GPM/TRMM
データ読み込みプログラムガイド
(FORTRAN 編)

2026/06/01

第九版

本書は全球降雨観測衛星(GPM)のデータを読み込むプログラム（FORTRAN）の作成方法についてまとめたものです。
本書で解説するサンプルプログラムは、GPM/TRMM はプロダクトバージョン０８、GSMaP はプロダクトバージョン０５で動作を確認しています。

目次

1. はじめに

本書は GPM/TRMM データに対して FORTRAN 言語を用いて読み込む方法について解説します。
GPM 及び TRMM はバージョン 06 プロダクト（TRMM バージョン 8 相当）からフォーマットを統一しており、最新のアルゴリズムはバージョン 08 となっています。本サンプルプログラムにて同様に読むことができます。
GPM データを読み込むには FORTRAN の他にも表 1.1 に示すような方法があります。どの方法で読み込むかについては、次頁の「読み込み方法判断フロー」を参考にして判断してください。
また、本資料で使用しているサンプルプログラムの動作を確認したＯＳの一覧を表 1.2 に示します。

表 1.1 データ読み込み方法

	データ読み込み方法	資料名	備考
1	THOR を使用する	GPM/TRMM データ読み込みプログラムガイド(THOR 編)	
2	QGIS や Panoply、HDFView などの市販ツール（COTS）を使用する	GPM/TRMM データ読み込みプログラムガイド(COTS 編)	
3	GeoTIFF を使用する	GPM/TRMM データ読み込みプログラムガイド(GeoTIFF 変換編)	
4	IDL を使用する	GPM/TRMM データ読み込みプログラムガイド(IDL 編)	
5	C を使用する	GPM/TRMM データ読み込みプログラムガイド(C 言語編)	
6	FORTRAN を使用する	GPM/TRMM データ読み込みプログラムガイド(FORTRAN 編)	
7	Python を使用する	GPM/TRMM データ読み込みプログラムガイド(Python 編)	

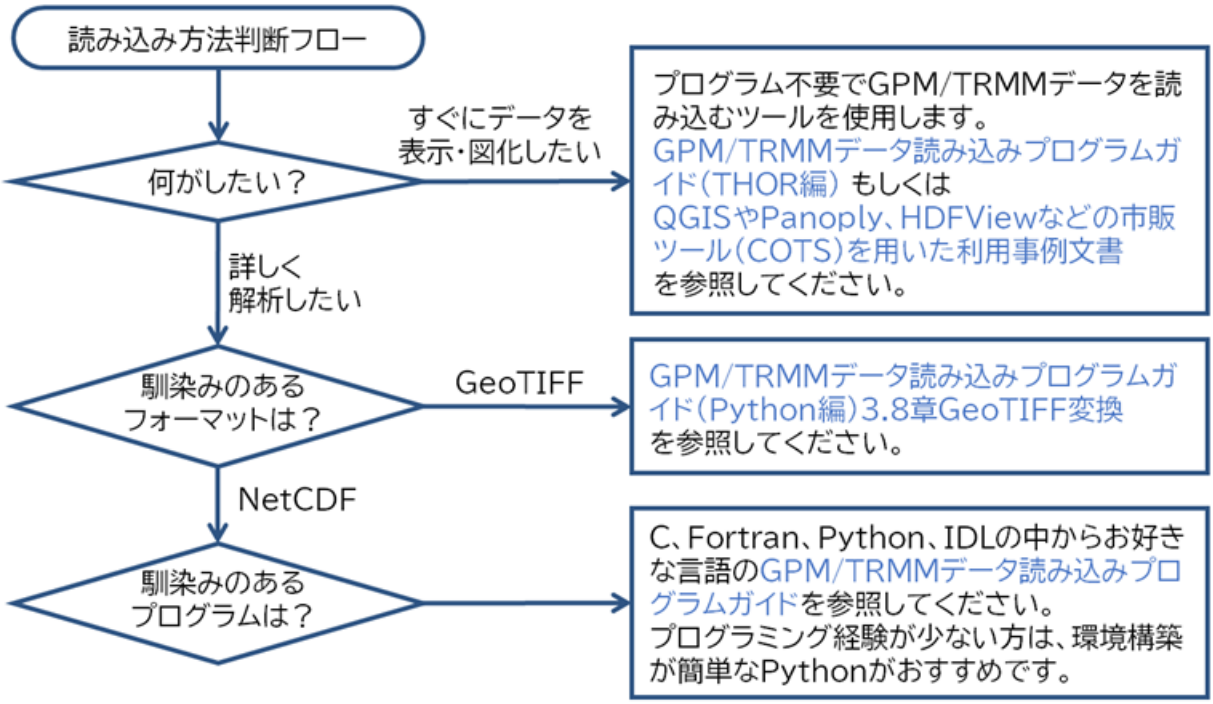
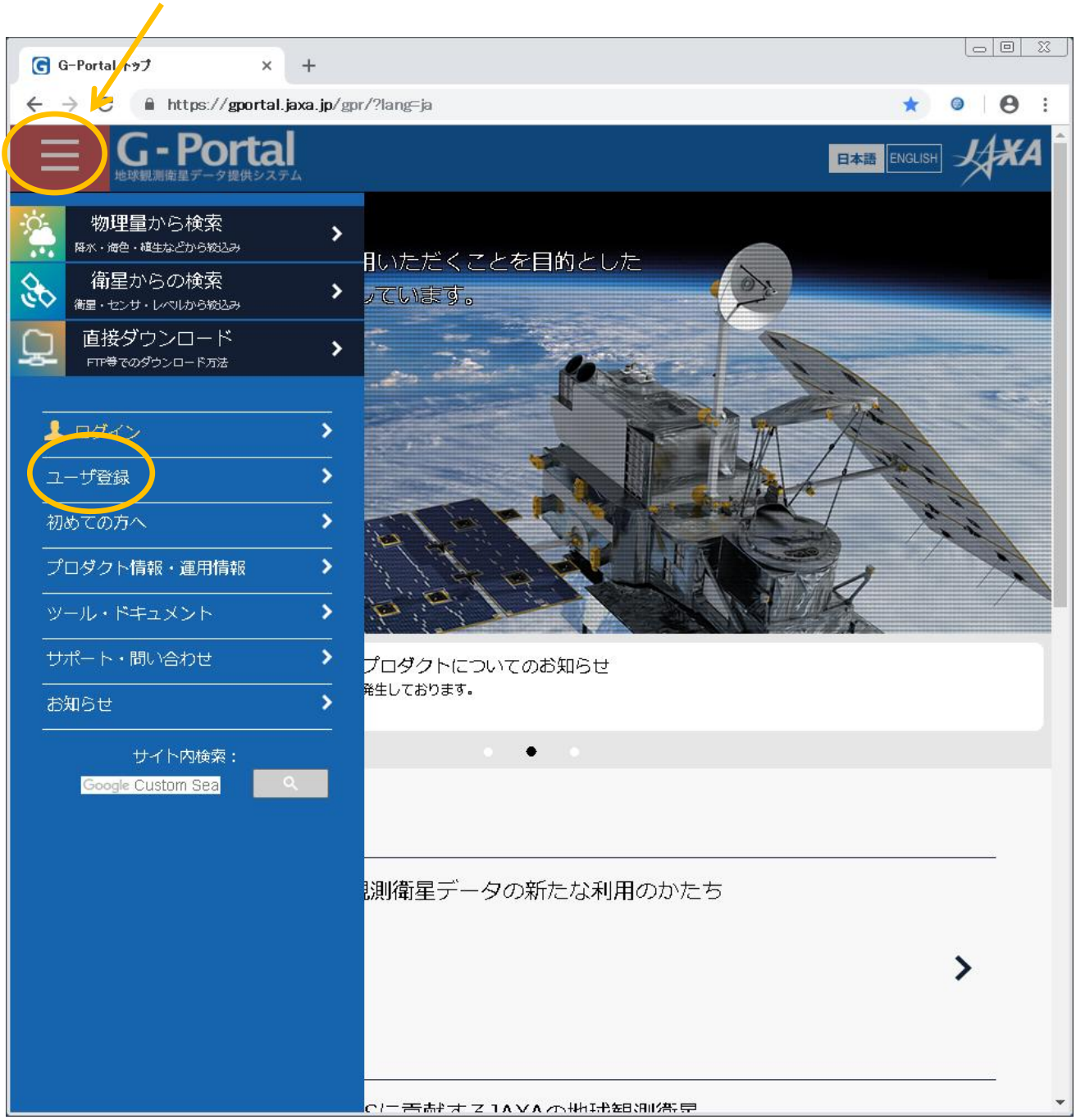


表 1.2 サンプルプログラム動作確認表

	サンプルプログラム	Linux	Windows	備考
1	C	○	—	
2	FORTRAN	○	—	
3	Python	○	○	
4	IDL	○	○	

2. GPM/TRMM データの入手方法

GPM/TRMM データは、G-Portal のサイト(https://gportal.jaxa.jp/gpr/)から取得することができます。取得の際にはユーザ登録が必要になりますので、G-Portal のサイトのメニューから「ユーザ登録」を選択してユーザ登録を行ってください。



規約を読み「同意して次へ」をクリックします。

ユーザ登録画面になりますので、ユーザ登録を行います。

以降の手順や、ユーザ登録後のデータ取得方法については、「GPM データ利用ハンドブック」の「5.2 データ提供サービスの使い方」を参照してください。「GPM データ利用ハンドブック」の入手方法については「3. 関連文書、サンプルプログラムの入手方法」を参照してください。

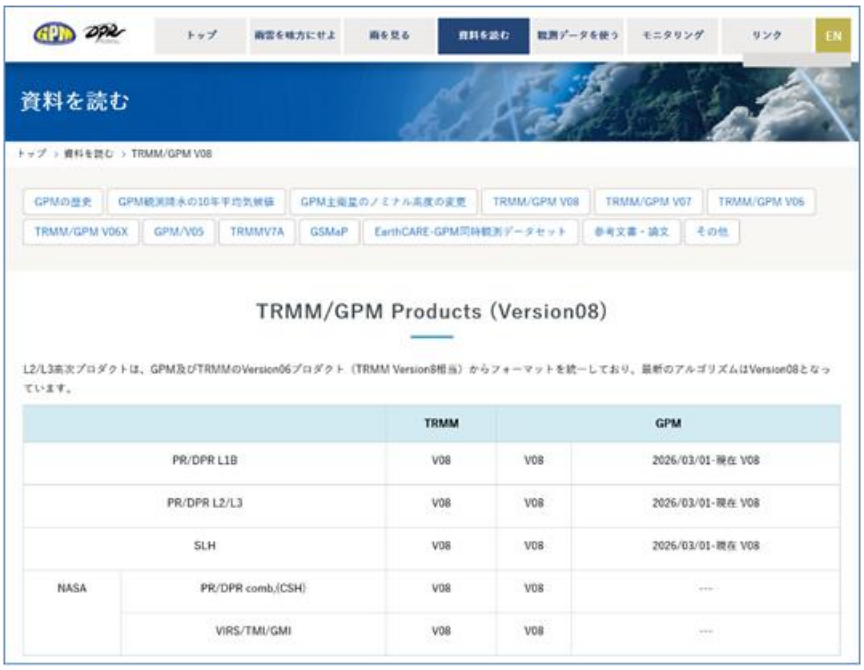
3. 関連文書、サンプルプログラムの入手方法

GPM/TRMM データの関連文書には、データ利用に関する文書と、プロダクトに関する文書があります。どちらも全球降水観測計画 GPM のサイト(https://www.eorc.jaxa.jp/GPM/index.html)のトップページ ＞ 資料を読む からダウンロードできます。また、本書で解説しているサンプルコードについてはトップページ ＞ 観測データを使う からダウンロードできます。

GPM データ利用に関する文書には以下のものがあります。

GPM データ利用ハンドブック

ファイル命名規約



「TRMM/GPM V08」をクリックするとプロダクトバージョン 08 の文書一覧が表示されます。Format Specification は各プロダクトのデータ仕様が記載されたドキュメントです。

本書で解説するプロダクトとプログラム、サンプルデータは以下の通りです。

表 3.1 サンプルプログラム一覧

プロダクト	サンプルプログラム	サンプルデータ
L1Ku	sample_L1_Ku_F.f90	GPMCOR_KUR_2604010211_0345_068594_1BS_DUB_08A.nc
L2DPR	sample_L2_DPR_F.f90	GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc
	sample_HDF5_L2_DPR_F.f90	
	sample_h5dump_L2_F.f90	
L3DPR	sample_L3_DPR_F.f90	GPMCOR_DPR_2603_M_L3S_D3M_08A.nc
	sample_HDF5_L3_DPR_F.f90	
	sample_h5dump_L3_F.f90	
L2GMI	sample_L2_GMI_F.f90	GPMCOR_GMI_2604010211_0345_068594_L2S_GL2_08A.nc
L3GMI	sample_L3_GMI_F.f90	GPMCOR_GMI_2603_M_L3S_GL3_08A.nc
L2SLP	sample_L2_SLP_F.f90	GPMCOR_DPR_2604010211_0345_068594_L2S_SLP_08A.nc
L3SLM	sample_L3_SLM_F.f90	GPMCOR_DPR_2603_M_L3S_SLM_08A.nc
L3SLG	sample_L3_SLG_F.f90	GPMCOR_DPR_2604010211_0345_068594_L3S_SLG_08A.nc

4. ライブラリ・ツールのインストール

FORTRAN で GPM データを読み込むには、表 4.1 で示すように 3 種類の方法があり、方法によってはツールをインストールする必要があります。本書ではそれぞれについてプログラム作成の解説を行います。

表 4.1 GPM データ読み込み方法

	GPM データ読み込み方法	必要なライブラリ、ツール	備考
1	PPS Toolkit(TKIO)	HDF5、NetCDF、PPS TKIO	表 4.2 参照
2	HDF5 ライブラリ	HDF5、NetCDF	
3	h5dump	HDF5	

また PPS Toolkit(TKIO)を利用する場合、GPM データのプロダクトバージョンと、対応する PPS Toolkit(TKIO)バージョンの関係を表 4.2 に示します。

表 4.2 プロダクトバージョンと PPS Toolkit(TKIO)の対応バージョン

プロダクト	プロダクトバージョン	PPS ツールキットのバージョン	備考
L1Ku	０８	３．１０２	
L2DPR	０８	３．１０２	
L3DPR	０８	３．１０２	
L2GMI	０８	３．１０２	
L3GMI	０８	３．１０２	
L2SLP	０８	３．１０２	
L3SLM	０８	３．１０２	
L3SLG	０８	３．１０２	

注）PPS Toolkit(TKIO)は基本的には上位互換ですが、一部で正常に読み込めない場合があります。その場合は「5.4 PPS Toolkit(TKIO)のバージョンについて」を参照してください。

本書のサンプルプログラムは以下の環境で動作確認を行っています。

表 4.2 動作環境

項目	環境	備考
計算機	Intel(R) Xeon(R) CPU E5-2665 0 @ 2.40GHz	
OS	AlmaLinux release 8.8	
FORTRAN コンパイラ	GNU Fortran (GCC) 8.5.0 20210514	HDF5 ライブラリを使用して読み出す場合
	ifx (IFX) 2025.3.2 20260112	PPS Toolkit(TKIO)を使用して読み出す場合
HDF5	Hdf5-1.10.8	
NetCDF	netcdf-4.9.2	
PPS TKIO	tkio-3.102	

なお、FORTRAN コンパイラにつきまして、ifort が利用可能であればそちらの利用を推奨します。

4.1 HDF5 のインストール

fortran モジュール（include/*.mod）が必要になるため、ソースコードよりビルドします。

ビルド時に使用するコンパイラは、サンプルプログラムをビルドするコンパイラと同一である必要があります。

※以下では、gfortran を使用しておりますが、ifort が利用可能であればそちらの利用を推奨します。なお、ifort の後継コンパイラ（ifx）では、ビルドできません。

4.1.1 ダウンロード

The HDF Group ホームページ(http://www.hdfgroup.org/)から HDF5 のソースインストール版の圧縮ファイルをダウンロードします。

※以下では hdf5-1.10.8.tar.gz をダウンロードしたものとして説明します。

4.1.2 解凍

適当な作業ディレクトリで圧縮ファイルを解凍します。 以下のコマンドで解凍できます。

```
$ tar -xzvf hdf5-1.10.8.tar.gz
```

解凍すると、hdf5-1.10.8 のようなディレクトリが作成されるので、その配下へ移動します。

```
$ cd hdf5-1.10.8
```

4.1.3 コンパイルとインストール

以下のコマンドを順番に実行して、コンパイルとインストールを行います。

--prefix=には、インストール先ディレクトリを指定します。

※この例の場合、hdf5 のバージョンは 1.10.8 なので、hdf5_1.10.8 としています。

バージョン文字部分は実際に使用するバージョンに置き換えてください。

```
$ ./configure --enable-static-exec --prefix=/home/user1/util/hdf5_1.10.8 --enable-fortran
FC=gfortran
$ make
$ make install
```

4.2 NetCDF のインストール

インストールした HDF5 ライブラリをリンクするために、ソースコードよりビルドします。

4.2.1 ダウンロード

Unidata ホームページ(https://www.unidata.ucar.edu/)から NetCDF のソースインストール版の圧縮ファイルをダウンロードします。

※以下では netcdf-c-4.9.2.tar.gz をダウンロードしたものとして説明します。

4.2.2 解凍

適当な作業ディレクトリで圧縮ファイルを解凍します。 以下のコマンドで解凍できます。

```
$ tar -xzvf netcdf-c-4.9.2.tar.gz
```

解凍すると、netcdf-c-4.9.2 のようなディレクトリが作成されるので、その配下へ移動します。

```
$ cd netcdf-c-4.9.2
```

4.2.3 コンパイルとインストール

以下のコマンドを順番に実行して、コンパイルとインストールを行います。

--prefix=には、インストール先ディレクトリを指定します。

※この例の場合、NetCDF のバージョンは 4.9.2 なので、netcdf-c-4.9.2 としています。

バージョン文字部分は実際に使用するバージョンに置き換えてください。

```
$ ./configure --prefix=/home/user1/util/netcdf-c-4.9.2 --enable-fortran CC=gcc ¥
            CPPFLAGS=-I/home/user1/util/hdf5_1.10.8/include ¥
            LDFLAGS=-L/home/user1/util/hdf5_1.10.8/lib
$ make
$ make install
```

4.3 PPS Toolkit(TKIO)のインストール

PPS Toolkit(TKIO)とは、GPM の NetCDF ファイルを読み込むプログラムを作成する際に使用するライブラリです。HDF5 ライブラリを使用して読み出す場合や、h5dump を使用して読み出す場合にはインストールする必要はありません。

4.3.1 ダウンロード

以下の URL から 、自分の環境に合った圧縮ファイルをダウンロードします。
https://gpmweb2https.pps.eosdis.nasa.gov/pub/PPStoolkit/GPM/

4.3.2 解凍

適当な作業ディレクトリを作成してダウンロードしたファイルを移し、圧縮ファイルを解凍します。
以下のコマンドで解凍できます。

```
$ mkdir tkio.xxx
$ mv tkio.xxx.tar.gz tkio.xxx/.
$ cd tkio.xxx
$ tar zxf tkio.xxx.tar.gz
```

4.3.3 前提条件の確認

docs ディレクトリにある tkioINSTALL.txt ファイルを参照し、ダウンロードした PPS Toolkit(TKIO)が動作する前提条件を確認します。必要なライブラリがインストールされていない場合や、バージョンが古い場合はインストールを行います。

・libxml2 ライブラリのインストール

```
必要なバージョンは docs/tkioINSTALL.txt を確認！
./configure --prefix=[インストール DIR]
make
make install
```

・zlib ライブラリのインストール

```
./configure --prefix=[インストール DIR]
make
make install
```

・jpeg ライブラリのインストール

```
./configure --prefix=[インストール DIR] --enable-shared
make
make install
make install-lib
```

4.3.4 環境設定ファイルの編集

環境変数を定義するファイルを作成します。以下に作成例を示します。自分の環境に合った環境変数を定義してください。

fortran コンパイラオプション（FFLAGS）に、最適化無効（-O0）を指定してください。サンプルプログラムビルドにて、リンクエラー発生の可能性があります。

```
ulimit -s unlimited
export TKDEBUG="-g"

export TKIO=/home/user1/util/tkio-3.102_gcc_ifx/tkio_v8
export HDF5_INC=/home/user1/util/hdf5-1.10.8/include
export HDF5_LIB=/home/user1/util/hdf5-1.10.8/lib
export NETCDF_INC=/home/user1/util/netcdf-c-4.9.2/include
export NETCDF_LIB=/home/user1/util/netcdf-c-4.9.2/lib
export CLASSPATH=$TKIO/classes
export JPEG_INC=/usr/include
export JPEG_LIB=/usr/lib64
export ZLIB=/usr/lib64
export LIBXML2_INC=/usr/include/libxml2
export LIBXML2_LIB=/usr/lib64

export
LD_LIBRARY_PATH=/home/user1/util/jdk1.8.0_60/lib:${ZLIB}:/home/user1/util/libtoo
l-2.4/lib:/home/user1/util/flex-2.5.39/lib:${LD_LIBRARY_PATH}

export
PATH=./:/home/user1/util/jdk1.8.0_60/bin:/home/user1/util/byacc-20150711/bin:/ho
me/user1/util/libtool-2.4/bin:/home/user1/util/flex-2.5.39/bin::${PATH}

export CC=gcc
export CFLAGS='-fPIC -mcmodel=medium'
export CXXFLAGS='-fPIC -mcmodel=medium'
export FFLAGS='-O0 -fPIC -mcmodel=medium'
export FC=ifx
export F77=ifx
export F90=ifx
export FORTC=ifx

export PERL5LIB=/usr/share/perl5/vendor_perl/CPAN
```

4.3.5 環境設定ファイルの読み込み

以下のコマンドで環境設定ファイルを読み込みます。

$ source 環境設定ファイル名

4.3.6 コンパイル

以下のコマンドでコンパイルを実行します。

$ ./INSTALL.pl compileJAVA

$ ./INSTALL.pl buildRW

$ ./INSTALL.pl compileRW

5. PPS Toolkit(TKIO)で GPM データ読み込み

PPS Toolkit(TKIO)を使用した Fortran プログラムの作成方法について説明します。PPS Toolkit(TKIO)を使用する場合は、予め PPS Toolkit(TKIO)をインストールしておく必要があります。

また、PPS Toolkit(TKIO)を使用してプログラムを作成する場合、予めアルゴリズム ID を知っておく必要があります。アルゴリズム ID とはプロダクト（データの種類）毎にある ID で、HDF5 ファイルのファイルヘッダに格納されています。PPS Viewer THOR でファイルヘッダの情報を確認することができます。

表 5.1 プロダクトとアルゴリズム ID、TKIO ヘッダファイルの対応

レベル	プロダクト	アルゴリズム ID	TKIO ヘッダファイル	備考
1	L1Ku	1BKu	TK_1BKu.h	
	L1PR	1BPR	TK_1BPR.h	
2	L2DPR	2ADPR	TK_2ADPR.h	
	L2GMI	2AGPROFGMI	TK_2AGPROFGMI.h	
	L2CMB	2BCMB	TK_2BCMB.h	
	L2PR	2APR	TK_L2APR.h	
	L2SLH	2HSLH	TK_2HSLH.h	
	L2LHP	2HSLHT	TK_2HSLHT.h	
3	L3DPR	3DPR	TK_3DPR.h	
	L3GMI	3GPROF	TK_3GPROF.h	
	L3CMB	3CMB	TK_3CMB.h	
	L3PR	3PR	TK_3PR.h	
	L3HSLH	3HSLH	TK_3HSLH.h	
	L3GSLH	3GSLH	TK_3GSLH.h	
	L3HSLHT	3HSLHT	TK_3HSLHT.h	
	L3GSLHT	3GSLHT	TK_3GSLHT.h	
	GSMaP	3GSMAPH	TK_3GSMAP.h	

プログラムを作成する際、読み込むデータにあわせて格納する領域を定義する必要があります。GPM/TRMM データは、スキャン、アングルビン、レンジビンという名称の次元で構成されています。このスキャン、アングルビン、レンジビンの関係については、「GPM/TRMM データ読み込みプログラムガイド(付録)」の「1.2 シーン定義」を参照してください。また、読み込むデータの構成については 「3. 関連文書、サンプルプログラムの入手方法」で示したサイトから「GPM/DPR TRMM/PR L1 プロダクトフォーマット説明書」/ 「GPM/DPR TRMM/PR L2/L3 プロダクトフォーマット説明書」をダウンロードして参照してください。

次項よりデータ読み込みプログラムの作成例を示します。
プログラムの説明は以下のように色分けしています。

赤字の解説はサンプルプログラムについて説明しています。

青字の解説は PPS Toolkit または衛星基礎知識について説明しています。

5.1 L1 データ読み込み

5.1.1 ソースプログラム

以下は L1Ku を読み込むプログラム例です。ジョブ名と、L1Ku の NetCDF ファイル名を引数として、引数として指定されたファイルから、日時情報、緯度経度情報、echoPower、noisePower というデータを読み込んでいます。

```
PROGRAM SAMPLE

#include "TKHEADERS.h"
#include "TK_1BKu.h"
#include "tkiofortdeclare.h"


RECORD /TKINFO/ granuleHandle1BKu


RECORD /L1BKu_FS/ L1BKu

        PARAMETER( NANGLE=49, NRANGE=260 )

        INTEGER status, numOfScan
        INTEGER iscan, iangle, irange
        CHARACTER*255 jobname, file
        REAL*8 lat(NANGLE), lon(NANGLE)
        INTEGER*2 noisePower(NANGLE),
echoPower(NRANGE,NANGLE)
        INTEGER*2 year, msec, dayOfYear
        BYTE month, dayOfMon, hour, min, sec
        REAL*8 secOfDay

        call getarg( 1, jobname )
        call getarg( 2, file )

        ! Open the file for reading.
        status = TKopen( file,'1BKu,TKREAD, 'NETCDF4', jobname, granuleHandle1BKu, 0 )
        IF ( status .NE. TK_SUCCESS ) THEN
          status = TKmessage( granuleHandle1BKu.jobname, TKERROR,'message to print' )
        ENDIF


        status = TKgetMetaInt( granuleHandle1BKu, 'SwathHeader',
'NumberScansGranule', numOfScan )
```

該当するアルゴリズム ID のヘッダファイルをインクルードします(表 5.1 プロダクトとアルゴリズム ID、TKIO ヘッダファイルの対応を参照)。。L1Ku はアルゴリズム ID が"1BKu"なので"TK_1BKu.h"をインクルードします。

Fortran の場合必ずインクルードします。

granuleHandle1BKu は NetCDF ファイルの情報を格納する構造体で PPS Toolkit を使用する際に使用します。

L1BKu は NetCDF ファイルの構造体です。1 スキャン分のデータを格納します。

NetCDF ファイルのオープン
file:NetCDF ファイル名(引数で指定した文字列)
1BKu:アルゴリズム ID
TKREAD:読み込み指定
NETCDF4:フォーマットタイプ
jobname:ジョブ名(引数で指定した文字列)、
granuleHandle1BKu:ファイルポインタ(TKINFO 構造体を指定)
1 :ファイルの内部圧縮(1 を指定)

メタデータ読み込み（スキャン数読み出し）
granuleHandle1BKu：ファイルポインタ(TKINFO 構造体を指定)
"SwathHeader"：NetCDF ファイルにある項目名で、読み出すデータの情報が格納されています。予め項目名を「L1 プロダクトフォーマット説明書」か PPS Viewer THOR で確認しておく必要があります。
"NumberScansGranule"：スキャン数を指定
numOfScan：読み出したスキャン数を格納する変数

スキャン数分ループ

スキャンデータ読み込み
granuleHandle1BKu：ファイルポインタ(TKINFO 構造体を指定)
L1BKu：読み出したデータを格納する領域

```
      do iscan=1,numOfScan
        status = TKreadScan( granuleHandle1BKu, L1BKu )

        ! ScanTime Read
        year      = L1BKu.ScanTime.Year
        month     = L1BKu.ScanTime.Month
        dayOfMon  = L1BKu.ScanTime.DayOfMonth
        hour      = L1BKu.ScanTime.Hour
        min       = L1BKu.ScanTime.Minute
        sec       = L1BKu.ScanTime.Second
        msec      = L1BKu.ScanTime.MilliSecond
        dayOfYear = L1BKu.ScanTime.DayOfYear
        secOfDay  = L1BKu.ScanTime.SecondOfDay

        do iangle=1,NANGLE
      ! Latitude and Longitude Read
      lat(iangle)  = L1BKu.Latitude(iangle)
      lon(iangle)  = L1BKu.Longitude(iangle)

      ! noisePower Read
      noisePower(iangle) = L1BKu.Receiver.noisePower(iangle)

      ! echoPower Read
        do irange=1,NRANGE
        echoPower(irange, iangle) = L1BKu.Receiver.echoPower(irange, iangle)
      enddo

       ! print the value
       IF(iscan .eq. 3947 .and. iangle .eq. 20 ) THEN
         write(*,*) "scan=",iscan-1,",angle=",iangle-1
         write(*,*) "lat=", lat(iangle), "lon=", lon(iangle)
         write(*,*) "echoPower(260)=",echoPower(260,iangle)
         write(*,*) "noisePower=",noisePower(iangle)
       ENDIF
     enddo
   enddo

   ! Close NetCDF file.
      status = TKclose( granuleHandle1BKu )
      STOP
END
```

スキャン日時の読み込み

緯度経度情報読み込み

noisePower 読み込み

echoPower 読み込み

正しく読み込めているか確認するため、一部分(このケースは、スキャンが 3947 番目のアングル 20、レンジビン 260 のデータ）を出力しています。

NetCDF ファイルのクローズ
granuleHandle1BKu：ファイルポインタ(TKINFO 構造体を指定)

5.1.2 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=ifx
MACHINE=LINUX
FFLAGS=-O0 -g -mcmodel=medium -fPIC -fpp -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_L1_Ku_F

INC = -I$(HDF5_INC) ¥
       -I$(NETCDF_INC) ¥
       -I$(TKIO)/inc/fortcode ¥
       -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
       -L$(NETCDF_LIB) ¥
       -L$(TKIO)/lib

LIBES   = -ltkcselect ¥
          -ltkchdf5algs -ltkchdf5 ¥
          -ltkcnetcdf4 -ltkcnetcdf4algs -ltkfortalgmodules -ltkfortselect -ltkc ¥
          -lm -lhdf5_hl -lhdf5 -ljpeg -lz -lxml2 -lnetcdf

$(MAIN): $(MAIN).o
       $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
       $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
       rm -f *.o $(MAIN)
```

5.1.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

PPS Toolkit(TKIO)の Fortran のヘッダファイルがあるディレクトリのパスです

PPS Toolkit(TKIO)のライブラリがあるディレクトリのパスです

5.1.3 実行結果

5.1.1 で説明したプログラムの実行結果を示します。

第 1 引数の"TEST"はジョブ名です。(任意の文字列で構いません)

第 2 引数は NetCDF ファイル名です。

```
$ ./sample_L1_Ku_F
TEST ../data/GPM/DPR/GPMCOR_KUR_2604010211_0345_068594_1BS_DUB_08A.nc
 scan=        3946 ,angle=          19
 lat=   64.7641448974609      lon=   57.4538764953613
 echoPower(260)= -29999
 noisePower= -11130
```

5.2 L2 データ読み込み

5.2.1 ソースプログラム

以下は L2DPR を読み込むプログラム例です。ジョブ名と、L2DPR の NetCDF ファイル名を引数として、引数として指定されたファイルから、日時情報、緯度経度情報、precipRateESurface 及び precipWater というデータを読み込んでいます。

```
PROGRAM SAMPLE

#include "TKHEADERS.h"
#include "TK_2ADPR.h"

#include "tkiofortdeclare.h"


RECORD /TKINFO/ granuleHandleL2DPR


RECORD /L2ADPR_SWATHS/ L2ADPR

        PARAMETER( NANGLE=49, NRANGE=176 )

        INTEGER status, numOfScan
        INTEGER iscan, iangle, irange
        CHARACTER*255 jobname, file
        REAL*8 lat(NANGLE), lon(NANGLE)
        REAL*4 precipRateESurface(NANGLE)
        REAL*4 precipWater(NRANGE,NANGLE)
        INTEGER*2 year, msec, dayOfYear
        BYTE month, dayOfMon, hour, min, sec
        REAL*8 secOfDay

        call getarg( 1, jobname )
        call getarg( 2, file )

        !Open the file for reading.
        status = TKopen( file,'2ADPR,TKREAD,'NETCDF4',jobname, granuleHandleL2DPR, 0 )
        IF ( status .NE. TK_SUCCESS ) THEN
          status = TKmessage( granuleHandleL2DPR.jobname, TKERROR,'message to print' )
        ENDIF


        status = TKgetMetaInt( granuleHandleL2DPR, 'FS_SwathHeader',
'NumberScansGranule', numOfScan )
```

該当するアルゴリズム ID のヘッダファイルをインクルードします(表 5.1 プロダクトとアルゴリズム ID、TKIO ヘッダファイルの対応を参照)。L2DPR はアルゴリズム ID が"2ADPR"なので"TK_2ADPR.h"をインクルードします。

Fortran の場合必ずインクルードします。

granuleHandleL2DPR は NetCDF ファイルの情報を格納する構造体で PPS Toolkit を使用する際に使用します。

L2ADPR は NetCDF ファイルの構造体です。1 スキャン分のデータを格納します。

NetCDF ファイルのオープン
file:NetCDF ファイル名(引数で指定した文字列)
2ADPR:アルゴリズム ID
TKREAD:読み込み指定
NETCDF4:フォーマットタイプ
jobname:ジョブ名(引数で指定した文字列)
granuleHandleL2DPR：ファイルポインタ
(TKINFO 構造体を指定)
1 ：ファイルの内部圧縮(1 を指定)

メタデータ読み込み（スキャン数読み出し）
granuleHandle2ADPR：ファイルポインタ(TKINFO 構造体を指定)
FS_SwathHeader" : NetCDF ファイルにある項目名で、読み出すデータの情報が格納されています。予め項目名を「L1 プロダクトフォーマット説明書」か PPS Viewer THOR で確認しておく必要があります。
"NumberScansGranule" : スキャン数を指定
numOfScan：読み出したスキャン数を格納する変数

スキャン数分ループ

スキャンデータ読み込み
granuleHandleL2DPR：ファイルポインタ(TKINFO 構造体を指定)
L2ADPR：読み出したデータを格納する領域

```
      do iscan=1,numOfScan
        status = TKreadScan( granuleHandleL2DPR, L2ADPR )

        ! ScanTime Read
        year      = L2ADPR.FS.ScanTime.Year
        month     = L2ADPR.FS.ScanTime.Month
        dayOfMon  = L2ADPR.FS.ScanTime.DayOfMonth
        hour      = L2ADPR.FS.ScanTime.Hour
        min       = L2ADPR.FS.ScanTime.Minute
        sec       = L2ADPR.FS.ScanTime.Second
        msec      = L2ADPR.FS.ScanTime.MilliSecond
        dayOfYear = L2ADPR.FS.ScanTime.DayOfYear
        secOfDay  = L2ADPR.FS.ScanTime.SecondOfDay

        do iangle=1,NANGLE

      ! Latitude and Longitude Read
      lat(iangle)  = L2ADPR.FS.Latitude(iangle)
      lon(iangle)  = L2ADPR.FS.Longitude(iangle)

      ! precipRateESurface Read
      precipRateESurface(iangle) = L2ADPR.FS.SLV.precipRateESurface(iangle)


        do irange=1,NRANGE
        precipWater(irange, iangle) = L2ADPR.FS.SLV. precipWater (irange, iangle)
      enddo



      ! print the value
      IF(iscan .eq. 5211 .and. iangle .eq. 44 ) THEN
        write(*,*) "scan=",iscan-1,",angle=",iangle-1
        write(*,*) "lat=", lat(iangle), "lon=", lon(iangle)
        write(*,*) "precipRateESurface=",precipRateESurface (iangle), "(mm/hr)"
        write(*,*) "precipWater(171,44)=", precipWater(171, iangle), "(g/m~3)"
      ENDIF
    enddo
  enddo

      status = TKclose( granuleHandleL2DPR )
      STOP
END
```

スキャン日時の読み込み

緯度経度情報読み込み

precipRateESurface 読み込み

precipWater 読み込み

正しく読み込めているか確認するため、一部分（このケースは、スキャンが 5211 番目のアングル 44 のデータ）を出力しています。

NetCDF ファイルのクローズ
granuleHandleL2DPR：ファイルポインタ(TKINFO 構造体を指定)

5.2.2 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=ifx
MACHINE=LINUX
FFLAGS=-O0 -g -mcmodel=medium -fPIC -fpp -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_L2_DPR_F

INC = -I$(HDF5_INC) ¥
       -I$(NETCDF_INC) ¥
       -I$(TKIO)/inc/fortcode ¥
       -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
       -L$(NETCDF_LIB) ¥
       -L$(TKIO)/lib

LIBES   = -ltkcselect ¥
          -ltkchdf5algs -ltkchdf5 ¥
          -ltkcnetcdf4 -ltkcnetcdf4algs -ltkfortalgmodules -ltkfortselect -ltkc ¥
          -lm -lhdf5_hl -lhdf5 -ljpeg -lz -lxml2 -lnetcdf

$(MAIN): $(MAIN).o
       $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
       $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
       rm -f *.o $(MAIN)
```

5.2.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

PPS Toolkit(TKIO)の Fortran のヘッダファイルがあるディレクトリのパスです

PPS Toolkit(TKIO)のライブラリがあるディレクトリのパスです

5.2.3 実行結果

5.2.1 で説明したプログラムの実行結果を示します。

第 1 引数の”TEST”はジョブ名です。(任意の文字列で構いません)

第 2 引数は NetCDF ファイル名です。

```
$ ./sample_L2_DPR_F
TEST ../data/GPM/DPR/GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc
 scan=        5210 ,angle=          43
 lat=   32.0528602600098      lon=   132.983184814453
 precipRateESurface=   2.261208     (mm/hr)
 precipWater(171,44)=  0.1282614     (g/m~3)
```

5.3 L3 データ読み込み

5.3.1 ソースプログラム

以下は L3DPR 読み込みプログラム例です。ジョブ名と、L3DPR の HDF5 ファイル名を引数として、引数として指定されたファイルから、precipRateESurface.mean というデータを読み込んでいます。

```
PROGRAM SAMPLE

#include "TKHEADERS.h"
#include "TK_3DPR.h"
#include "tkiofortdeclare.h"

RECORD /TKINFO/ granuleHandle3DPR
RECORD /L3DPR_FS/ L3DPR

        INTEGER status
        INTEGER st, rt, chn, lnL, ltL
        CHARACTER*255 jobname, file
    REAL*4 precipEsurf(28,72,3,3,3)
    REAL*4 lat, lon
        call getarg( 1, jobname )
        call getarg( 2, file )

    ! Open the file for reading.
    status = TKopen( file, '3DPR', TKREAD, 'NETCDF4', jobname, granuleHandle3DPR, 0 )
    IF ( status .NE. TK_SUCCESS ) THEN
      status = TKmessage( granuleHandle3DPR.jobname, TKERROR,'message to print' )
    ENDIF

    ! Grid data read
    status = TKreadGrid( granuleHandle3DPR, L3DPR )

    do ltL=1, 28
      do lnL=1, 72
        do chn=1, 3
          do rt=1, 3
            do st=1, 3
              ! precipRateESurface.mean Read
              precipEsurf(ltL, lnL, chn, rt, st) =
L3DPR.G1.precipRateESurface.mean(ltL, lnL, chn, rt, st)

              ! print the value
              IF(lnL .eq. 64 .and. ltL .eq. 15 .and. chn .eq. 1 .and. st .eq. 1) THEN
                write(*,*) "st=",st-1,"rt=",rt-1
                lat = (140.0/28.0) * (ltL-1) - 70.0 + (140.0/28.0/2)
                lon = (360.0/72.0) * (lnL-1) - 180.0 + (360.0/72.0/2)
                write(*,*) "lat=",lat,"lon=",lon
                write(*,*) "chn=0 lnL=63 ltL=14
precipRateEsurface.mean=",precipEsurf(ltL, lnL, chn, rt, st), "(mm/hr)"
              ENDIF
```

該当するアルゴリズム ID のヘッダファイルをインクルードします(表 5.1 プロダクトとアルゴリズム ID、TKIO ヘッダファイルの対応を参照)。L3DPR はアルゴリズム ID が"3DPR"なので"TK_3DPR.h"をインクルードします。

Fortran の場合必ずインクルードします。

granuleHandle3DPR は NetCDF ファイルの情報を格納する構造体で PPS Toolkit を使用する際に使用します。

L3DPR は NetCDF ファイルの構造体です。グリッドデータを格納します。

NetCDF ファイルのオープン
file:NetCDF ファイル名(引数で指定した文字列)
3DPR:アルゴリズム ID、 TKREAD:読み込み指定
NETCDF4:フォーマットタイプ
jobname:ジョブ名(引数で指定した文字列)
granuleHandle3DPR：ファイルポインタ(TKINFO 構造体を指定)
1 ：ファイルの内部圧縮(1 を指定)

グリッドデータ読み込み
granuleHandle3DPR：ファイルポインタ(TKINFO 構造体を指定)
L3DPR: 読み出したデータを格納する領域

precipRateESurface.mean 読み込み

正しく読み込めているか確認するため、一部分を出力しています。

```
            enddo
          enddo
        enddo
      enddo
    enddo

    ! HDF file close
        status = TKclose( granuleHandle3DPR )
        STOP
END
```

NetCDF ファイルのクローズ
granuleHandle3DPR：ファイルポインタ(TKINFO 構造体を指定)

5.3.2 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=ifx
MACHINE=LINUX
FFLAGS=-O0 -g -mcmodel=medium -fPIC -fpp -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_L3_DPR_F

INC = -I$(HDF5_INC) ¥
       -I$(NETCDF_INC) ¥
       -I$(TKIO)/inc/fortcode ¥
       -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
       -L$(NETCDF_LIB) ¥
       -L$(TKIO)/lib

LIBES   = -ltkcselect ¥
          -ltkchdf5algs -ltkchdf5 ¥
          -ltkcnetcdf4 -ltkcnetcdf4algs -ltkfortalgmodules -ltkfortselect -ltkc ¥
          -lm -lhdf5_hl -lhdf5 -ljpeg -lz -lxml2 -lnetcdf

$(MAIN): $(MAIN).o
       $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
       $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
       rm -f *.o $(MAIN)
```

5.3.1 のプログラムの名前です。

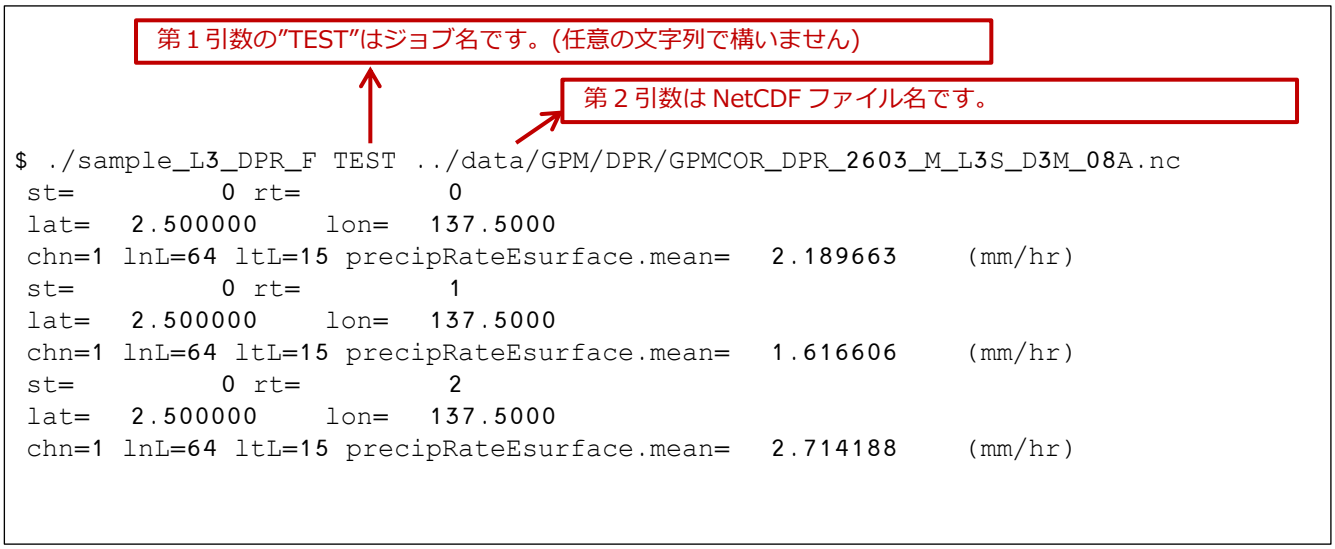
NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

PPS Toolkit(TKIO)の Fortran 用のヘッダファイルがあるディレクトリのパスです

PPS Toolkit(TKIO)のライブラリがあるディレクトリのパスです

5.3.3 実行結果

5.3.1 で説明したプログラムの実行結果を示します。




```
$ ./sample_L3_DPR_F TEST ../data/GPM/DPR/GPMCOR_DPR_2603_M_L3S_D3M_08A.nc
 st=          0 rt=          0
 lat=   2.500000     lon=   137.5000
 chn=1 lnL=64 ltL=15 precipRateEsurface.mean=   2.189663     (mm/hr)
 st=          0 rt=          1
 lat=   2.500000     lon=   137.5000
 chn=1 lnL=64 ltL=15 precipRateEsurface.mean=   1.616606     (mm/hr)
 st=          0 rt=          2
 lat=   2.500000     lon=   137.5000
 chn=1 lnL=64 ltL=15 precipRateEsurface.mean=   2.714188     (mm/hr)
```

5.4　PPS Toolkit(TKIO)のバージョンについて

インストールした PPS Toolkit(TKIO)のバージョンと、NetCDF ファイルを作成したバージョンが異なると正常に読み込めない場合があります。その場合、NetCDF ファイルのバージョンを調べ、バージョンに合わせたヘッダファイル、アルゴリズム ID にプログラムを変更する必要があります。

NetCDF ファイルのバージョンを調べるには PPS Viewer THOR を使用して NetCDF ファイルの FileInfo を読出し、「DataFormatVersion」と「TKCodeBuildVersion」の値を確認します。

DataFormatVersion=bk

TKCodeBuildVersion=2

の場合、バージョンは bk2 となります。

プログラムの変更は以下の 3 箇所です。

1) インクルードするヘッダファイル名
2) アルゴリズム ID
3) ヘッダファイルの参照箇所

ヘッダファイルとアルゴリズム ID の変更はバージョンの表記を追加します。ヘッダファイルが「TK_2ADPR.h」、アルゴリズム ID が「2ADPR」の場合、ヘッダファイルは「TK_2DPR_bk2.h」、アルゴリズム ID は「2ADPR_bk2」となります。

ヘッダファイルの変更に伴い、ヘッダファイルの内容を参照している箇所も変更後のヘッダファイルに合わせた内容に変更する必要があります。

以下に L2DPR データ読み込みサンプルプログラムの修正例を示します。

```
PROGRAM SAMPLE

#include "TKHEADERS.h"
#include "TK_2ADPR.h"

#include "tkiofortdeclare.h"


RECORD /TKINFO/ granuleHandleL2DPR


RECORD /L2ADPR_SWATHS/ L2ADPR

        PARAMETER( NANGLE=49, NRANGE=176 )

        INTEGER status, numOfScan
        INTEGER iscan, iangle, irange
        CHARACTER*255 jobname, file
        REAL*8 lat(NANGLE), lon(NANGLE)
        REAL*4 precipEsurf(NANGLE)
        REAL*4 zFactorCor(NRANGE,NANGLE)
        INTEGER*2 year, msec, dayOfYear
        BYTE month, dayOfMon, hour, min, sec
        REAL*8 secOfDay
```

インクルードするヘッダファイルで、バージョンに合わせて変更するのはこのファイルです。（TK_xxxx.h）
“TK_2ADPR_bk2.h”に変更します。

TK_2ADPR.h の内容を参照しているのはこの部分です。
“TK_2ADPR_bk2.h”の内容を調べて対応する定義の
“L2ADPR_SWATHS_bk2”に変更します。

```
       call getarg( 1, jobname )
       call getarg( 2, file )

   !Open the file for reading.
       status = TKopen(file,'2ADPR',TKREAD,'NETCDF4',jobname,granuleHandleL2DPR, 0 )
       IF ( status .NE. TK_SUCCESS ) THEN
         status = TKmessage( granuleHandleL2DPR.jobname, TKERROR,'message to print' )
       ENDIF

       status = TKgetMetaInt( granuleHandleL2DPR, 'NS_SwathHeader',
'NumberScansGranule', numOfScan )

       do iscan=1,numOfScan
         status = TKreadScan( granuleHandleL2DPR, L2ADPR )

         ! ScanTime Read
         year      = L2ADPR.NS.ScanTime.Year
         month     = L2ADPR.NS.ScanTime.Month
         dayOfMon  = L2ADPR.NS.ScanTime.DayOfMonth
         hour      = L2ADPR.NS.ScanTime.Hour
         min       = L2ADPR.NS.ScanTime.Minute
         sec       = L2ADPR.NS.ScanTime.Second
         msec      = L2ADPR.NS.ScanTime.MilliSecond
         dayOfYear = L2ADPR.NS.ScanTime.DayOfYear
         secOfDay  = L2ADPR.NS.ScanTime.SecondOfDay

         do iangle=1,NANGLE

       ! Latitude and Longitude Read
       lat(iangle)  = L2ADPR.NS.Latitude(iangle)
       lon(iangle)  = L2ADPR.NS.Longitude(iangle)

       ! precipRateESurface Read
       precipEsurf(iangle) = L2ADPR.NS.SLV.precipRateESurface(iangle)


       do irange=1,NRANGE
         zFactorCor(irange, iangle) = L2ADPR.NS.SLV.zFactorCorrected(irange, iangle)
       enddo

       ! print the value
       IF(iscan .eq. 3947 .and. iangle .eq. 20 ) THEN
         write(*,*) "scan=",iscan-1,",angle=",iangle-1
         write(*,*) "lat=", lat(iangle), "lon=", lon(iangle)
         write(*,*) "precipEsurf=",precipEsurf(iangle), "(mm/hr)"
       ENDIF
     enddo
   enddo

       status = TKclose( granuleHandleL2DPR )
       STOP
END
```

アルゴリズム ID を指定しているのは、
この部分です。バージョンを追加して
“2ADPR_bk2”に変更します。

6. HDF ライブラリで GPM データ読み込み

HDF ライブラリを使用した Fortran プログラムの作成方法について説明します。

赤字の解説はサンプルプログラムについて説明しています。

青字の解説は HDF ライブラリまたは衛星基礎知識について説明しています。

6.1 L2DPR データ読み込み

6.1.1 ソースプログラム

以下のサンプルプログラムは、filename で指定されたファイルから、precipRateESurface というデータを読み込んでいます。

```
PROGRAM SAMPLE

      use hdf5   ! This module contains all necessary modules

      CHARACTER(LEN=34), PARAMETER :: filename = "../data/GPM/DPR/
GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc"

      CHARACTER(LEN=26), PARAMETER :: dsetname = "/HS/SLV/precipRateESurface"

      INTEGER(HID_T) :: file_id      ! File identifier
      INTEGER(HID_T) :: dset_id      ! Dataset identifier
      REAL*4, DIMENSION(49,7988) :: precipRateESurface ! Data buffers
      INTEGER(HSIZE_T), DIMENSION(2) :: data_dims
      INTEGER error

      ! Initialize FORTRAN interface.
      CALL h5open_f(error)

      ! Open an existing file.
      CALL h5fopen_f (filename, H5F_ACC_RDWR_F, file_id, error)
```

読み込む NetCDF のファイル名です。

NetCDF ファイルの中から読み込むデータ名です。

HDF5 ライブラリ初期化
error：エラー情報(0:正常)

NetCDF ファイルオープン
filename：NetCDF ファイル名
H5F_ACC_RDWR_F：アクセス指定
file_id：ファイル ID(出力情報)
error：エラー情報(0:正常)

データセットのオープン
file_id：h5fopen_f で出力した値を指定します。
dsetname：読み込むデータの名前です。
dset_id：データセット ID(出力情報)
error：エラー情報(0:正常)

```
        ! Open an existing dataset.
        CALL h5dopen_f(file_id, dsetname, dset_id, error)

        ! Read precipRateESurface.
```

データ読み込み
dset_id：h5dopen_f で出力した値を指定します。

```
        data_dims(1) = 49
        data_dims(2) = 7988
       CALL h5dread_f(dset_id, H5T_NATIVE_REAL, precipRateESurface, data_dims, error)

         ! print the value
         write(*,*) "filename=",filename,""
         write(*,*) "dataset name=",dsetname,""
         write(*,*) "precipRateESurface(44,5211)=",precipRateESurface(44,5211),
"(mm/hr)"

         ! Close the dataset.
         CALL h5dclose_f(dset_id, error)

         ! Close the file.
         CALL h5fclose_f(file_id, error)

         ! Close FORTRAN interface.
         CALL h5close_f(error)

        STOP
  END
```

6.1.2 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=gfortran
MACHINE=LINUX
FFLAGS=-g -mcmodel=medium -fPIC -cpp -fdec-structure -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_HDF5_L2_DPR_F

INC = -I$(HDF5_INC) ¥
      -I$(NETCDF_INC) ¥
      -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
      -L$(NETCDF_LIB)

LIBES = -lnetcdf -lm -lhdf5_hl_fortran -lhdf5_fortran -lhdf5_hl -lhdf5 -ljpeg -lxml2

$(MAIN): $(MAIN).o
      $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
      $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
      rm -f *.o $(MAIN)
```

6.1.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

6.1.3 実行結果

6.1.1 で説明したプログラムの実行結果を示します。

```
$ ./sample_HDF5_L2_DPR_F
 filename=../data/GPM/DPR/GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc
 dataset name=/FS/SLV/precipRateESurface
 precipRateESurface(44,5211)=   2.261208     (mm/hr)
```

6.2 L3DPR データ読み込み

6.2.1 ソースプログラム

以下のサンプルプログラムは、filename で指定されたファイルから、precipRateESurface というデータを読み込んでいます。

```
PROGRAM SAMPLE

    use hdf5   ! This module contains all necessary modules

    CHARACTER(LEN=38), PARAMETER :: filename =
"../data/GPM/DPR/GPMCOR_DPR_2603_M_L3S_D3M_08A.nc"

    CHARACTER(LEN=34), PARAMETER :: dsetname = "/FS/G1/precipRateESurface/mean"

    INTEGER(HID_T) :: file_id       ! File identifier
    INTEGER(HID_T) :: dset_id       ! Dataset identifier
    REAL*4, DIMENSION(28,72,3,3,3) :: precipRateESurface ! Data buffers
    REAL*4 lat, lon
    INTEGER(HSIZE_T), DIMENSION(5) :: data_dims
       INTEGER error

    ! Initialize FORTRAN interface.
    CALL h5open_f(error)

    ! Open an existing file.
    CALL h5fopen_f (filename, H5F_ACC_RDWR_F, file_id, error)

    ! Open an existing dataset.
    CALL h5dopen_f(file_id, dsetname, dset_id, error)
```

読み込む NetCDF のファイル名です。

NetCDF ファイルの中から読み込むデータ名です。

HDF5 ライブラリ初期化
error：エラー情報(0:正常)

NetCDF ファイルオープン
filename：NetCDF ファイル名
H5F_ACC_RDWR_F：アクセス指定
file_id：ファイル ID(出力情報)
error：エラー情報(0:正常)

データセットのオープン
file_id：h5fopen_f で出力した値を指定します。
dsetname：読み込むデータの名前です。
dset_id：データセット ID(出力情報)
error：エラー情報(0:正常)

```
     ! Read precipRateESurface.
     data_dims(1) = 28
     data_dims(2) = 72
     data_dims(3) = 3
     data_dims(4) = 3
     data_dims(5) = 3
     CALL h5dread_f(dset_id, H5T_NATIVE_REAL, precipRateESurface, data_dims, error)

     ! print the value
     write(*,*) "filename=",filename
     write(*,*) "dataset name=",dsetname
     lat = (140.0/28.0) * (15-1) - 70.0 + (140.0/28.0/2)
     lon = (360.0/72.0) * (64-1) - 180.0 + (360.0/72.0/2)
     write(*,*) "lat=",lat,"lon=",lon
     write(*,*)
"precipRateESurface.mean(15,64,1,1,1)=",precipRateESurface(15,64,1,1,1)
     write(*,*)
"precipRateESurface.mean(15,64,1,2,1)=",precipRateESurface(15,64,1,2,1)

     ! Close the dataset.
     CALL h5dclose_f(dset_id, error)

     ! Close the file.
     CALL h5fclose_f(file_id, error)

     ! Close FORTRAN interface.
     CALL h5close_f(error)

       STOP
 END
```

データ読み込み
dset_id：h5dopen_f で出力した値を指定します。

6.2.2 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=gfortran
MACHINE=LINUX
FFLAGS=-g -mcmodel=medium -fPIC -cpp -fdec-structure -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_HDF5_L3_DPR_F

INC = -I$(HDF5_INC) ¥
       -I$(NETCDF_INC) ¥
       -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
       -L$(NETCDF_LIB)

LIBES = -lnetcdf -lm -lhdf5_hl_fortran -lhdf5_fortran -lhdf5_hl -lhdf5 -ljpeg -lxml2

$(MAIN): $(MAIN).o
       $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
       $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
       rm -f *.o $(MAIN)
```

6.2.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

6.2.3 実行結果

6.2.1 で説明したプログラムの実行結果を示します。

```
$ ./sample_HDF5_L3_DPR_F
 filename=../data/GPM/DPR/GPMCOR_DPR_2603_M_L3S_D3M_08A.nc
 dataset name=/FS/G1/precipRateESurface/mean
 lat=   2.50000000     lon=   137.500000
 precipRateESurface.mean(15,64,1,1,1)=   2.18966341
 precipRateESurface.mean(15,64,1,2,1)=   1.61660624
```

7. h5dump で GPM データ読み込み

h5dump を使用して、NetCDF ファイルから読み込みたいデータのバイナリファイルを作成し、そのバイナリファイルを読み出す Fortran プログラムの作成方法について説明します。

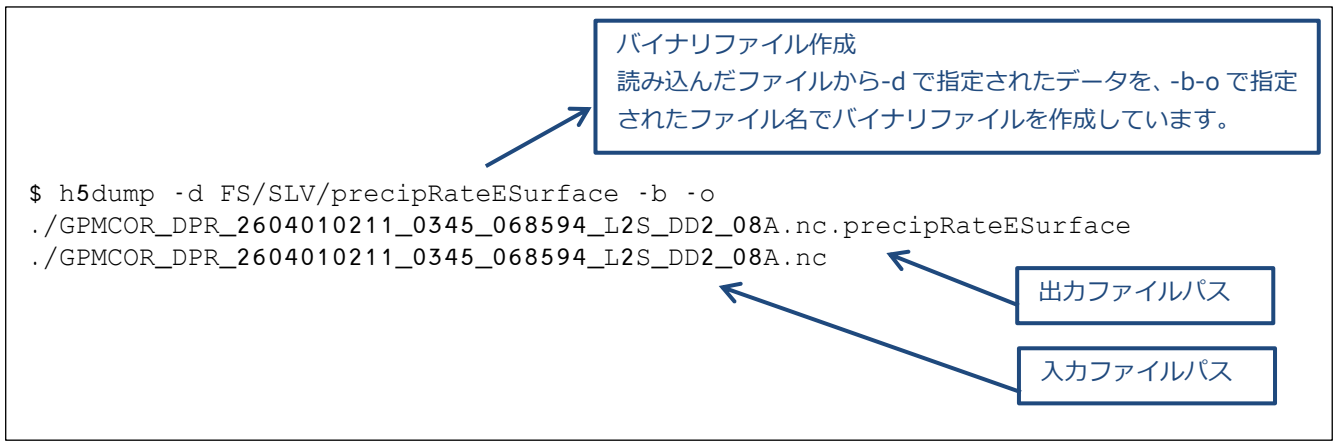
赤字の解説はサンプルプログラムについて説明しています。

青字の解説は HDF ライブラリまたは衛星基礎知識について説明しています。

7.1 L2 データ読み込み

7.1.1 バイナリファイルの作成

h5dump を使用してバイナリファイルの作成を行う例を示します。



上記コマンドを実行すると以下のように表示され、出力ファイルパスにバイナリファイルが作成されます。

```
HDF5 "./GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc" {
DATASET "FS/SLV/precipRateESurface" {
   DATATYPE  H5T_IEEE_F32LE
   DATASPACE  SIMPLE { ( 7988, 49 ) / ( H5S_UNLIMITED, 49 ) }
   DATA {
   }
}
}
```

7.1.2 ソースプログラム

以下のサンプルプログラムは、inputfile で指定されたバイナリファイルから情報を読み込んでいます。

```
program sample

    CHARACTER(LEN=66), PARAMETER :: filename =”../data/GPM/DPR/
GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc.precipRateESurface ”

    ! read data area
    real*4 precipRateESurface(49,7988)



    ! binary file open
    open(10,file=filename,access=’direct’,status=’old’,recl=4*49*7988)


    ! binary file read
    read(10,rec=1) precipRateESurface

    ! print the value
    write(*,*) ”filename=”,filename,””
    write(*,*) ”precipRateESurface(44,5211)=”,precipRateESurface(44,5211),
”(mm/hr)”
    write(*,*) ”precipRateESurface(45,5211)=”,precipRateESurface(45,5211),
”(mm/hr)”

    ! binary file close
    close(10)

end
```

7.1.1 で作成したバイナリファイルを指定しています。

バイナリファイルのオープン
recl：レコード長(precipRateESurface のサイズを指定)

バイナリファイルの読み込み
1 回で全データを precipRateESurface に読み込んでいます。

7.1.3 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=gfortran
MACHINE=LINUX
FFLAGS=-g -mcmodel=medium -fPIC -cpp -fdec-structure -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_h5dump_L2_F

INC = -I$(HDF5_INC) ¥
      -I$(NETCDF_INC) ¥
      -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
      -L$(NETCDF_LIB)

LIBES = -lnetcdf -lm -lhdf5_hl_fortran -lhdf5_fortran -lhdf5_hl -lhdf5 -ljpeg -lxml2
-ltirpc

$(MAIN): $(MAIN).o
      $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
      $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
      rm -f *.o $(MAIN)
```

7.1.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

7.1.4 実行結果

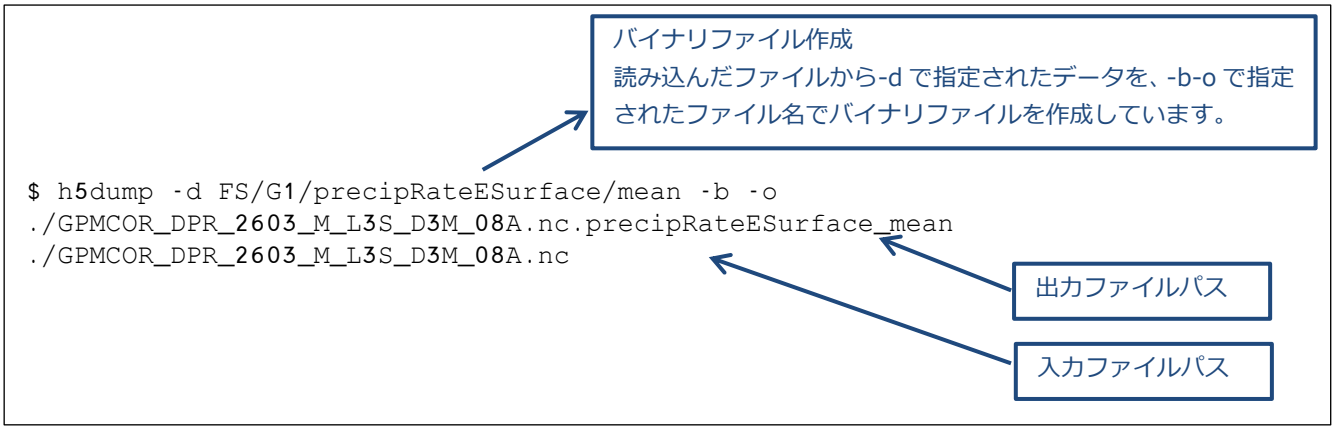
7.1.2 で説明したプログラムの実行結果を示します。

```
$ ./sample_h5dump_L2_F

filename=../data/GPM/DPR/GPMCOR_DPR_2604010211_0345_068594_L2S_DD2_08A.nc.precipR
ateESurface

 precipRateESurface(44,5211)=   2.26120758     (mm/hr)
 precipRateESurface(45,5211)=   2.95193911     (mm/hr)
```

7.2 L3 データ読み込み

7.2.1 バイナリファイルの作成

h5dump を使用してバイナリファイルの作成を行うシェルの作成例を示します。



上記コマンドを実行すると以下のように表示され、出力ファイルパスにバイナリファイルが作成されます。

```
HDF5 "./GPMCOR_DPR_2603_M_L3S_D3M_08A.nc" {
DATASET "FS/G1/precipRateESurface/mean" {
   DATATYPE  H5T_IEEE_F32LE
   DATASPACE  SIMPLE { ( 3, 3, 3, 72, 28 ) / ( 3, 3, 3, 72, 28 ) }
   DATA {
   }
}
}
```

7.2.2 ソースプログラム

以下のサンプルプログラムは、inputfile で指定されたバイナリファイルから情報を読み込んでいます。

```
program sample


    CHARACTER(LEN=74), PARAMETER :: filename =" ../data/GPM/DPR&
&/GPMCOR_DPR_2603_M_L3S_D3M_08A.nc.precipRateESurface_mean"

    ! read data area
    real*4 precipRateESurface(28,72,3,3,3)



    ! binary file open
    open(10,file=filename,access='direct',status='old',recl=4*28*72*3*3*3)



    ! binary file read
    read(10,rec=1) precipRateESurface

    ! print the value
    write(*,*) "filename=",filename
    lat = (140.0/28.0) * (15-1) - 70.0 + (140.0/28.0/2)
    lon = (360.0/72.0) * (64-1) - 180.0 + (360.0/72.0/2)
    write(*,*) "lat=",lat,"lon=",lon
    write(*,*) "precipRateESurface.mean(15,64,1,1,1)=",
precipRateESurface(15,64,1,1,1)
    write(*,*) "precipRateESurface.mean(15,64,1,2,1)=",
precipRateESurface(15,64,1,2,1)

    ! binary file close
    close(10)

end
```

7.2.1 で作成したバイナリファイルを指定しています。

バイナリファイルのオープン
recl：レコード長(precipRateESurface のサイズを指定)

バイナリファイルの読み込み
1 回で全データを precipRateESurface に読み込んでいます。

7.2.3 コンパイル方法

コンパイル時に使用する makefile の例を説明します。

```
F90=gfortran
MACHINE=LINUX
FFLAGS=-g -mcmodel=medium -fPIC -cpp -fdec-structure -D$(MACHINE) -DLANGUAGE_FORTRAN
-DWRITEPIAINFO

MAIN=./sample_h5dump_L3_F

INC = -I$(HDF5_INC) ¥
      -I$(NETCDF_INC) ¥
      -I$(LIBXML2_INC)

LIB = -L$(HDF5_LIB) ¥
      -L$(NETCDF_LIB)

LIBES = -lnetcdf -lm -lhdf5_hl_fortran -lhdf5_fortran -lhdf5_hl -lhdf5 -ljpeg -lxml2
-ltirpc

$(MAIN): $(MAIN).o
      $(F90) $(FFLAGS) -o $@ $^ $(LIB) $(LIBES)

$(MAIN).o: $(MAIN).f90
      $(F90) $(FFLAGS) $(INC) -c $< -o $@

clean:
      rm -f *.o $(MAIN)
```

7.2.1 のプログラムの名前です。

NetCDF のインクルードディレクトリ、ライブラリディレクトリのパスです。

7.2.4 実行結果

7.2.2 で説明したプログラムの実行結果を示します。

```
$ ./sample_h5dump_L3_F
 filename=../data/GPM/DPR/GPMCOR_DPR_2603_M_L3S_D3M_08A.nc.precipRateESurface_mean

 lat=   2.50000000     lon=   137.500000
 precipRateESurface.mean(15,64,1,1,1)=   2.18966341
 precipRateESurface.mean(15,64,1,2,1)=   1.61660624
```

8. 改訂履歴

改版履歴

版数	日付	改版内容	備考
1	2016/1/26		
2	2017/9/13	1. はじめに : 表 1.1 に python の記載を追加、それに伴いフローチャート修正。 表 1.2 サンプルコード動作確認表を追加。 4. ライブラリ・ツールのインストール:表 4.2 プロダクトバージョンと PPS Toolkit(TKIO)の対応バージョンを追加。 表 4.3 動作環境の tkio-3.70.7 と記載している箇所を tkio-x.xx.x に変更 4.2.4 環境設定ファイルの編集: tkio-3.70.7 と記載している箇所を tkio-x.xx.x に変更。	
3	2018/3/15	2. 関連文書、サンプルプログラムの入手方法 : 表 3.1 サンプルプログラム一覧を追加	
4	2019/3/8	1.～3 . TRMM 追加及び GPM サイトリニューアルに伴う修正 4．表 4.2 プロダクトバージョンと TKIO 対応バージョンをプロダクト毎に記載するように修正 5．アルゴリズム ID の一覧表を追加、またサンプルプログラムはレベル毎に 1 つ記載するように変更	
5	2021/12/6	1. GSMaP プロダクトバージョン 5 、GPM/TRMM プロダクトバージョン 7 に修正。 3．関連文書とサンプルプログラムの入手方法修正 4．TKIO ダウンロードの URL 修正	
6	2021/12/24	3.～5. SLP/SLM/SLG/LHP/LHM/LHG のサンプルコード追加に伴い表 3.1、表 4.2、表 5.1 に記載を追加 5.～7.プログラム中の V7 変更部分を修正	
7	2022/1/21	5.～7．サンプルプログラム修正（表示位置見直しや表示項目に lat/lon 追加等）に伴う修正	
8	2022/2/4	誤記修正	
9	2026/6/1	サンプルデータを V8 に更新 コードの説明を V8、NetCDF に合わせて修正	