

SOT-SP MEKSY フィッティングプログラム  
マニュアル & Level-2データの使い方

MEKSY チーム

Ver. 0.7-J

# 目次

第0章 改版履歴	3
第1章 はじめに	4
1.1 この文書について	4
1.2 Level-2データとは	4
1.3 ソフトウェアのありか	4
第2章 設定	6
2.1 コマンド探索パス	6
第3章 IDLによるデータ読み込みと表示	7
3.1 sotsp用のidlの立ち上げ	7
3.2 Level-2データの読み込み	7
3.3 mapへの変更	8
3.4 ベクトル磁場表示	8
第4章 IDLによるフィッティング	9
4.1 概要	9
4.2 基本プロシジャ	9
4.3 PCクラスターによる並列実行	9
4.4 フィッティングプログラムにわたすパラメータ	11
第5章 コマンドラインによるフィッティング	12
5.1 基本	12
5.1.1 準備：入力ファイルの編集	12
5.1.2 前処理	12
5.1.3 初期推定値の作成	13
5.1.4 フィッティングの実行	13
5.1.5 後処理	13
5.2 オプションの与え方	14
5.3 PCクラスターによる並列実行	15

付録 A	16
A.1 フィッティングパラメータ	16
A.2 フィット属性の意味	17
A.3 実行コマンドのオプション	17
A.3.1 共通	17
A.3.2 <code>sotsp_invprep</code>	18
A.3.3 <code>sotsp_mksigma</code>	19
A.3.4 <code>sotsp_guesspk</code>	19
A.3.5 <code>sotsp_fitme</code>	21

## 第0章 改版履歴

- Ver. 0.0, 2007. 1. 10. 横山央明
- Ver. 0.1, 2007. 1. 25. 下条圭美
- Ver. 0.2, 2007. 2. 2. 勝川行雄
- Ver. 0.3, 2007. 2. 3. 横山央明
- Ver. 0.4, 2007. 2. 15. 勝川行雄
- Ver. 0.5, 2007. 3. 5. 横山央明
- Ver. 0.6, 2013. 2. 13. 下条圭美 for SDAS
- Ver. 0.7, 2018. 1. 26. 下条圭美 for ADC

# 第1章 はじめに

## 1.1 この文書について

このマニュアルは、ひので SOT/SP 観測について、Level-2 データの作成と解析とについて述べたものである。

## 1.2 Level-2 データとは

Level-2 データは、Level-1 データから、以下の作業をおこなったものである。

- 波長 Stokes プロファイル=フィット (Milne-Eddington モデルにもとづく)
- 磁場方位角の 180 度不定性に対する推測

なお、Level-1 データは観測 Stokes プロファイルにキャリブレーションを施したものの、Level-0 データが生データ (をフォーマット変換したもの)。

## 1.3 ソフトウェアのありか

国立天文台 ADC 共同利用データ解析システムの以下のディレクトリにデータやソフトウェアが置かれる。MEKSY パッケージ用ディレクトリーは

```
/solar/share/soft_develop/sot/sp/sotsp_inv
```

であり、その下に以下のディレクトリーが配置されている。

bin/ 実行形式コマンドやシェルスクリプト

data/ データ

idl/ IDL プロシージャ

doc/ ドキュメント

setup/ IDL 設定ファイル

src/ フィッティングプログラム等のソースコード

## 第2章 設定

MEKSY パッケージを利用する為には、以下の設定が必要である。

注：ADC システム上ではデフォルトにて MEKSY パッケージの設定されているため、以下の作業は不要である。

### 2.1 コマンド探索パス

.cshrc ファイルなどでコマンド探索パスを設定する。たとえば以下のような行を書き加える。

```
set path = (/solar/share/soft_develop/sot/sp/sotsp_inv/bin ${path})
```

注：ADC システム上ではデフォルトで MEKSY パッケージの設定されているため、この作業は不要である。

## 第3章 IDLによるデータ読み込みと表示

### 3.1 sotsp用のidlの立ち上げ

SDASではSSWに対して適切な拡張設定を行ったSSWidlコマンドが用意されている。このSSWidlを起動する。

```
sagami$ SSWidl
```

### 3.2 Level-2データの読み込み

データは、拡張FITSファイルなのでreadfitsなどを使えば読めるが、専用のプロシジャを使うのが便利である。

```
IDL> dir='/solar/share/soft_develop/sot/sp/sotsp_inv/data/'
IDL> file=dir+'level2/2006/12/11/sprst20061211_031010_007342.fits'
IDL> sotsp_read_rst,file,index,data,idata,coordinates,time,slitpos $
IDL> ,par_type=par_type,par_unit=par_unit $
IDL> ,ipar_type=ipar_type,ipar_unit=ipar_unit,straylight=straylight
% READFITS: EOF encountered attempting to read extension 42
```

注：「EOF encountered ...」の警告メッセージは無視してだいじょうぶ。  
ここでデータの中身を説明する。配列dataとidataとがデータで、

```
IDL> help,data,idata
DATA          FLOAT      = Array[998, 512, 4]
IDATA         LONG       = Array[998, 512]
```

前者が浮動小数点型のもの（磁場強度など）、後者が整数型のもの（フィット属性など）。  
配列の添え字の意味は次のとおり



- 1 番目：スキャン方向（基本的には太陽の東西方向）
- 2 番目：ピクセル方向（基本的には太陽の南北方向）
- 3 番目：フィッティングパラメータ

プロシジャ `sotsp_read_rst` には、「`read_level`」というキーワードが用意されている。この値を変えることで読み込むデータの種類を調整できる。

- 「0 (デフォルト)」: 磁場強度・磁場仰角・磁場方位角・迷光比率とフィット属性
- 「1」: 上に加えて、Doppler 速度 (6301.5A ラインと 6302.5A ラインとそれぞれ)、Doppler 幅、ライン/連続光オパシティ比、減衰幅/Doppler 幅比、放射源泉関数、放射源泉関数勾配、マクロ乱流幅、迷光 Doppler シフト、連続光強度、偏光率
- 「2」: Level-2 データに含まれる全情報。上に加えて、各フィット値の誤差幅、フィットの評価関数値 (カイ 2 乗)、Stokes-V 概要値などを含む。

### 3.3 map への変更

SSW の `map` 構造体を作るには

```
IDL> sotsp_index2map,index,data,map,par_type,par_unit $
IDL> ,idata,imap,ipar_type,ipar_unit
```

### 3.4 ベクトル磁場表示

磁場 3 成分を表示するには以下のようにすればよい。

```
IDL> sotsp_mkmag3map,map[0],map[1],map[2],mapbx,mapby,mapbz
IDL> plot_map,mapbz
IDL> plot_vmap,/over,mapbx,mapby $
IDL> ,limit=100,scale=0.002,iskip=10,jskip=10 $
IDL> ,/sample,index_size=1000
```

オプション「`limit`」で表示する最小強度、「`scale`」で矢印の大きさ、「`iskip`」「`jskip`」で出力点間隔、「`sample`」「`index_size`」で見本矢印 (右上に表示される) が調整できる。

## 第4章 IDLによるフィッティング

### 4.1 概要

IDLによるフィッティングは以下の手順で行う。実際の実行コマンドは Fortran プログラムであり、IDLからそれらの中から実行する。ユーザー自身は意識する必要がない(はずである)。

### 4.2 基本プロシジャ

以下のようにおこなう。ユーザーが入力として与える情報は、観測データ (Level-1) のファイルリストのみ。

注:ここではフォアグラウンドジョブとして実行する手順を説明する。しかしそのまま実行すると、とてつもなく時間がかかる。実際には、並列実行するべきなので、次項を参照のこと。

```
IDL> files_prof=findfile('./SP3D*.fits')
IDL> sotsp_invprep,files_prof,outfile_stray=file_stray ; 前処理
IDL> sotsp_fit,files_prof,outfiles_fitparams=file_level1_5 ; フィッティング
IDL> sotsp_write_rst,file_level1_5,$
IDL> files_prof,file_stray,outfile=file_level2 ; 後処理
IDL> sotsp_mkq1,file_level2 ; (必要なら)早見 PNG 画像の作成
```

### 4.3 PC クラスタによる並列実行

**ADC 共同利用計算機システムでは、この方法は利用不可能である。**

sotsp\_fit の手続きは、とてつもなく時間がかかる。旧 HSC のマシン 1CPU (x86 3GHz) で 1 ピクセルあたり、0.05 秒ぐらい。1k x 1k の画像であれば、5 万秒つまり 14 時間かかる。なので画像を処理するときは以下の並列処理を行う。

```

IDL> files_prof=findfile('./SP3D*.fits')
IDL> sotsp_invprep,files_prof,outfile_stray=file_stray
    ; ここまでは基本と同じ
IDL> sotsp_fit,files_prof,/gridengine $
IDL> ,outfiles_fitparams=files_fitparams ; 並列フィッティング

```

ここで並列ジョブが投入される。プロンプトがすぐ戻ってくるが、うしろでジョブが走っている。UNIX コマンド bjobs で実行状態を確認できる。

```

IDL> $ bjobs
JOBID  USER   STAT  QUEUE          FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
1980   shimojo RUN    all_japan_     sagami     solar-pc1  MEKSY_p000 Feb 14 08:54
1982   shimojo RUN    all_japan_     sagami     solar-pc1  MEKSY_p002 Feb 14 08:54
1981   shimojo RUN    all_japan_     sagami     solar-pc2  MEKSY_p001 Feb 14 08:54
1983   shimojo RUN    all_japan_     sagami     solar-pc2  MEKSY_p003 Feb 14 08:54
1984   shimojo PEND   all_japan_     sagami                 MEKSY_p004 Feb 14 08:54
1985   shimojo PEND   all_japan_     sagami                 MEKSY_p005 Feb 14 08:54
1986   shimojo PEND   all_japan_     sagami                 MEKSY_p006 Feb 14 08:54
1987   shimojo PEND   all_japan_     sagami                 MEKSY_p007 Feb 14 08:54
1988   shimojo PEND   all_japan_     sagami                 MEKSY_p008 Feb 14 08:54
1989   shimojo PEND   all_japan_     sagami                 MEKSY_p009 Feb 14 08:54
1990   shimojo PEND   all_japan_     sagami                 MEKSY_p010 Feb 14 08:54
1991   shimojo PEND   all_japan_     sagami                 MEKSY_p011 Feb 14 08:54
1992   shimojo PEND   all_japan_     sagami                 MEKSY_p012 Feb 14 08:54
1993   shimojo PEND   all_japan_     sagami                 MEKSY_p013 Feb 14 08:54
1994   shimojo PEND   all_japan_     sagami                 MEKSY_p014 Feb 14 08:54
1995   shimojo PEND   all_japan_     sagami                 MEKSY_p015 Feb 14 08:54

```

というようなメッセージがでて、3つめの項目「STAT」のどれかが「RUN(実行中)」になれば、無事にフィッティングのプログラムが走っている。

数分～十数分後、ふたたび bjobs を実行し、

```

IDL> $ bjobs
No unfinished job found

```

のような表示がでたら、並列ジョブが全て終了している。並列ジョブが「完全に終了したのち」に以下の後処理を行う。

```
IDL> sotsp_merge_fmd,files_fitparams, $
IDL> outfile=file_level1_5 ; 複数ファイルの合体
IDL> sotsp_write_rst,file_level1_5,files_prof, $
IDL> file_stray,outfile=file_level2
IDL> sotsp_mkql,file_level2
```

sotsp\_merge\_fmd は、並列処理で分割して出力された結果ファイルをマージするための手続き。

#### 4.4 フィッティングプログラムにわたすパラメータ

前処理やフィッティングのプログラムへの入力パラメータの値を デフォルトから変更するには以下のようにする。

```
IDL> sotsp_fit,files_prof,verbose=1,i_slit1=3,i_slit2=5, $
IDL> /flag_output_syn_prof
```

これは一例で、出力メッセージの詳しさの度合いを 1 に設定すると、フィッティング実行の範囲をスリット番号 3 番から 5 番の範囲に指定し、フィットした波長 Stokes プロファイルを出力する、という設定を行っている。

また次のオプションもある。

```
IDL> sotsp_fit,/norun, ...
; Fortran コマンドへの入力ファイルを作るだけで実行しない。
```

```
IDL> sotsp_fit,/gridengine,nprocs=16,queue='all-japan-short', ...
; PC クラスターの並列実行で、プロセスを 16 並列とし、
; キュー「all-japan-short」に投入する。
```

```
IDL> sotsp_fit,/gridengine,/show_message
; PC クラスターへのジョブ投入後、実行を監視するメッセージを出力する。
```

## 第5章 コマンドラインによるフィッティング

コマンドラインによるフィッティングは以下の手順で行う。

1. 準備：入力ファイルの編集
2. 前処理。迷光プロファイルとノイズ=プロファイルとの作成
3. 初期推定値の作成
4. フィッティング
5. 後処理：磁場方位角推定とフォーマット変換

以下順に説明する。

### 5.1 基本

#### 5.1.1 準備：入力ファイルの編集

入力ファイルは以下。

`datalist.txt` : 観測 Level-1 データのリスト

1行に Level-1 ファイルひとつ。( スキャンスリットデータ1列分になっているはず、gzip 圧縮されたもので構わない。)

なお、以下説明する各プログラムの動作を制御する各種パラメータがあり、それをプログラムにわたすこともできる。設定ファイルをそれぞれに 作る必要があり、詳細は後述する。

#### 5.1.2 前処理

フィッティングに使うための、迷光プロファイル、重み付け、ノイズ=プロファイルとを作る。コマンドは「`sotsp_invprep`」と「`sotsp_mksigma`」とを順に使う。実行は以下のように行う。

```
sagami% sotsp_invprep  
sagami% sotsp_mksigma
```

### 5.1.3 初期推定値の作成

フィッティングの初期値の推定を行う。コマンドは「sotsp\_guesspk」を使う。実行は以下のように行う。

```
sagami% sotsp_guesspk
```

結果、作業ディレクトリに

```
guess.fits
```

というファイルができあがる。初期推定値が格納されている。

### 5.1.4 フィッティングの実行

前節で得られた初期推定値を使ってフィッティングを実行する。コマンドは「sotsp\_fitme」を使う。実行は以下のように行う。

```
sagami% sotsp_fitme
```

結果、作業ディレクトリに

```
fitparams.fits
```

というファイルができあがる。フィット値が格納されている。

### 5.1.5 後処理

フィッティング作業のあと、次の後処理をおこなう。

- データの再フォーマット（ひので SOT/SP の Level-2 フォーマットに変換）
- 磁場方位角の 180 度不定性に対する推測

作業には IDL を使う。

```
IDL> files_prof=rd_text('datalist.txt')
IDL> sotsp_write_rst,'fitparams.fits',files_prof,'stray.fits'
```

結果、作業ディレクトリに

```
sprst.fits
```

というファイルができあがる。ひので SOT/SP の Level-2 フォーマットによる、フィット値が格納されている。

## 5.2 オプションの与え方

フィッティングに際してデフォルト値と異なるオプションを与えるには、入力ファイルを以下のように作成する。ファイル名は任意でよい。実行は、たとえば

```
sagami% sotsp_invprep input_invprep.txt
satuma% sotsp_mksigma input_mksigma.txt
sagami% sotsp_guesspk input_guesspk.txt
sagami% sotsp_fitme input_fitme.txt
```

とする。

中身は、たとえば以下のようなフォーマット。それぞれのキーワードの意味は付録 A.2 で詳述する。

```
sagami% cat input_guesspk.txt
verbose: 2
i_slit1: 300
i_slit2: 320
i_pixel1: 250
i_pixel2: 270
file_datalist: dlist000.txt
file_sigma: sigma000.fits
file_stray: stray000.fits
file_guess: guess000_p000.fits
```

### 5.3 PCクラスターによる並列実行

datalist.txt の中身を分割し、別々のファイルにする。たとえば4プロセス並列で実行するのであれば、おおざっぱに4分割すればよい。それぞれの分割ファイルにたいして上記の内容を実行する。ファイル名が重複しないようにするか、ディレクトリを分けて実行すればよい。sotsp\_fitmeの結果生成されたフィットパラメータファイルをIDLで合体したあと、フォーマット変換する必要がある。

```
IDL> files=[ ..., ... ]
IDL> sotsp_merge_fmd,files ; ファイル合体
IDL> files_prof=rd_text('datalist.txt')
IDL> sotsp_write_rst,'fitparams.fits',files_prof,'stray.fits'
```

ここで files には、フィットパラメータファイル名をいれる。



# 付録 A

## A.1 フィッティングパラメータ

注：磁場仰角は、太陽から観測者に向かうほうがゼロ度注：磁場方位角は、右側（ $x$ 軸正方向、一般に太陽の西向）がゼロ度

fieldstrength:	磁場強度（単位 G）
inclination:	磁場仰角（単位度）
azimuth:	磁場方位角（単位度）
linestrength:	ライン/連続光オパシティ比
dopplerwidth:	Doppler 幅（単位 mÅ）
damping:	減衰幅/Doppler 幅比
dopplervelocity:	Doppler 速度（単位 km/s）
sourcefunct:	放射源泉関数
sourcegrad:	放射源泉関数勾配
macroturbulence:	マクロ乱流（単位 km/s）
strayfraction:	迷光比率
strayshift:	迷光 Doppler シフト（単位 mÅ）

弱偏光ピクセルでは以下のパラメータだけがフィットされる。これ以外はゼロに固定。

linestrength:	ライン/連続光オパシティ比
dopplerwidth:	Doppler 幅（単位 mÅ）
damping:	減衰幅/Doppler 幅比
dopplervelocity:	Doppler 速度（単位 km/s）
sourcefunct:	放射源泉関数
sourcegrad:	放射源泉関数勾配
macroturbulence:	マクロ乱流（単位 km/s）

## A.2 フィット属性の意味

- 0: フィッティング成功
- 1: ユーザー指定(マスクファイル)によりフィッティングを実行していない
- 2: フィッティング失敗
- 3: 波長プロファイル異常または初期値推測失敗
- 4: 波長プロファイル異常または初期値推測失敗
- 5: 明度不足
- 10: 弱偏光率ピクセル専用フィッティング成功
- 12: 弱偏光率ピクセル専用フィッティング失敗

## A.3 実行コマンドのオプション

オプションは、実行コマンドごとの入力ファイル中に記述する。

注意1: コメントアウトは未実装

注意2:「:」の前に「空白」をいれてはならない。

### A.3.1 共通

()内がデフォルト値

```

verbose:          出力メッセージの詳細度 ( 0 )
file_datalist:   Level-1 データリストのファイル名 ( datalist.txt )
file_weight:    データ点の重み付けのファイル名 ( weight.fits )
file_sigma:     計測誤差 ( シグマ ) のファイル名 ( sigma.fits )
file_stray:     迷光波長プロファイルのファイル名 ( stray.fits )
file_guess:     推測初期値のファイル名 ( guess.fits )
file_fitparams: フィット結果値のファイル名 ( fitparams.fits )
flag_usemask:   マスクを使うか? Yes なら 1. ( 0 )
file_mask:     マスク ( フィッティングを行わない点指定 ) ファイル名 ( mask.fits )
wl1_pdeg:      偏光率計算用の波長範囲 ( 単位 mÅ )( 6302.3d3 )
wl2_pdeg:      偏光率計算用の波長範囲 ( 単位 mÅ )( 6302.7d3 )
wl1_cct:       連続光計算用の波長範囲 ( 単位 mÅ )( 6302.9d3 )
wl2_cct:       連続光計算用の波長範囲 ( 単位 mÅ )( 6303.1d3 )
cct_lim:       明度の最低しきい値 ( 単位 DN )( 1000. )
i_pixel1:     フィッティングするピクセル座標 ( y 方向 ) 下限
i_pixel2:     フィッティングするピクセル座標 ( y 方向 ) 上限
i_slit1:      フィッティングするスリット座標 ( x 方向 ) 下限
i_slit2:      フィッティングするスリット座標 ( x 方向 ) 上限
fixpdegfit:   偏光率下限値を固定するか? Yes なら 1. ( 0 )
pdeg_fit:     フィッティング実行の偏光率下限値。fixpdegfit が 1 の
              時しか使わない。( 0.002 )
sigma_fit:    フィッティング実行の偏光率下限値をシグマから判断する
              ときのファクター。「この値 × シグマ」が下限値 ( 2 )
n_zerointpixel_lim: プロファイル異常を判断するパラメータ。
                  明度がゼロのピクセルがこの数を超えると異常と判断。( 0 )
linedepth_lim: プロファイル異常を判断するパラメータ。
                  Stokes-Iでのライン深さがこの値より浅いと異常と判断。( 1.e-5 )

```

### A.3.2 sotsp\_invprep

() 内がデフォルト値

```
weight_i:      IQUV 間での重み付けの相対比 ( 0.1 )
weight_q:      IQUV 間での重み付けの相対比 ( 1.0 )
weight_u:      IQUV 間での重み付けの相対比 ( 1.0 )
weight_v:      IQUV 間での重み付けの相対比 ( 1.0 )
wl1_63015:     重みを与える波長範囲 ( 単位 mA ) 6301.5A 付近の下限值 ( -350 )
wl2_63015:     重みを与える波長範囲 ( 単位 mA ) 6301.5A 付近の上限値 ( +350 )
wl1_63025:     重みを与える波長範囲 ( 単位 mA ) 6302.5A 付近の下限值 ( -350 )
wl2_63025:     重みを与える波長範囲 ( 単位 mA ) 6302.5A 付近の上限値 ( +350 )
flag_telluric: 地球大気起源の波長範囲を取り除くか? ( 0 )
pdeg_stray:    迷光成分に加えるデータの偏光率の上限値 ( 0.002 )
```

### A.3.3 sotsp\_mksigma

() 内がデフォルト値

```
fixsigma:      計測誤差を固定値にするか? Yes なら 1 を与える。 ( 0 )
sigma_value:   計測誤差の固定値。fixsigma が 1 の時に使う。 ( 0.001 )
noise_photon:  計測誤差計算パラメータ。光子雑音分。 ( 0.02 )
noise_readout: 計測誤差計算パラメータ。読出雑音分。 ( 8.4 )
```

### A.3.4 sotsp\_guesspk

() 内がデフォルト値

wl1\_fit: フィッティング波長範囲(単位 mA) 6302.5A 付近の下限值( -350 )  
 wl2\_fit: フィッティング波長範囲(単位 mA) 6302.5A 付近の上限值( +350 )  
 pdeg\_b1: 偏光率が pdeg\_b1 より小さなピクセルでは、磁場強度の上限値を  
 field\_b1: field\_b1 に設定する。( 0.1, 3000 )  
 pdeg\_b2: 偏光率が pdeg\_b2 より小さなピクセルでは、磁場強度の上限値を  
 field\_b2: field\_b2 に設定する。( 0.03, 2000 )  
 field\_tryagain: フィットした磁場強度がこの値より大きい時は、  
 もう一度やりなおしてみる。カイ 2 乗のよい方を採用する。( 2000 )  
 n\_pointspergeneration: 遺伝的アルゴリズムの各世代ごとの個体数( 20 )  
 n\_generations: 遺伝的アルゴリズムの世代数( 40 )  
 floormin\_fieldstrength: 磁場強度(単位 G) 下限値( 500 )  
 floormax\_fieldstrength: 上限値( 5000 )  
 floormin\_inclination: 磁場仰角余弦( -1 )  
 floormax\_inclination: ( +1 )  
 floormin\_azimuth: 磁場方位角(単位度)( 0 )  
 floormax\_azimuth: ( 180 )  
 floormin\_linestrength: ライン/連続光オパシティ比( 5 )  
 floormax\_linestrength: ( 30 )  
 floormin\_dopplerwidth: Doppler 幅(単位 mA)( 20 )  
 floormax\_dopplerwidth: ( 50 )  
 floormin\_damping: 減衰幅/Doppler 幅比( 0 )  
 floormax\_damping: ( 2 )  
 floormin\_dopplervelocity: Doppler 速度(単位 km/s)( -10 )  
 floormax\_dopplervelocity: ( +10 )  
 floormin\_sourcefunct: 放射源泉関数( 0 )  
 floormax\_sourcefunct: ( 1 )  
 floormin\_sourcegrad: 放射源泉関数勾配( 0 )  
 floormax\_sourcegrad: ( 1 )  
 floormin\_macroturbulence: マクロ乱流(単位 km/s)( 0 )  
 floormax\_macroturbulence: ( 3 )  
 floormin\_strayfraction: 迷光比率( 0.01 )  
 floormax\_strayfraction: ( 0.99 )  
 floormin\_strayshift: 迷光 Doppler シフト(単位 mA)( -40 )  
 floormax\_strayshift: ( +40 )

### A.3.5 `sotsp_fitme`

() 内がデフォルト値

