

## Postprint: Applied Research on FITS Variable-Length Arrays for Dark Matter Satellite Data Storage

**Authors:** Xiaoyan Yang, Shi Tao, Li Bing, Sun Xiaojuan, Lu Xiaojun

**Date:** 2018-05-29T00:00:00+00:00

### Abstract

FITS is a widely utilized data format in the field of space astronomy, wherein space astronomy data files are typically stored as FITS files employing fixed-length data structures. As China's first launched space science satellite, the Dark Matter Particle Explorer (DAMPE) satellite's scientific data packets are characterized by variable lengths. In the DAMPE satellite data processing workflow, there exists an urgent requirement to design storage structures and processing methodologies that support variable-length data. This study proposes a FITS format that supports variable-length arrays and implements data processing, storage, and management based on this data structure. When applied to the DAMPE satellite ground processing system, verification results indicate that this approach achieves edit-level product data storage, substantially reduces product data volume, conserves storage space, and concurrently improves processing efficiency.

### Full Text

#### Application Research of FITS Variable-Length Arrays in DAMPE Data Storage

Yang Xiaoyan<sup>[1,2]</sup>, Shi Tao<sup>[1,2]</sup>, Li Bing<sup>[1,2]</sup>, Sun Xiaojuan<sup>[1,2]</sup>, Lu Xiaojun<sup>[3]</sup>

1. Key Laboratory of Technology in Geo-spatial Information Processing and Application System, Chinese Academy of Sciences, Beijing 100190, China
2. (Affiliation not specified in original)
3. China International Engineering Consulting Corporation, Beijing 100048, China

---

## Abstract

FITS (Flexible Image Transport System) is a data format widely used in space astronomy, where astronomical data files are typically stored using fixed-length data structures. As China's first space science satellite, the Dark Matter Particle Explorer (DAMPE) satellite produces science data source packets with variable lengths. The DAMPE data processing pipeline urgently requires a storage structure and processing method that supports variable-length data. This paper designs a FITS format that supports variable-length arrays and implements data processing, storage, and management using this data structure. Applied to DAMPE ground processing, validation results demonstrate that this method successfully enables editing-level product data storage, significantly reduces product data volume, conserves storage space, and improves processing efficiency.

**Keywords:** FITS; variable-length data structure; space astronomy; DAMPE

---

## 1 FITS Data Format

FITS has achieved widespread adoption in international astronomy due to its self-describing nature and flexibility. A standard FITS file consists of a primary HDU and a number of extension HDUs, each comprising a header unit and a data unit. The primary HDU's header contains general descriptive information such as satellite name and production date, supports extended definitions, and has an empty data unit. Extension HDU headers contain metadata including start/end times, reference coordinate systems, and column data types, also supporting extended definitions, while data units store information in fixed-length ASCII or binary tables. illustrates the standard FITS file structure.

The data unit stores two-dimensional arrays as ASCII or binary tables with fixed row lengths and consistent column data types. FITS supports logical (L), bit (X), byte (B), 16-bit integer (I), 32-bit integer (J), 64-bit integer (K), single-precision float (E), and double-precision float (D) data types. Additionally, FITS flexibility enables storage of variable-length arrays through special keywords in the header that specify offset positions and total data lengths. The actual variable-length array data is not stored in the data unit but in a heap area following it.

---

## 2 DAMPE Data Characteristics and Storage Requirements

The Dark Matter Particle Explorer (DAMPE) is one of the first five scientific satellites selected under the Chinese Academy of Sciences' Strategic Priority

Program on Space Science. Launched in December 2015, DAMPE aims to detect dark matter particles through high-precision measurements of high-energy electron and gamma-ray spectra and their spatial distributions, searching for and studying the possible cosmic distribution of dark matter particles.

Scientific observation data from DAMPE' s payloads and engineering data collected by the satellite platform are transmitted to ground via telemetry channels. After processing through frame synchronization, virtual channel separation, source packet extraction, verification and sorting, deduplication, and splicing/splitting, the data is output as editing-level products in FITS format for archival storage and distribution to the scientific application system. The main processing flow is shown in [Figure 1: see original paper].

In DAMPE telemetry data, engineering data representing payload and platform status has fixed lengths and can be stored using conventional fixed-length methods. However, science data source packets consist of 30 data frame queues with variable total lengths. The source packet structure and maximum lengths of each data frame are shown in .

With approximately 600,000 science data source packets transmitted per orbit, storing each at its maximum length of 20,998 bytes would produce single files of 11.73 GB. In reality, source packet lengths vary significantly depending on payload detection modes and particle hit statuses, with onboard data compression further reducing sizes. Particularly for the silicon array detector with over 70,000 channels, most channels remain inactive without major hit events, outputting no scientific data. Storing at maximum length would substantially inflate DAMPE science data files, increasing processing difficulty, reducing efficiency, and wasting storage resources.

Addressing these characteristics and storage requirements, a FITS file storage scheme supporting variable-length data must be designed and implemented for DAMPE.

---

### 3 FITS Variable-Length Storage Scheme Design and Implementation

Given the substantial length variations in DAMPE science data source packets, we designed a storage scheme tailored to its data characteristics and requirements, implementing it in C++ using CFITSIO. The design comprises the following steps:

**(1) Header Modification.** In conventional fixed-length storage schemes, the header keyword `tform` specifies data types (L, X, B, I, J, etc.). For DAMPE' s variable-length data, `tform` is assigned special types `rPt(emax)` or `rQt(emax)`, where `r` is a repeat count (0, 1, or omitted), `P` and `Q` are array descriptor types representing 32-bit and 64-bit signed integers respectively, `t` is the data type code, and `emax` is the maximum byte length. For DAMPE data, we set

`tform=1QB(emax)`, indicating 64-bit signed integer descriptors with byte-type entity data, where `emax` is extracted from actual data.

Additional keywords include `theap`, specifying the heap area start position (defaulting to the data unit length if omitted), and `pcount`, representing the sum of gap size and heap area size. In our scheme, `theap` uses the default value, while `pcount` equals the science source packet byte count, extracted from actual data.

**(2) FITS File Creation via `fits_create_tbl`.** The function is called as:

```
fits_create_tbl(fitsfile *fptr, int tbltype, LONGLONG naxis2, int tfields, char *ttype[], ch
```

Where `fptr` is the target DAMPE editing-level product FITS file; `tbltype` specifies the table type (`ASCII_TBL` or `BINARY_TBL`), with DAMPE using binary tables; `naxis2` is the total number of science source packet rows, extracted from actual data; `tfields` is the parameter count (only 1 parameter column for CCSDS source packets); `ttype` specifies parameter names (CCSDS); `tform` sets the parameter type as `1QB(emax)`; `tunit` defines units (none for CCSDS source packets); and `extname` names the extension HDU (`Sci_Src`).

**(3) Array Descriptor Writing.** Array descriptors form an  $N \times 2$  matrix where  $N$  is the total number of variable-length array rows. Column 1 defines each row's data length, and column 2 defines the offset of each row's start position within the heap area, both stored as positive integers in the data unit. Our scheme creates two  $N$ -dimensional index arrays: `index_len[N]` stores each row's length, and `index_offset[N]` stores each row's start position offset, both extracted from actual data.

**(4) Variable-Length Array Data Writing to Heap.** Variable-length array entities are stored row-by-row in the heap area. With `theap` at its default value, storage begins at the byte immediately following the data unit terminator. Based on descriptor values confirming each row's start position and length, the `fits_write_col` function writes DAMPE science source packet data row-by-row into the FITS file heap area:

```
int fits_write_col(fitsfile *fptr, int datatype, int colnum, LONGLONG firstrow, LONGLONG fi
```

Where `fptr` is the target DAMPE editing-level product FITS file; `datatype` is set to `TBYTE` for byte-wise writing; `colnum` starts from row 1; `firstrow` is the starting row number (1-based); `firstelem` is the first element of the row; `nelements` is the row data length from `index_len[N]`; and `*array` points to the CCSDS source packet data location at `p+index.index[N].offset`, with `p` as the position pointer.

Under this scheme, the extension HDU storage structure in DAMPE data FITS files is:

[Header] [Data Unit] Stores DAMPE data array [heap] Stores DAMPE science source packages

## 4 Performance Validation

To evaluate performance in file size, storage efficiency, and processing speed, we analyzed editing-level product files from five orbits of DAMPE telemetry data (July 25, 2017 - orbit 8923; August 30, 2017 - orbit 9478). compares file sizes between fixed-length and variable-length FITS storage schemes.

---

## 5 Results and Discussion

Validation with real data demonstrates that the proposed FITS-based variable-length array storage scheme reduces file data volume by approximately 88% for DAMPE' s product storage characteristics, dramatically saving storage space. The reduced file size significantly improves data processing speed, product archival speed, and application efficiency.

This FITS-based variable-length array storage scheme can be extended to other astronomical data with varying lengths, effectively reducing storage requirements depending on the average ratio of actual data length to maximum data length.

### References

- [1] Hanisch R J, Farris A, Greisen E, et al. Definition of the Flexible Image Transport System (FITS) [J]. *Astronomy & Astrophysics*, 2001, 376: 359-380.
- [2] Murphy T, Lamb P, Owen C, et al. Data storage, processing, and visualization for the Australia telescope compact array [J]. *Publications of the Astronomical Society of Australia*, 2006, 23(1): 2.
- [3] Zhao Haisheng, Ge Mingyu, Li Zhengheng, et al. A method of data pre-processing for astronomical satellite [J]. *Astronomical Research & Technology*, 2017, 14(6): 376-381.
- [4] Cui Chenzhou, Zhao Yongheng. Architecture of Chinese Virtual Observatory[J]. *Astronomical Research & Technology –Publications of National Astronomical Observatories of China*, 2004, 1(2): 140-151.
- [5] Cui Chenzhou, Li Wen, Yu Ce, et al. Search and location of FITS data files [J]. *Astronomical Research & Technology –Publications of National Astronomical Observatories of China*, 2008, 5(2): 116-123.
- [6] Zhang Hailong, Ye Xinchun, Li Huijuan, et al. Astronomical data query and release review[J]. *Astronomical Research & Technology*, 2017, 14(2): 212-228.

*Note: Figure translations are in progress. See original paper for figures.*

*Source: ChinaXiv –Machine translation. Verify with original.*