

# FITS I/O library "SFITSIO" version 1.2.0 の開発・公開

○山内 千里 (NAOJ, ISAS)  
chisato.yamauchi@nao.ac.jp

松崎 恵一 (ISAS)

石原 大助 (名古屋大学)

<http://www.ir.isas.jaxa.jp/~cyamauch/sli/>

SFITSIO は、ゼロからの新規開発による C/C++用の FITS I/O ライブラリである。C++のクラスによって FITS の構造を表現し、マニュアルの頻繁な参照を必要としない直感的なコーディングを可能にした。オンメモリ処理を基本とし、ディスク I/O の最小化とライブラリ側での完全なメモリ管理により、プログラマの負担を極限まで小さくする事を目標としてきた。

2012年4月にリリースした version 1.2.0 では、画像処理 API の拡充、各種高速化、ヘッダ・バイナリテーブルに関する API の機能強化を行なった。今回のバージョンは、宇宙研の C-SODA が SFITSIO を使って開発している「L1TSD TLM2FITS(本年会 W58b)」にあわせたもので、TLM2FITS 向けの拡張としては、FITS テンプレートへの対応、CFITSIO 互換のチェックサム、バイナリテーブルの可変長配列のサポート等があげられる。現在、複数のプロジェクトで SFITSIO の採用が検討されている。このライブラリがどのような用途に適しており、どのように開発効率を高めているかをご覧ください。ただければ幸いです。

## ユーザプログラムの例

### 課題 A: バイナリテーブルへのアクセス

複数のバイナリテーブル HDU を持つ FITS ファイル「foo.fits.gz」がある。それを読み込んで、バイナリテーブル「TBL-A」のカラム「X」の最初の行の値が NULL 値ではなく、かつバイナリテーブル「TBL-B」のカラム「Y」の最初の行の値が 0 を越える場合に、変数 flag に 1 をセットするコードを作成せよ。ただし、ヘッダには TZERO<sub>n</sub>, TNULL<sub>n</sub> がセットされているのでその事も考慮する必要がある。

### 課題 A の解答例

```
fitscc fits; ←FITSを表現するオブジェクト(インスタンス)
long row_idx = 0; ←テーブルの行番号
double v0, v1;
/* ファイルを読む */
fits.read_stream("foo.fits.gz"); ←FITSファイルの全内容をオブジェクトにぶちこむ
/* 値を読んでテスト */
v0 = fits.table("TBL-A").col("X").dvalue(row_idx); ←APIがFITSの構造を素直に表現している事に注目
v1 = fits.table("TBL-B").col("Y").dvalue(row_idx); ←dvalue()はdoubleに変換された値を返す
if ( isfinite(v0) && isfinite(v1) && 0 < v1 ) flag = 1;
/* カラムのデータ型が何であってisfinite()でNULLチェックが可能
```

### 課題 B: 簡単なリダクション

処理済みのバイアスフレーム「bias.fits.gz」、フラットフレーム「flat.fits.gz」と、未処理の観測フレーム「obj1.fits」がある。観測フレームについて、オーバースキャン(平均値)引き、バイアス引き、フラット補正をした後、画像の上 1/4 と下 1/4 を削除した FITS を作成せよ。オーバースキャンの領域情報は、ヘッダの BIASSEC キーワードの値に IRAF 準拠の形式で保存されている。ただし、観測フレームは 16-bit 整数型で保存され、ヘッダには BZERO がセットされているのでその事も考慮する必要がある。

### 課題 B の解答例

```
fitscc bias_fits, flat_fits, obj1_fits;
long y_len, sect[4];
double v_stat[1];
/* バイアス、フラット、観測フレームを読む */
bias_fits.read_stream("bias.fits.gz"); ←圧縮ファイル(.gz, .bz2)もそのまま読込OK
flat_fits.read_stream("flat.fits.gz");
obj1_fits.read_stream("obj1.fits");
/* 長いので alias を定義 */
fits_image &obj1pri = obj1_fits.image("Primary"); ←image HDUの場合は、image() でアクセス
/* データ型の変換、overscan 情報の取得、 */
obj1pri.convert_type(FITS::DOUBLE_T); ←sect[0]-sect[3]に領域情報が返る
obj1pri.header("BIASSEC").get_section_info(sect, 4);
/* overscan 領域の平均値を求める */
obj1pri.stat_pixels(v_stat, 1, "results=mean"); ←v_stat[0]に統計値が返る。IRAFのimstatと同様、median等も取得可能
/* overscan(平均値)引き、バイアス引き、フラット補正 */
obj1pri.subtract(v_stat[0]);
obj1pri.subtract(bias_fits.image("Primary")); ←C言語の「/=」と同じ感覚。一つの値or画像と演算が可能(データ型は一致してなくて良い)
obj1pri.divide(flat_fits.image("Primary"));
/* 画像の縦の長さを求め、上下 1/4 を削除して保存 */
y_len = obj1pri.row_length();
obj1pri.data_array().crop(1, y_len/4, y_len/2); ←引数は、次元番号、開始点、長さ
obj1_fits.write_stream("obj1_done.fits");
```

各解答例では、各種エラーチェックと#include から main 関数までのコードは省略している。

### こっそり情報

SFITSIO version 1.3.0 では pthread を使った高速化とマルチスレッドに対応した API を追加する予定。

## 1 概要

### 開発の動機

「多数の HDU からなる複雑な構造を持つ FITS ファイル」を扱う「複雑なデータ処理」のためのコード開発を効率化したい。「あたり」のデータ処理がキッカケ。

### 設計思想

IDL の AstroLib と同様のオンメモリ処理を前提。C++特有の作法を使わず、C 言語の良さを継承した API。  
✗ cout << "foo" << endl; printf("foo");

### ライブラリの位置付け

中~高レベルライブラリ。かなり高度な処理も可能。オブジェクト内メモリへの直接アクセスも可能。

### 想定しているユーザ

C 言語の経験がある方。C++の経験は無くても OK。

### 扱う事ができる HDU, FITS 規約, WCS

Image, ASCII・Binary Table, CONTINUE, CHECKSUM などの非公式規約に対応。WCSTools との連携で多彩な WCS に対応。

### サポート・ライセンス

C-SODA/ISAS の公式サポート。MIT ライセンス。

## 2 御利益

「API の形」が「FITS の構造」を反映しており、API が覚えやすい。多数の HDU を含む FITS ファイルでも、日本語で考えたことおりの「1 行のコード」で目的のデータにアクセス。

BSCALE, BZERO, BLANK 等の変換処理はすべてライブラリ側で行なうため、API は必要最小限の引数しか持たない。

プログラマによるメモリ管理 (malloc ~ free) は不要。これもライブラリ側で行なう。恐怖のメモリリークとおさらば!

HDU やテーブルへは「名前」でも高速にアクセス可能。

データ編集のための API が充実。画像の演算、テーブルの結合、一部削除等も関数呼び出し一発で OK。

ネットワーク経由の圧縮ファイルでも直接アクセス可能。

FITS テンプレートに対応。ヘッダコメント辞書を搭載。

日本語マニュアルを完備。親切なチュートリアル。

C/C++だから、特殊処理の高速化も容易。

FITS I/O のコード書きを可能な限り脇役に追いやり、プログラマはデータ処理の本質的な部分に集中できる

## 3 採用事例

- JAXA 宇宙科学研究所 (C-SODA, 赤外, 月惑星 等) L1TSD TLM2FITS (本年会 W58b)、「かぐや」HDTV 画像の加工, AKARI FIS 全天マップ作成プロジェクト, 他
- 東広島天文台 HONIR, ASTE 望遠鏡, 他 で採用を検討中。

## References

- Yamauchi, C., 2010, ASPC, 434, 469Y
- Yamauchi, C. et al. 2010, SLLIB-1.1.0 ユーザーズガイド
- Yamauchi, C. et al. 2012, SFITSIO-1.2.0 ユーザーズガイド