively?
(find = no) Find apertures?
(recente= yes) Recenter apertures?
(resize = yes) Resize apertures?
(edit = yes) Edit apertures?
(trace = no) Trace apertures?
(fittrac= no) Fit traced points interactively?
(normali= yes) Normalize spectra?
(fitspec= yes) Fit normalization spectra interactively?
(line = INDEF) Dispersion line
(nsum = 10) Number of dispersion lines to sum or median
(cennorm= no) Normalize to the aperture center?
(thresho= 10.) Threshold for normalization spectra
(backgro= none) Background to subtract
(weights= none) Extraction weights (none|variance)
(pfit = fit1d) Profile fitting type (fit1d|fit2d)
```

(clean = no) Detect and replace bad pixels?
 (skybox = 1) Box car smoothing length for sky
 (saturat= INDEF) Saturation level
 (readnoi= 0.) Read out noise sigma (photons)
 (gain = 1.) Photon gain (photons/data number)
 (lsigma = 4.) Lower rejection threshold
 (usigma = 4.) Upper rejection threshold
 (functio= spline3) Fitting function for normalization spectra
 (order = 3) Fitting function order
 (sample = *) Sample regions
 (naverag= 1) Average or median
 (niterat= 5) Number of rejection iterations
 (low_rej= 3.) Lower rejection sigma
 (high_re= 3.) High upper rejection sigma
 (grow = 0.) Rejection growing radius
 (mode = ql)

Они также аналогичны параметрам команды **apall**, и изображение объекта, к которому экстракция уже применялась, используется как опорное изображение (параметр **referen**). Поскольку изображения объекта и **flat** получаются при одинаковых настройках спектрографа, то положения спектральных порядков на изображении в принципе одинаковое. (Положения спектров на CCD-изображении записываются в файл как **apNAME** в директории **database**, поэтому не следует удалять или перемещать эту директорию.)

Для ссылки на файл, имя файла указывается в качестве параметра **referen**. Эталонное изображение представляет из себя двумерные данные перед экстракцией, а не извлеченные одномерные спектры. Эталонное изображение используется для поиска апертур и трассировки спектра. В этом случае параметры **find** и **fittrac** должны быть установлены как **no**. Положение и размер апертуры определяются для **flat** дополнительно, потому что изображение объекта и плоского поля в целом между собой различаются. Следовательно, параметры **recente** и **resize** должны быть установлены как **yes**. К сожалению, в списке параметров **apnormalize** не существует **peak** и **ylevel**. Они приведены отдельно в списке параметров **apresize**, который вызывается из **apnormalize** следующим образом:

R A F

Image Reduction and Analysis Facility

PACKAGE = echelle
 TASK = apresize
 input = List of input images
 (apertur=) Apertures
 (referen=) Reference images
 (interac= no) Run task interactively?
 (find = yes) Find apertures?
 (recente= no) Recenter apertures?
 (resize = yes) Resize apertures?
 (edit = yes) Edit apertures?
 (line = INDEF) Dispersion line
 (nsum = 1) Number of dispersion lines to sum or median
 (llimit = -20.) Lower aperture limit relative to center
 (ulimit = 20.) Upper aperture limit relative to center
 (ylevel = 0.4) Fraction of peak or intensity for automatic wi
 (peak = yes) Is ylevel a fraction of the peak?
 (bkg = no) Subtract background in automatic width?
 (r_grow = 0.) Grow limits by this factor
 (avglimi= yes) Average limits over all apertures?
 (mode = ql)

Извлечение **flat** производится **apnormalize** как в **apall** для спектров объекта. Тем не менее, так как используются эталонные данные для трассировки спектра, не нужно ничего делать руками. Первый экстрагированный спектр отображается как рисунок 14. Также на диаграмме показан профиль, аппроксимирующий данные. Чтобы улучшить подгонку, можно изменить функцию и порядок фитирования, например **:function spline3** и **:order 9** соответственно. Можно также изменять диапазон фитирования двойным нажатием клавиши **s**, чтобы исключить неподходящие части спектра из процесса фитирования, н-р пиксели, затронутые «плохими» столбцами. Нажатие **f** выполняет перефитирование с новыми параметрами. Нажатие **q** осуществляет завершение процедуры с данным спектром и осуществляет переход к следующему. Эта процедура применяется ко всем эшелле-порядкам спектрам.

Результат можно увидеть с помощью команды **implot**. Нормализованный **flat** выглядит как рисунок 15. Часть, в которой отсчет равен точно единице, является внешней частью апертуры, т.е. областью между двумя спектрами, где свет (существенным образом) не обнаружен. Разброс, обнаруженный в апертурах, указывает на неоднородность чувствительности детектора. Деление изображения объекта на **flat** производится с помощью команды

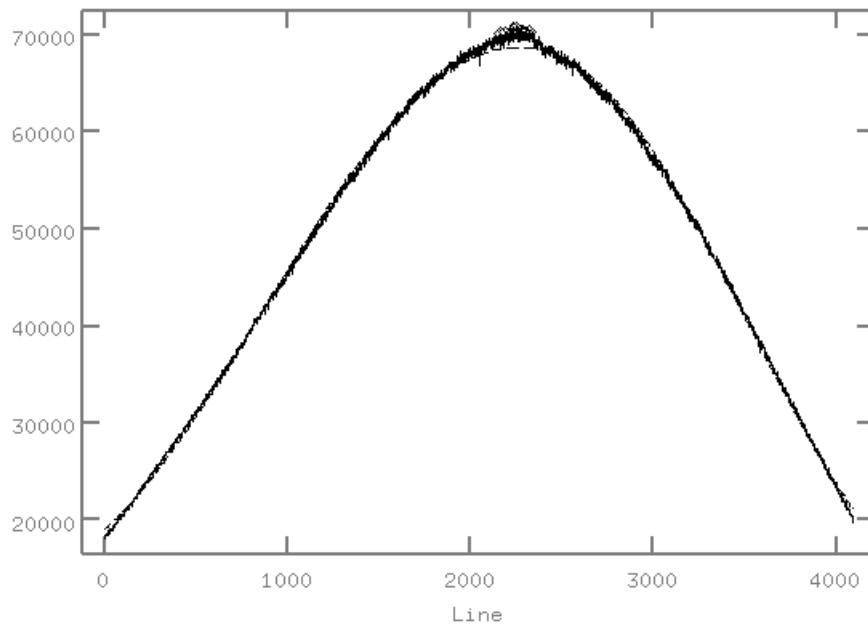


Рис. 14: Подгонка фитирующей кривой к выделенному спектру **flat**.

imarith:

echelle>imarith Xs / flatn Xsf

, где нормализованный **flat** — **flatn.fits**, а результат деления на **flat** - **Xf**.

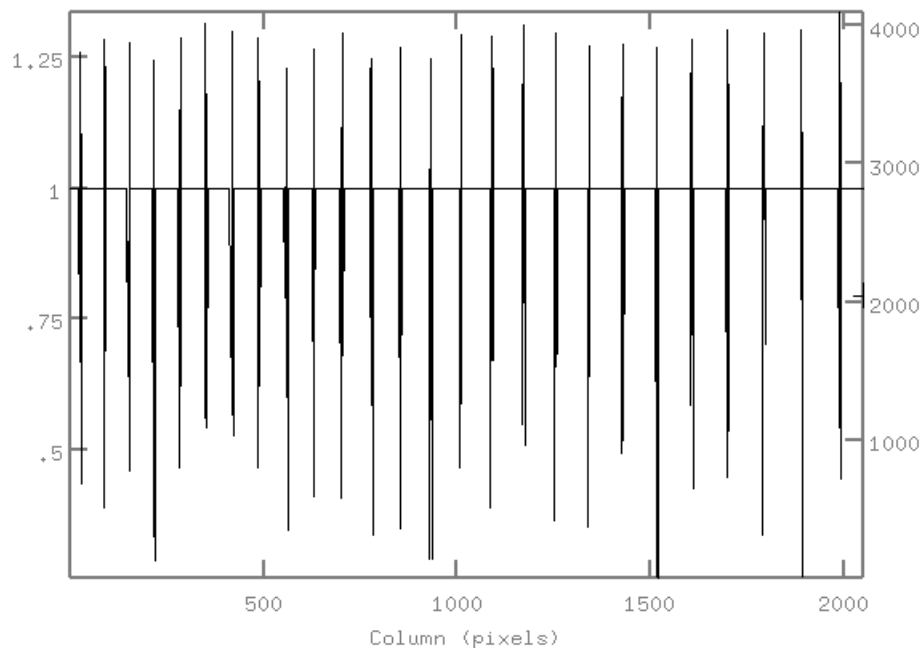


Рис. 15: Кросс разрез **flat** вдоль направления щели.

8 Экстракция одномерных спектров

Экстракция спектра после вычитания рассеянного света из спектра объекта и деления на **flat** снова производится с использованием команды **apall**. Ниже приведен пример настройки параметров для первых 16 строк:

I R A F

Image Reduction and Analysis Facility

```
PACKAGE = echelle
TASK = apall
input = Xsf List of input images
(output = Xsfc)Listofoutputspectra
(apertur = )Apertures
(format = echelle)Extractedspectraformat
(referen = Y)Listofaperturereferenceimages
(profile = )Listofapertureprofileimages
(interac = yes)Runtaskinteractively?
(find = no)Findapertures?
(recente = yes)Recenterapertures?
(resize = yes)Resizeapertures?
(edit = yes)Editapertures?
(trace = no)Traceapertures?
(fittrac = no)Fitthetracedpointsinteractively?
(extract = yes)Extractspectra?
(extras = no)Extractsky, sigma, etc.?
(review = yes)Reviewextractions?
```

Эту часть не нужно запускать в интерактивном режиме (установите для параметра **interac** значение **no**), однако, нужно контролировать процесс на этом этапе, в случае данных с низким соотношением сигнал-шум.

9 Калибровка длин волн

Вышеупомянутая процедура представляет одномерные спектры отдельных эшелле-порядков как функцию от номера пикселя. Калибровка по длине волны производится с использованием спектра сравнения (н-р, ThAr лампа), полученного при тех же настройках спектрографа, что и спектр объекта. Лабораторные длины волн отдельных линий калибровочного спектра известны.

9.1 Идентификация спектральных линий спектра сравнения

Возьмем для работы спектр **ThAr**. Сначала спектры **ThAr** экстрагируются с использованием **apall** как и спектры объекта. В этом случае положение и размер апертуры должны быть такими же как и для объекта. Т.е. параметры **find**, **recente**, ..., **fittrac** определяются как **no**, а файл объекта (перед экстракцией) задается как эталонный файл (параметр **referen**). Для оптоволоконных спектрографов эталонный файл может быть таким же, как и для других шагов. Затем получается спектр, как на рисунке 16.

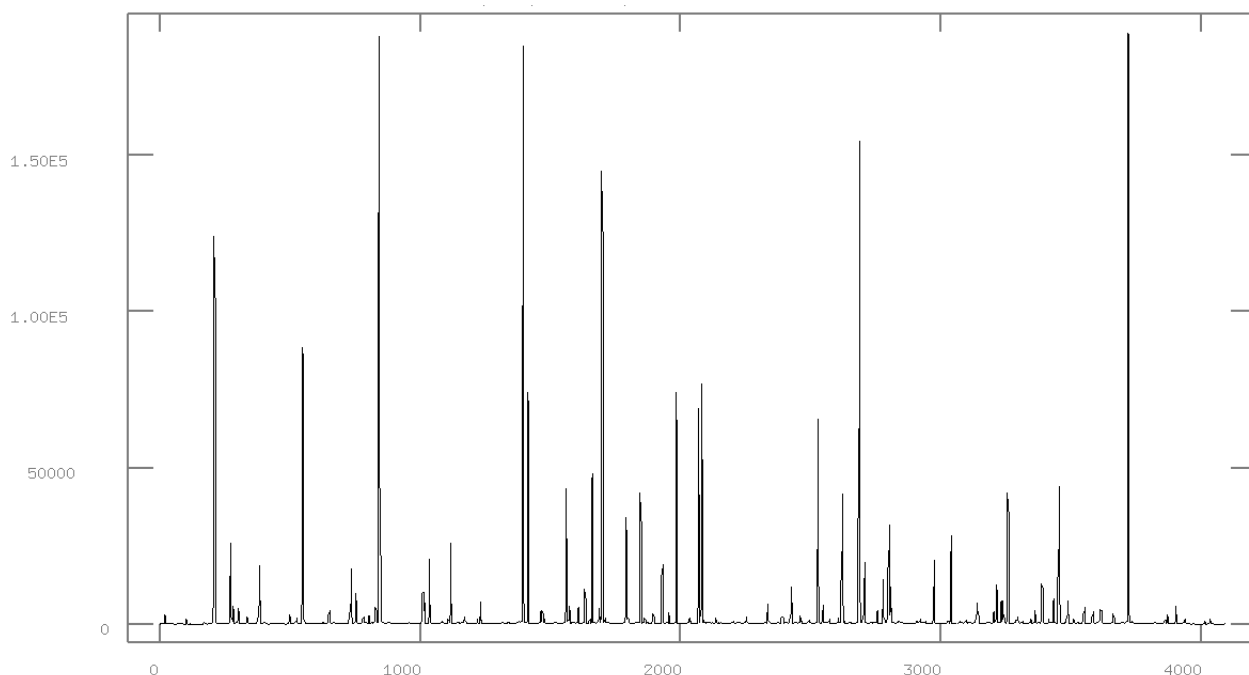


Рис. 16: Пример спектра сравнения. Эта диаграмма появляется в процедуре **ecidentify**.

Следующим шагом является идентификация спектральных линий и определение длин волн. Для этих целей полезно знать охват данных по длинам волн. Обычно, для каждого спектрографа есть свои атласы, в которых отмечены наиболее яркие линии спектра-сравнения и их длины волн. Длина волны каждой ThAr-линии определяется сравнением приведенного выше графика и атласа.

9.2 Идентификация длин волн линий и масштабирование длин волн

Идентификация длины волны для каждой линии ThAr производится с помощью команды **ecidentify**:

```
echelle> epar ecidentify
```

Название спектра-сравнения дается как параметр **images**, в то время как **coodli** настроен на **linelists\$thar.dat**, что означает список линий ThAr реализованный в IRAF. В процессе выполнения этой задачи отображается спектр, как на рисунке 16. Можно увеличить график используя режим **window** (клавиша **w**), как в случае **splot**. Наведите курсор на одну эмиссионную линию и нажмите **m**. Затем появляется отметка над эмиссионной линией. Укажите ее длину волны, указанную в атласе ThAr данных. Для каждого эшелле-порядка следует выделить несколько линий, распределенных как можно более равномерно по всему порядку. Для удаления ошибочно идентифицированной линии наведите на нее курсор и нажмите **d** (отметка может остаться на окне. В этом случае повторно постройте график с помощью клавиши **r**). Можно перейти к следующему порядку спектра используя клавишу **k** (и возврат к предыдущему порядку с помощью **j**). Эта процедура производится для каждого порядка спектра. На практике некоторые порядки можно пропустить.

Если идентификация линий завершена, нажмите **f**, чтобы подогнать функцию к идентифицированным линиям, обеспечивая связь между положением линии на детекторе и длиной волны. Результат аппроксимации вдоль направления дисперсии отобразится как на рисунке 17. Точки с сильными отклонениями можно удалить клавишей **d** (указав на нее курсором). Порядок функции фитирования изменяется путем ввода **:xorder 4, :yorder 3**, которые соответствуют функциям вдоль направления дисперсии и вдоль направления щели. На рисунке 17 показан результат соответствующей подгонки (количество данных велико потому что это результат автоматического поиска линий). График в направлении щели показывается нажатием клавиши **x** и **o** (клавиши **x** и **p** снова показывают подгонку в направлении дисперсии). Если результат фитирования вам нравится (среднеквадратичное отклонение обычно меньше **0,01** для **xorder=yorder=3**), то выйдите из этого режима нажатием **q** (график подобный графику 13 появится снова). Клавиши **l** выполняет автоматический поиск линий Th используя список линий, включенный в IRAF. Максимальное число автоматической идентификации указано в файле параметров **ecidentify** как **maxfeat**. Результат фитирования снова показывается нажатием клавиши **f** (на этот раз многие

линии идентифицированы как показано на рисунке 16). Изменяем параметры **xorder** и **yorder** чтобы получить удовлетворительное соответствие. Обычно значения **xorder = yorder = 4** дает хорошие результаты. Завершение программы инициализируется клавишей **q**.

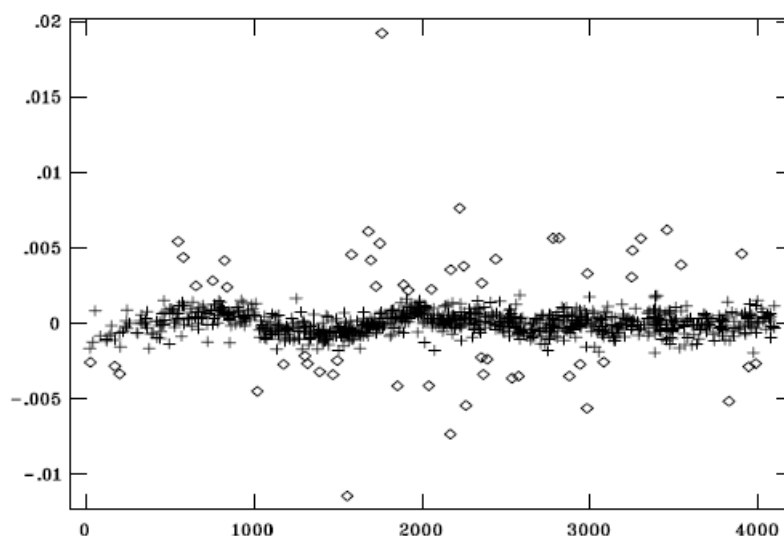


Рис. 17: Результат идентификации линий ThAr и фитирования. Результат фитирования (единицы измерения Å) показан как функция от номера пикселя в направлении дисперсии.

Если при первой идентификации вручную возникают проблемы, то автоматическая подгонка может привести к неверным результатам. Полученные графики идентифицированных линий и фитирования должны быть подробно изучены. Решение по привязке к длинам волн в целом отличается при каждой настройке спектрографа, но в пределах одной настройки должно быть одинаковым. Чтобы применить его ко многим спектрам **ThAr** нужно использовать задачу **ecreidentify**:

```
echelle> ecreidentify newThAr_ec refe=oldThAr_ec
```

, здесь **newThAr_ec.fits** - это новая экспозиция **ThAr**, для которой мы хотим найти решение, а **oldThAr_ec.fits** — та, для которой решение мы нашли ранее. В итоге должно появиться:

```
ECREIDENTIFY: NOAO/IRAF V2.12.1 aokiwk@sb31 Wed 16:15:48
20-Jun-2019
Reference image = oldThAr_ec, Refit = yes
Image Found Fit Pix Shift User Shift Z Shift RMS
newThAr_ec 784/784 784/784 0.0381 0.0634 1.06E-7 6.3E-4
```

Это означает, что процедура нашла и подобрала все линии, которые использовались для эталонного спектра **ThAr**. Можно пересмотреть и

исправить решение для нового спектра командой **ecidentify**.

9.3 Калибровка спектров по длине волны

Теперь шкала длин волн, полученная **ecidentify** или **ecreidentify** применяется к звездным спектрам. Сначала звездный спектр связывается с приведенным выше спектром командой **refspectra**:

```
echelle> refspect Xsf_ec refe=oldThAr_ec
```

, здесь **Xsf_ec.fits** спектр звезды с вычтенным рассеянным светом, поделенный на **flat** и экстрагированный, а **oldThAr_ec.fits** спектр сравнения.

Обычно спектр объекта соседствует с двумя экспозициями ThAr, из которых один был сделан до, а второй после спектра объекта. В **refspectra** можно указывать более одной ссылки чер запятую. В таком случае параметры **sort** и **select** определяются после того как будет выполнена итоговая калибровка, т.е. с каким весом какой эталонный спектр берется. Для более подробной информации смотрите **help refspectra**. Калибровка спектра по длине волны производится командой **dispcor**:

```
echelle> dispcor Xsf_ec Xsf_ecw
```

Затем получается спектр, откалиброванный по длине волны **Xsf_ecw.fits** (рис. 18).

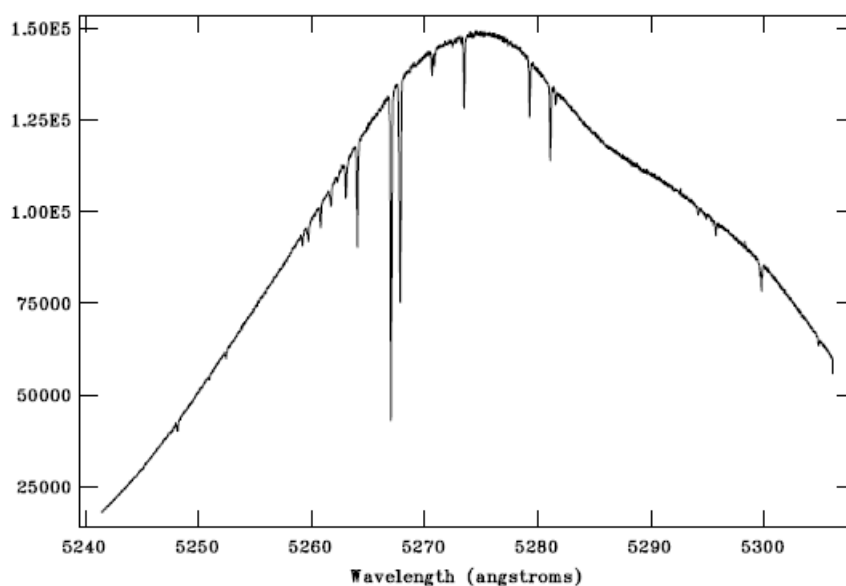


Рис. 18: Спектр откалиброванный по длине волны. Построен с помощью команды **dispcor**.

9.4 Нормализация континуума

Следующим шагом является нормализация спектра путем подбора фитирующей кривой. Эшелле-спектры обычно имеют пик эффективности вблизи центра ПЗС, и низкие отсчеты по краям. Пример показан на рисунке 18, где отсчеты в области с меньшим значением длины волны довольно низкие. В таких случаях, некоторая часть с очень низкими отсчетами может быть вырезана. Обрезку первых 600 пикселей, например, можно выполнить с помощью команды **imcopy**:

```
echelle>imcopy Xsf_ecw[601:4100,*] Xsf_ecwt
```

Нормализация проводится с помощью задачи **continuum**:

```
echelle>continuum Xsf_ecwt Xsf_ecwt.n.fits
```

На рисунке 19 показан пример подгонки функции к спектру, где линии поглощения исключены из подгонки. Функция фитирования и ее порядок изменяются путем ввода **:func spline3** и **:order 5** соответственно. Диапазон фитирования задается так же как в случае **apnormalize** клавишей **s**. Фитирование запускается клавишей **f**. Клавиши **h**, **j** и **k** показывают спектр с подгонкой, невязками и отношением (спектр с нормализованным континуумом) соответственно. Если результат удовлетворительный, то нажимаем **q**, тогда появится спектр следующего порядка. Процедура повторяется для каждого порядка. На рисунке 20 показан пример нормализованного спектра.

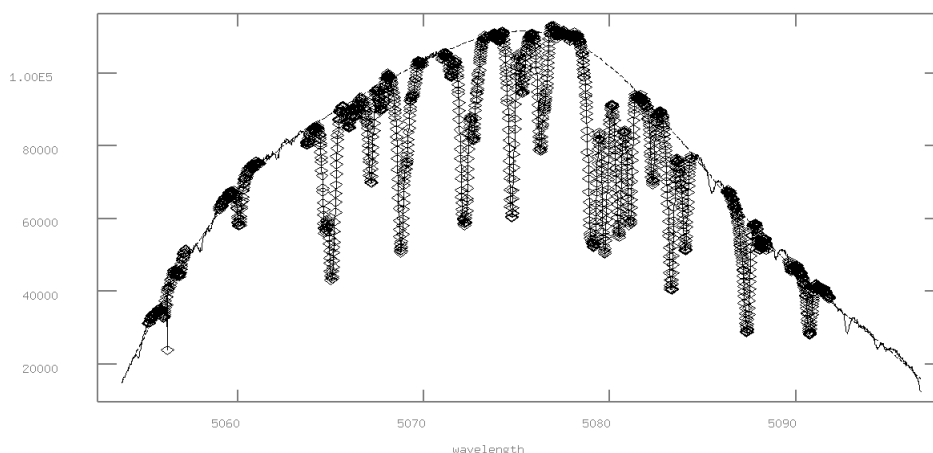


Рис. 19: Пример фитирования континуума.

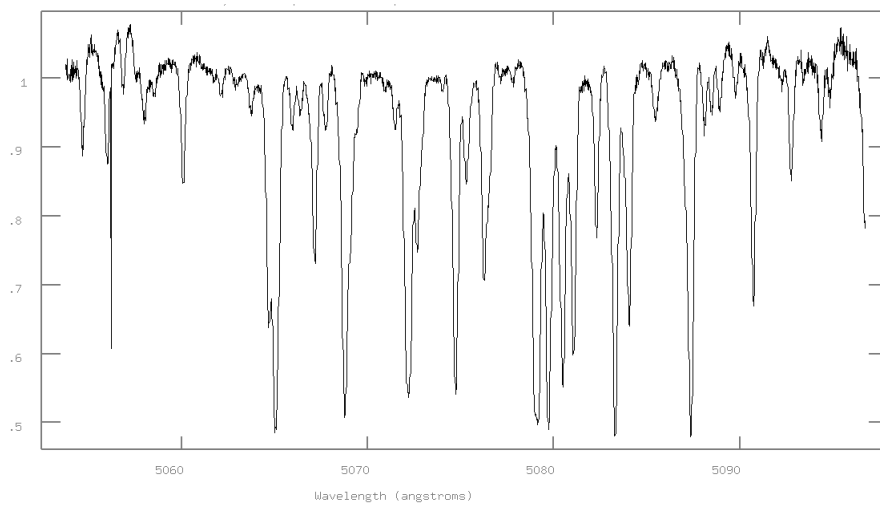


Рис. 20: Пример нормализованного спектра.