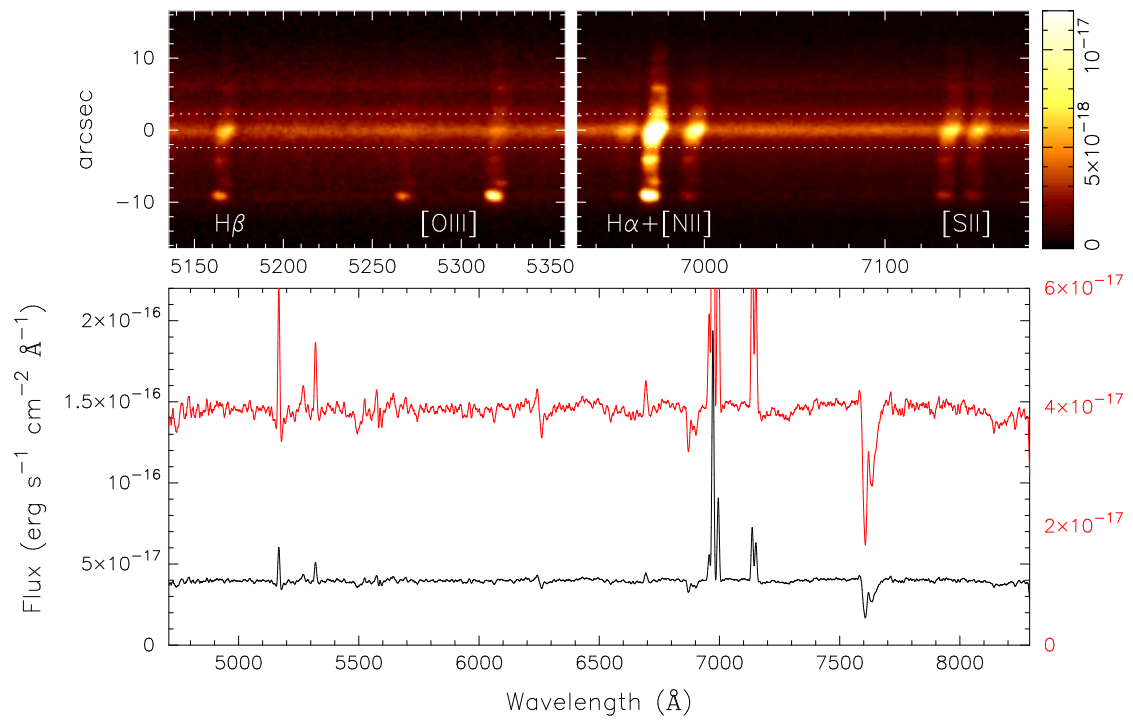


すばる/FOCAS データ解析講習会資料

服部堯、柏川伸成

平成 21 年 5 月 11 日



1 はじめに

このテキストは、FOCAS のロングスリット分光データを IRAF (Image Reduction and Analysis Facility) ¹を用いて解析する手順の一例を示したものです。一部のプロセス (overscan を用いての bias 引き、ゆがみ補正) を除けば全て IRAF に標準でついているタスクを使用しているので、他の装置による低分散スリット分光データ解析の参考にもなると思います。

このテキストに関する御意見、質問、コメントなどがありましたら、下記までお寄せください。

服部 堯

181-8588 東京都三鷹市大沢 2-21-1

国立天文台ハワイ観測所

E-mail : hattori@subaru.naoj.org

2 FOCAS 分光データ

2.1 FOCAS について

FOCAS (Faint Object Camera and Spectrograph, 微光天体分光撮像装置) は、すばる望遠鏡のカセグレン焦点に取り付けられ、可視域での撮像、分光、偏光撮像、偏光分光観測をするための装置です。また多数の天体を一度に分光観測する多天体分光の機能を持ち、主として暗い天体の分光観測に使用されています。機能などの詳しい情報については

<http://www.naoj.org/Observing/Instruments/FOCAS/>
を参照して下さい。

2.2 FOCAS 分光観測の流れ

FOCAS には 2 つの CCD(以下 chip1, chip2) があり、図 1 のような直径 6' の視野を持っています。一度の露出に対して二つの画像ファイルが作成され、奇数番号が chip1、偶数番号が chip2 に対応しています。また図 1 の右に示されているように、ロングスリット分光観測ではスリット位置として”center”と”offset”の 2 箇所があります。これらは観測したい波長に合わせて使い分けられています。

図 2 の左は FOCAS の撮像モードで取った R-band 画像 (片方の CCD のみ) で、中心付近にいる渦巻銀河がターゲットです。目的の天体のスリット位置に来よう望遠鏡を動かし、スリットを入れ、グリズム (分散素子) を入れてスリットと垂直な方向に光を分散させる事で、ロングスリット分光データを得ます。

2.3 calibration データ

FOCAS 観測での calibration データの取り方は目的や観測モードによって多少変わりますが、大体以下のようになっています。

¹詳しくは <http://iraf.noao.edu/> もしくは <http://iraf.net> 参照

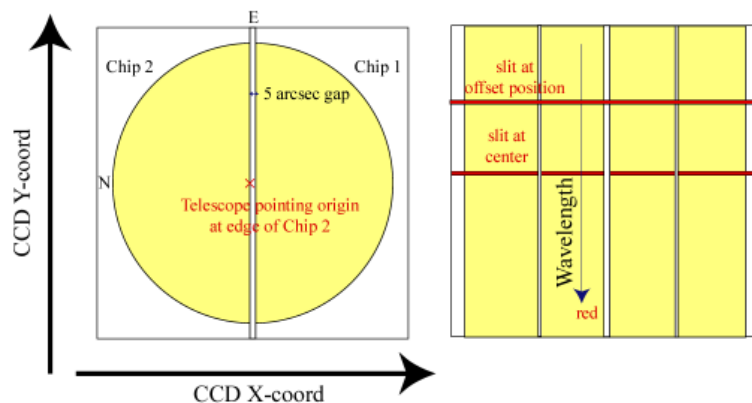


図 1: FOCAS field of view

bias 明け方か夕方にまとめて取ります。通常は overscan 領域を使えば問題ありませんが、視野の広い範囲を使う場合は取る事が望ましいです。

dark 取りません

flat 明け方か夕方にドームフラットをとります。同一ランの別の日に取得している場合もあるので、アーカイブデータを使用する際は注意して下さい。

comparison 波長較正用の比較光源データです。短い波長 ($< 5000\text{\AA}$) の観測ではスカイ輝線が波長較正に使えないため、ターゲットと同じ姿勢で取った comparison フレームがある事が望ましいです。もし無ければ、明け方か夕方に取ったものを使う事になりますが、姿勢が異なるため数 pixel 程度のずれは覚悟する必要があります。

standard star 大抵は明け方と夕方の薄明の時間帯に取得しています。スリットは、標準星の光を全て取り込むために幅が広いもの ($2''$) を使う場合と、ターゲットと同じ幅のものを使う場合とがあります。

2.4 テストデータについて

表 1 に、観測の内容とデータの中身について簡単にまとめておきます。

3 解析の流れ

スリット分光データの解析方法はデータの内容、目的、人によって変わり、やり方は多岐に渡ります。ここでは、

- 暗い天体を分光した場合、一回の露出では何も見えず、何枚も露出を重ねてようやく信号が見えてくる場合があること
- スリット方向の情報 (空間構造など) も保っておきたい場合がある事

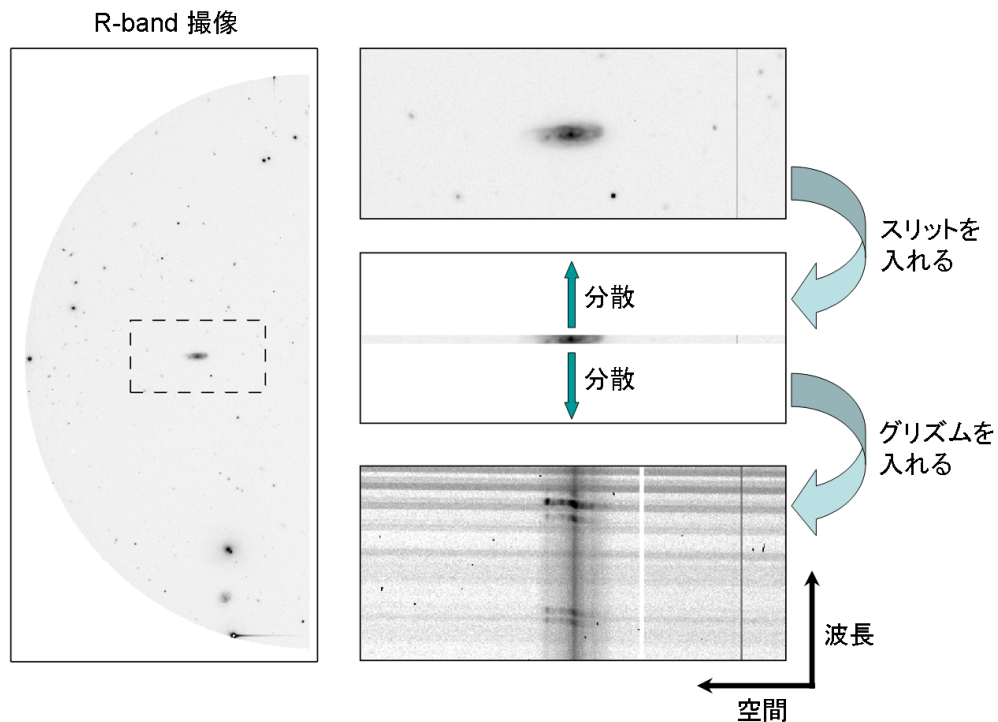


図 2: ロングスリット分光

を考慮して、最後までスペクトルの抽出を行わずに 2 次元のまま解析します。
この場合、

- x 軸を空間方向、y 軸を波長方向に対応させ、その対応関係を求める事
- カウントをエネルギー ($\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1}$) に変換する事

の 2 つが一次処理の目的です。

大まかな手順を並べると以下ようになります。

1. bias 引き (§ 5.2)

CCD の読み出しに伴う、カウントの offset 量を取り除きます。

露出をせずに読み出した画像 (bias 画像) を用い、全ての画像から引き算します。

2. dark 引き

FOCAS の場合ダークは無視できるため、行いません。

3. フラットフィールド (§ 5.3)

装置の光学系や CCD などに起因する感度のムラを取り除きます。

一様な光源 (ドームフラット) を、天体を観測した時と同じ設定で露出し、その画像で割り算します。

観測 : 2006 年 10 月 14 日
 ターゲット名 : SDSS J000347.01-000350.3
 スリット : 0".8 center
 グリズム : 300B (SCFCGRMB01)
 フィルター : Y47 (SCFCFLSY47)
 露出時間 : 600 秒 $\times 3$ ($\pm 3''$ dither)
 ビニング : 3×1 (空間方向 3pixel, 波長方向 1pixel)

Frame 番号	オブジェクト
FCSA00079163–172	ドームフラット (5 枚 1 セット)
FCSA00079183, 184	標準星 (BD+28d4211)
FCSA00079261–268	SDSS J000347.01 導入画像
FCSA00079269–274	SDSS J000347.01 分光データ
FCSA00079567–570	comparison
FCSA00079961–970	bias (5 枚 1 セット)

表 1: 観測、データ内容の概略

4. ゆがみ補正 (§ 5.4)

一般的に、装置内部の光学系を通して CCD 上にできる画像にはゆがみが生じます。分光データの場合スペクトルが曲がってしまうため、これを補正してまっすぐにします (図 3)。FOCAS データ専用のスクリプトを使います。

5. 波長較正 (§ 5.5)

あらかじめ波長の分かっている輝線を使って CCD 上の座標と波長の関係を求め、波長が y 座標の一次関数になるように画像を変換します。

6. スカイ引き (§ 5.6)

背景光を引きます。

通常可視のスリット分光では、天体の両隣の領域を使って内挿します。

7. フラックス補正 (§ 5.7)

標準星のデータを使ってカウントとエネルギーの関係を波長の関数として求めます。その結果をターゲットの画像に適用します。

8. 足し合わせ (§ 5.8)

取得した複数のデータを足し合わせます。

9. スペクトルの抽出、その他 (§ 6)

上にも書いたように、解析の手順は目的などに応じて変わります。例えば明るい星のスペクトルであれば、その星自身の光を使って初期段階にスペクトルの抽出 (1 次元化) を行うことで、手順を簡略化することができます。上記はあくまで一例であると捉えて下さい (§ 5.5.2)。

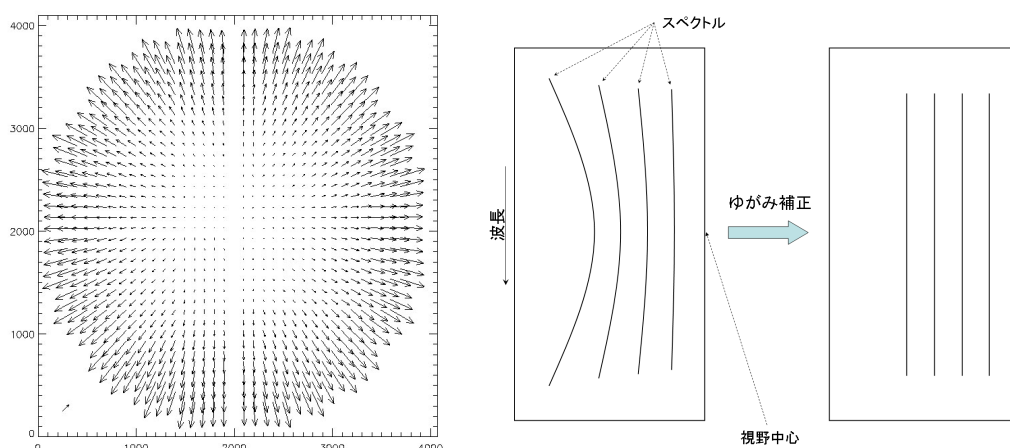


図 3: (左) FOCAS 撮像モードの distortion パターン ([2])。格子状のパターンを撮像し、CCD 上での分布が等間隔格子からどれだけずれているかを示したもの。(右) 分光データのゆがみ補正の概念図。

4 解析の準備

4.1 IRAF

sb 系マシンで初めて IRAF を使う場合、コマンド `mkiraf` を実行して初期設定を行う必要があります。terminal type は `xgterm` を選択して下さい。

```
ana03{hattiak}: mkiraf
-- creating a new uparm directory
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.
```

`mkiraf` を実行したディレクトリに `login.cl` というファイルと `uparm/` というディレクトリが作成されます。emacs 等のエディタを使って、`login.cl` に下の 1 行を追加して下さい。FOCAS データ用の解析スクリプトを使用するための設定です²。

```
task $focasred.pkg = "/home/hattiak/FOCASRED/IRAF/focasred.cl"
```

以下のように、あらかじめ設定が書かれたファイルを `login.cl` と同じディレクトリにコピーしても良いです。

```
ana03{hattiak}: cp /home/hattiak/temp/loginuser.cl ./
```

準備が出来たら

```
ana03{hattiak}: xgterm -sb &
```

²自分の解析環境で使いたい場合は Support Astronomer まで問い合わせして下さい

で xterm を立ち上げ³、xterm 内で login.cl のあるディレクトリに行き、cl と打ち込んで IRAF を立ち上げます。

```
ana03{hattiak}: cl
NOAO PC-IRAF Revision 2.12.2a-EXPORT Wed Jul 14 20:45:34 MST 2004
This is the EXPORT version of PC-IRAF V2.12 supporting most 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. The following
commands or packages are currently defined:
```

apropos	dbms.	language.	obsolete.	softtools.	tables.
color.	focasred.	lists.	plot.	stdas.	utilities.
dataio.	images.	noao.	proto.	system.	

FOCAS データ用のスクリプトを使うので、focasred を読み込んでおいて下さい。

```
cl> focasred
      bigimage      fsenscalib      fwavecalib_sky      ovsb
      distcalib     fskysub        lacos_im          resume_wcs
      flatnorm      fwavecalib     lacos_spec
fo>
```

以下では、xterm (IRAF) で実行するコマンド (IRAF では通常タスクと呼びます) を fo> で、別のターミナルで実行するそれ以外の UNIX/Linux コマンドを ana03{hattiak}: で示します。

解析のために実行したコマンドは、全てテキストファイルにログとして残すようにすると便利です。emacs 等を立ち上げておいて、実行したコマンドは全てコピー&ペーストで記録して置いて下さい。後でやり直さなければならなくなった時や、同じような事を別のデータで行う場合に役に立ちます⁴。

IRAF には、過去に実行したコマンドを記憶する機能もあります。これを利用するためには

```
fo> e
```

を実行します (e は ehistory の略)。過去に実行したコマンドが表示されるので、↑↓ キーで選択し、←→ や Delete キーなどで必要な変更を加えてから実行できます。

4.2 データの準備

まず作業用のディレクトリを準備します。自分のユーザー名のディレクトリを /mfs01i/等を作成して下さい。

³”-sb”はスクロールバーを出すためのオプション

⁴実行したすべてのコマンドを自動的に記録したい場合は、
fo> cl.logfile = "logfile.cl"
fo> cl.keeplog = yes
とする。

```
ana03{hatttriak}: mkdir /mfs01i/hatttriak
```

テストデータは/home/hatttriak/testdata.tar.gz に置いてあるので、以下の要領で自分の作業ディレクトリに展開して下さい。

```
ana03{hatttriak}: cd /mfs01i/hatttriak
ana03{hatttriak}: tar zxvf /home/hatttriak/testdata.tar.gz ./
./testdata
./testdata/FCSA00079273.fits
./testdata/FCSA00079261.fits
.
.
.
```

テストデータは以下からも取得可能です。

```
http://optik2.mtk.nao.ac.jp/subaru\_red/FOCAS/testdata\_20061204.tar.gz
```

4.3 その他

すばる望遠鏡 (に限らず多くの望遠鏡) で取られたデータは、FITS (Flexible Image Transport System) と呼ばれる形式のファイルに保存されます。この FITS ファイルを表示したり、IRAF を用いたインタラクティブな解析をするために ds9 と呼ばれるソフトウェアを使用します。先ほどデータを展開したディレクトリに入り、ds9 を立ち上げて下さい。

```
ana03{hatttriak}: cd testdata
ana03{hatttriak}: ds9 &
```

メニューもしくはボタンで”File”→”Open”を実行して FITS ファイルを選ぶか、IRAF の display コマンドを使って画像を表示することができます。

```
fo> display chip2/FCSA00079184.fits
```

初期設定では、中心部分の 512 × 512pixel のみが表示されます。画像全体を表示する場合は、最初に画像の大きさを指定する必要があります。

```
fo> reset stdimage= imt1x4
```

5 解析

5.1 準備 (リスト作成)

まず、解析に使用するリストを作成します。ここで言うリストとはファイル名の一覧が書かれたテキストファイルの事で、複数のファイルをまとめて処理するために使用します。

まず chip 1 と chip 2 にファイルを分けます。

```
ana03{hatttriak}: mkdir chip1
ana03{hatttriak}: mkdir chip2
ana03{hatttriak}: mv *[13579].fits chip1/
ana03{hatttriak}: mv *.fits chip2/
```

3 行目では、”.fits”の直前の文字が 13579 のいずれかであるファイル、つまり frame 番号が奇数であるファイルを全て chip1/ 以下に移動しています。

今回の解析では chip2 しか使用しないので、例えば以下のようにしてリストを作成します (表 1 参照)。

```
ana03{hatttriak}: ls chip2/FCSA000799*.fits > list.bias
ana03{hatttriak}: ls chip2/FCSA000791[67]*.fits > list.flat
ana03{hatttriak}: ls chip2/FCSA0007918*.fits > list.obj
ana03{hatttriak}: ls chip2/FCSA0007927*.fits >> list.obj
ana03{hatttriak}: ls chip2/FCSA000795*.fits >> list.obj
```

ここでは標準星、比較光源も list.obj に含めています。

IRAF のタスク hselect を使って以下のようにデータの内容を確認しておいてから行う事もできます。

```
fo> hselect chip2/*.fits $I,DATA-TYP,OBJECT yes
FCSA00079164.fits      DOMEFLAT      DOMEFLAT
FCSA00079166.fits      DOMEFLAT      DOMEFLAT
FCSA00079168.fits      DOMEFLAT      DOMEFLAT
.
.
.
fo> hselect chip2/*.fits $I 'OBJECT=="DOMEFLAT"' > list.flat
fo> hselect chip2/*.fits $I 'DATA-TYP=="OBJECT"' > list.obj
```

など。

noao.imred.ccdred.ccdgroups を利用するのも良いでしょう。

```
fo> imred
im> ccdred
cc> ccdgroups chip2/*.fits output="list." group=title
```

5.2 bias 引き

標準バイアスフレームを作成します。

```
fo> imcombine @list.bias stdbias combine=median
```

imcombine は画像を足し合わせるタスクです。

list.bias	:	足し合わせを行う画像のリストが書いてあるファイル名 先頭の@は、これが画像ではなくリストファイルである事を示しています。
stdbias	:	出力画像のファイル名
combine=median	:	各ピクセル位置で median(中央値) を求めるというオプション

上記以外にも多数のパラメーターがあります。以下のように、`epar` を使ってパラメーターの編集ができます。

```
fo> epar imcombine
```

```

                                I R A F
                        Image Reduction and Analysis Facility

PACKAGE = immatch
TASK = imcombine

input      =      @list.bias  List of images to combine
output     =      stdbias    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
More
ESC-? for HELP
```

ここで編集することによって、パラメーターの既定値を変更することができます。変更を保存して編集を終了するには `:q`、編集したパラメーターでタスクを実行するには `:go` と入力します。このテキストでは、コマンドラインに IRAF タスクのパラメーターを書いていきますが、実際は `epar` を使った方が便利な場合が多いです。どんどん活用して下さい。

出来た `stdbias.fits` を `ds9` で表示して、画像を確認して下さい。x=530 付近の dead column 以外は、大体 11700 カウントくらいの一様な画像になっているはず⁵。

```
fo> imstat stdbias[201:500,1001:2000]
```

とすることで、画像のカウントの平均値、標準偏差などをチェックすることもできます。

次に以下のコマンドを実行して、バイアスを引いた後のファイル名が書かれたリストを作ります。

⁵ここだけに限らず、正常に処理が進んでいるかどうかを確認するために、できる限り画像を目で見て確認することをお勧めします

```

ana03{hattriak}: sed -e 's/chip2/bs/' -e 's/fits/bs.fits/' list.flat > list.flat.bs
ana03{hattriak}: sed -e 's/chip2/bs/' -e 's/fits/bs.fits/' list.obj > list.obj.bs
ana03{hattriak}: mkdir bs
ana03{hattriak}: cat list.flat.bs
bs/FCSA00079164.bs.fits
bs/FCSA00079166.bs.fits
.
.
.

```

ここでは sed (Stream EEditor) を使って文字列”chip2”を”bs”に、”fits”を”bs.fits”に変換していますが、ファイルをコピーして emacs 等で編集しても良いです。バイアスを引いた画像を入れるためのディレクトリ bs を作り、cat でファイルの中身を表示させて確認しています。

タスク imarith を用いて stdbias.fits を引き算し、結果を bs/*.bs.fits として保存します。

```

fo> imarith @list.flat - stdbias @list.flat.bs
fo> imarith @list.obj - stdbias @list.obj.bs

```

ディレクトリ bs/ 内にバイアスを引いた画像が出来ている事を確認して下さい。

次に overscan 領域を引きます。overscan とは、CCD から電荷を読み出す際に、実際にあるピクセル数よりも余分に読み出しを行うことを言います。この領域を調べる事で電荷がない場合にどのようなピクセル値 (カウント) になるのかを知ることができます。今回使用するデータでは全て、右端の幅約 20pixel の領域が overscan 領域になっています。

実際にバイアスを引いた後の画像 (例えば FCSA00079184.bs.fits) を見るとわかるのですが、右端の overscan 領域など、光の当たっていない部分のピクセル値が優位に 0 からずれています。以下のようにして、カウントの分布を確認してみましょう。

```

fo> implot bs/FCSA00079184.bs.fits

```

implot を実行すると、”irafterm”と書かれた window が立ち上がります。”Line 2048”と書かれていますと思いますが、これは y=2048 でのピクセル値の分布になっている事を示しており、横軸は x 座標になっています。この window の上で”:c 665 683”と入力して、overscan 領域のピクセル値を y の関数として表示させてみましょう (図 4)。バイアスを引いたにも関わらず値が 0 になっていないのは、バイアスフレームを取得した時と FCSA00079184.fits を取得した時の間でバイアスレベルに違いがあるためです。この違いを補正するために、overscan 領域のピクセル値が 0 になるように調整します。implot は”q”を押して終了させて下さい。

```

fo> wcsreset @list.flat.bs world
fo> wcsreset @list.obj.bs world
fo> ovsub @list.flat.bs @list.flat.bs
fo> ovsub @list.obj.bs @list.obj.bs

```

上ではまず wcsreset で画像の座標を初期化し⁶、focared 内のタスク ovsub を使って overscan の引き算を行っています。ここでは結果を同じ名前の FITS ファイルに上書きしています。

⁶FOCAS の分光データの FITS ヘッダーに不具合があるためです。撮像データの場合は使用可能な座標の情報が含まれているので、初期化しないで下さい。

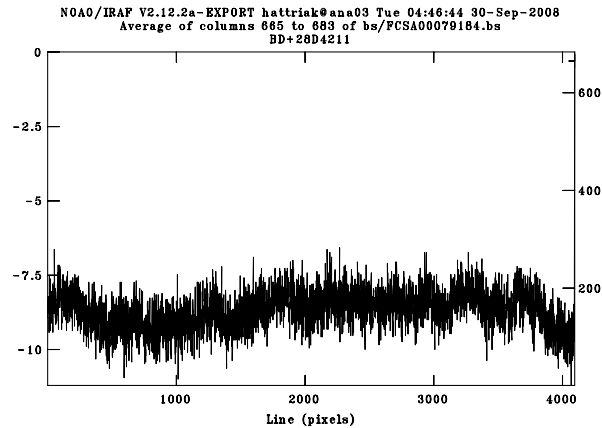


図 4: stdbias.fits を引いた後の overscan 領域

5.3 フラットフィールド

バイアスと同様の手順で、標準フラットフレームを作成します。

```
fo> imcombine @list.flat.bs flat combine=median
```

次に、このフラットフレームの規格化を行います。ds9 で flat.fits を表示すると同時に、implot で y 方向のカウントの分布を確認して下さい。

```
fo> implot flat
```

”:c 350”と入力して中心付近 y 方向をプロットします。右から左に向かってカウントが大きくなっているのが分かります。この分布はドームフラットの光源が赤い色をしているため長波長側が明るく写っているのが原因です。このようなパターンを補正する解析方法もあります⁷が、ここではフラットフレームのカウントの平均値が大体 1 になるように、定数で割り算するにとどめます。

```
fo> imarith flat / 10000 flat.nr
```

この規格化されたフラットを使って、オブジェクトを割り算します。

```
ana03{hattriak}: sed 's/bs/ff/g' list.obj.bs > list.obj.ff
ana03{hattriak}: mkdir ff
fo> imarith @list.obj.bs / flat.nr @list.obj.ff
```

上の例では、list.obj.bs に含まれる文字列”bs”を全て”ff”に変換したリストを作成し、出力ファイルのリストとして使っています。この結果、フラット後の画像は ff/*_ff.fits として出力されます。

5.4 ゆがみ補正

図 3 に示したように、FOCAS には視野中心から外側に向かって広がるような形の distortion パターンがあります。分光データの場合、これはスペクトルの曲がりとして見えてきます。

⁷付録 C 参照。

実際のデータで確認するために、ds9 で ff/FCSA00079270.ff.fits を開いて下さい。メニューの”Zoom”→”Pan Zoom Rotate Parameters...”を選択します。出てきたウィンドウで”Zoom”を”2 0.05”に変えて”Apply”ボタンを押します。y 方向に圧縮して表示されるので、スペクトルが曲がっている様子が見えやすくなると思います。

focasred 内のタスク distcalib を使って、この曲がりを補正します。

```
ana03{hattriak}: sed 's/bs/dc/g' list.obj.bs > list.obj.dc
ana03{hattriak}: mkdir dc
fo> distcalib @list.obj.ff @list.obj.dc obsmode=spec
```

今回取り扱うのは分光データなので”obsmode=spectroscopy”を指定します。撮像データの場合は”obsmode=imaging”です。結果は dc/FCSA*.dc.fits として出力されます。

5.5 波長較正

今回の解析では、ターゲット SDSS J000347.01 の波長較正には夜光の輝線を使い、標準星には比較光源を使います。

5.5.1 SDSS J000347.01 の波長較正

これまでは得られた画像をそのまま使用してきましたが、ここで必要な領域だけを切り取ります。使わない事が明らかな部分はできるだけ取り去って、解析上の困難を避けるのが目的です。dc/FCSA0007927[024].dc.fits を ds9 で確認しながら、例えば x 方向は 201 から 500、y 方向は 41 から 2620 までを切り出す事にすると、

```
fo> imcopy dc/FCSA00079270.dc.fits[201:500,41:2620] obj1
fo> imcopy dc/FCSA00079272.dc.fits[201:500,41:2620] obj2
fo> imcopy dc/FCSA00079274.dc.fits[201:500,41:2620] obj3
```

という風にします。3 行コマンドを並べましたが、もちろんリストを使っても良いです。

夜光輝線の S/N を上げるためにこの 3 枚を重ね⁸、その画像に対してラインの同定を行うタスク identify を走らせます。

```
fo> imcombine obj?.fits sky combine=median
fo> identify sky section="middle column"
      coordli="FOCASREDHOME$/IRAF/skyline_midres.dat" fwidth=8 order=3
      function=chebyshev

section   : ラインの同定を行う区画
coordli   : 輝線の波長が書かれているファイル
fwidth    : 同定に使われる夜光輝線の幅。ピクセル単位
order     : ピクセル-波長関係を求める際のフィットの次数 (初期値)
           : ここでは次数が低い場合にどうなるかを確認しながら進めるために、
           : 低めの設定で始める形にしています
function  : フィッティングに使用する関数
```

⁸これは今回使用するデータの露出時間が、10 分間と比較的短いからです。例えば 20-30 分の露出であれば一枚で十分ですし、観測時間の離れたデータを重ねてしまうと姿勢差 (装置の姿勢によって検出器上での像の位置が移動する現象) の影響が出て、正確な較正ができなくなってしまう。

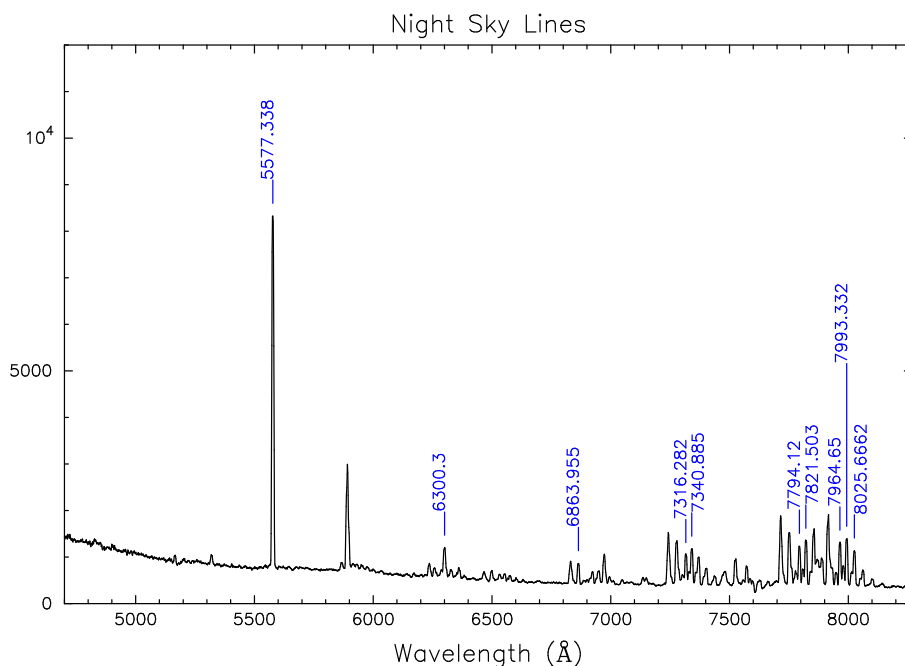


図 5: 夜光輝線の同定

fwidth の値が適切ではない場合、輝線の位置検出精度などに影響が出てしまいます。常に適切な値を使うよう注意して下さい。上では、紙面節約のためにパラメーターをコマンドの後ろに並べていますが、epar で設定しても構いません。

identify を走らせると、irafterm ウィンドウに図 5 を左右逆にしたようなグラフが出てきます。マウスを波長が分かっているラインの近くに持って行った状態で”m”(mark) を押して下さい。そこで波長を聞かれるので数値を入力する、というのが基本操作になります。identify を走らせる際に使用するラインのリストを”coordli”として指定しているので、整数部分だけ入力すれば波長の正確な値は補完してくれます。

以下に主要な操作を挙げるので、図 5 に書かれたラインを全て同定し、fit を行ってください。強い輝線をいくつか同定した段階で fit を行くと、その段階でのデータを元に横軸が波長に変換されます。すると、他の輝線が見つけやすくなり、また輝線の波長も自動的にリストから選ばれるようになります。identify のオプションで与えている”order=3”はフィットの次数ですが⁹、輝線の同定を進めて行くとこの次数では足りなくなり、残差にパターンが見えてきます。このような場合は fit の画面で”:order 4”などと入力して次数を変更します。全体をうまく fit するにはどれくらいの order が必要なのかを確認しておいて下さい。残差 (rms) は 0.2(Å) 程度になるはずです。この identify の結果は、database/idsky に保存されています。

夜光輝線のより詳しい情報については FOCAS の HP や文献 [3], [4] を参照して下さい。

identify では画像の中心部 (x=150) でだけ同定を行いました。他の x 座標でも同様の事を行うために、reidentify を使います。

⁹この”order”の値は多項式の項数を表しています。例えば直線（一次式）をフィットする場合、定数項と一次項の二つで order=2 となります。以下で出てくる xorder, yorder についても同様です。

key	動作
m	マウスカーソル付近で feature を探す
d	マウスカーソルに一番近い feature を消す
f	ピクセル vs 波長での fitting 画面に入る
X	マウスカーソルを中心に x 方向に拡大
Y	マウスカーソルを中心に y 方向に拡大
Z	マウスカーソルを中心に x と y 方向に拡大
>	カーソル位置を y 座標の最大値に設定
<	カーソル位置を y 座標の最小値に設定
r	redraw (上記の拡大等の操作をキャンセル)
c	カーソル近くの feature の座標表示
q	identify 終了
C	カーソル位置の座標を表示
?	help 表示
fitting 画面の操作	
f	再 fit
d	feature の消去
c	カーソル近くの feature の座標表示
q	fitting 画面から抜ける
?	help 表示
:order 4	fitting の order を変更
:func chebyshev	fitting に使用する関数を変更

表 2: identify におけるキー操作

```
fo> reidentify sky sky nlost=1 ver+ section="middle column"
```

```
REIDENTIFY: NOAO/IRAF V2.12.2a-EXPORT hattriak@ana03 Tue 02:02:47 30-Sep-2008
```

```
Reference image = sky, New image = sky, Refit = yes
```

Image Data	Found	Fit Pix	Shift	User Shift	Z Shift	RMS
sky[140,*]	10/10	10/10	0.249	-0.353	-4.8E-5	0.155
sky[130,*]	10/10	10/10	0.409	-0.576	-7.9E-5	0.155
sky[120,*]	10/10	10/10	0.361	-0.508	-6.9E-5	0.155
sky[110,*]	10/10	10/10	0.389	-0.548	-7.5E-5	0.186
.						
.						
.						

middle column (x=150) での同定結果を元にして、画像全体で夜光輝線の位置を測定していきます。”ver+”とする事によって、上記のように各 x 座標での identify の結果が表示されます。途中で止まってしまうたり、右端の”RMS”の値が急に大きくなったりしていないか、しっかり確認して下さい。

次に、identify/reidentify で測定した座標を用いて、波長を x と y の関数として 2 次元の fitting を行います。

```
fo> fitcoords sky func=chebyshev
```

”Fit sky interactively (yes):”と聞かれるので yes と答えて (リターンキーを押せば OK) 下さい。

key	動作
x,y	グラフの x,y に割り当てる座標を変更する x : sky の x 座標、 y : sky の y 座標 z : 同定された feature の波長、 s : $z=f(x,y)$ でフィットした時の $f(x,y)$ の値 r : fitting からの残差 (s-z) 例 1: x に引き続いて x を押せば、グラフの横方向の座標が sky の x 座標になる 例 2: y に引き続いて z を押せば、グラフの縦方向の座標が波長になる
d	データ点の消去 (fit に使わなくなる)
	d を押した後に p を押すと、マウスに最も近いデータを消去します
r	再描画
f	fit、再 fit
q	fitcoords 終了
:xorder 4	x 方向の fitting order を変更
:yorder 4	y 方向の fitting order を変更
:func chebyshev	fitting に使用する関数を変更

表 3: fitcoords におけるキー操作

まず”xxyy”と順番に押して、横方向が x 座標、縦方向が y 座標のグラフになるよう設定して下さい。この表示で、同定されたラインが画像上でどのように分布してるのかを見ることができます。画像全体に渡って (x = 0-300) 検出されている事を確認して下さい。

次に、fitting の残差を確認しながら次数を調整していきます。まず、”yr”と押してから再描画し、縦軸の表示を残差に変更して下さい。”x”や”y”で設定を変更した場合は、”r”で再描画しないと反映されないので注意して下さい。横軸は x 座標のままです。初期設定では x 方向の次数 (xorder) が 6 になっていますが、これは高すぎてデータ点がまばらな場所での振る舞いがおかしくなってしまう可能性があります。再 fit と再描画で残差を確認しながら、xorder を 1 つずつ下げて行って下さい。次数を下げすぎると、残差に大局的なパターンが出てきて大きくなってしまいます。そうならない範囲でできるだけ xorder を下げて下さい。通常は xorder=2-3 で十分なはずです。xorder が決まったら”xy”と押して横軸を y 座標に変更し、同様に yorder をチェックします。yorder は identify の時の order と同じで良いと思いますが、最終的には残差のパターンを見て判断して下さい。問題なければ”q”を押して終了し、保存するかどうかの問いに”yes”と答えます。

最後に、fitcoords で行った fitting の結果を用いてターゲット画像を変換します。

```
fo> transform obj1 obj1.wc sky
fo> transform obj2 obj2.wc sky
fo> transform obj3 obj3.wc sky

Transform obj1 to obj1.wc.
Conserve flux per pixel.
User coordinate transformations:
    sky
Interpolation is linear.
Using edge extension for out of bounds pixel values.
Output coordinate parameters are:
    x1 =          1., x2 =        300., dx =          1., nx =    300, xlog = no
    y1 =       4687., y2 =       8301., dy =        1.401, ny =   2580, ylog = no
```

以上で波長較正は終了です。

図 6 は波長較正前後の画像を示したものです。波長較正によって、斜めになっていたスカイの輝線がまっすぐになり、y 座標と波長の関係が求められています。上記の transform の出力を見ると、

$$\lambda = (y - 1) \times 1.401 + 4687$$

という関係になっている事がわかります¹⁰。

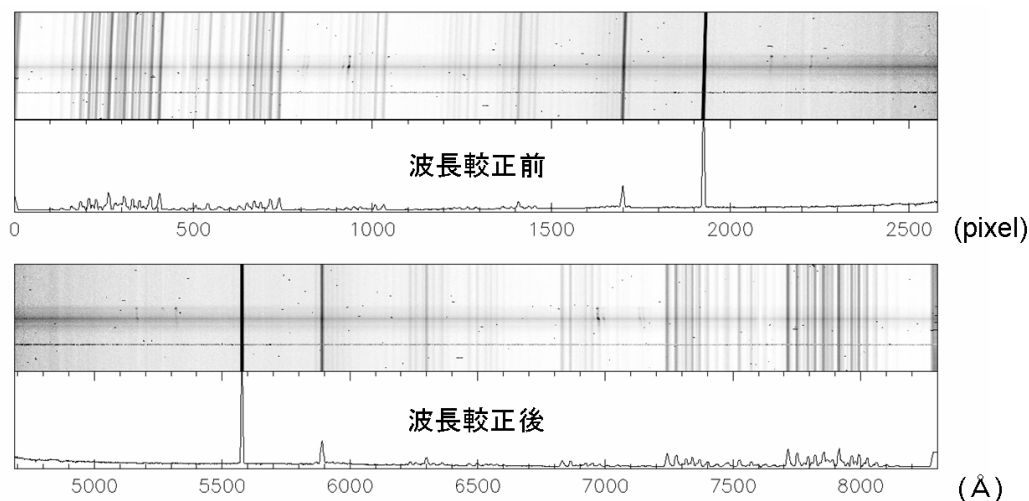


図 6: 波長較正の前と後

5.5.2 標準星の波長較正

標準星の波長較正は、比較光源画像を使用するという事以外はターゲットと一緒になので、全く同じようにできます。が、標準星の場合は 2 次元で fit する必要は全くないので、別の方法でやってみます。標準星に限らず、対象が明るくて積分されたスペクトルだけが必要な場合などは、同様のやり方でやる事ができます¹¹。

この場合ゆがみ補正は必要ないので、フラット済みの標準星フレーム (ff/FCSA00079184.ff.fits) を使用します。

```
fo> imcopy ff/FCSA00079184.ff[501:650,6:2650] std
fo> imcopy ff/FCSA00079568.ff[501:650,6:2650] arc
```

標準星のスペクトルを抽出するためにタスク apall を使いますが、非常に多くのパラメータを持っているので、epar を使ってその確認から行います。

```
fo> epar apall
```

interac から review までは yes、ただし resize は no に設定して下さい。また、"EXTRACTION PARAMETERS" のところで backgro=fit に設定して下さい。":q" で edit parameter を終了します。

¹⁰実際の数値はもっと精度があり、fits ヘッダーに CRVAL2 や CDELTA2 として書き込まれています。

¹¹ただし、通常は background の引き算だけは自分でい (次節参照)、結果を目で確認するようにした方が良いです。

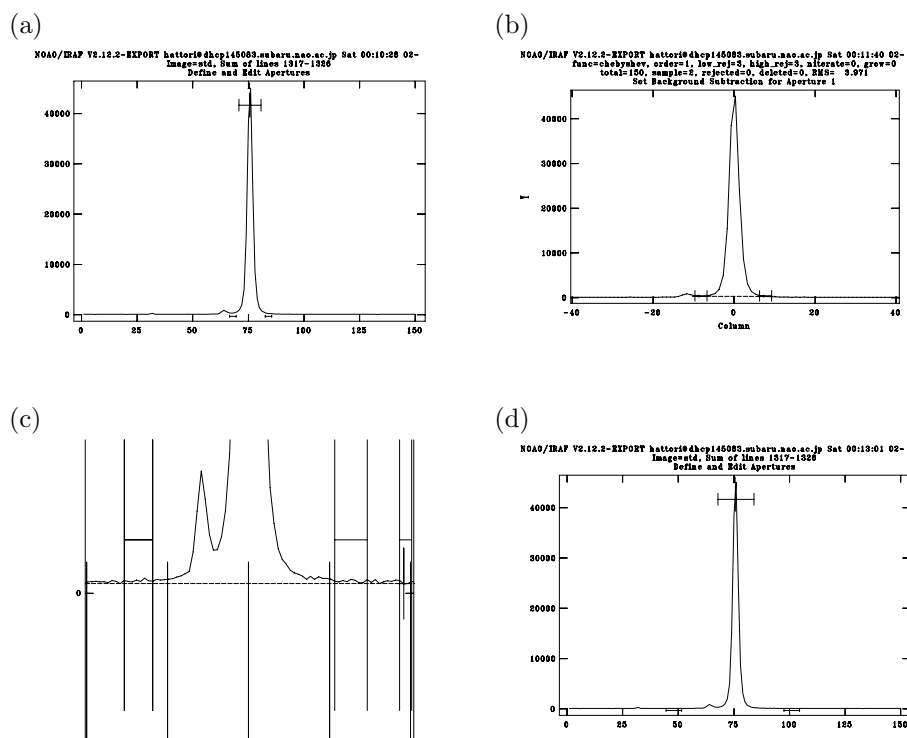


図 7: apall 画面

```
fo> apall std format=oned nfind=1
Find apertures for std? (yes):
Edit apertures for std? (yes):
```

上と同じ表示であれば、リターンキーを押し続けるだけで良いです。() の中身が違っていれば、正しい値を入力して下さい。

irafterm ウィンドウに図 7(a) のようなグラフが表示されたらスペクトル抽出の始まりです。手で調整するのはスペクトルを切り出す aperture の大きさと、background の引き算に使用する場所です。図 7(a) はスリット方向の断面図ですが、既に標準星が自動的に検出されている状態になっています。標準星のすぐ左、 $x=65$ 付近にいるのは隣の星です。今回はこの星からの影響が無視できる範囲でできるだけ aperture を広めに取りするという操作をしてみましょう。identify の時と同様なズーム操作を適宜行いながら aperture の左端にカーソルを持って行って”l”(エル) を、右端を持って行って”u”を押す事で aperture の大きさを調整できます。

aperture の大きさが決まったら、”b”を押して background 領域の調整に入ります (図 7(b))。すでに標準星の両サイドに background の設定領域が表示してありますが、この場所では標準星と隣の星からの光が入って来てしまうため、もっと離れた場所に設定しましょう。まず”t”を押して領域を初期化して下さい。次にマウスを $I=0$ 付近に持って行き、”Y”を何度か押して縦に拡大していきます (図 7(c))。background 領域を設定するには、領域の左端と右端で”s”を押します。図に示されているような形で、標準星の両サイドに設定して下さい。設定が終わったら”f”を押して fit をやりなおし、”q”で background 調整モードを終了します。

図 7(d) の状態に戻ったら”q”を押して設定を終了し、聞かれる質問には全て”yes”で答えてスペク

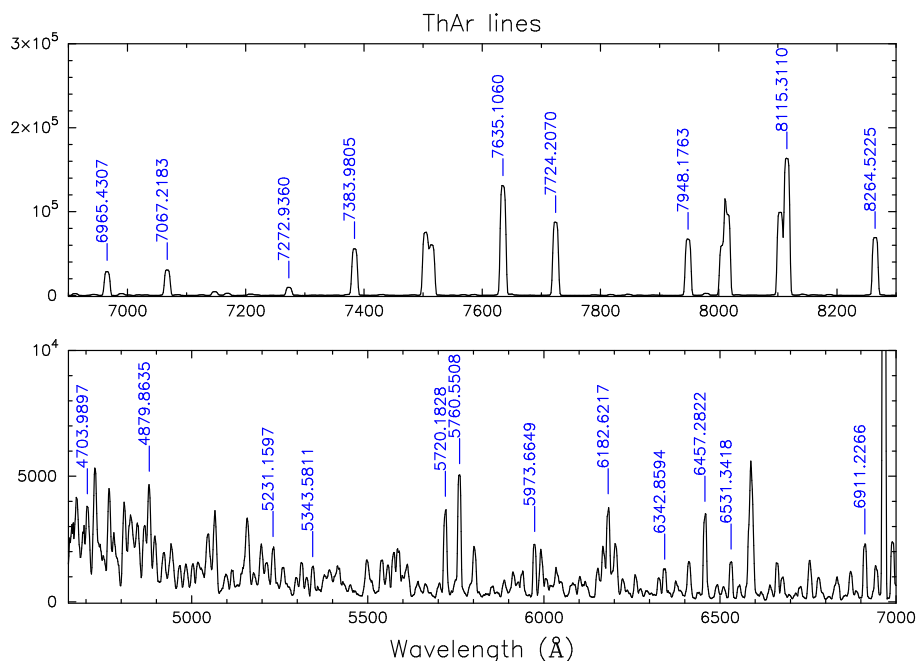


図 8: ThAr ランプ輝線の同定

トルの抽出を行います。途中で天体を trace した結果が表示されるので、order を変更して ("order 4" など) fit が正常に行われるように調整して下さい。結果は std.0001.fits という名前で出力されます。

次に標準星用の comparison を抽出します。

```
fo> apall arc format=oned reference=std find- recen- resize- edit-
      trace- backgro-
```

標準星と全く同じ抽出をしたいので、"reference=std" とし、centering 等の調整は一切しないオプションで行います。結果は arc.0001.fits に出力されます。

```
fo> identify arc.0001 coordli="FOCASREDHOME$/IRAF/thar.300b_y47.dat"
      fwidth=8 order=3 function=chebyshev
```

identify は前節と同様です。図 8 を参考に同定と fitting を行って下さい。6900Å 付近を境に急激にラインの明るさが変わっているので注意して下さい。

波長較正には、1 次元スペクトル用のパッケージ noao.onedspec の dispcor を使います。まず refspec を用いて比較光源スペクトルを指定し (ヘッダーに REFSPEC1='arc.0001' が追加される)、dispcor で波長較正を行います。この時、std.0001[* ,1,1] に抽出スペクトル、std.0001[* ,1,2] に抽出した際の sky スペクトルが含まれているので、抽出スペクトルだけを波長較正します。

```
fo> refspec std.0001 reference=arc.0001 select=match sort="" group=""
fo> dispcor std.0001[* ,1,1] std.wc
```

最後に `splot` を使って波長較正されたスペクトルを確認して下さい。

```
fo> splot std.wc
```

`splot` の使い方については後ほど (§ 6.1) 説明します。

5.6 スカイ引き

タスク `background` を用いて、波長較正した SDSS J000347 のスペクトルからスカイを引き算します。

```
fo> background obj1.wc obj1.bg axis=1 order=2 low_rej=3 high_rej=3 nit=3
```

```
obj1.wc  : 入力画像
obj1.bg   : 出力画像
axis      : スリット方向を指定する。FOCAS は x 方向なので 1
           (以下のオプションはフィッティング時に変更可能)
order     : fitting 関数の次数+1(=パラメーターの数)
low_rej   : データ点が  $\text{fit} - \text{low\_rej} \times \sigma$  より小さければ、その点を除いてフィットする
high_rej  : データ点が  $\text{fit} + \text{high\_rej} \times \sigma$  より大きければ、その点を除いてフィットする
niterate  : low_rej, high_rej で指定されるリジェクションを何回繰り返すか
```

まず、“Fit line”を聞かれますが、ここでは `fit` 領域を決定するのに用いる `y` 座標を入れます。`y` 方向のサイズは 2600 ピクセル程度になっているので、中心付近に相当する“1000 1500”や“500 2000”を指定して下さい。その後に表示されるグラフは、指定した `y` 座標の範囲で平均を取ったスリット方向断面図になっています。`apall` の時の `background` と同じ要領で“s”を使って `sky` の領域を決定して下さい (図 9)。領域設定が終わったら“f”で `fit` し、“q”で抜け、再び `line` 番号を聞かれるのをリターンキーでキャンセルして終了です。

この作業がどれだけうまく行っているかは、データの質を大きく左右します。必ず `ds9` で画像を見て問題がないかどうかを確認して下さい。ある程度うまくいった場合でも、`fitting` の次数や `fit` 領域を変えるなどして、試行錯誤を試みるのが良いと思います。画像の一部でのみうまくいっていない場合は、以下の要領で特定の `y` 座標だけやりなおす事ができます。

```
fo> background obj1.bg[*,1000:1100] obj1.bg[*,1000:1100] axis=1 order=2 ....
```

`obj2`, `obj3` についても同様にして、スカイ引きを行って下さい。

5.7 フラックス補正

まず波長較正された標準星データを使って、各波長でのカウントとフラックスの関係を求めます。タスク `standard` のパラメーターで、`extinct` と `caldir` を以下のような設定にして下さい。`input` と `output` も設定したら“go”でタスクを走らせます。

```
fo> epar standard
```

```
PACKAGE = onedspec
```

```
TASK = standard
```

```
input    =          std.wc  Input image file root name
```

NOAO/IRAF V2.14EXPORT hattori@athelas.subaru.nao.ac.jp Tue 21:37:23 28-Oct
 func=chebyshev, order=2, low_rej=3, high_rej=3, niterate=3, grow=0
 total=300, sample=95, rejected=4, deleted=0, RMS= 0.4039
 obj1.wc.fits: Fit line = 500 - 2000
 SDSSJ000347

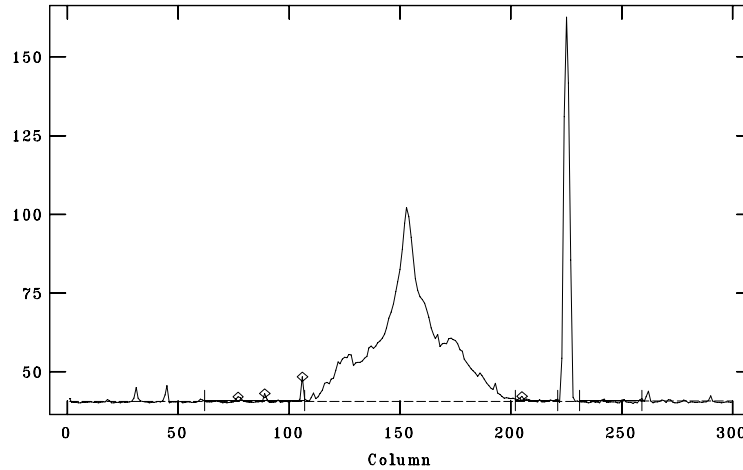


図 9: タスク background の画面。この例では x=60-110, 205-220, 230-260 が背景光を計算する領域として選択されています。

```

output =          std.dat  Output flux file (used by SENSFUNC)
(samesta=         yes) Same star in all apertures?
(beam_sw=         no) Beam switch spectra?
(apertur=         ) Aperture selection list
(bandwid=         INDEF) Bandpass widths
(bandsep=         INDEF) Bandpass separation
(fnuzero= 3.68000000000000E-20) Absolute flux zero point
(extinct= FOCASREDHOME$/IRAF/mkoextinct.dat) Extinction file
(caldir = onedstds$spec50cal/) Directory containing calibration data
(observa=        )_.observatory) Observatory for data
(interac=         yes) Graphic interaction to define new bandpasses
(graphic=         stdgraph) Graphics output device
(cursor =         ) Graphics cursor input
star_nam=         Star name in calibration list
airmass =         Airmass
exptime =         Exposure time (seconds)

```

まず星の名前を聞かれますが、何も入れずにリターンキーを押すと、“caldir”で指定したディレクトリに含まれている星の一覧が出てきます。ここでは”bd284211”を選択します。”Edit bandpasses?”と聞かれたら”yes”と答えます。

```

std.wc(1): BD+28D4211
Star name in calibration list:
Standard stars in onedstds$spec50cal/ (3200A - 8100 A)

```

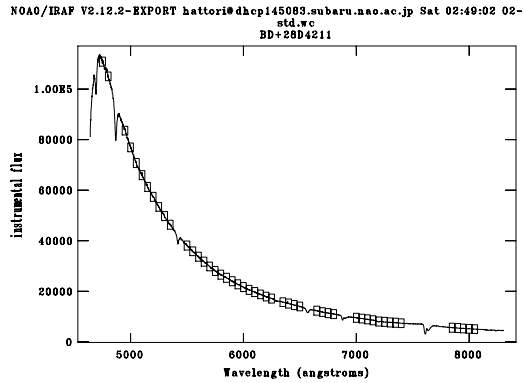


図 10: standard の画面

bd284211	eg247	feige34	hd192281	pg0205134	pg0934554	wolf1346
cygob2no9	eg42	feige66	hd217086	pg0216032	pg0939262	
eg139	eg71	feige67	hilt600	pg0310149	pg1121145	
eg158	eg81	g191b2b	hz14	pg0823546	pg1545035	
eg20	feige110	gd140	hz44	pg0846249	pg1708602	

Standard stars in onedstds\$spec50cal/ (3200A - 10200A)

bd284211	eg247	feige34	g191b2b	hz44
eg139	eg71	feige66	gd140	pg0823546
eg158	feige110	feige67	hilt600	wolf1346

Star name in calibration list: bd284211

std.wc[1]: Edit bandpasses? (no|yes|NO|YES|NO!|YES!) (no): yes

その後、図 10 のようなグラフが表示されますが、実線が観測データ (標準星のカウント) を、四角がライブラリの中で標準星のフラックスが測定されている波長とバンド幅を示しています。この際、星の吸収線 ($H\alpha$ 6563Å, $H\beta$ 4861Å 等) や大気吸収帯 (6900Å, 7600Å 付近) は”d”を押して取り除いておきます (図 10 参照)。

standard では、各バンド内で積分したカウント値を計算し、波長や対応するフラックスなどと共に結果をファイル std.dat に出力します。その測定結果に対し、タスク sensfunc を用いて fitting を行い、カウントをフラックスに変換する関数を求めます。

```
fo> sensfunc std.dat sens extinct="FOCASREDHOME$/IRAF/mkoextinct.dat"
```

”Fit paerture 1 interactively?” と聞かれたら”yes”と答えて下さい。

図 11 のようなプロットが出てきます。すでにドームフラットで割り算されているので (§ 5.3)、原理的には、この結果はドームフラットランプのスペクトルから実際のフラックス ($\text{erg s}^{-1} \text{cm}^{-2} \text{\AA}^{-1}$) への変換に対応しているはずですが、従ってここで滑らかな曲線が出てこなければ、それまでの解析に問題がある事を意味します。図 11 では、測定点をいくつか除いて (× 印)fit していま

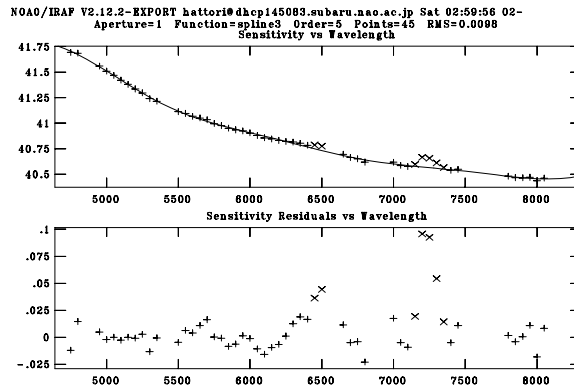


図 11: sensfunc の画面

すが、このうち 7200-7300Å 付近の盛り上がりは、ライブラリ側の問題だと思われます。結果は sens.0001.fits に出力されます。

sensfunc によって作成された sens.0001.fits を使って、オブジェクトフレームのカウントをフラックスに変換します (fluxcalib)。また、マウナケア標準の大気減光曲線を用いて大気吸収の補正を行います (extinct)。

```
fo> fluxcalib obj1.bg obj1.fc sens exposur="EXPTIME"
fo> extinct obj1.fc obj1.ex extinct="FOCASREDHOME$/IRAF/mkoextinct.dat"
```

5.8 その他

重ね合わせ

3 枚あるオブジェクトフレームを重ね合わせます。

位置合わせのやり方は色々ありますが、ここでは比較的構造が少なくてシーイング変化などの影響を受けにくい、銀河本体の連続光ピークを使う事にします。

```
fo> implot obj1.ex
:l 1300 1400
```

これで 6500Å 付近の連続光での断面図が表示されます。”X”を使って銀河中心付近を拡大し、”p”を 2 回押して中心の x 座標を測定します (図 12)。

obj2.ex と obj3.ex が obj1.ex に対して ± 9.7pixel 程度ずれているという結果になるはずですが、FOCAS の 1pixel は 0".1038 なので、このデータのビニングでは 0".3114/pixel になります。3" dither の場合 3/0.3114=9.63pixel が期待される移動量なので、大体その通りに動いていると言えます。imshift で画像を動かし、imcombine で重ね合わせます。

```
fo> imshift obj2.ex obj2.sh -9.7 0
fo> imshift obj3.ex obj3.sh 9.7 0
fo> imcombine obj1.ex,obj2.sh,obj3.sh obj.comb combine=median
```

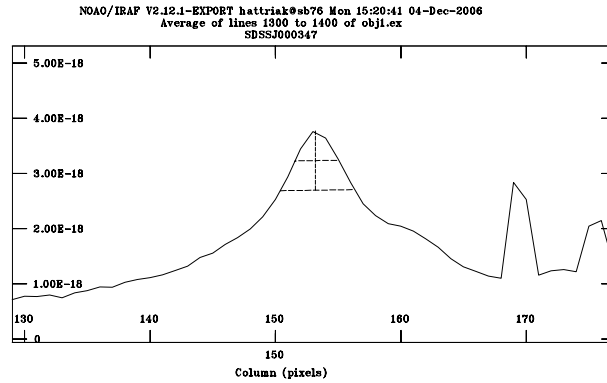


図 12: 銀河中心位置の測定。下側の破線の左端で”p”を押し、右端で再び”p”を押した結果。

ここでは cosmic ray の除去を意識して median を使っています。

星間減光

HI 21cm 線や遠赤外線放射などから求められたダスト量を用いて、銀河系内でどの程度の星間減光を受けているかを調べる事ができます。NED¹²で SDSS J000347.01-000350.3 を検索すると、この銀河の方向の星間減光が $A_V=0.106$ となっています。これを、タスク `deredden` を用いて補正します。

```
fo> deredden obj.comb obj.comb.dred 0.106 type="A(V)"
```

減光量の波長依存性は、参考文献 [1] によって求められた関数形が使われています。

helio centric velocity

低分散分光では通常問題になりませんが、観測されたデータには地球の自転や公転による赤方 (青方) 偏移が含まれています。その大きさを調べるにはタスクパッケージ `astutil` に含まれる `rvcorrect` を使います。日付や UT(Universal Time)、座標 (RA, Dec) は、ターゲットデータの FITS ヘッダーにあります。

```
fo> astutil
as> rvcorrect observa=subaru vobs=0.
# RVCORRECT: Observatory parameters for Subaru Telescope, NAOJ
#       latitude = 19:49.5302375
#       longitude = 155:28.5611198
#       altitude = 4163
Year of observation: 2006
Month of observation (1-12) (1:12): 10
Day of observation (1:31): 14
UT of observation (hours) (0.:24.): 7.6
Right ascension of observation (hours) (0.:24.): 0.063
Declination of observation (degrees) (-90.:90.): -0.064
```

¹²NASA/IPAC Extragalactic Database, <http://nedwww.ipac.caltech.edu>

##	HJD	VOBS	VHELIO	VLSR	VDIURNAL	VLUNAR	VANNUAL	VSOLAR
2454022.82208		0.00	-10.27	-10.26	0.381	0.012	-10.667	0.013

観測された視線速度が 0km/s の時 (vobs=0.)、太陽から見た視線速度 (VHELIO) が-10.27 になっているので、観測されたスペクトルを少し (10.27km/s) 青側にずらさなければいけない事になります。タスク dopcor を使って補正します。

```
as> dopcor obj.comb.dred obj.fin 10.27 isveloc+ add- disp+ flux-
```

6 その後の解析

6.1 redshift 測定

この観測の目的の一つは、この銀河の redshift を測定することでした。H α 輝線を使って測定してみましょう。ここではタスク splot を使います。

```
fo> splot obj.fin
Image line/aperture to plot (0:) (1): 153
:nsum 11
```

splot を走らせるとまずどの列を表示するか聞かれるので、銀河中心付近 (x=153) を指定しています。表示後、パラメーター nsum の値を変更して、11 列分を足し合わせたスペクトルを表示しました。H α は 7000Å 付近にあります。周辺を拡大表示して (例えば表示したい範囲の左端と右端で”a”を一回ずつ押す)、H α 輝線の左下と右下で”k”を一回ずつ押して下さい。Gauss 関数を fit した結果が表示されます。H α の静止波長 6562.8Å を使って redshift を計算して下さい。

$$z = (\lambda - 6562.8)/6562.8$$

この銀河は、SDSS でも redshift が計られています (図 13)。aperture の違い、スリット効果、fitting 誤差などの影響で正確には一致しないと思いますが、1pixel 以下の精度では一致するはずで (1pixel~1.4Å~60km/s)。自分の測定結果と SDSS の結果とを比べてみて下さい (図 13)。

splot には表 4 のように非常にたくさんの機能があります。全ての機能は書き切れないので、詳細はヘルプを参照して下さい。

6.2 1次元化、テキスト化

例えば x=153 を中心として 5pixel(=1".5) のスペクトルを抽出 (1次元化) する場合、

```
fo> blkavg obj.fin[151:155,*] obj.nucleus 5 1 option=sum
fo> imcopy obj.nucleus[1,*] obj.nucleus
```

とします。なんらかの rejection 等をかけたい場合には

```
obj.fin[151,*]
obj.fin[152,*]
obj.fin[153,*]
obj.fin[154,*]
obj.fin[155,*]
```

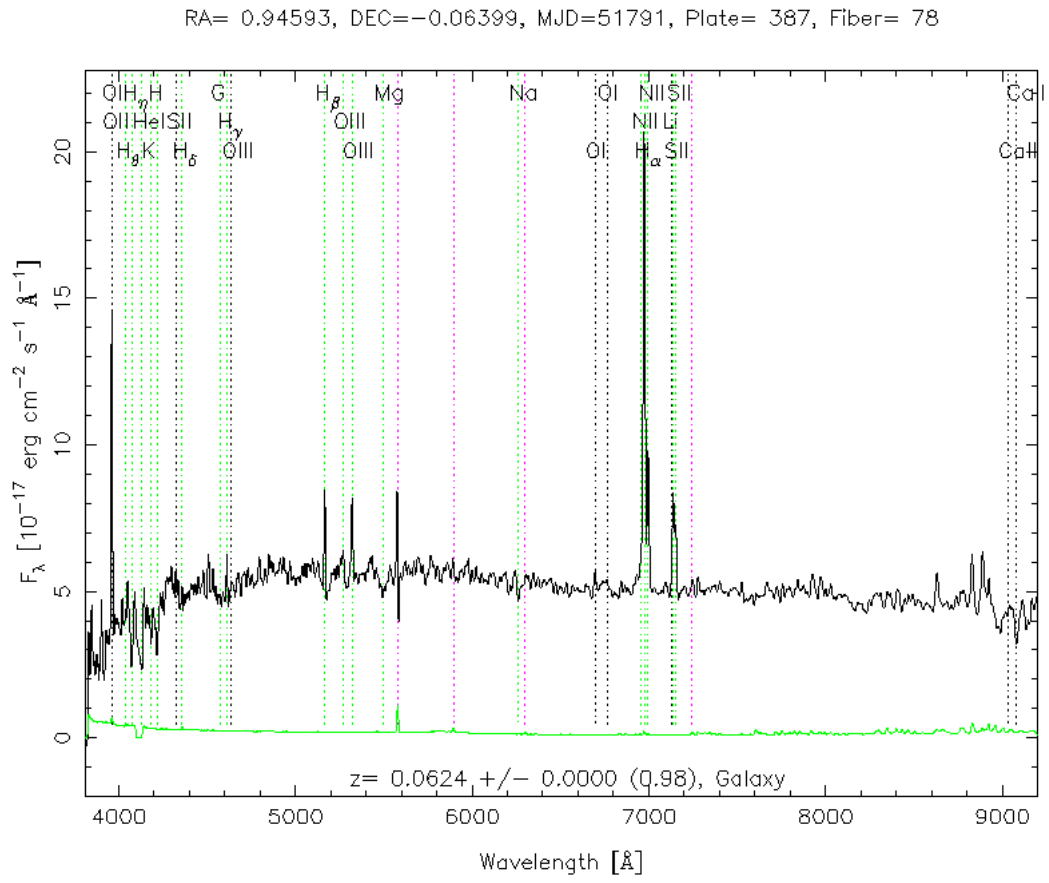


図 13: SDSS J000347.01-000350.3 の SDSS のスペクトル (黒) とそのエラースペクトル (緑)。破線で各原子の輝線、吸収線。

key	動作
a	異なる 2 箇所では”a”を押すと、カーソル位置を左右の端としたプロットが表示される。 上下のスケールは表示範囲のデータに応じて自動的に調整される。 元に戻すには、”c”を押すかカーソルを動かさずに”a”を 2 回押す。
b	現在のプロットの y 座標の最小値を 0 にする
c	表示範囲を元に戻す
d	deblending(複数のラインに対して同時に fit) モードに入る
e	”e”を 2 回押すと、カーソルに囲まれた領域の等価幅などを調べる
f	y 軸の値を変換する。対数、2 乗根、定数倍など
g	新しいファイルを読み込む
k	2 回押す事によってプロファイルフィッティングを行う。2 回目の key が l,v の場合は それぞれ Lorentzian、Voigt profile のフィッティングになる。それ以外は Gaussian。
l	縦軸が単位周波数あたりエネルギーになっているプロットを単位波長あたりのエネルギー に変換
m	2 回押す事で、その範囲の平均や分散などを測定する
n	l の逆
o	次のプロットを現在のプロットに重ねて書く
q	splot 終了
r	redraw
s	現在表示しているスペクトルにスムージングをかける
u	横軸のスケールを変換する。例えば”u”を押した後、H α にカーソルを合わせて ”d”を押し、6562.8 と入力すれば rest frame の波長で表示される。
t	現在表示されているスペクトルへの連続光フィットモードに入る。フィットした 結果を引いたり、フィットした結果で規格化する等のオプションがある。
z	カーソル位置を中心に横軸方向を拡大。縦軸のスケールは表示範囲のデータに応じて 自動的に調整される。元に戻すには”c”を押すかカーソルを動かさずに”a”を 2 回。
)	次のスペクトルに進む。今回のデータだと、x=153 から x=154 に移動という風に。
(	”)”の逆
#	何番目のスペクトルを表示するか、数字を入力して選択
-	”d”でフィットした結果を引き算する
,	短波長方向に移動
.	長波長方向に移動
?	ヘルプ表示
:nsum 3	2 次元のスペクトル画像のみ。空間方向に何 pixel 足し合わせるかを設定。 この例だと 3pixel 足し合わせる。

表 4: splot のキー操作

というファイルを用意しておいて、scombine を使うのも良いでしょう。

例えば既存のツールを使ったり、モデルスペクトル等と比較するためにデータをテキストファイルに落とすには、タスク `wspectext` を使います。

```
fo> wspectext obj.nucleus.fits obj.nucleus.dat header-
```

これによって、第一列が波長、第二列がフラックスになっているテキストデータが `obj.nucleus.dat` に出力されます。

参考文献

- [1] Cardelli et al. (1989), ApJ, **345**, 245
- [2] Kashikawa et al. (2002), PASJ, **54**, 819
- [3] Osterbrock et al. (1996), PASP, **108**,277
- [4] Osterbrock et al. (1997), PASP, **109**,614

A グリズム、フィルター、スリット ID

ID	グリズム名
SCFCGREL01	75/mm
SCFCGRLD01	150/mm
SCFCGRMB01	300B
SCFCGRMR01	300R
SCFCGRHDEC	Echelle
SCFCGRHD45	VPH450
SCFCGRHD52	VPH520
SCFCGRHD65	VPH650
SCFCGRHD68	VPH680
SCFCGRHD80	VPH800
SCFCGRHD95	VPH950

表 5: 分散素子 (グリズム) の ID。各グリズムの詳細については FOCAS HP 参照。

ID	スリット
SCFCSLLC02	0".2 center
SCFCSLLC04	0".4 center
SCFCSLLC06	0".6 center
SCFCSLLC08	0".8 center
SCFCSLLC10	1".0 center
SCFCSLLC20	2".0 center
SCFCSLLO02	0".2 offset
SCFCSLLO04	0".4 offset
SCFCSLLO06	0".6 offset
SCFCSLLO08	0".8 offset
SCFCSLLO10	1".0 offset
SCFCSLLO20	2".0 offset
SCFCMS0???	MOS マスク

表 6: スリットの ID

ID	フィルター名
SCFCFLBU01	U-band
SCFCFLBB01	B-band
SCFCFLBV01	V-band
SCFCFLBR01	R-band
SCFCFLBI01	I-band
SCFCFLN373	373nm narrow-band
SCFCFLN386	386nm narrow-band
SCFCFLN487	487nm narrow-band
SCFCFLN502	502nm narrow-band
SCFCFLN512	512nm narrow-band
SCFCFLN642	642nm narrow-band
SCFCFLN670	670nm narrow-band
SCFCFLBSZ1	SDSS z'
SCFCFLSO58	O58
SCFCFLSY47	Y47
SCFCFL600	L600
SCFCFL550	L550
SCFCFLC50	C50

表 7: フィルターの ID。各フィルターの詳細については FOCAS HP 参照。

B MOS のスペクトル切り出し –fcsmosred の使い方– β 版

IDL の `fcsmosred.pro` を用いて FOCAS/MOS で取得された複スペクトルデータを 1 つ 1 つのスペクトルに切り出す方法について解説します。この操作では切り出した後にフラットフィールドイング、distortion 補正、簡単な cosmic ray 除去を行います。切り出した後の操作（波長較正から）は基本的に long slit の解析の場合と同じです。

なお、本ソフト及び解説にはバグがある可能性があるので、もし不都合が生じた場合には適宜ご報告ください。

B.1 FOCASRED(IDL 版) のインストール

http://optik2.mtk.nao.ac.jp/subaru_red/FOCAS/FOCASRED_IDL.tar.gz

を適当な dir(以下/home/user/を仮定します) にコピーして解凍、展開して下さい。

B.2 IDL の設定

本プロセスには IDL を用います。IDL はすばるの sb マシンにはインストールされています。

- ・ FOCASRED/setup_focasred_local に例があるように、各 dir を setenv しておくとその後の作業が楽になります。

FOCASRED_DATA_IN, FOCASRED_DATA_OUT は /home/user/FOCASRED/DATA/ にしておきます。

- ・ /home/user/FOCASRED/setup_focasred を起動します。IDL の window が現れます。

注: 以下のやり方でも IDL を起動できます。

```
> idlde
```

で IDL の window が立ち上がります。

sb 系のマシンはなぜか毎回 IDL の path が切れるので毎回設定しなおす必要があります。

File>Preferences>Paths>で

```
/home/user/FOCASRED
```

```
/usr/local/rsi/idl_6.0
```

に path を切ります (チェックボックスにチェックする)。

B.3 fcsmosred 使用方法

1. wmdp_mos による sbr ファイル作成

マスクデザインの際に既に*_ccd.sbr を作成している場合はスキップして下さい。

IDL を起動し、window の一番下のコマンドラインで wmdp3 を起動します。

```
IDL> wmdp3
```

wmdp3 の使い方については以下のページ

[http://subarutelescope.org/Observing/Instruments/FOCAS/Detail/
UsersGuide/Observing/MOS/MDPman-e.html](http://subarutelescope.org/Observing/Instruments/FOCAS/Detail/UsersGuide/Observing/MOS/MDPman-e.html)

に記述されていますが、*_ccd.sbr ファイルの作り方を以下に簡単に述べます。

- ・ まず、Input Distortion Data File という window が現れますので、DistChipDat ファイルの最新のもの (2006 年 12 月 7 日現在では DistChipDat.MIT.090904.dat) を選びます。
- ・ 次に Input CCD Bad Pixel Table という window が現れるので、ここでは (一番下にある) badpixelst.dat を選びます。次に上部バーメニューの File で、マスクデザインに使用した bigimage と mdp ファイルを読み出します。
- ・ 上部メニュー Option の Grism select で観測の時に用いた grism+filter セットを選びます。左横メニューの Redraw ボタンを押すと、擬似スペクトルが現れます。
- ・ File のところで*.sbr を CCD 座標で保存します。この時、distortion 補正済みのイメージを用いている場合は、CCD Dist. Corrected Img を選択、distortion 補正していないイメージを用いている場合は CCD Dist. UN-Corrected Img を選択します。Save の段階で FOV 中心の座標を尋ねられますので、デザイン時の FOV 中心を入力します。これで wmdp3 は終了しましょう。

注: 実際の分光取得 MOS データは実際のデザインとマスク alignment の際に生じるオフセットによりずれています。これが気になる場合は、CCD 座標で保存する場合に、適宜 FOV 中心座標の x にオフセットをはかせます。*_ccd.sbr ファイルをそのまま awk など edit して

もよいですし、wmdp.mos 上で FOV の中心位置をずらして*_ccd.sbr ファイルを save し直すことによって作成できます。

2. 分光 bigimage 作成

バイアス補正をした flat、comparison、object の MOS 画像の bigimage(2 枚の CCD 画像をつなぎ合わせて 1 枚の画像にする)を作成します。

この際、distortion 補正のオプション (wmdp.mos 上の distortion 補正は imaging データについてのものであって分光データのものではないため) は選ばないように気をつけてください。

```
IDL> focasmenu
```

で、"combine chip1&2"を選択します。処理したい 2 枚のファイルを選びます。

3. fcsmosred による分光イメージ切り出し

IDL を起動し、

```
IDL> fcsmosred '切り出したデータを保存するディレクトリ名'  
              '用いるファイルを記述したファイル名 (フルパスで)'  
              'CCD 座標の sbr ファイル名 (フルパスで)'
```

という書式で、下の起動例のように fcsmosred を動作させると、あとは自動的にすべてのスペクトルを切り出してくれます。

「用いるファイルを記述したファイル」とは

```
flat1.fits comp1.fits obj1.fits /home/user/FOCASRED/DATA/DistChipDat.MIT.090904.dat 1  
flat2.fits comp2.fits obj2.fits /home/user/FOCASRED/DATA/DistChipDat.MIT.090904.dat 1  
flat3.fits comp3.fits obj3.fits /home/user/FOCASRED/DATA/DistChipDat.MIT.090904.dat 1
```

のように、flat フレーム、comparison フレーム、object フレームのファイル名と、distortion データ名 (必ずフルパスで)、グリズムの番号 (1 は 300B) を記述したファイルです。comparison に sky line を使う場合は comparison フレーム=object フレームで良いです。複数セットを一度に切り出したい時には各行ごとに 1 セット、それを数行にわたって書けば OK です。

起動例:

```
IDL> fcsmosred, '/work/user/test/', '/work/user/test/fcsmosred.dat', '/work/user/test/test.sbr'
```

この結果、/mb21b/user/test のディレクトリ下に obj#_*.fits, comp#_*.fits というように object/comparison のデータセットができあがります。obj の後の番号#は sbr ファイル上の行番号に相当します。この後 sky line を用いて波長較正を行う場合は comp#_*.fits は必要ありません。処理が終わると "Program caused arithmetic error" というメッセージが現れますが、正常終了です、気にしないでください。

この後 IRAF での波長較正に移る場合には、作業に入る前に

```
cl> wcsreset * world
```

を実行してください。

途中でエラーが生じた場合:(TBD)

B.4 演習

以下のデータを使って実際に MOS スペクトルデータを解析してみよう。

http://optik2.mtk.nao.ac.jp/subaru_red/FOCAS/mostest_20061207.tar.gz

をコピーして解凍、展開してください。以下のデータが含まれています。

test.obj[a,b,c].bs_[1,2].fits	オブジェクトデータ
test.flat.bs_[1,2].fits	オブジェクトのドームフラットデータ (imcombine 済)
test.feige34.bs.fits	標準星 (星の名前: Feige34) データ
test.feige34.dome.bs.fits	標準星のドームフラットデータ (imcombine 済)
test.arc.bs.fits	comparison データ
test.mdp	MDP ファイル (マスクデザインファイル)
test.image.fits	マスクデザインの際に用いた direct image

観測: 300R グリズム+O58 フィルター

FOV 中心 (X, Y) = (1950, 2001)

注 1: *_1.fits は chip1、*_2.fits は chip2 のデータ

注 2: 簡単のため今回のデータはすべてバイアス・オーバースキャンが差し引かれて (解析マニュアル §3.4 参照) います。またドームフラットデータは 5 枚のデータが imcombine されて (§3.5 参照) います。

long slit と異なるところ

- chip1 と chip2、2 枚の CCD データを使います。
- 標準星 (long slit で取得) のドームフラットとオブジェクト (MOS で取得) のドームフラットは異なります。
- 一般的に long slit と比べると空間方向のピクセル数が少ないです。パラメーターを注意深く選ばないとオブジェクトを失くしてしまったり、余計なノイズを混入させてしまったりします。画像上で確認しながら作業を進めましょう。
- long slit の演習の時とはグリズム (波長域) も違うことに注意してください。

問題

初級編 : オブジェクト NO=23 を解析してみましょう。楕円銀河に特徴的な吸収線が見えてくるはずです。何の吸収線かわかりますか? 赤方偏移はいくつでしょう?

上級編 1: オブジェクト NO=1 を解析してみましょう。z=5.7 のライマン α 銀河が映っています。連続光はほとんど見えず、ある狭い波長範囲にしかシグナルが検出されません。どこにシグナルがあるかわかりますか?

上級編 2: オブジェクト NO=9 を解析してみましょう。z=6.5 のライマン α 銀河が映っています。連続光は全く見えません。どこにシグナルがあるかわかりますか?

C flat field スペクトルの規格化

§ 5.3 では、ドームフラットの規格化は定数による割り算のみを行いました。しかしフラット用光源のスペクトルが一様で *featureless* とは限らず、フラットを行う前にフラット画像自身のスペクトルパターンを取り除きたい場合があります。

FOCAS の場合の例として、図 14 に 300R グリズムと O58 フィルターで取得したドームフラットスペクトルを示します。赤い矢印で示されているように、反射板による吸収パターンが見えています。このフラット画像で割り算すると、これらの波長では天体のスペクトルに *excess* が生じてしまいます。これは、その後のフラックス補正などで取り除く事も原理的には可能ですが、別の方法として、フラットスペクトルの大局的なパターンを予め取り除いておくやり方があります。

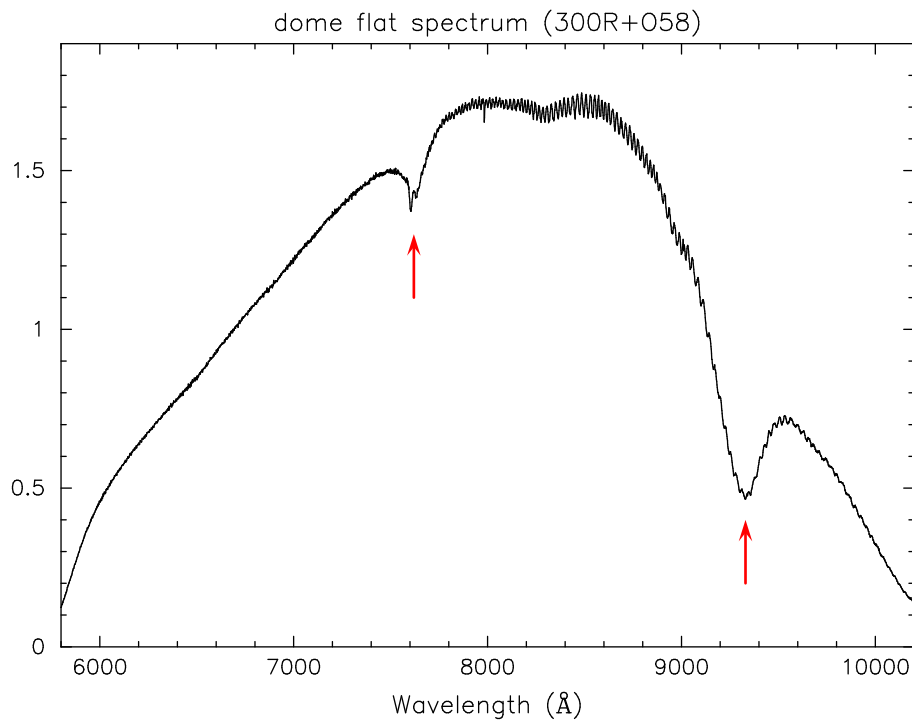


図 14: 300R グリズムと O58 フィルターで取得したドームフラットスペクトル。赤い矢印は、ドームフラット反射板による吸収パターン。

C.1 response を用いた規格化

この節ではタスク *response* を使って規格化する方法を紹介します。

```
fo> epar response
```

```
PACKAGE = longslit
```

```
TASK = response
```

```
calibrat=
```

```
Longslit calibration images
```

```

normaliz=          Normalization spectrum images
response=          Response function images
(interac=          yes) Fit normalization spectrum interactively?
(thresho=          INDEF) Response threshold
(sample =          *) Sample of points to use in fit
(naverag=          1) Number of points in sample averaging
(functiono=        spline3) Fitting function
(order =           1) Order of fitting function
(low_rej=          0.) Low rejection in sigma of fit
(high_re=          0.) High rejection in sigma of fit
(niterat=          1) Number of rejection iterations
(grow =            0.) Rejection growing radius
(graphic=          stdgraph) Graphics output device
(cursor =          ) Graphics cursor input
(mode =            q)

```

例として、§ 5.3 で作った `flat.fits` を規格化してみましょう。

```

fo> response flat[201:500,5:2700] flat[431:500,5:2700] flat.nr2
      low_rej=3 high_rej=3 niter=3

```

```

flat[201:500,5:2700] : 入力画像
                      flat.fits の一部分を入力画像として与えています。
                      ここで指定された部分以外の領域では、出力画像のピクセル値は 1.0 になります。
flat[431:500,5:2700] : 規格化するスペクトルを求める画像
                      ここでは入力画像の一部を使います。
                      flat.fits の左下には光学系によるケラレの影響が見えていますが、(図 16)
                      そのケラレの影響がない領域を選んでいきます。
flat.nr2              : 出力画像名

```

上記のコマンドを実行すると、`x=431-500` の範囲で平均を取った `flat.fits` が表示されます (図 15a)。

”k”を押すとフィットした関数に対する比が表示されます (図 15b)。この表示でどの程度うまくフィットできているのかを確認しながら、次数を調整して下さい。これまで使ってきたタスクと同じように、フィットの次数は”order”で変更できます。表示を元に戻すには”h”を押します。

また、`y=500` 付近の凹みに対して fit を行うのは困難なので、”s”を使って fit する領域から `y=500` 付近を取り除きます (図 15c)。フラットスペクトルのパターンを消すために、ここでは大きめの次数 (50) を使用しています。

`response` を終了するには”q”を押して下さい。こうして求められた関数で各 column (`x=201` から `500`) を割り算した結果が、`flat.nr2.fits` として出力されます。この結果 `flat.nr2.fits` は、ピクセル値が 1 付近で、細かいピクセルごとの感度むらやスリット方向の感度むらを保持した画像になっています (図 16)。

C.2 別のアプローチ

もう一つの方法として、ここでは `focared` 内のタスク、`flatnorm` を使用します。`response` との違いは、関数のフィットを行わずに、指定された領域で平均したスペクトルをそのまま使用する

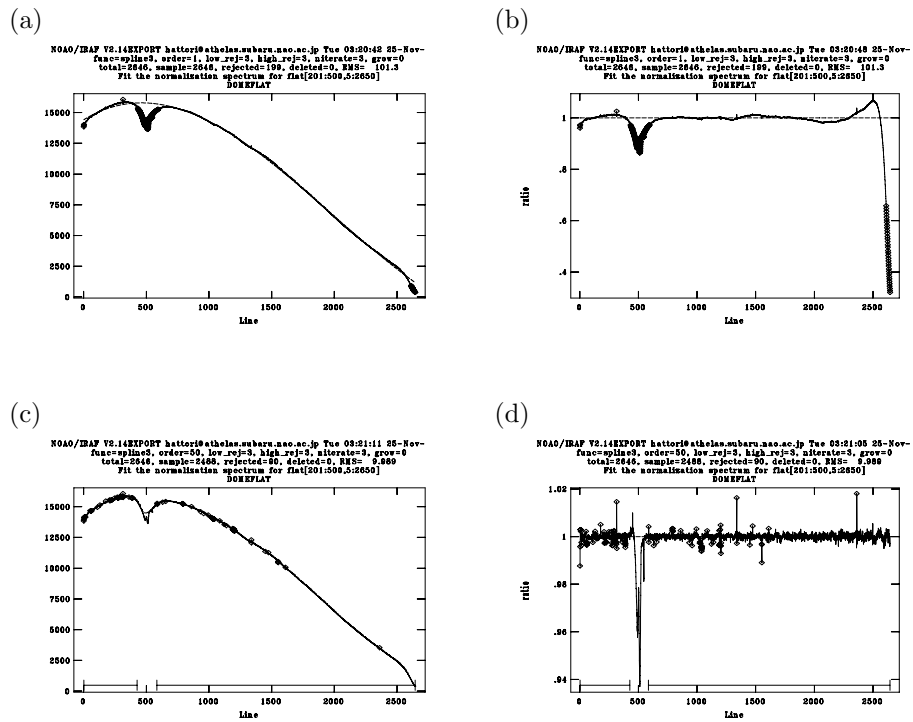


図 15: (a)response の初期画面。(b)”k”を押して縦軸をフィットに対する比に変更した画面。(c,d) フィットする範囲や次数を変更した結果。

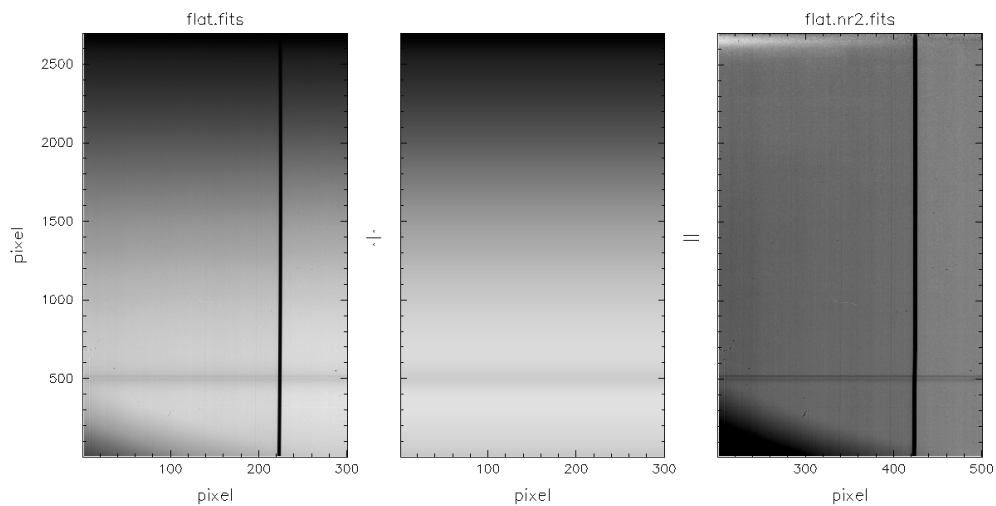


図 16: 分光用 flat field 画像の規格化

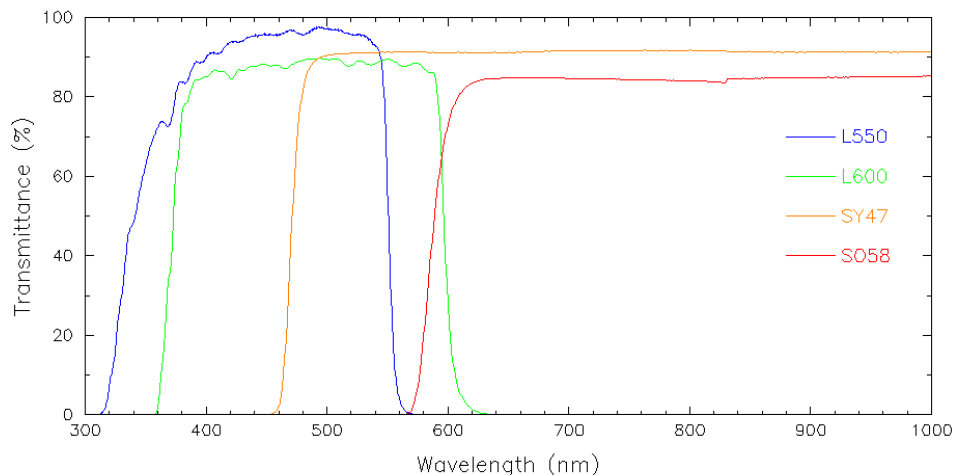


図 17: FOCAS で使用している次数分離フィルター

点です。

この方法は response と比べると、フィットの次数などの調整が必要ない、 $x=500$ 付近の吸収のような複雑なパターンでも簡単に補正できる、などといった利点があります。一方で、フリンジのように本来であれば補正せずにそのまま残しておくべきパターンを、(部分的に) 補正してしまう場合があるという欠点があります。

```
fo> imcopy flat[201:500,5:2700] flat.tr
fo> flatnorm flat.tr flat.nr3 x1=231 x2=300
```

上の例ではまず必要な領域だけを取り出してから、flatnorm を使って規格化しています。前節の結果と比べてみて下さい。

C.3 指針

ここでは、FOCAS のログスリット分光観測について、flat field スペクトルの規格化をどのように行うべきかについて解説します。

図 17 は、FOCAS で使用している次数分離フィルターの透過曲線を示しています¹³。

短波長側の観測に使用される L550 や L600 は干渉フィルターで、透過率に数%の凸凹 (リップルパターン) があります。これらのフィルターを使った場合、このリップルパターンは天体のスペクトルにも乗っているはずなので、フラットフィールドの過程で補正されるべきものです。従って、この節で行ったような規格化によって消されるのは望ましくありません。一方で、すばる望遠鏡のドームフラットは $< 7000 \text{ \AA}$ では単調なスペクトルをしています。つまり、L550 や L600 を使って行うような短波長側の分光観測では、装置の感度に補正すべきパターンがあるのに対し、フラットスペクトルには邪魔なパターンがないことになります。このような場合、flat field 画像の規格化

¹³<http://www.naoj.org/Observing/Instruments/FOCAS/Detail/UsersGuide/PartsDetail/Filters/FilterList.html>

は定数による割り算だけ (§ 5.3) で行うのが良いと言えます。また Y47 や O58 を使った観測でも、フィルターの効率が変化する $< 4900 \text{ \AA}$ や $< 6100 \text{ \AA}$ を精度よく解析したい場合は、同様のやり方をするのが良いと考えられます。

Y47 や O58 フィルターは長波長側の観測に使用されますが、カットされる波長 (それぞれ $4700 \text{ \AA}$, $5800 \text{ \AA}$) の周辺を除けば、ほぼ様な透過率をしています。またすでに図 14 で見たように、 $> 7500 \text{ \AA}$ ではドームフラット自身がスペクトルにパターンを持っています。これらの波長域を観測する場合は、この節で行ってきたように、関数フィットなどを使用してフラットスペクトルを規格化する方が解析が簡単になるでしょう。

Source	: SDSS J084119.51+290504.4
Slit	: 0".8 center
Grism	: 300 R (SCFCGRMR01)
Filter	: O58 (SCFCFLSO58)
Exposure Time	: 1200 sec $\times$ 3
Binning	: 3 pixels along spatial axis, no-bining along wavelength axis

Frame No.	Object Category
FCSA00072705–714	Dome flat
FCSA00072735, 736	Standard star (G191B2B)
FCSA00072737, 738	Comparison
FCSA00072849, 850	SDSS J084119.52+290504.4
FCSA00072899–902	SDSS J084119.52+290504.4

1. copy fits files from /home/hattriak/qso/
2. the target is at x~580
3. no bias frames (only overscan subtraction)
4. distortion correction is not necessary
5. normalize the domeflat spectrum using the task **response** as described in Appendix B
6. wavelength calibration should be performed for each exposure
7. reference : Goto 2006, MNRAS, 371, 769