

# MAXI データ解析の手引き

GSC/SSC Software/Calibration Team  
Archive Team

2015/07/28

Ver.1.2.6

## 更新履歴

日付	バージョン	改訂者	コメント
2012/05/25	1.0		release ver.1.0
2012/06/05	1.0.1		maxiutil mxsscrmfspec bug fix
2013/05/23	1.1	杉崎	release ver.1.1
2013/07/01	1.2.0	中川	解析準備 (§3) を改訂し、configure バージョンを用いたコンパイル・インストールの手順を追記した。
2013/07/23	1.2.1	中川	§2 の一部を改訂した。ユーザーからのコメントに基づき §3 を改訂した。
2013/08/01	1.2.2	中川	§1 の解析フロー図を差し替えた。§2、§3 を改訂した。§3 の一部を §A へ移動した。標準解析に関わる記述を §4 へ統合して、改訂した。
2013/08/02	1.2.3	中川	§2 の GSC の DP Mode の記述を整理した。§3 と §A において HEAsoft の記述を修正した。
2013/10/26	1.2.4	中川	§3 に追記した。用語の統一のために全体的に修正した。§4.3.4 の SSC のデフォルトのパラメータを修正した。
2015/03/02	1.2.5	中川	xselect.mdb へ追記する、MAXI:wmapcoord のデフォルトを「DETECTOR」から「SKY」へ変更した。
2015/07/28	1.2.6	中川	apt-get、yum のパッケージ名に関する情報を 3.1 節に追記した。

# 目 次

<b>第 1 章 Introduction</b>	<b>3</b>
1.1 目的	3
1.2 MAXI のデータ処理と科学解析の流れ	3
<b>第 2 章 データファイルの種類</b>	<b>4</b>
2.1 GSC	4
2.1.1 Event File の DP Mode	4
2.1.2 Raw Event File	4
2.1.3 Processed Event File	4
2.1.4 Cleaned Event File	4
2.1.5 HK file	5
2.1.6 STDGTI	5
2.2 SSC	5
2.2.1 Cleaned event file	5
2.3 Auxiliary file	5
2.3.1 Time (DPTC)	5
2.3.2 Attitude	5
2.3.3 Orbit	5
2.3.4 MKF	6
2.3.5 ISSANC	6
2.3.6 GSC-E FRC	6
2.4 CALDB file	6
<b>第 3 章 解析準備</b>	<b>7</b>
3.1 外部ソフトウェア・ライブラリのインストール	7
3.2 MAXI ソフトウェアのインストール	12
3.3 CALDB のインストール	14
3.4 GSC RMF Database の取得	16
3.5 Cleaned Event File と Auxiliary File の取得	17
3.6 セットアップスクリプト	18
<b>第 4 章 標準解析</b>	<b>20</b>
4.1 天体周辺の Event File の抽出	20
4.2 Image File の生成、表示	22
4.3 ライトカーブ解析	23
4.3.1 バックグラウンドレベルの見積り方法の決定	23
4.3.2 観測条件ファイルの生成	24
4.3.3 ライトカーブ生成中間ファイルの作成	26
4.3.4 ライトカーブファイルの作成	30

4.3.5	ライトカーブのプロット	32
4.4	スペクトル解析	34
4.4.1	ソース、バックグラウンドのエネルギースペクトル (FITS 形式) の抽出	34
4.4.2	Exposure weight map の生成	34
4.4.3	ソーススペクトル用の RMF の生成	36
4.4.4	モデルフィッティング	38
4.5	周期解析	39
4.5.1	天体周辺の Event File の抽出	39
4.5.2	バリセントリック補正	39
4.5.3	パワースペクトル解析	39
4.5.4	周期探索	40
4.5.5	パルス波形の導出	40
付 録 A	オプション・ソフトウェア	43
A.1	外部ソフトウェアのインストール	43
A.2	MAXI ソフトウェアのインストール	44
付 録 B	役に立つツール	46
B.1	時刻変換	46
B.2	Processed Event File の再処理	46
付 録 C	位置決め	47
C.1	Event File の取得	47
C.2	位置決め手順	48
C.3	位置決め結果のチェック	50

# 第1章 Introduction

## 1.1 目的

本文書は MAXI メンバーがデータ解析を始めるに当り、必要なデータ及び解析ソフトウェアの情報を記したものである。理研で行っている変動監視ターゲットの解析プロセスを基にしている。2013 年 5 月現在 も、解析ソフトウェアは頻繁に更新されている状態なので、始める前に Wiki<sup>1</sup> などで最新の情報を確認すること。

## 1.2 MAXI のデータ処理と科学解析の流れ

日々の天体監視解析でのプロセスの概要を図 1.1 に示した。ASCA, Suzaku などの X 線衛星データで用いられている Headas software の枠組みを採用している<sup>2</sup>。MAXI のソフトウェアでは、イベントデータ処理や応答関数など MAXI 固有の問題に対応して標準解析に乗せるために必要なツールを開発している。

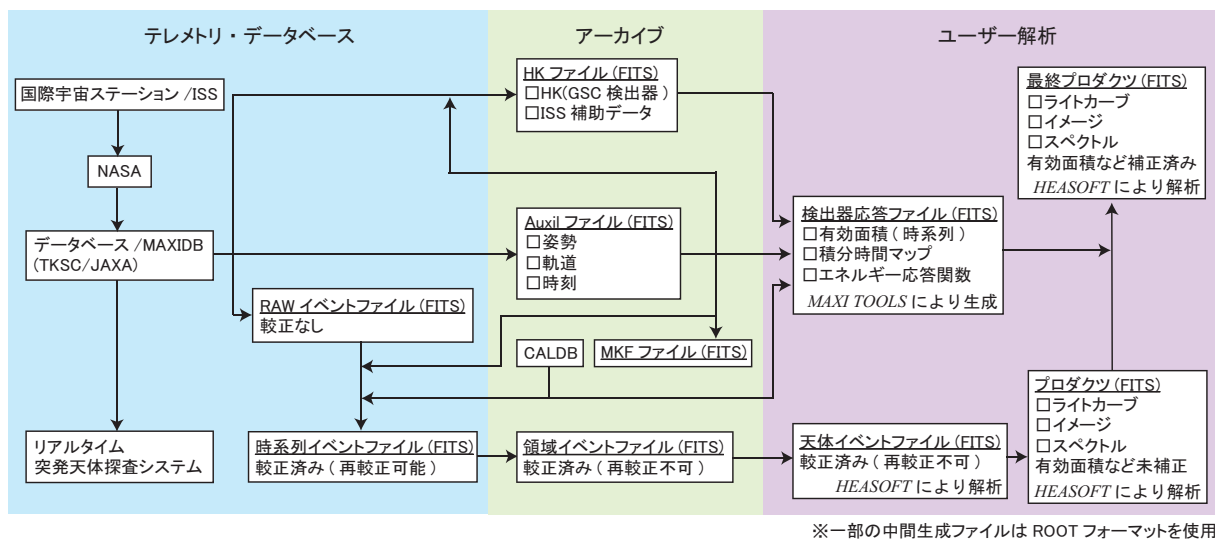


図 1.1: MAXI のデータの流れ、および解析フローを示す。

<sup>1</sup>[http://www.maxi.jaxa.jp/pm/mission\\_team/pukiwiki/index.php?MaxiSoft%2FStandardAnalysis](http://www.maxi.jaxa.jp/pm/mission_team/pukiwiki/index.php?MaxiSoft%2FStandardAnalysis)

<sup>2</sup>現在は開発ツールとして ROOT Libray、ROOT 形式のファイルを使っている部分があるが、最終的には除く予定である。

## 第2章 データファイルの種類

本章では、MAXI の標準解析で用いるデータファイルについて述べる。

### 2.1 GSC

#### 2.1.1 Event File の DP Mode

DP Mode に応じて 32-bit と 64-bit の 2 種類の形式がある。両者の違いは時刻のタグ付けの精度であり、32-bit は 1s、64-bit は FRC(GSC-MDP の  $50\mu\text{s}$  Free-Run Clock Counter) があるため  $50\mu\text{s}$  である。2009 年 9 月 30 日までは低速系は 32-bit、中速系は 64-bit となっていたが、それ以降は基本的に低速系、中速系ともに 64-bit である。ただし、再起動の直後は低速系が 32-bit になっていることがある。なお、低速系と中速系では Software LD/UD が異なる。

64-bit からは 32-bit へ変換できる。§3.5 に記載してある理化学研究所に置いてある Cleaned Event File は、2009 年 9 月 30 日以前のデータとの整合性を重視して、64-bit から 32-bit へ変換したものである。

#### 2.1.2 Raw Event File

MAXI-DB から Rev(Raw Event) データテーブルを抽出し、fits binary table へ変換したファイルである。camera, anode, dptc, pha(L,R) などテレメトリに含まれる生データのみで構成される。camera 番号には A 系 B 系の区別がない。

Raw Event File の内容はすべて後段の Processed Event File に引き継がれるので、通常のユーザー解析で Raw Event File まで戻ることはない。

#### 2.1.3 Processed Event File

Raw Event File に検出器の校正データ (CALDB) と補助データ (Auxiliary data: DPTC, Attitude, ...) を使って時刻付け、座標付け、PI 付け処理を行って TIME, detx, bex, RA, DEC, X, Y, PI などの column を付けたもの。cameraID は 0-11 の通し番号になっている。

CALDB、Auxiliary Data のバージョンに依存する。再処理プログラム mxgevt2evt で最新 CALDB による Processed Event File に更新できる。

RIKEN では Raw Event File と同じ切り分けで生成した Processed Event File と、カウンター毎に分割して一日毎にまとめたファイルを生成している。

#### 2.1.4 Cleaned Event File

Processed Event File に HK を使って Filter をかけたものである。かけている条件は以下の通りである。

- 端カット ( $|\text{DETX}| < 135.0\text{mm}$ )
- HV-on/off cut ( $15.0\text{ sec} < \text{T\_HV\_ON} \ \&\& \ 15.0\text{ sec} < \text{TN\_HV\_OFF}$ )

これは緩い条件なので、標準解析ではきつい Filter をかけた方が多い場合が多い。Daily LC Analysis の出発点として使われている。

### 2.1.5 HK file

GSC power, HV monitor, LD count monitor, RBM monitor, count/RBM flags などの情報を集めたファイルで、xselect のイベント選別で使う。データ欠損がない限り、1 秒ステップである。

追加情報として、T\_HV\_ON(HV-on からの時間)、TN\_HV\_OFF(HV-off までの時間) を加えている。

RIKEN の標準プロセスで日々生成している。

### 2.1.6 STDGTI

Cleaned Event File の GTI Extension を抽出して 1 ファイルまとめたものである。通常の日体解析では、Event File に付随した STDGTI Extension HDU を使う。長期間の全天データを扱うなど特殊な用途向け。

RIKEN の標準プロセスで日々生成している。

## 2.2 SSC

### 2.2.1 Cleaned event file

GRADE Format、PHA Format の 2 種類の Format がある。

## 2.3 Auxiliary file

ISS の軌道、姿勢、MAXI DP 時刻クロックなど、観測データの解析の時に必要になる補助データからなる。\$MXSOFT/auxil のディレクトリに置く。

### 2.3.1 Time (DPTC)

DP 時刻補正ファイル (DPTIME と UTC の差分) で、JAXA で日々生成している。

### 2.3.2 Attitude

MAXI の姿勢データのファイルで、JAXA で日々生成している。

### 2.3.3 Orbit

ISS の軌道データのファイルで、JAXA で日々生成している。

### 2.3.4 MKF

時刻毎の ISS の緯度、経度、高度、地球磁場の Cutoff-rigidity、昼夜の情報を記したファイルで、(xselect の) イベント選別で使う。

Attitude, Orbit, DPTC file から生成される。

イベント選別パラメータファイルとしては HK file との切り分けの取り決めはしていないが、MKF では GSC counter の情報に寄らず、SSC と共用していくことを想定している。

MAXI ソフトウェアの mxmkf を使って生成する。JAXA でも日々生成している。

### 2.3.5 ISSANC

ISS 補助ファイルの太陽パドル角のデータを抽出したもの。 $\alpha$  角と  $\beta$  角用で 2 種類。視野遮蔽判定を行う program で使う。

RIKEN の標準プロセスで日々生成している。

### 2.3.6 GSC-E FRC

GSC MDP 20kHz(50 $\mu$ s) free-run clock counter の DPTime 正秒時の数値のリストで、A 系 B 系で別である。64bit event data の処理 (時刻付け) で使う。

RIKEN の標準プロセスで日々生成している。

## 2.4 CALDB file

Heasarc CALDB(Calibration database) のフォーマットに基づいて作られている、観測装置の校正情報のファイル。GSC、SSC チームがそれぞれ管理、提供している。



## 第3章 解析準備

本章では解析を始めるために必要なソフトウェア・ライブラリのコンパイル・インストール (§3.1, §3.2)、各種データのインストール・取得 (§3.3, §3.5)、環境変数の設定 (§3.6) について順番に説明する。ここでは、C シェル環境を例にして説明するので、Bourne シェルの場合は、適宜、読み替えて頂きたい。

### 3.1 外部ソフトウェア・ライブラリのインストール

本節では表 3.1 に示す外部ソフトウェア・ライブラリのコンパイル・インストールの手順を説明する。MAXI-DB からイベントデータを直接に取得したい場合、つまり mxdbtool を使いたい場合は、§A.1 を参考にして、Postgresql をインストールする。また、mxkwtool を使う場合は、§A.1 を参考にして、ROOT のインストールの前に fftw、gsl、HEAsoft 6.12 をインストール、また ROOT の configure の際にオプションを追加する。ここでは、ダウンロードしたソースファイルは “/opt/maxi/src” に置き、作業は “/opt/maxi/local\_20130505/install\_work” で行い、“/opt/maxi/local\_20130505” にインストールするとして説明する。ただし、healpix のみ §3.2 の MAXI ソフトウェアと同じ “/opt/maxi/mxsoft\_20130505” にインストールするとして説明する。

tcl-devel, tk-devel, readline-devel, pcre, pcre-devel, openssl, openssl-devel, libtool

apt-get、yum を使用した場合のインストールの例を示す。Ubuntu などディレクトリビューションによっては “devel” ではなく、“dev” とする必要があるので、パッケージ名に注意する必要がある。

```
$ apt-get install tcl-devel tk-devel readline-devel pcre pcre-devel openssl openssl-devel libtool
$ yum install tcl-devel tk-devel readline-devel pcre pcre-devel openssl openssl-devel libtool
```

§3.1 と §3.2 のソフトウェアのコンパイルにおいて、configure を実行する環境によっては、libtool に関連した以下のようなエラーが出る場合がある。

```
checking build system type... Invalid configuration ‘/opt/maxi/local_20130505/
install_work/heasoft-6.12/x86_64-unknown-linux-gnu-libc2.12/lib’: machine
‘/opt/maxi/local_20130505/install_work/heasoft-6.12/x86_64-unknown-linux-gnu-libc2.12/lib’
not recognized configure: error: /bin/sh ./config.sub /opt/maxi/local_20130505/install_work/
heasoft-6.12/x86_64-unknown-linux-gnu-libc2.12/lib failed
```

このエラーが出た場合には、libtool がインストールされていない可能性がある。インストールされている場合は、以下のファイルを configure が存在するディレクトリへコピーして、configure を実行する。

```
$ cp /usr/share/libtool/config/config.guess .
$ cp /usr/share/libtool/config/config.sub .
```

**python 2.7.3** – <http://www.python.org/ftp/python/2.7.3/Python-2.7.3.tgz>

コンパイル (make) を行った後に以下のようなメッセージが表示されるので、メッセージに続くリストの中に “\_ssl” が無いことを確認する。“\_ssl” が表示されている場合は、openssl と openssl-devel がインストールされていない可能性があるので、インストールを行った後に再び python のコンパイルを行う。

```
Python build finished, but the necessary bits to build these modules were not found:
```

```
bsddb185          dl                gdbm
imageop           sunaudiodev
```

```
$ cd /opt/maxi/local_20130505/install_work
$ tar zxvf /opt/maxi/src/Python-2.7.3.tgz
$ cd Python-2.7.3
$ ./configure --prefix=/opt/maxi/local_20130505 --enable-shared
$ make
$ make install
$ pwd
/opt/maxi/local_20130505/install_work/Python-2.7.3
$ cd ../
$ setenv PATH /opt/maxi/local_20130505/bin:$PATH [CSH の場合]
$ setenv LD_LIBRARY_PATH /opt/maxi/local_20130505/lib:$LD_LIBRARY_PATH [CSH の場合]
$ export PATH=/opt/maxi/local_20130505/bin:$PATH [BASH の場合]
$ export LD_LIBRARY_PATH=/opt/maxi/local_20130505/lib:$LD_LIBRARY_PATH [BASH の場合]
```

**numpy 1.6.2** – <http://sourceforge.net/projects/numpy/files/NumPy/1.6.2/numpy-1.6.2.tar.gz/download>

```
$ tar zxvf /opt/maxi/src/numpy-1.6.2.tar.gz
$ cd numpy-1.6.2
$ /opt/maxi/local_20130505/bin/python setup.py build
$ /opt/maxi/local_20130505/bin/python setup.py install
$ pwd
/opt/maxi/local_20130505/install_work/numpy-1.6.2
$ cd ../
```

**pyfits 3.1** – <http://pypi.python.org/packages/source/p/pyfits/pyfits-3.1.2.tar.gz>

pyfits のコンパイルの際に以下のようなエラーが出る場合には openssl と openssl-devel がインストールされていない可能性があるので、インストールを行った後にコンパイルを行う。

```
urllib2.URLError: urlopen error unknown url type: https
```

```
$ tar zxvf /opt/maxi/src/pyfits-3.1.2.tar.gz
$ cd pyfits-3.1.2
$ /opt/maxi/local_20130505/bin/python setup.py build
$ /opt/maxi/local_20130505/bin/python setup.py install
$ pwd
```

表 3.1: 外部ソフトウェア、ライブラリの一覧を示す。

ソフトウェア	バージョン	コメント
tcl-devel	–	yum や apt-get でインストール
tk-devel	–	yum や apt-get でインストール
readline-devel	–	yum や apt-get でインストール
pcre, pcre-devel	–	yum や apt-get でインストール, swig に必要
openssl, openssl-devel	–	yum や apt-get でインストール, python(pyfits) に必要
libtool	–	configure を使用するために必要
python	2.7.3	–
numpy	1.6.2	pyfits に必要
pyfits	3.1	python fits library
swig	2.0.3	maxiutil に必要
Postgresql	8.1 or 8.2	mxdbtool に必要。§A.1 を参照。
fftw	3.3.2	mxkwtool に必要。§A.1 を参照。
gsl	1.15	mxkwtool に必要。§A.1 を参照。
HEAsoft	6.13, 6.12	6.12 は mxkwtool に必要。§A.1 を参照。
ds9	7.1	–
ROOT	v5.34.07	ライトカーブ補正スクリプトで使用
Healpix	3.11	全天 map で使用

```
/opt/maxi/local_20130505/install_work/pyfits-3.1.2
$ cd ../
```

**swig 2.0.3** – <http://sourceforge.net/projects/swig/files/swig/swig-2.0.3/swig-2.0.3.tar.gz/download>

```
$ tar zxvf /opt/maxi/src/swig-2.0.3.tar.gz
$ cd swig-2.0.3
$ ./configure --prefix=/opt/maxi/local_20130505
--with-python=/opt/maxi/local_20130505/bin/python --without-python3
$ make
$ make install
$ pwd
/opt/maxi/local_20130505/install_work/swig-2.0.3
$ cd ../
```

**HEAsoft 6.13** – <http://heasarc.gsfc.nasa.gov/docs/software/lheasoft><sup>1</sup>

HEAsoft はソースからインストールを行う。またパッケージは、少なくとも ASCA、Xspec を選択すれば良い。make をしている時に python に関するエラーが出る場合は、前述の python の環境変数が設定されているかを確認する。

```
$ tar zxvf heasoft-6.13src.tar.gz
$ cd heasoft-6.13/BUILD_DIR $ ./configure >& config.out
$ make >& build.log
```

<sup>1</sup>もし新バージョンがリリースされた場合、”[ftp://heasarc.gsfc.nasa.gov/software/lheasoft/lheasoft6.13](http://heasarc.gsfc.nasa.gov/software/lheasoft/lheasoft6.13)” から取得できる。ここに  
ある heasoft-6.13 を wget などコピーするか、heasoft-6.13src.tar.gz が用意されていればダウンロードする。

```
$ make install >& install.log
```

ROOT のコンパイルのために、以下のように環境変数を設定し、シンボリックリンクを作成する。以下の“x86\_64-unknown-linux-gnu-libc2.12”は使用する PC 環境によって変わるので、適宜、読み替えて頂きたい(以後、同様)。

```
$ setenv HEADAS /opt/maxi/local_20130505/install_work/heasoft-6.13/x86_64-unknown-linux-gnu-libc2.12
[CSH の場合]
$ source $HEADAS/headas-init.csh [CSH の場合]
$ export HEADAS=/opt/maxi/local_20130505/install_work/heasoft-6.13/x86_64-unknown-linux-gnu-libc2.12
[BASH の場合]
$ source $HEADAS/headas-init.sh [BASH の場合]
$ cd $HEADAS/lib
$ ln -s libcfitsio_3.32.so libcfitsio.so
$ pwd
/opt/maxi/local_20130505/install_work/heasoft-6.13/x86_64-unknown-linux-gnu-libc2.12/lib
$ cd ../../..
```

xselect で MAXI のデータを扱うために、以下の MAXI Mission のパラメータの情報を”\$HEADAS/bin/xselect.mdb”に記述する。

```
!
! MAXI
!
MAXI:submkey          NONE
MAXI:instkey          INSTRUME
MAXI:dmodekey          DATAMODE
MAXI:mkf_def_expr      mx*.mkf*
MAXI:mkf_rel_dir       .
MAXI:time             TIME
MAXI:tunits            s
MAXI:binsize           16.
MAXI:x                 X
MAXI:y                 Y
MAXI:xsiz              TLMAX
MAXI:ysiz              TLMAX
MAXI:detx              DETX
MAXI:dety              DETY
MAXI:detxsiz           TLMAX
MAXI:detysiz           TLMAX
MAXI:rawx              NONE
MAXI:rawy              NONE
MAXI:rawxsiz           NONE
MAXI:rawysiz           NONE
MAXI:phamax            TLMAX
MAXI:gti               STDGTI
MAXI:events            EVENTS
MAXI:timeorder         yes
!MAXI:instruments      GSC SSC
MAXI:instruments       GSC GSC_0 GSC_1 GSC_2 GSC_3 GSC_4 GSC_5 GSC_6 GSC_7 GSC_8 GSC_9 GSC_A GSC_B SSC SSC_H SSC_Z
MAXI:spbn              1
MAXI:ecol              PI
MAXI:ccol              NONE
MAXI:gcol              NONE
MAXI:imagecoord        SKY
MAXI:wmapcoord         SKY
MAXI:catnum            1
MAXI:wmapb             yes
MAXI:wmapfix           no
MAXI:lststr            *.evt
MAXI:fbin              1
MAXI:hbin              1
MAXI:GSC:modes         64BIT 32BIT 16BIT
MAXI:SSC:modes         64BIT
MAXI:extract           extractor
MAXI:catcol            OBJECT TELESCOP INSTRUME DATAMODE DATE-OBS DATE-END TIME-OBS
MAXI:dispcol           OBJECT DATE-OBS DATE-END TIME-OBS
MAXI:extendresp        yes
```

**ds9 7.1** – <http://hea-www.harvard.edu/RD/ds9/site/Download.html><sup>2</sup>

使用する PC 環境に合わせて、64bit 版か 32bit 版のどちらが必要かを判断してインストールする。

```
$ tar zxvf /opt/maxi/src/ds9.linux64.7.1.tar.gz -C /opt/maxi/local_20130505/bin
```

**ROOT 5.34.07** – [ftp://root.cern.ch/root/root\\_v5.34.07.source.tar.gz](ftp://root.cern.ch/root/root_v5.34.07.source.tar.gz)

mxkwtool を使用しない場合は、configure のオプションは全て必要無い。

```
$ tar zxvf /opt/maxi/src/root_v5.34.07.source.tar.gz
$ mv root root_v5.34.07
$ cd root_v5.34.07
$ ./configure
  --with-cfitsio-incdir=$HEADAS/include
  --with-cfitsio-libdir=$HEADAS/lib
  --with-python-incdir=/opt/maxi/local_20130505/include/python2.7
  --with-python-libdir=/opt/maxi/local_20130505/lib
$ make
$ setenv ROOTSYS /opt/maxi/local_20130505/install_work/root_v5.34.07 [CSH の場合]
$ source $ROOTSYS/bin/thisroot.csh [CSH の場合]
$ export ROOTSYS=/opt/maxi/local_20130505/install_work/root_v5.34.07 [BASH の場合]
$ source $ROOTSYS/bin/thisroot.sh [BASH の場合]
$ pwd
/opt/maxi/local_20130505/install_work/root_v5.34.07
$ cd ../
```

**healpix 3.11** –

[http://sourceforge.net/projects/healpix/files/Healpix\\_3.11/Healpix\\_3.11\\_2013Apr24.tar.gz/download](http://sourceforge.net/projects/healpix/files/Healpix_3.11/Healpix_3.11_2013Apr24.tar.gz/download)

```
$ setenv MXSOFT /opt/maxi/mxsoft_20130505 [CSH の場合]
$ export MXSOFT=/opt/maxi/mxsoft_20130505 [BASH の場合]
$ cd $MXSOFT
$ mkdir healpix
$ cd healpix
$ tar zxvf /opt/maxi/src/Healpix_3.11_2013Apr24.tar.gz
$ cd Healpix_3.11/src/C/subs
$ sed 's/lcfitsio/lcfitsio_3.32/g' Makefile > Makefile.local
$ make -f Makefile.local shared OPT=-fPIC CFITSIO_INCDIR=$HEADAS/include CFITSIO_LIBDIR=$HEADAS/lib
$ mkdir $MXSOFT/include $MXSOFT/lib
$ make -f Makefile.local install LIBDIR=$MXSOFT/lib INCDIR=$MXSOFT/include
$ pwd
/opt/maxi/mxsoft_20130505/healpix/Healpix_3.11/src/C/subs
$ cd ../../../../
```

---

<sup>2</sup>古いバージョンは”<http://hea-www.harvard.edu/RD/ds9/archive>” から取得できる。

## 3.2 MAXIソフトウェアのインストール

本節では表 3.2 に一覧を示す MAXI ソフトウェアの configure バージョン<sup>3</sup>を用いたコンパイル・インストールの手順を説明する。開発は Cent-OS 5.X x86\_64(64-bit)で行っており、i386(32-bit)でも動作テストを行っている。それぞれの MAXI ソフトウェアにバージョン番号はあるが、まとめてバージョン番号を付けており、表 3.2 は Ver.20130505 である。これらは 2013 年 5 月におけるの暫定バージョンである。MAXI-DB からイベントデータを直接に取得したい場合は、§A.2 を参考にして mxdbtool のインストールを行う。位置決めを行いたい場合は、§A.2 を参考にして mxkwtool のインストールを行う。なお、これらの mxdbtool と mxkwtool は、他の MAXI ソフトウェアをインストールした後に、インストールを行う。ここでは、“/opt/maxi/mxsoft\_20130505”にインストールするとして説明する。MAXI ソフトウェアは表 3.2 のバージョン番号の組み合わせで動作させる必要があり、それを configure スクリプトでチェックしているため、ディレクトリ構造は本節での説明と厳密に同一にする<sup>4</sup>。

Subversion のアカウントは Wiki と同じであり、“svn”を実行する際にユーザー名を指定する必要がある。以降の説明ではユーザー名を“USERNAME”を表記するので、自身のユーザ名に読み替えて頂きたい。

以下の環境変数が設定されていることを確認する。設定されていなくても configure に影響は無いが、以下に説明するように手順が簡単になる。MXSOFT が定義されていると、configure を実行した際に、prefix に MXSOFT が設定される。また、HEADAS と ROOT の環境変数を設定しておくと、configure の様々なオプションを省略できる。

```
$ setenv PATH /opt/maxi/local_20130505/bin:$PATH [CSH の場合]
$ setenv LD_LIBRARY_PATH /opt/maxi/local_20130505/lib:$LD_LIBRARY_PATH [CSH の場合]
$ setenv HEADAS /opt/maxi/local_20130505/install_work/heasoft-6.13/x86_64-unknown-linux-gnu-libc2.12
[CSH の場合]
$ source $HEADAS/headas-init.csh [CSH の場合]
$ setenv ROOTSYS /opt/maxi/local_20130505/install_work/root_v5.34.07 [CSH の場合]
$ source $ROOTSYS/bin/thisroot.csh $ROOTSYS [CSH の場合]
$ setenv MXSOFT /opt/maxi/mxsoft_20130505 [CSH の場合]

$ export PATH=/opt/maxi/local_20130505/bin:$PATH [BASH の場合]
$ export LD_LIBRARY_PATH=/opt/maxi/local_20130505/lib:$LD_LIBRARY_PATH [BASH の場合]
$ export HEADAS=/opt/maxi/local_20130505/install_work/heasoft-6.13/x86_64-unknown-linux-gnu-libc2.12
[BASH の場合]
$ source $HEADAS/headas-init.sh [BASH の場合]
$ export ROOTSYS=/opt/maxi/local_20130505/install_work/root_v5.34.07 [BASH の場合]
$ source $ROOTSYS/bin/thisroot.sh [BASH の場合]
$ export MXSOFT=/opt/maxi/mxsoft_20130505 [BASH の場合]
```

### maxitool 3.26a

```
$ cd $MXSOFT
$ mkdir maxitool
$ cd maxitool
```

<sup>3</sup>開発者がリリースしたものは Makefile.in を書き換える必要があるが、簡単にインストールが出来るように中川 (ISAS/JAXA) が configure スクリプトを追加したバージョンを用意している。リリースされたバージョン番号の末尾に“\_c”を付加している。

<sup>4</sup>ディレクトリ構造を厳密に同一にするという条件は不便な点もあるだろうが、現状では一部のヘッダーファイルは、“\$MXSOFT/include”ではなく、例えば“maxitool/3.26a/include”に置かれているため、これに対処する目的もある。

表 3.2: MAXI ソフトウェア (Ver.20130505) の一覧を示す。

ソフトウェア	バージョン	コメント
maxitool	3.26a	–
gsc software	1.36.2a	–
obsjud	0.2.2mx	–
maxiutil	1.45a	–
f2root	0.5	FITS 形式の Event File を ROOT 形式に変換するツール
mxdatabase	2.0a	MAXI-DB から Event File を FITS 形式で取得するツール。§A.2 を参照。
mxkwtool	2.1.2	位置決めツール。§A.2 を参照。

```
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/maxitool/branches/3.26a_c
$ cd 3.26a_c
$ ./configure
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/maxitool/3.26a_c
$ cd ../../
```

#### **gsc software 1.36.2a**

```
$ mkdir gsc
$ cd gsc
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/gsc/resp/branches/1.36.2a_c
$ cd 1.36.2a_c
$ ./configure
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/gsc/1.36.2a_c
$ cd ../../
```

#### **obsjud 0.2.2mx**

```
$ mkdir obsjud
$ cd obsjud
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/obsjud/branches/0.2.2mx_c
$ cd 0.2.2mx_c
$ ./configure
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/obsjud/0.2.2mx_c
```

```
$ cd ../../
```

### **maxiutil 1.45a**

```
$ mkdir maxiutil
$ cd maxiutil
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/maxiutil/branches/1.45a_c
$ cd 1.45a_c
$ ./configure
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/maxiutil/1.45a_c
$ cd ../../
```

### **f2root 0.5**

```
$ mkdir f2root
$ cd f2root
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/f2root/branches/0.5_c
$ cd 0.5_c
$ ./configure
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/f2root/0.5_c
$ cd ../../
```

### **pfiles のシンボリックリンク**

“\$MXSOFT/pfiles” にあるファイルのシンボリックリンクを “\$HEADAS/syspfiles” に作成する。

```
$ cd $HEADAS/syspfiles
$ ln -s $MXSOFT/pfiles/*.par .
```

## **3.3 CALDB のインストール**

本節では表 3.3 に示す CALDB のインストールの手順について説明する。“/opt/maxi/mycaldb\_20130225” にインストールするとして説明する。

**Setup Files** – [http://heasarc.gsfc.nasa.gov/FTP/caldb/software/tools/caldb\\_setup\\_files.tar.Z](http://heasarc.gsfc.nasa.gov/FTP/caldb/software/tools/caldb_setup_files.tar.Z)

```
$ cd /opt/maxi/mycaldb_20130225
$ tar zxvf /opt/maxi/src/caldb_setup_files.tar.Z
```



表 3.3: CALDB の一覧を示す。

ミッション	検出器	バージョン	コメント
(Setup)	—	—	CALDB Setup Files
(Generic)	—	—	
MAXI	MIS	20090809	
MAXI	GSC	20130225	
MAXI	SSC	20111205	

“/opt/maxi/mycaldb\_20130225/software/tools” の下にある caldbinit.csh(Bourne シェルなら caldbinit.sh) の CALDB を設定する部分を書き換える。

変更前：setenv CALDB /FTP/caldb [CSH の場合]

変更後：setenv CALDB /opt/maxi/mycaldb\_20130225 [CSH の場合]

変更前：export CALDB=/FTP/caldb [BASH の場合]

変更後：export CALDB=/opt/maxi/mycaldb\_20130225 [BASH の場合]

“/opt/maxi/mycaldb\_20130225/software/tools/caldb.config” に以下の内容を追記する。

```
#
# MAXI
#
MAXI GSC      CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_0    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_1    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_2    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_3    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_4    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_5    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_6    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_7    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_8    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_9    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_A    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
MAXI GSC_B    CALDB data/maxi/gsc  caldb.indx  CALDB  data/maxi/gsc
#
MAXI SSC      CALDB data/maxi/ssc  caldb.indx  CALDB  data/maxi/ssc
MAXI SSC_H    CALDB data/maxi/ssc  caldb.indx  CALDB  data/maxi/ssc
MAXI SSC_Z    CALDB data/maxi/ssc  caldb.indx  CALDB  data/maxi/ssc
#
MAXI VSC      CALDB data/maxi/mis  caldb.indx  CALDB  data/maxi/mis
MAXI INS      CALDB data/maxi/mis  caldb.indx  CALDB  data/maxi/mis
```

**Generic** – [http://heasarc.gsfc.nasa.gov/FTP/caldb/data/gen/goodfiles\\_gen\\_ins\\_20120127.tar.Z](http://heasarc.gsfc.nasa.gov/FTP/caldb/data/gen/goodfiles_gen_ins_20120127.tar.Z)

```
$ tar zxvf /opt/maxi/src/goodfiles_gen_ins_20120127.tar.Z
```

**MAXI/MIS** –

[http://www.maxi.jaxa.jp/pm/mission\\_team/pukiwiki/index.php?plugin=attach&refer=MaxiSoft%2Fmaxi\\_caldb&openfile=caldb\\_maxi\\_mis\\_ht20090809.tar.gz](http://www.maxi.jaxa.jp/pm/mission_team/pukiwiki/index.php?plugin=attach&refer=MaxiSoft%2Fmaxi_caldb&openfile=caldb_maxi_mis_ht20090809.tar.gz)

```
$ tar zxvf /opt/maxi/src/caldb_maxi_mis_ht20090809.tar.gz
```

**MAXI/GSC** –

[http://www.maxi.jaxa.jp/pm/mission\\_team/web/sugizaki/maxi/caldb\\_archive/caldb\\_maxi\\_gsc\\_ms20130225.tar.gz](http://www.maxi.jaxa.jp/pm/mission_team/web/sugizaki/maxi/caldb_archive/caldb_maxi_gsc_ms20130225.tar.gz)

```
$ tar zxvf /opt/maxi/src/caldb_maxi_gsc_ms20130225.tar.gz
```

**MAXI/SSC** –

[http://www.maxi.jaxa.jp/pm/mission\\_team/pukiwiki/index.php?plugin=attach&refer=MaxiSoft%2Fmaxi\\_caldb&openfile=ssc\\_caldb\\_20111205.tgz](http://www.maxi.jaxa.jp/pm/mission_team/pukiwiki/index.php?plugin=attach&refer=MaxiSoft%2Fmaxi_caldb&openfile=ssc_caldb_20111205.tgz)

```
$ cd data/maxi
```

```
$ tar zxvf /opt/maxi/src/ssc_caldb_20111205.tgz
```

### 環境変数の設定

以下のコマンドを実行して CALDB の環境変数を設定する。

```
$ source /opt/maxi/mycaldb_20130225/software/tools/caldbinit.csh [CSH の場合]
```

```
$ source /opt/maxi/mycaldb_20130225/software/tools/caldbinit.sh [BASH の場合]
```

## 3.4 GSC RMF Database の取得

GSC RMF Database は HV 毎、芯線毎、detx 毎に用意されており、110GB 程度のディスク容量になる。GSC のスペクトル解析で使う RMF Builder を走らせるために必要になる。GSC のスペクトル解析をしなければ取得しなくて良い。理化学研究所の以下の場所から取得する。アクセス方法は理化学研究所の関係者に問い合わせる。ここでは “/opt/maxi/gscrmfdb” に置くとする。

```
gsc@maxireceive.riken.jp:/dx1/gscrmfdb
```

gscrmfdb の直下にあるインデックスファイルの gscrmfdbidx.fits を PC 環境に合わせて書き換える。具体的には FILEDIR を書き換えるのだが、これは “gscrmfdb” の直下にある fparkey\_filedir.sh で実施できる。以下のように fparkey\_filedir.sh の記述を変更する。

変更前 : /opt/gscrmfdb

変更後 : /opt/maxi/gscrmfdb

以下のコマンドを実行する。

```
$ cd /opt/maxi/gscrmfdb
$ ./fparkey_filedir.sh
```

以下の環境変数を設定する。

```
$ setenv GSCRMFDB /opt/maxi/gscrmfdb [CSH の場合]

$ export GSCRMFDB=/opt/maxi/gscrmfdb [BASH の場合]
```

### 3.5 Cleaned Event File と Auxiliary File の取得

理化学研究所の以下の場所から GSC の Cleaned Event File、および Auxiliary File を取得する。アクセス方法 (パスワード) は理化学研究所の関係者に問い合わせる。これらは毎日更新しているので、cron で rsync しておくと便利である。ここでは “/opt/maxi/mxdata” に置くとする。

```
gsc@maxireceive.riken.jp:/dx1/mxdata-jax/gsc.v1.4
gsc@maxireceive.riken.jp:/dx1/mxdata-jax/auxil
gsc@maxireceive.riken.jp:/dx1/mxdata-jax/mkf
gsc@maxireceive.riken.jp:/dx1/mxdata-jax/rbmhk
```

auxil の直下にあるインデックスファイルの FILEDIR を “/opt/maxi/mxdata” へ書き換える。HEAsoft に含まれている fparkey を使用して書き換えることが出来るので、以下のコマンドを実行する。

```
$ fparkey /opt/maxi/mxdata attlist.fits FILEDIR
$ fparkey /opt/maxi/mxdata dptclist.fits FILEDIR
$ fparkey /opt/maxi/mxdata geafrcelist.fits FILEDIR
$ fparkey /opt/maxi/mxdata gebfrcelist.fits FILEDIR
$ fparkey /opt/maxi/mxdata issancpm.fits FILEDIR
$ fparkey /opt/maxi/mxdata issancpmlist.fits FILEDIR
$ fparkey /opt/maxi/mxdata issancsarj.fits FILEDIR
$ fparkey /opt/maxi/mxdata issancsarjlist.fits FILEDIR
$ fparkey /opt/maxi/mxdata orblist.fits FILEDIR
```

最後に以下の環境変数を設定する。

```
$ setenv MXDATA /opt/maxi/mxdata [CSH の場合]

$ export MXDATA=/opt/maxi/mxdata [BASH の場合]
```

## 3.6 セットアップスクリプト

§3.1 ~ §3.5 において、コンパイル・インストールを行いながら、MAXI ソフトウェアを実行するために必要な環境変数を設定した。これらをまとめたセットアップスクリプトを以下に示す。mxkwtool を使用したい場合は、最後の 8 行のコメントアウト(＃)を削除する。また、mxkwtool は HEAsoft 6.13 だけでなく、WCS に関連するライブラリは HEAsoft 6.12 を使用しているため、このセットアップスクリプトの通りに環境変数を整えないと、全ての MAXI ソフトウェアにおいてライブラリに関するエラーが出る場合がある<sup>5</sup>。

### CSH のセットアップスクリプト

```
#=====
# Setup File for MAXI Software Ver.20130505
#=====
#
# set versions
setenv MXSOFT_VER 20130505
setenv HEADAS_VER 6.13
setenv CALDB_VER 20130225
setenv ROOT_VER v5.34.07
#
# set path
setenv PATH /usr/local/bin:/usr/bin:/bin
setenv LD_LIBRARY_PATH /usr/local/lib
#
# heasoft
setenv HEADAS /opt/maxi/local_20130505/install_work/heasoft-$HEADAS_VER/x86_64-unknown-linux-gnu-libc2.12
source $HEADAS/headas-init.csh
#
# root
source /opt/maxi/local_20130505/install_work/root_${ROOT_VER}/bin/thisroot.csh
#
# caldb
source /opt/maxi/mycaldb_${CALDB_VER}/software/tools/caldbinit.csh
#
# mxsoft
setenv MXSOFT /opt/maxi/mxsoft_${MXSOFT_VER}
setenv PATH $MXSOFT/bin:$PATH
setenv LD_LIBRARY_PATH $MXSOFT/lib:$LD_LIBRARY_PATH
setenv PFILES "$HOME/pfiles;$MXSOFT/pfiles:$HEADAS/syspfiles"
#
# mxdata
setenv MXDATA /opt/maxi/mxdata
setenv GSCRMFDB /opt/maxi/gscrmfdb
#
# python
setenv PYTHONPATH $MXSOFT/pylib:$PYTHONPATH
#
# mxkwtool
#setenv PATH $MXSOFT/mxkwtool/2.1.2/script:$PATH
#setenv MXKWTOOL $MXSOFT/mxkwtool/2.1.2
#setenv MXKW_TMPDIR $HOME/maxi/tmp
#setenv MXKW_DS9_BIN_DIR /opt/maxi/local_20130505/bin
#setenv MXKW_MXFILE_BUSY_FLAG $HOME/maxi/exec_flag_rsync_jax_data_from_riken
#setenv MXKW_GRBTOOL_CAT $MXSOFT/catalog/weblistv6_detect091209_merge_t.csv
#setenv MXKW_GRBTOOL_CAT_LOCAL $HOME/maxi/srcinfo.list
#setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/opt/heasoft/6.12/x86_64-unknown-linux-gnu-libc2.12/lib
```

<sup>5</sup>基本的に MAXI ソフトウェアは HEAsoft 6.13 でコンパイルしているため、LD\_LIBRARY\_PATH を設定する際に、HEAsoft 6.12 を先に書いてしまうと、ライブラリのバージョンの違いによりエラーが出る。

## BASH のセットアップスクリプト

```
#=====
# Setup File for MAXI Software Ver.20130505
#=====
#
# set versions
export MXSOFT_VER=20130505
export HEADAS_VER=6.13
export CALDB_VER=20130225
export ROOT_VER=v5.34.07
#
# set path
export PATH=/usr/local/bin:/usr/bin:/bin
export LD_LIBRARY_PATH=/usr/local/lib
#
# heasoft
export HEADAS=/opt/maxi/local_20130505/install_work/heasoft-$HEADAS_VER/x86_64-unknown-linux-gnu-libc2.12
source $HEADAS/headas-init.sh
#
# root
source /opt/maxi/local_20130505/install_work/root_${ROOT_VER}/bin/thisroot.sh
#
# caldb
source /opt/maxi/mycaldb_${CALDB_VER}/software/tools/caldbinit.sh
#
# mxsoft
export MXSOFT=/opt/maxi/mxsoft_${MXSOFT_VER}
export PATH=$MXSOFT/bin:$PATH
export LD_LIBRARY_PATH=$MXSOFT/lib:$LD_LIBRARY_PATH
export PFILES="$HOME/pfiles;$MXSOFT/pfiles:$HEADAS/syspfiles"
#
# mxdata
export MXDATA=/opt/maxi/mxdata
export GSCRMFDB=/opt/maxi/gscrmfdb
#
# python
export PYTHONPATH=$MXSOFT/pylib:$PYTHONPATH
#
# mxkwtool
#export PATH=$MXSOFT/mxkwtool/2.1.2/script:$PATH
#export MXKWTOOL=$MXSOFT/mxkwtool/2.1.2
#export MXKW_TMPDIR=$HOME/maxi/tmp
#export MXKW_DS9_BIN_DIR=/opt/maxi/local_20130505/bin
#export MXKW_MXFILE_BUSY_FLAG=$HOME/maxi/exec_flag_rsync_jax_data_from_riken
#export MXKW_GRBTOOL_CAT=$MXSOFT/catalog/weblistv6_detect091209_merge_t.csv
#export MXKW_GRBTOOL_CAT_LOCAL=$HOME/maxi/srcinfo.list
#export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/heasoft/6.12/x86_64-unknown-linux-gnu-libc2.12/lib
```

## 第4章 標準解析

理化学研究所で生成している Cleaned Event File (§3.5) を使用して、天体の標準解析を行う事を目標として、その手順を説明する。GSC と SSC で手順はほぼ同じなので、並列して説明する。なお、Quick Look 解析の場合は MAXI-DB から直接に Event File の抽出を行うが、Event File を得た後は本章と同様な手順である。

### 4.1 天体周辺の Event File の抽出

カメラ毎に時系列に蓄積した全天データから中心座標と周辺半径、XY Pixel Format を指定して天体周辺の Event File を生成する。プログラムは GSC では `mxextract`、SSC では `mxsextract` を用いる。使い方はほぼ同じだが、SSC の `mxsextract` では最後に出力ファイルのフォーマットを指定するパラメータ `outform` がある。

出力ファイルの XY 座標は、指定した Pixel Format に従って TAN Projection で付け直される。条件に合う Event がなかった場合は Event File は生成されない。STDGTI Extension は入力ファイルの Event File のものが継承される。

実行時、以下のパラメータを入力する。

`infile` 入力ファイル名 (例: `gsc0_evtfile.fits`)、またはリストのファイル名 (例: `@evtfile.list`)。複数の Event File を読み込みたい場合、ファイル名をテキストファイルにリストしておいて、`@evtfile.list` と入力ファイルの所で入れると、リストの順に読み込んでくれる。異なるカメラの Event File を一度にリストで処理することもできるが、今のところ STDGTI Extension が正しく処理されないので、カメラ毎に処理する。

`outfile` 出力ファイル (fits binary table) 名。

`ra` ターゲットの Right Ascension (degree)。

`dec` ターゲットの Declination (degree)。

`radi_i` 抽出内半径。通常  $0^\circ$  を指定する。

`radi_o` 抽出内外径。ライトカーブ、スペクトル解析でバックグラウンド領域が取れるように PSF より大きめに  $8^\circ$  くらいとっておくと良い。

`nx` XY 投影座標の X Pixel 数。

`ny` XY 投影座標の Y Pixel 数。

`pixsize` XY 投影座標の Pixel サイズ (degree/pixel)。

`attlist` 姿勢ファイルリストのファイル名。所定の場所に所定の名前でインストールされていれば、DEF と入れれば良い。

`outform` SSC のみ 出力イベントファイルフォーマット。10:GRADE, 11:PHA, 20:EXT。EXT は PHA フォーマットにイベント毎に月 (moon) からの離角のパラメータを加えたもの。

実行スクリプトは例えば以下の通り。

#### GSC の場合

```
#!/bin/sh
mjd=55501
gscid=4
infile="/nfs/m/maxir1/jax_data/rev1.1/low/evt_camid_cl/low32/gsc${gscid}_r2pcl_MJD${mjd}.evt"
outfile="crab_gsc${gscid}_mjd${mjd}.evt"
object="Crab"
ra=83.63
dec=22.01
radi_i=0.0
radi_o=10.0
nx=200
ny=200
pixsize=0.1
attlist="DEF"
```

```
mxextract $infile $outfile $object $ra $dec $radi_i $radi_o $nx $ny $pixsize $attlist
```

#### SSC の場合

```
#!/bin/sh
mjd=55501
hz="h"
infile="/nfs/m/maxir1/ssc/evt_cleaned/ssc${hz}_allcl_mjd${mjd}.evt"
outfile="crab_ssc${hz}_mjd${mjd}.evt"
object="Crab"
ra=83.63
dec=22.01
radi_i=0.0
radi_o=10.0
nx=200
ny=200
pixsize=0.1
attlist="DEF"
outform=20
```

```
mxsextract $infile $outfile $object $ra $dec $radi_i $radi_o $nx $ny $pixsize $attlist $outform
```

生成した Event File は FITS 形式であるが、ROOT Library を用いたツールで解析を行うために、ftbl2root により FITS 形式から ROOT 形式へ変換する。

```
$ ftbl2root (fits file name) (root file name)
```

## 4.2 Image File の生成、表示

§4.1 で生成した Event File を `xselect` で読み込み、Image を生成し、`ds9` で表示し、保存するには以下のコマンドを実行する (見易さのため一部省略してある)。なお、一度に複数の Event File を読み込めば、すべて足し合わせたイメージが出来る。複数の Event File を読み込みたい場合、Event File のリストをテキスト形式で作成して (例: `evtfile.list`)、Event File の名前を指定する代わりに `"@evtfile.list"` を指定する。また、エネルギーバンドを指定したい場合は `"pha_cutoff"`<sup>1</sup> を指定する。

```
$ xselect
```

```
> Enter session name >[]gsc4
```

```
gsc4:SUZAKU > read events crab_gsc4_mjd55501.evt <-- もしくは"@evtfile.list"を指定する。
```

```
> Enter the Event file dir >[.] .
```

```
Got new mission: MAXI
```

```
> Reset the mission ? >[yes] yes
```

```
gsc4:MAXI-GSC-32BIT > filter pha_cutoff 40 200 <-- 2-10keV のエネルギーバンドを指定する。
```

```
gsc4:MAXI-GSC-32BIT > extract image <-- Image を生成する。
```

```
gsc4:MAXI-GSC-32BIT > sao <-- ds9 で表示する。
```

```
gsc4:MAXI-GSC-32BIT > save image crab_gsc4_mjd55501_2_10keV.img <-- ファイルに保存する。
```

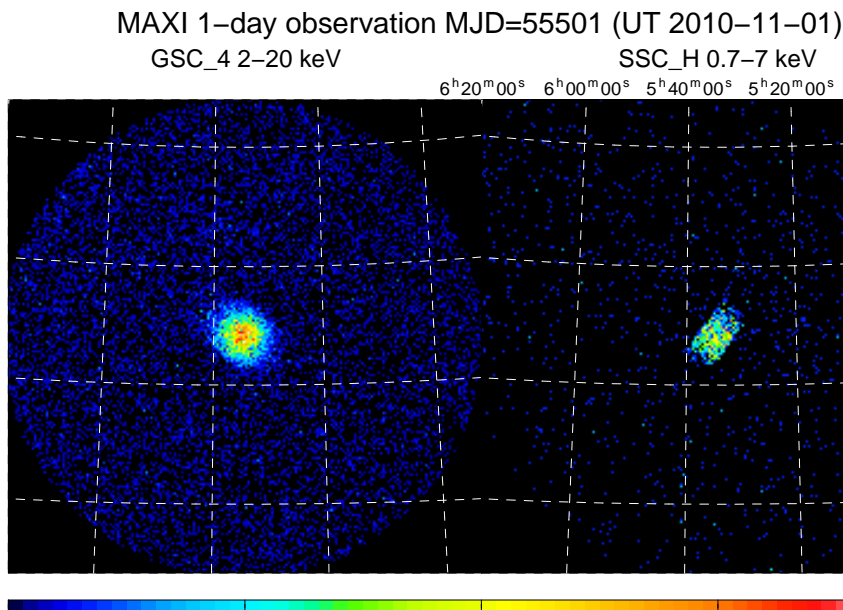


図 4.1: 2010 年 11 月 1 日における Crab の 1 日積分した Image を示す。左図は GSC-4 の 2-20 keV であり、右図は SSC-H の 0.7-7 keV の Image である。

<sup>1</sup> エネルギー ( $E$ ) とチャンネル ( $N$ ) には  $E = 50N$  [eV] の関係がある。



## 4.3 ライトカーブ解析

座標がわかっている X 線源に対して、MAXI の軌道スキャン周期 (~90 分) を超えた時間スケールのライトカーブを導出することを目標として、観測時間、有効面積、バックグラウンドを補正する手順を述べる。

### 4.3.1 バックグラウンドレベルの見積り方法の決定

§4.2 で生成した Image を見て、バックグラウンドレベルをどのように見積もるかを決める。以下の 2 種類の方法がある。

ターゲットの周りに混入ソースが無い場合

各スキャン毎に同じコリメータ入射角の検出器領域で、ソースが視野に入る前後の時刻のイベントレートからソースに含まれるバックグラウンドレートを見積もる。どの領域からソースのイベントレート、バックグラウンドレートを見積もるかは §4.3.3 のプロセスで指定する。図 4.2 に領域を指定するパラメータを示した。

ターゲットの周りに混入ソースある場合

天球座標を用いてソースとバックグラウンドの領域を決めて、バックグラウンドは立体角当り一様であると仮定して、ソースに含まれるバックグラウンドレートを見積もる。そのために最初に、ソース領域とバックグラウンド領域、それぞれの Region File(ここではファイル名は src.reg, bkg.reg とする)を作成する。§4.2 で生成した Image を ds9 で表示し、Region を決定し<sup>2</sup>、SKYXY 座標で保存する (形式:ds9, 座標系:fk5)。

§4.1 で生成した天体周辺の Event File を xselect で読み込み、作成した Region File を読み込んで、ソース領域とバックグラウンド領域、それぞれの Event File を生成する。以下に xselect での作業の例を示す。Event File は §4.1 で説明した ftbl2root を用いて、FITS 形式から ROOT 形式へ変換しておく。特殊な Region File でなければ、ソース領域とバックグラウンド領域、それぞれの面積は、抽出した Event File の “Header Key Parameter: NPIXSOU” に記述されている。

```
$ xselect
```

```
> Enter session name >[]gsc4
gsc4:SUZAKU > read events crab_gsc4_mjd55501.evt
> Enter the Event file dir >[.] .
Got new mission: MAXI
> Reset the mission ? >[yes] yes

gsc4:MAXI-GSC-32BIT > set XYNAME X Y

gsc4:MAXI-GSC-32BIT > filter region src.reg
gsc4:MAXI-GSC-32BIT > extract events
gsc4:MAXI-GSC-32BIT > save events crab_gsc4_mjd55501_src.evt
> Use filtered events as input data file ? >[yes] no
gsc4:MAXI-GSC-32BIT > clear events
```

<sup>2</sup>Region についてはインターネットに情報があるので詳細は説明しないが、例えば「すざく」衛星の”ファーストステップガイド”(http://cosmic.riken.jp/suzaku/help/guide/fstep-web/node8.html) が参考になる。

```

gsc4:MAXI-GSC-32BIT > clear region

gsc4:MAXI-GSC-32BIT > filter region bkg.reg
gsc4:MAXI-GSC-32BIT > extract events
gsc4:MAXI-GSC-32BIT > save events crab_gsc4_mjd55501_bkg.evt
> Use filtered events as input data file ? >[yes] no
gsc4:MAXI-GSC-32BIT > clear events
gsc4:MAXI-GSC-32BIT > clear region

```

#### 4.3.2 観測条件ファイルの生成

解析する天体に対する観測条件ファイル (OBSCUR ファイル) を作成する。プログラムは GSC と SSC で共通で、mxobscur を用いる。mxobscur は、CALDB データ、GTI データ、姿勢データ、ISS パドル回転角、視野解析ライブラリ<sup>3</sup>のパラメータを読み、それぞれカメラ毎に表 4.1 のデータを計算し、FITS ファイルに保存する。GSC(instr=gsc) なら最大で 12 個、SSC(instr=ssc) なら最大で 2 個の Extension HDU が 1 つの FITS ファイルに保存される<sup>4</sup>。“gtifile0=NONE”などを指定すると、そのカメラは計算されないで、その分の Extension HDU が少なくなる。

入力パラメータは、以下の通り。

instr GSC または SSC。

outfname 出力観測条件ファイル名。FITS ファイル。

ra ターゲットの Right Ascension (degree)。

dec ターゲットの Declination (degree)。

tstart 観測開始時刻 (MJD)。

tstop 観測終了時刻 (MJD)。

dt 計算時間ステップ (秒)。通常は 1(秒)。

colphi\_max 結果をファイルに出力する、天体入射角  $|\phi_{\text{col}}|$  の上限。これより大きいと天体が視野から外れているとみなし、保存しない。GSC では  $42^\circ$ 、SSC では  $46^\circ$  程度。

coltha\_max 結果をファイルに出力する、天体入射角  $|\theta_{\text{col}}|$  の上限。後でソースだけでなくバックグラウンドに対する exposure の計算でも使うので、視野より少し大きめ、 $8^\circ$  程度まで取っておく。

attlist 姿勢ファイルリストのファイル名。通常は DEF を指定する。

obsjud\_dname 視野解析ライブラリのパラメータファイルがあるディレクトリ名。パラメータの順番が ver.1.0 から変わった。通常は DEF でよく、NONE と入れると、視野判定計算をスキップする。

sarjfnamelist 太陽パドル  $\alpha$  角のリストファイル名。通常は DEF を指定する。

pmfnamelist 太陽パドル  $\beta$  角のリストファイル名。通常は DEF を指定する。

<sup>3</sup>視野解析ライブラリのパラメータに NONE と入れると、パドル遮蔽判定をスキップできる。

<sup>4</sup>以前の mxgeaobscur、mxseaobscur とは異なる。

gtifile0 ... gtifile11 カメラ毎の GTI File 名を指定する。GTI File は FITS 形式で、Event File の STDGTI Extension HDU と同じ Format を持つことを想定している。通常は、Event File を指定する。GSC(instr=gsc) の場合、gtifile0 ... gtifile11 の 12 個の入力パラメータが読まれる。SSC(instr=ssc) の場合は最初の 2 つ (gtifile0: SSC-H, gtifile1: SSC-Z) だけが読まれる。GTI File が無い、もしくはそのカメラのデータを以降の解析に使わない場合は “NONE” を指定する。GSC は、カメラ毎に全ての Cleaned Event File の STDGTI だけを抽出したファイルを日々生成している。

表 4.1: 観測条件ファイル (OBSCUR ファイル) の表データに含まれるパラメータ名を示す。

TIME	MAXI Time
GTI	TIME が GTI file の good time interval に入っていれば 1。それ以外は 0 か、GTI ファイルの範囲外なら 255。
OBSJUD	視野判定ライブラリが返す値。ターゲットが遮蔽物に隠されていなければ 0。
COLTHA	天体の Collimator Theta 角度。
COLPHI	天体の Collimator Phi 角度。
AREA	天体の入射角に対する Collimator Area。

#### GSC の実行スクリプトの例

```
#!/bin/sh

mjd=55058
mjd_end=55059

instr=gsc    # SSC の場合はここを"ssc"とする。
outfname=crab_gobscur_mjd${mjd}.fits

ra=83.63
dec=22.01
tstart=$mjd
tstop=$mjd_end
dt=1.0
colphi_max=42.0
coltha_max=10.0
attlist=DEF
obsjud_dname=DEF
sarjfnamelist=DEF
pmfnamelist=DEF

gtifile0=NONE
gtifile1=crab_gsc1_mjd${mjd}.evt
gtifile2=crab_gsc2_mjd${mjd}.evt
gtifile3=NONE
gtifile4=crab_gsc4_mjd${mjd}.evt
gtifile5=crab_gsc5_mjd${mjd}.evt
```

```
gtifile6=NONE
gtifile7=crab_gsc7_mjd${mjd}.evt
gtifile8=crab_gsc8_mjd${mjd}.evt
gtifile9=NONE
gtifilea=crab_gsca_mjd${mjd}.evt
gtifileb=crab_gscb_mjd${mjd}.evt

cmdstr="mxobscur $instr $outfname $ra $dec $tstart $tstop $dt $colphi_max $coltha_max \
$attlist $obsjud_dname $sarjfnamelist $pmfnamelist \
$gtifile0 $gtifile1 $gtifile2 $gtifile3 $gtifile4 $gtifile5 \
$gtifile6 $gtifile7 $gtifile8 $gtifile9 $gtifilea $gtifileb"

echo $cmdstr
$cmdstr
```

生成した OBSCUR ファイルは、後でライトカーブ生成プログラムで処理できるように、fits2trees<sup>5</sup> で FITS 形式から ROOT 形式へ変換する。

```
$ fits2trees crab_gsc_obscur.fits crab_gsc_obscur.root
```

#### 4.3.3 ライトカーブ生成中間ファイルの作成

Event File(ROOT 形式) と観測条件ファイル (ROOT 形式) から、ライトカーブを生成するために必要な種々のヒストグラムから成るファイル (ROOT 形式) を生成する。プログラムは GSC では mxglchist、SSC では mxslchist を使う。

§4.3.1 で解説したバックグラウンドの見積り方法は、入力パラメータの bgdscale で指定する。bgdscale は、図 4.2 のようにスキャンに沿って同じ検出器領域で決める場合は負の値、天球座標で決める場合は正の値を指定する。bgdscale の値によって使われるパラメータが違うが、実行時にはすべてのパラメータを入力する必要がある。

X 線エネルギーに依って有効面積の入射角依存性が異なるため、エネルギーバンド (pibandfname)、およびソースのスペクトルを wabs\*powerlaw (phindx, nh) で指定する。phindx と nh は天体のエネルギースペクトルを冪関数で再現した際のパラメータであるので、あたりをつけて入力する。

以下が実行時に入力するパラメータで、GSC(mxglchist) と SSC(mxslchist) で若干異なる。

**GSC と SSC で共通の入力パラメータ**

**obsflist** §4.3.2 で生成した観測条件ファイル (ROOT 形式)。複数のファイルを読み込みたい場合、テキストファイルに観測条件ファイルをリスト (例:obsfile.list) しておき、@obsfile.list を指定すれば、リストの順に観測条件ファイルを読んでもくれる。

**outfname** 出力ファイル (ROOT 形式)。

**evtflist** (evtflist.s0, evtflist.s1) Event File(ROOT 形式) またはそのリスト。bgdscale が正の場合、ソース領域で切り抜いたの Event File、またはそのリストとみなされる。入力ファイルは GSC は 1 つだが、SSC の場合は、SSC\_H(evtflist.s0)、SSC\_Z(evtflist.s1) で別々に入力する。

<sup>5</sup> §4.1 で説明した ftbl2root と違い、FITS File に複数の Table HDU があった場合、1 つの ROOT File の中にすべての Table HDU の Tree Data を保存する。

bgdflist (bgdflist.s0, bgdflist.s1) バックグラウンド領域で切り抜いた Event File(ROOT 形式)、またはそのリスト。bgdscale が負の場合は無視される。evtflist と同様に入力ファイルは、GSC は 1 つ、SSC の場合は SSC\_H、SSC\_Z で別々に入力する。

bgdscale バックグラウンド領域のソース領域に対する面積比。天球座標でバックグラウンドを決めた場合は、Event File(FITS 形式) の NPIXSOU (§4.3.1) から計算する。つまり、バックグラウンドの NPIXSOU を、ソースの NPIXSOU で割る。負の場合、§4.3.1 で解説した方法、つまり検出器上の領域で天体が視野に入る前後のイベントレートからバックグラウンドを見積もる。

tmin ライトカーブのピンの始端 (MJD)。

tmax ライトカーブのピンの終端 (MJD)。

binsize ライトカーブのピン幅 (秒)。通常は大体 1 ピン 1 スキャンになるように 5400(=90 分) とする。

obs\_colphi\_min ターゲット入射角の  $\phi_{\text{col}}$  の下限。GSC の場合、真ん中にサポート構造によるギャップがあるので、 $1.5\text{--}2.0^\circ$  程度を入れる。SSC の場合は 0.0 とする。なお SSC では、天体が CCD ギャップ付近に当たる時にはその時間帯を GTI から除くプログラムになっている。

obs\_colphi\_max ターゲット入射角の  $\phi_{\text{col}}$  の上限。カメラの端ではバックグラウンドレートがうまく見積もれないので、余分を持ってカットしたりする。GSC では、 $37\text{--}38^\circ$  を指定する。

obs\_coltha\_max ターゲット入射角の  $\theta_{\text{col}}$  の上限。有効面積が 0 の領域のデータを足しても意味がないので、通常、三角山の幅に合わせて  $1.5^\circ$  程度にする。意図的に三角山の裾をカットしたい時にはもっと小さい値にする。

pibandfname エネルギーバンドの上限と下限を PI Channel で記したファイル。例えば、GSC の 2–6, 6–12 keV のエネルギーバンドのライトカーブを抽出したい場合、エネルギーと PI Channel の変換係数である 50 eV/ch を用いて PI Channel を計算して、以下の内容が書かれたテキストファイルを作成する。

```
40   120   # 2--6 keV
120   240   # 6--12 keV
```

最大で 10 個のエネルギーバンドまで指定できる。SSC の場合も同様だが、変換係数は 3.65 eV/ch である。GSC\_NOM と入れると、GSC Nominal 4 Bands(2–4, 4–10, 10–20, 2–20 keV) となる。同様に SSC\_NOM と入れると、SSC Nominal 4 Bands(0.7–2, 2–4, 4–7, 0.7–7 keV) となる。

phindx エネルギースペクトルの冪関数指数  $\Gamma$ 。

nh エネルギースペクトルの吸収水素柱密度  $N_{\text{H}}$  [ $10^{22} \text{ cm}^{-2}$ ]。

srcreg\_dcolphi (図 4.2  $\Delta\phi_{\text{src}}$ ) 天体のスキャン前後でバックグラウンドを見積もる場合に使われるソース、バックグラウンド領域指定パラメータで、スキャン方向と直行する方向 ( $\phi_{\text{col}}$ ) の天体からの離角  $\Delta\phi_{\text{col}}$  の上限。SSC はスリット幅に対して位置分解能が十分高いので、原理的には  $1.0^\circ$  程度で良い。GSC はカウンターの位置分解能を考えて、 $1.5\text{--}1.7^\circ$  程度を指定する。

srcreg\_dcoltha (図 4.2  $\Delta\theta_{\text{src}}$ ) 天体のスキャン前後でバックグラウンドを見積もる場合に使われるソース領域指定パラメータ。PSF の断面が三角山になるスキャン方向 ( $\theta_{\text{col}}$ ) の天体からの離角  $\Delta\theta_{\text{col}}$  の上限。通常  $1.5^\circ$  を指定する。

bgdreg\_dcoltha (図 4.2  $\Delta\theta_{\text{bgd}}$ ) 天体のスキャン前後でバックグラウンドを見積もる場合に使われるバックグラウンド領域指定パラメータで、 $\Delta\theta_{\text{col}}$  の上限。通常  $4.5^\circ$  を指定する。

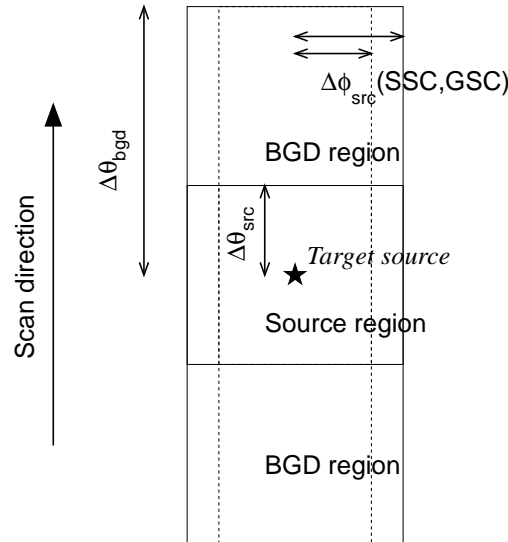


図 4.2: 検出器上の領域でターゲットが視野に入る前後のイベントレートからバックグラウンドを見積もる場合に指定するパラメータ。コリメーター入射角球面座標  $(\theta_{\text{col}}, \phi_{\text{col}})$  で、ターゲットからの離角でソース領域、バックグラウンド領域を定義する。 $\Delta\phi_{\text{src}}$  (srcreg.dcolphi),  $\Delta\theta_{\text{src}}$  (srcreg.dcoltha),  $\Delta\theta_{\text{bgd}}$  (bgdreg.dcoltha)

#### GSC のみの入力パラメータ

anodelist\_g0 ... anodelist\_gb GSC のカメラ毎に、使用するアノード番号 (012345) のリストを指定する。例えば、C1、C2 芯線を除く場合は"0345" と入力する。

dlpath Data Downlink Path(0:LOW, 1:MED) を指定する。例えば\$MXDATA/gsc.v1.4/low32 のデータを使うなら 0(LOW) を指定する。

#### SSC のみの入力パラメータ

moon\_colphi\_limit 月の入射角の  $\phi_{\text{col}}$  の下限。通常  $10^\circ$  程度を指定する。

moon\_coltha\_limit 月の入射角の  $\theta_{\text{col}}$  の下限。通常  $10^\circ$  程度を指定する。

#### GSC の実行スクリプトの例

```
#!/bin/sh
obsflist="@gsc_obscurfile.list"
outfname="gsc_lchist.root"
#pibandfname="GSC_NOM"
pibandfname="pibands.list"
evtflist="@gsc_srcfile.list"
bgdflist="@gsc_bgdfile.list"

bgdscale=2.686122

tmin=55400.0
```

```

tmax=56040.0
binsize=5400.0
phindx=2.1
nh=0.35

srcreg_dcolphi=1.7
srcreg_dcoltha=1.5
bgdreg_dcoltha=4.5

obs_colphi_min=2.0
obs_colphi_max=37.0
obs_coltha_max=2.0
evt_colphi_max=38.5

anodelist0=012345
anodelist1=012345
anodelist2=012345
anodelist3=012345
anodelist4=012345
anodelist5=012345
anodelist6=012345
anodelist7=012345
anodelist8=012345
anodelist9=012345
anodelista=012345
anodelistb=012345

dlpath=0

cmdstr="mxglchist $obsflist $outfname $pibandfname \
$evtflist $bgdflist $bgdscale\
$tmin $tmax $binsize \
$phindx $nh\
$srcreg_dcolphi $srcreg_dcoltha $bgdreg_dcoltha $evt_colphi_max\
$obs_colphi_min $obs_colphi_max $obs_coltha_max\
$anodelist0 $anodelist1 $anodelist2 $anodelist3 $anodelist4 $anodelist5\
$anodelist6 $anodelist7 $anodelist8 $anodelist9 $anodelista $anodelistb\
$dlpath"

echo $cmdstr
$cmdstr

```

## SSC の実行スクリプトの例

```
#!/bin/sh
```

```

obsflist=@ssc_obscur.list
outfname=ssc_lchist_pibands.root
#pibandfname=SSC_NOM
pibandfname=pibands.list
evtflist_s0=@ssc0_evtfile.list
evtflist_s1=@ssc1_evtfile.list
bgdflist_s0="NONE"
bgdflist_s1="NONE"

bgdscale=-1.0

tmin=55400.0
tmax=56040.0
binsize=5400.0
phindx=2.1
nh=0.35

srcreg_dcolphi=1.0
srcreg_dcoltha=1.5
bgdreg_dcoltha=4.5
evt_colphi_max=1.0

obs_colphi_min=0.0
obs_colphi_max=45.0
obs_coltha_max=2.0
evt_colphi_max=46.0

moon_colphi_limit=8.0
moon_coltha_limit=8.0

cmdstr="mxslchist $obsflist $outfname $pibandfname\
$evtflist_s0 $evtflist_s1 $bgdflist_s0 $bgdflist_s1\
$bgdscale $tmin $tmax $binsize $phindx $nh\
$srcreg_dcolphi $srcreg_dcoltha $bgdreg_dcoltha $evt_colphi_max\
$obs_colphi_min $obs_colphi_max $obs_coltha_max\
$moon_colphi_limit $moon_coltha_limit"

echo $cmdstr
$cmdstr

```

#### 4.3.4 ライトカーブファイルの作成

§4.3.3 で生成した、ライトカーブ生成中間ファイル (ROOT 形式) から種々のデータ選別条件を指定して、ライトカーブファイル (FITS 形式) を生成する。プログラムは GSC では `mxglcur`、SSC では `mxslcur` を使用



する。

入力パラメータは以下の通りである。

lchistfile §4.3.3 で生成したライトカーブ生成中間ファイル (ROOT 形式)。

outfname 出力ファイル (FITS 形式)。

pibandfname エネルギーバンドの上限と下限を PI Channel で記したファイル。§4.3.3 の処理で使ったものと同じでなければならない。

eatime\_min 1 ビン当たりの有効面積時間 ( $\text{cm}^2 \text{ s}$ ) の下限値。通常、GSC は  $40.0 \text{ cm}^2 \text{ s}$ 、SSC は  $10.0 \text{ cm}^2 \text{ s}$  を指定する。

srctime\_min 1 ビン当たりのソース領域の観測時間 (s) の下限値。通常は  $15.0 \text{ s}$  を指定する。

bgdtime\_min 1 ビン当たりのバックグラウンド領域の観測時間 ( $\text{cm}^2 \text{ s}$ ) の下限値。通常は  $15.0 \text{ cm}^2 \text{ s}$  を指定する。

cnts\_min 1 ビン当たりのイベント数の下限値。通常、GSC は 10.0、SSC は 0 を指定する。

rebin 出力するライトカーブをビンまとめする場合、まとめるビン数を整数値で入れる。ビンまとめしない場合は 1 を指定する。

### GSC のみの入力パラメータ

gsclist 使う GSC カメラの ID を hex 値 [0123456789ab] で入れる。全 12 カメラを使う時には "0123456789ab"。特に質の良いカメラのみに選別したい時や、悪いカメラのデータを見たいときなどに使う。

hvsel データの HV 設定値での選別。1: 1650V のデータのみ。 2: 1550V のデータのみ。その他: 選別なし。標準解析では 1650 V のデータのみを使用を推奨している。

### SSC のみの入力パラメータ

sschz SSC のみ 使う SSC カメラの ID を [hz] または [01] で入れる。

### GSC の実行スクリプトの例

```
#!/bin/sh

lchistfile=gsc_lchist.root
pibandfname=pibands.list
hvsel=0
gsclist=012456789a
eatime_min=40.0
srctime_min=15.0
bgdtime_min=15.0
cnts_min=10.0
rebin=4

outfname=gsc_lcur_hv${hvsel}.fits

cmdstr="mxg1cur\
```

```

$!lchistfile $outfname $pibandfname $hvsel $gsclist\
$eatime_min $srctime_min $bgdtime_min $cnts_min $rebin"

echo $cmdstr
$cmdstr

```

## SSC の実行スクリプトの例

```

#!/bin/sh
lchistfile=ssc_lchist_pibands.root
pibandfname=pibands.list

#ssclist=hz
ssclist=h
outfname=test_slcur_pibands_${ssclist}.fits

eatime_min=10.0
srctime_min=15.0
bgdtime_min=15.0
cnts_min=0.0
rebin=16

cmdstr="mxslcur \
  $!lchistfile $outfname $pibandfname $ssclist\
  $eatime_min $srctime_min $bgdtime_min $cnts_min $rebin"

echo $cmdstr
$cmdstr

```

### 4.3.5 ライトカーブのプロット

ライトカーブの FITS ファイルは、`fplot` でプロットできる。

```

$fplot crab_glcur.fits
Name of X Axis Parameter[error] [MJD[XAX_E]]
Name of Y Axis Parameter[error] up to 8 allowed[RATE1[ERROR1] RATE2[ERROR2]]
Lists of rows[-]
Device: /XWindow, /XTerm, /TK, /PS, etc[/xw]
Any legal PLT command[-]
...

```

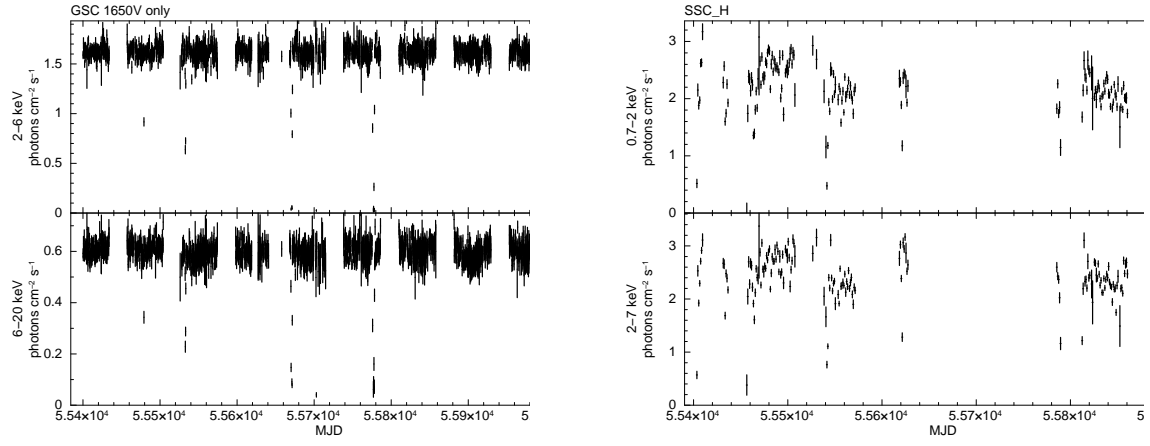


図 4.3: Crab の 2010 年 7 月 23 日 (MJD=55400) から 2012 年 4 月 23 日 (MJD=56040) までの約 1 年半のライトカーブ。(左): GSC 1650V カウンター、(右): SSC\_H。

## 4.4 スペクトル解析

### 4.4.1 ソース、バックグラウンドのエネルギースペクトル (FITS 形式) の抽出

§4.3.1 の説明を参考にして、ソースとバックグラウンドの Region File を作成し、(SKY)XY 座標で保存する。下記のように xselect でソースとバックグラウンドのエネルギースペクトルを作成・保存する。

```
$ xselect
> Enter session name >[gsc4] gsc4
gsc4:SUZAKU > read events crab_gsc4_mjd55501.evt
> Enter the Event file dir >[.] .
Got new mission: MAXI
> Reset the mission ? >[yes] yes
gsc4:MAXI-GSC-32BIT > set XYNAME X Y
gsc4:MAXI-GSC-32BIT > filter region src.reg
gsc4:MAXI-GSC-32BIT > extract spec
gsc4:MAXI-GSC-32BIT > save spec crab_src.pha
Wrote spectrum to crab_src.pha
gsc4:MAXI-GSC-32BIT > clear region

gsc4:MAXI-GSC-32BIT > filter region bkg.reg
gsc4:MAXI-GSC-32BIT > extract spec
gsc4:MAXI-GSC-32BIT > save spec crab_bkg.pha
Wrote spectrum to crab_bkg.pha
```

(SKY)XY 座標で Region File を保存した場合には、選択領域の面積補正パラメータがエネルギースペクトルのファイル (FITS 形式) の BACKSCAL に反映されている。観測時間が違ったり、一般的でない Filter をかけると、後で違った方法で適宜で Exposure や Area Factor の補正をする必要がある<sup>6</sup>。今のところ §4.3.1 で説明したような、同じ検出器上でバックグラウンドを決める方法をエネルギースペクトル解析で使えるようにはなっていない。また、標準解析では GSC は 1650V のデータのみを使用することを推奨している。GSC の HV の変更履歴は”\$CALDB/data/maxi/gsc/bcf/mx\_gsc0\_hvhist\_20100425.fits” などに書いてあるが、分かり易いように表 4.2 に一覧をまとめた。これを見て解析したい時期において、どの GSC カウンタが 1650V で運用しているかを調べる。

### 4.4.2 Exposure weight map の生成

mxdetwmap<sup>7</sup> は §4.3.2 で生成した観測条件ファイル (OBSCUR ファイル) から、カメラ毎の検出器座標系での天体の観測露光時間の Weight Map = Effective Exposure Map [ $\text{cm}^2 \text{s}$ ] を計算する。プログラムは GSC と SSC で共通である。GSC では各カメラの各 HV 毎、SSC では H,Z 毎のテーブルデータユニットを生成する。GSC では、露光時間が 0 の検出器データは保存しない。

入力パラメータは以下の通りである。

infilename §4.3.2 で生成した観測条件ファイル (OBSCUR ファイル)、あるいはそのファイルのリスト (obscur.list) を@obscur.list として指定する。

---

<sup>6</sup>本マニュアルでは解説しない。

<sup>7</sup> mxobswmap から変わり、collimator 座標系ではなく検出器座標での exposure weight map を中間ファイルとして使うようにした。

表 4.2: 2013 年 7 月 30 日における GSC の運用状況 (HV の状況) を記載する。

GSC	1650V(終了時刻)	1550V(開始時刻)	コメント
0	2010-04-08 18:25:00	2010-04-08 18:25:00	1550V で運用中。
1	2010-11-12 15:00:00	2010-11-18 19:30:00	1550V で運用中。
2	2012-04-17 17:05:00	2012-04-17 17:05:00	1550V で運用中。
3	2010-04-08 18:25:00	2010-06-21 11:00:00	2010-04-08~2010-06-21 は停止期間。
4	—	—	1650V で継続運用中。
5	—	—	1650V で継続運用中。
6	2010-04-08 18:25:00	—	停止中。
7	2010-04-08 18:25:00	2010-04-08 18:25:00	1550V で運用中。
8	2010-04-08 18:25:00	2010-04-08 18:25:00	1550V で運用中。
9	2010-04-08 18:25:00	—	停止中。
a	2010-04-08 18:25:00	2010-04-08 18:25:00	1550V で運用中。
b	2010-04-08 18:25:00	2010-04-08 18:25:00	1550V で運用中。

outfname 出力ファイル。

mjd\_start 観測開始時刻 (MJD)。入力ファイルの開始時間より後の時刻を入れた時に有効。0 と入力すると、観測条件ファイルの開始時刻になる。

mjd\_end 観測終了時刻 (MJD)。入力ファイルの終了時間より早い時刻を入れた時に有効。0 と入力すると、観測条件ファイルの終了時刻になる。

theta\_min 入射角  $\theta_{\text{col}}$  の下端。通常  $-1.5^\circ$  を指定する。

theta\_max 入射角  $\theta_{\text{col}}$  の上端。通常  $1.5^\circ$  を指定する。

phi\_min 入射角  $\phi_{\text{col}}$  の下端。余裕を見て  $-50^\circ$  程度を指定する。

phi\_max 入射角  $\phi_{\text{col}}$  の上端。余裕を見て  $50^\circ$  程度を指定する。

#### GSC/SSC の実行スクリプトの例

```
#!/bin/sh

infile="crab_gobscur_mjd55058.fits"
outfile="crab_gwmap_mjd55058.fits"
tstart=0.0
tstop=0.0
theta_min=-2.5
theta_max=2.5
phi_min=-50.0
phi_max=50.0

cmdstr="mxdetwmap $infile $outfile\
$tstart $tstop\
```

```
$theta_min $theta_max\  
$phi_min $phi_max"  
  
echo $cmdstr  
$cmdstr
```

#### 4.4.3 ソーススペクトル用の RMF の生成

Weight Map (Effective Exposure Map) に応じて、異なる入射角に対する RMF File を重みを付けて足し合わせて、データの観測条件に見合った RMF を生成する。同時にソースとバックグラウンドのエネルギースペクトル (FITS 形式) の EXPOSURE Keyword を Weight Map から求まる Effective Exposure ( $\text{cm}^2 \text{s}$ ) へ書き換える。RMF の MATRIX の数値は Efficiency を意味するようになる<sup>8</sup>。プログラムは GSC では `mxgrmfgen`、SSC では `mxsmfgen` を使う<sup>9</sup>。GSC と SSC で入力パラメータが若干異なる。

##### GSC と SSC で共通の入力パラメータ

`wmapfname` §4.4.2 で生成した Exposure Weight Map のファイル。

`specfname` ソースのエネルギースペクトルファイル。なければ NONE を指定する。

`bgdfname` バックグラウンドのエネルギースペクトルファイル。なければ NONE を指定する。

`rmfname` 出力 RMF ファイル。

##### GSC のみの入力パラメータ

`gsclist` 使う GSC カメラの ID を hex 値 [0123456789ab] で入れる。全 12 カメラを使う時には "0123456789ab"。特に質の良いカメラのみに選別したい時や、悪いカメラのデータを見たいときなどに使う。現状、GSC.3 は HV 値にかかわらずエネルギースペクトル解析で使わない方がよい。

`gscrmfdbidx` GSC RMF Database のインデックスファイル名。

`dlpath` Data Downlink Path(0:LOW, 1:MED) を指定する。例えば `$MXDATA/gsc.v1.4/low32` のデータを使うなら 0(LOW) を指定する。

`ldcut` GSC のみ LDcut. 0 (disable) または 1 (enable)。ldcut=1 だと、LD 以下の efficiency を強制的に 0 にする。1650V の GSC カウンタだけのデータでは全く影響が無いので、0 と指定する。1550V のデータでは LD が  $\text{PI}=40(2 \text{ keV})$  を超える場合があり、カウンター毎の稼働モードに応じた LD 値を CALDB から読み込む。1650V と 1550V の混在データを解析する時には ldcut=1 とする。スペクトル抽出の段階で、LD 以下のイベントをカットするなど、まだ特別な手続きが必要である。そのため標準解析では 1650V のデータのみの使用を推奨している。

`hvsel` 1(1650V のみ)、2(1550V のみ)、0(selection なし)。標準解析では 1650V の GSC カウンタのデータのみの使用を推奨しているので、1 を指定する。

##### SSC のみの入力パラメータ

`sschz` 0: SSC-H または 1: SSC-Z。GSC と違い、異なるカメラの RMF を足し合わせることはしない。

<sup>8</sup>将来、この形式は変更の可能性あり。

<sup>9</sup>`mxgscrmfspec`、`mxsscrmfspec` から変わった。入力が検出器上での Weight Map に変更。GSC では、HV 選別、LD 無効/有効のパラメータが加わった。

quantefffile 量子効率の CALDB ファイル。SSC の CALDB がインストールされていれば CALDB と指定する。

rmfparamfile RMF Parameter の CALDB ファイル。SSC の CALDB がインストールされていれば CALDB と指定する。

### GSC の実行スクリプトの例

mxgrmfgen は、検出器上の芯線上の場所毎に生成した RMF Database から、Weight Map に応じて重み付けした RMF File を合成する。RMF Database の取得は §3.2 を参照する。Crab Calibration から求めた低エネルギー側の Effective Area の補正が、この段階で行われる。Software LD に対応するため、低速系/中速系を指定する。

```
#!/bin/sh
wmapfname=crab_gwmap.fits
specfname=crab_gsc_src.pi
bgdfname=crab_gsc_bgd.pi
rmfname=crab_gscrsp.rmf
gsclist=012456789a # exclude GSC_3 and GSC_B
gscrmfdbidx=$GSCRMFDB/gscrmfdbidx.fits
dlpath=1
ldcut=1
hvsel=0

mxgrmfgen $wmapfname $specfname $bgdfname $rmfname $gsclist $gscrmfdbidx $dlpath \
$ldcut $hvsel
```

### SSC の実行スクリプトの例

CCD 検出器の量子効率、応答関数パラメータ校正ファイルから、Weight Map に応じて重み付けした RMF File を合成する。SSC では、HZ カメラで別に RMF File を生成する。

```
infile=crab_swmap.fits
outfile=crab_ssch.rmf
specfname=crab_ssch_src.pi
bgdfname=crab_ssch_bgd.pi
sschz=0
quantefffile=CALDB
rmfparamfile=CALDB

mxsscrmfspec $infile $outfile $specfname $bgdfname $sschz $quantefffile $rmfparamfile
```

#### 4.4.4 モデルフィッティング

適宜 grppha で binning して、xspec でモデルフィッティングを行う。

```
$grppha crab_gscsum_src.pi crab_gscsum_src_bin.pi
GRPPHA[] group min 100
GRPPHA[] exit
...
$ xspec
XSPEC12> read data crab_gscsum_src_bin.pi
XSPEC12> read back crab_gscsum_bgd.pi
XSPEC12> read resp crab_gscsum.rmf
XSPEC12> model wabs*pow
XSPEC12> ignore 0.0-2.0 20.0-**
XSPEC12> notice 2.0-20.0
XSPEC12> fit
...
```

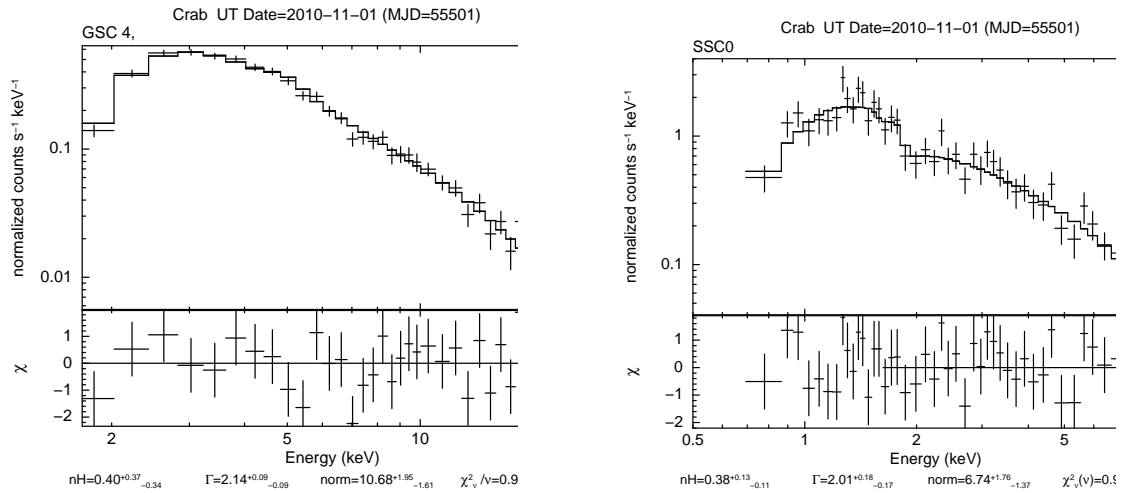


図 4.4: Crab の 2010 年 11 月 1 日のエネルギースペクトル。(左): GSC\_4、(右): SSC\_H。



## 4.5 周期解析

GSC のみ周期解析を行うことが出来る。ここでは Crab Pulsar を例にして、パルサーなどの時間変動する天体の周期解析の方法を説明する。基本的には Headas ftools パッケージの xronos<sup>10</sup> を用いる。

### 4.5.1 天体周辺の Event File の抽出

ライトカーブ解析の §4.3.1 と同じ方法で抽出できる。

### 4.5.2 バリセントリック補正

パルサーは天体ごとに固有のパルス周期を持っている。しかし、天体からのパルス周期は一定だが、MAXI の地球周回や地球の公転など、観測する側の問題でパルス周期は見かけ上変化してしまう。MAXI のデータから正しいパルス周期を求めるためには、バリセントリック補正をかけて太陽系重心座標時刻に変換する必要がある。mxbarycen で太陽系重心座標時刻に変換できる。

```
$ mxbarycen
Input/output file or @filelist to be corrected[crab64.evt]
Orbit file or @orblast of the observation[@orblast.txt]
leapsec file name[Headas/heasoft-6.6.2/i686-pc-linux-gnu-libc2.5/refdata/leapsec.fits]
Right Ascension of target (NNhNNmNNs or NNN.NNN deg)[83.63]
Declination of target (+NNdNNmNNs or +NN.NNN deg)[22.01]
```

現在、確認しているエラーは1つのみである。Event File の GTI にゼロが入っていると途中で落ちる<sup>11</sup>。その場合は、fv で Event File を表示して、GTI のコラムからゼロの行を消すことにより解決できる。

### 4.5.3 パワースペクトル解析

天体からの周期的な変動がある場合には FFT(Fast Fourier Transform) を行くと、特定の周波数にピークが現れる。xronos の powspec を使用してパワースペクトルを作成する。

```
$ powspec
Ser. 1 filename +options (or @file of filenames +options)[1001_1.evt] 1001_1.evt
Name of the window file ('-' for default window)[-] -
Newbin Time or negative rebinning[20] 0.008
Number of Newbins/Interval[8192] INDEF
Number of Intervals/Frame[15] INDEF
Rebin results? (>1 const rebin, <-1 geom. rebin, 0 none)[0] <- そのまま
Name of output file[default] <- そのまま
Do you want to plot your results?[yes] <- そのまま
Enter PGPLOT device[/XW] <- そのまま
```

xronos のデフォルトの値を採用する場合には “INDEF” を指定する。周期的な変動があれば図 4.5 のように出力される。 $(30 \text{ Hz})^{-1} \simeq 0.033 \text{ s}$  なので、この天体 (Crab Pulsar) は約 33 ms の周期的な変動を持つことが分かる。

<sup>10</sup><http://heasarc.gsfc.nasa.gov/docs/xanadu/xronos/xronos.html>

<sup>11</sup> その他のエラーが出た場合にはプログラム作成者へ報告する。

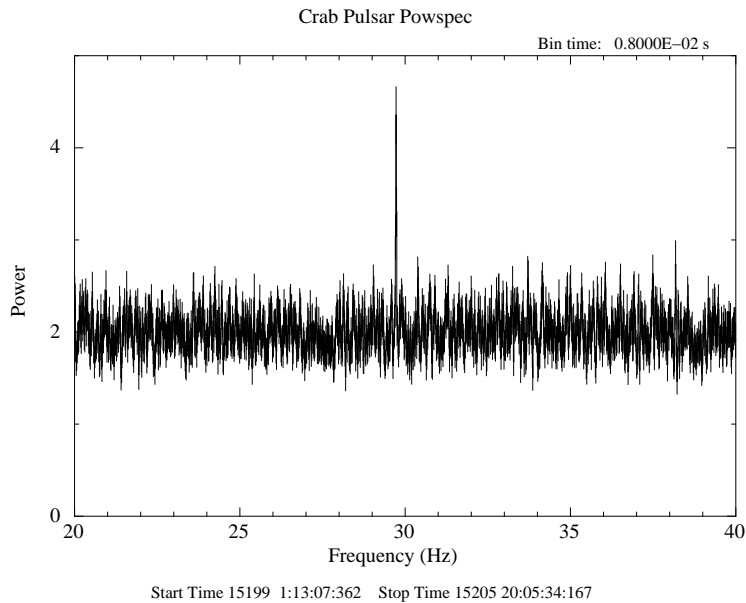


図 4.5: powspec によりパワースペクトルを作成した出力結果。30 Hz 付近にピークが見える。

#### 4.5.4 周期探査

powspec にて、おおまかな周期がわかったので、精度を上げてその周期周辺で周期探査を行う。xronos の efsearch を使用する。Crab Pulsar や 4U 1626 などのように、パルス周期が単調に伸びる天体の場合は、“efsearch dpdot=4.202E-13” というようにパルス周期の減衰率 ( $\text{s s}^{-1}$ ) を指定する必要がある。それ以外の場合は割愛する。

```
$efsearch dpdot=4.202E-13
Ser. 1 filename +options (or @file of filenames +options)[1001_1.evt] 1001_1.evt
Name of the window file ('-' for default window)[-] -
Epoch[0.1515500000E+05] INDEF
Period[10] 0.033
Phasebins/Period {value or neg. power of 2}[10] INDEF
Number of Newbins/Interval[38110] INDEF
Resolution for period search {value or neg. power of 2}[1E-2] 1e-10
Number of periods to search[256] 1024
Name of output file[default] <- そのまま
Do you want to plot your results?[yes] <- そのまま
Enter PGPLOT device[/xw] <- そのまま
```

正しい周期でたたみ込んでいれば、図 4.6 のように、ある  $\chi^2$  の値において、非常に強いピークが現れる。ただし、1 スキャン ( $\sim 50$  s) から 1 周回 ( $\sim 90$  min) の間にパルス周期がある場合、 $(90 \text{ min})^{-1}$ ,  $2(90 \text{ min})^{-1}$ , ... などの周期で偽者のピークが立つ可能性があるので、結果の吟味には注意する。

#### 4.5.5 パルス波形の導出

パルス周期がわかったので、周期の中で天体がどのような波形で変動しているのかを調べる。xronos の efold を使用して、バリセントリック補正をかたライトカーブをパルス周期で畳み込む。§4.5.4 の efsearch と同様に、必要に応じて “ddot” のオプションを指定する。

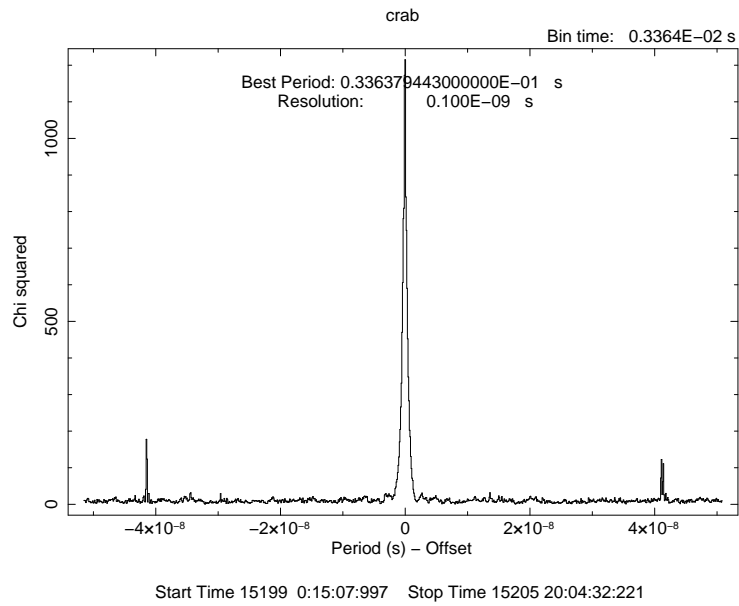


図 4.6: efsearch による周期探索の出力結果。

```
$efold dpdot=4.202E-13
Number of time series for this task[1] 1
Ser. 1 filename +options (or @file of filenames +options)[kouhan.evt] 1001_1.evt
Name of the window file ('-' for default window)[-] -
Epoch[0.1515500000E+05] INDEF
Period[249.01] 0.033
Phasebins/Period {value or neg. power of 2}[32]
Number of Newbins/Interval[121947] INDEF
Number of Intervals/Frame[1] INDEF
Name of output file[default] <- そのまま
Do you want to plot your results?[yes] <- そのまま
Enter PGPLOT device[/XW] <- そのまま
```

パルス周期が正しければ、図 4.7 のように天体に特有のパルス波形が出力される。

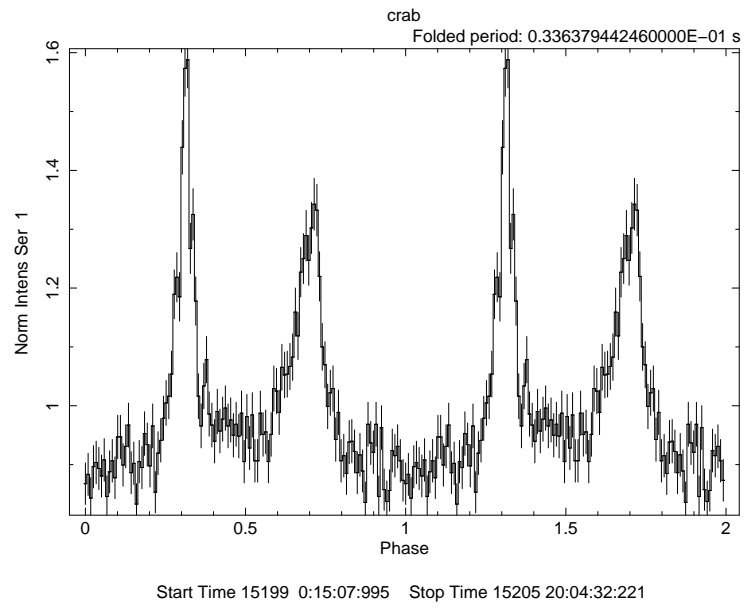


図 4.7: efold により、バリセントリック補正をかけたライトカーブをパルス周期で畳み込んだ出力結果。

# 付 録 A オプション・ソフトウェア

標準解析では必要としていないソフトウェアのインストールについて説明する。

## A.1 外部ソフトウェアのインストール

**Postgresql 8.1 or 8.2** – <http://www.postgresql.org/ftp/source>

```
$ tar zxvf /opt/maxi/src/postgresql-8.1.23.tar.gz
$ cd postgresql-8.1.23
$ ./configure --prefix=/opt/maxi/local_20130505
$ make
$ make install
$ cd ../
```

**fftw 3.3.2** – <ftp://ftp.fftw.org/pub/fftw/fftw-3.3.2.tar.gz>

```
$ tar zxvf fftw-3.3.2.tar.gz
$ cd fftw-3.3.2 $ ./configure --prefix=/opt/maxi/local_20130505 --enable-shared
$ make
$ make install
$ cd ../
```

**gsl 1.15** – <ftp://ftp.gnu.org/gnu/gsl/gsl-1.15.tar.gz>

```
$ tar zxvf gsl-1.15.tar.gz
$ cd gsl-1.15
$ ./configure --prefix=/opt/maxi/local_20130505
$ make
$ make check
$ make install
$ cd ../
```

**HEAsoft 6.12** – <ftp://heasarc.gsfc.nasa.gov/software/lheasoft/lheasoft6.12/heasoft-6.12src.tar.gz>

ソースからコンパイルする。またパッケージは、少なくとも ASCA、Xspec を選択すれば良い。

```
$ cd /opt/maxi/local_20130505/install_work $ tar zxvf /opt/maxi/src/heasoft-6.12src.tar.gz
$ cd heasoft-6.12/BUILD_DIR $ ./configure >& config.out
$ make >& build.log
```

```
$ make install >& install.log
```

**ROOT 5.34.07** – [ftp://root.cern.ch/root/root\\_v5.34.07.source.tar.gz](ftp://root.cern.ch/root/root_v5.34.07.source.tar.gz)

mxkwtool を使用する場合は、以下のようにオプションを与えて configure を実行する。

```
$ cd /opt/maxi/local_20130505/install_work $ tar zxvf /opt/maxi/src/root_v5.34.07.source.tar.gz
$ mv root root_v5.34.07
$ cd root_v5.34.07
$ ./configure --enable-minuit2 --enable-mathmore --enable-fftw3
  --with-fftw3-incdir=/opt/maxi/local_20130505/include
  --with-fftw3-libdir=/opt/maxi/local_20130505/lib
  --with-cfitsio-incdir=$HEADAS/include
  --with-cfitsio-libdir=$HEADAS/lib
  --with-gsl-incdir=/opt/maxi/local_20130505/include
  --with-gsl-libdir=/opt/maxi/local_20130505/lib
  --with-python-incdir=/opt/maxi/local_20130505/include/python2.7
  --with-python-libdir=/opt/maxi/local_20130505/lib
$ make
$ source /opt/maxi/local_20130505/install_work/root_v5.34.07/bin/thisroot.csh
/opt/maxi/local_20130505/install_work/root_v5.34.07
```

## A.2 MAXI ソフトウェアのインストール

### mxdbtool 2.0a

```
$ cd /opt/maxi/mxsoft_20130505 $ mkdir mxdbtool
$ cd mxdbtool
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/mxdbtool/branches/2.0a_c
$ cd 2.0a_c
$ ./configure
$ make
$ make install
```

### mxkwtool 2.1.2

一部の wcs に関するライブラリは HEAsoft 6.13 に含まれていないので、configure を実行する際に HEAsoft 6.12 を使用するように指定する。

```
$ cd /opt/maxi/mxsoft_20130505 $ mkdir mxkwtool
$ cd mxkwtool
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/svnrepo_mission/common/mxkwtool/branches/2.1.2_c
$ cd 2.1.2_c
$ ./configure
```

```
--with-wcs-libdir=/opt/maxi/local_20130505/install_work/heasoft-6.12/x86_64-unknown-linux-gnu-libc2.12/lib
$ make
$ make install
$ pwd
/opt/maxi/mxsoft_20130505/mxkwtool/2.1.2_c
$ cd ../../
$ svn export --username=USERNAME http://www.maxi.jaxa.jp/motoko_maxirepos/grbtool/catalog
```

mxkwtool を実行するために以下の環境変数を設定する。

```
$ setenv PATH $MXSOFT/mxkwtool/2.1.2_c/script:$PATH
$ setenv MXKWTOOL $MXSOFT/mxkwtool/2.1.2_c
$ setenv MXKW_TMPDIR $HOME/maxi/tmp
$ setenv MXKW_DS9_BIN_DIR /opt/maxi/local_20130505/bin
$ setenv MXKW_MXFILE_BUSY_FLAG $HOME/maxi/exec.flag_rsync_jax_data_from_riken
$ setenv MXKW_GRBTOOL_CAT $MXSOFT/catalog/weblistv6_detect091209_merge.t.csv
$ setenv MXKW_GRBTOOL_CAT_LOCAL $HOME/maxi/srcinfo.list
$ setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:/opt/heasoft/6.12/x86_64-unknown-linux-gnu-libc2.12/lib
```

## 付 録 B 役に立つツール

### B.1 時刻変換

`mxtimecalc` を使うと MAXI, DTPC, GPS, MJD, UTC, JST、それぞれの間の時刻の変換ができる。

```
$ mxtimecalc dptc 940808110
...
DPTC Auxiliary file=/nfs/m/maxir1/maxiuser/mxdata_jax.v1/auxil/dptclist.fits
DPTC list is FITS-format.
MAXITIME = 310088096.885635
DPTC      = 940808110.000000
GPS       = 940808109.885635
MJD       = 55132.982580
UTC Time  : 2009-10-28T23:34:54.885635
JST Time  : 2009-10-29T08:34:54.885635
```

### B.2 Processed Event File の再処理

`mxgevt2evt` を使って、地上データ処理をやり直して、最新版のソフトウェア、CALDB に相当する Processed Event File にアップデートすることができる (動作確認は十分していない)。32-bit の場合、GSC-E FRC List は無視される。

```
$ mxgevt2evt
Name of input event file[crab_gscall.evt]
Name of output event file[crab_gscall_new.evt]
List of attitude file[/nfs/m/maxidata/jax_data/auxil_jaxa/attitude/attlist_local.txt]
List of dptc file[/nfs/m/maxidata/jax_data/auxil_jaxa/dptc/dptclist_local.txt]
Nevnets[10000] -1
phamode(0/32/64). 0: default (input data), 32: 32-bit evt mode, 64: 64-bit evt mode[0]
List of sys-a dptc-frc list file[/nfs/m/maxid2/jax_data/low/gscfrc/geafrcfile.fits]
List of sys-b dptc-frc list file[/nfs/m/maxid2/jax_data/low/gscfrc/gebfrcfile.fits]
...
```



## 付 録 C 位置決め

mxkwtool を用いて、GSC の複数カメラ、複数スキャンを用いた位置決めの方法を説明する。ここでは Flash Report から取得した Event File を使用する。

### C.1 Event File の取得

2 次元フィッティングを行うために、Flash Report のデータをロカール PC に保存し、Rev 1.0 への Reprocess を行い、ROOT File への変換を行う必要がある。これらの作業を行うスクリプトが mxkwtool に含まれているので、以下のように実行する。

CSH の場合の例

```
set scriptdir = $MXSOFT/mxkwtool/2.1.2/script

set novaid = 6502776009
set usr = {FLASH REPORT の username}
set pass = {FLASH REPORT の password}

$scriptdir/get_flash_report_files.py $novaid $usr $pass |& tee getdata.log

cd $novaid
```

BASH の場合の例

```
scriptdir=$MXSOFT/mxkwtool/2.1.2/script

novaid=6502776009
user={FLASH REPORT の username}
pass={FLASH REPORT の password}

$scriptdir/get_flash_report_files.py $novaid $user $pass

cd id$novaid
```

スクリプトが正しく動いていれば、id\${novaid} というディレクトリが作成されるので、以降、id\${novaid} に居るとして説明するので、このディレクトリへ移動する。fits.list に scan/fits 以下にある 1 スキャン毎の Event File のリストができる。同様に att.list に att 以下にあるファイルのリスト、dptc.list に dptc 以下にあるファイルのリストが作成される。

## C.2 位置決め手順

複数カメラ、複数スキャンで位置決めを行う場合について手順を説明する。原理としては、1 カメラ、1 スキャンの結果を足し合わせている。位置決めを行うにあたり、§3.6 で示した環境変数の設定を行う。

### CSH の場合の例

```
set bindir = $MXSOFT/mxkwtool/2.1.2/script

set srcname = Aql_X-1
set ra = 287.816875
set dec = 0.584944
set rad = "auto"          # catalog を作成する半径 (deg) を指定する。auto だと 5 deg。
set catfile = cat.txt     # 出力ファイル名。
set regfile = cat.reg     # 出力ファイル名。
set mode = sky            # sky 座標 (天球に対する接平面) 上で位置決めするという意味。

$bindir/mk_catalog.pl \
. \
$srcname \
$ra $dec $rad \
$catfile \
$regfile \
$mode
```

### BASH の場合の例

```
bindir=$MXSOFT/mxkwtool/2.1.2/script

srcname=Aql_X-1
ra=287.816875
dec=0.584944
rad="auto"          # catalog を作成する半径 (deg) を指定する。auto だと 5 deg。
catfile=cat.txt     # 出力ファイル名。
regfile=cat.reg     # 出力ファイル名。
mode=sky            # sky 座標 (天球に対する接平面) 上で位置決めするという意味。

$bindir/mk_catalog.pl \
. \
$srcname \
$ra $dec $rad \
$catfile \
$regfile \
$mode
```

この段階で生成された “\$catfile” (cat.txt) をチェックして、不要な天体を除く。行頭の一文字を “#” にすることにより除くことも出来る。もし不要な天体を除いた場合は、インデックス (行頭の数字) を付け直す。ただし、最後にある “! bg” で始まる行は消さないようにする。

ROOT 形式に変換された Event File のリストは “root.list” に書かれている。これをそのまま使用するので、中身が書かれているか確認する。

#### CSH の場合の例

```
set procdir = .
set root_list = root.list

set dptcfname = "auto"
set attlistfile = "auto"

set nbin_delta_sky = "auto"
set fwidth_delta_sky = "auto"

set sel_ene_lo = 4.0
set sel_ene_up = 10.0

${bindir}/detpos_sky.pl \
$procdir \
$srcname \
$root_list \
$catfile \
$dptcfname \
$attlistfile \
$nbin_delta_sky \
$fwidth_delta_sky \
$sel_ene_lo \
$sel_ene_up \
|& tee detpos_sky.log
```

#### BASH の場合の例

```
procdir=.
root_list=root.list

dptcfname="auto"
attlistfile="auto"

nbin_delta_sky="auto"
fwidth_delta_sky="auto"

time ${bindir}/detpos_sky.pl \
```

```
$procdir \  
$srcname \  
$root_list \  
$catfile \  
$dptcname \  
$attlistfile \  
$nbin_delta_sky \  
$fwidth_delta_sky \  
| tee detpos_sky.log
```

前述の “detpos\_sky.pl” が動いている間に、途中経過を知りたい場合には、以下を実行する。

### CSH,BASH の例

```
${bindir}/merge_cont_mklist.pl \  
$procdir  
  
${bindir}/merge_cont.pl \  
$procdir \  
$catfile \  
$srcname
```

## C.3 位置決め結果のチェック

Web ブラウザで、list.html を開く。“detpos\_gpev\_id\${novaid}\_gti01\_0000\_04” などへのリンクが見られる。また、各スキャン毎の Contour Plot を確認できる。“\$srcname\_cam[CAMERA-ID]\_objs.png” には、Catalog File の 0 番目の天体の初期位置が × 印で表示される。周囲の天体は、○1、○2、○3 のように表示されるので、Catalog File と天体の数が一致しているかどうか確認する。Catalog File の編集をミスすると、これらの印が正しく表示されない場合がある。

Web ブラウザで “merge\_\${srcname}/pnglist.html” を開く。“\${srcname}\_cont\_add.png” は、全スキャンの  $c\text{-stat}(c\text{-stat} = -2\ln(\text{Likelihood}))$  の 2 次元ヒストグラムを足し合わせたものである。2 次元フィッティングが妥当であれば、ある部分において  $c\text{-stat}$  が小さくなっており、その外側へ向かうにつれて円形上に  $c\text{-stat}$  が大きくなっていくはずである。そのようになっているかチェックする。また、“\${srcname}\_cont\_good\_0000.png” などのファイルは各スキャン、各カメラの  $c\text{-stat}$  の 2 次元ヒストグラムであり、同様に妥当であるか確認する。ここで “\${srcname}\_cont\_good\_0000.png” などのファイルの数字が何を表すか説明する。複数スキャン、複数カメラを使用している場合、例えば 3 周回、および cam4 と cam5 のデータを使用している場合、“0000” は 1 周回目の cam4 であり、“0001” は 1 周回目の cam5 である。同様に “0002” は 2 周回目の cam4 というように並んでいる。

各スキャンのフィッティング結果を確認して、妥当でないスキャンがある場合は、それを取り除く必要がある。“cont\_hist.list” と “valid\_hist.list” にスキャンごと、カメラごとのファイルがリストされているので、妥当でないスキャンを “#” でコメントアウトする。その後、“merge\_cont.pl” を実行することにより、妥当でないスキャンを除いて、全スキャンを足し合わせた 2 次元ヒストグラムを作成できる。

## CSH,BASH の例

```
emacs cont_hist.list &  
emacs valid_hist.list &  
  
${bindir}/merge_cont.pl \  
$procdir \  
$catfile \  
$srcname
```

# References

1. Heasoft Home page <http://heasarc.nasa.gov/docs/software/lheasoft/>
2. CALDB Home page [http://heasarc.nasa.gov/docs/heasarc/caldb/caldb\\_intro.html](http://heasarc.nasa.gov/docs/heasarc/caldb/caldb_intro.html)
3. MAXI 低速系テレメトリデータ
4. MAXI 中速系テレメトリデータ