

DCPAM Reference Manual

GFD Dennou Club

平成 17 年 2 月 18 日

目次

1	Module axis_z_mod	2
1.1	Overview	2
1.2	Error Handling	2
1.3	Known Bugs	2
1.4	Note	2
1.5	Future Plans	2
1.6	Dependency	2
1.7	Public Interface	2
1.8	Procedure Interface	3
1.8.1	Initialize module and acquire NAMELIST	3
1.8.2	Return Sigma Full Level and Sigma Half Level (from NAMELIST) as Data of axis Z	7
1.8.3	Return Sigma Full Level and Sigma Half Level (from netCDF Data) as Data of axis Z	7
1.8.4	Return Data of axis Z from NAMELIST	8
1.8.5	Return Data of axis Z from netCDF file	8
1.8.6	Return Half Level Data of axis Z from NAMELIST	9
1.8.7	Return Half Level Data of axis Z from netCDF file	9
1.8.8	Generate Sigma Level from Sigma Half Level.	10
1.8.9	Terminate module	10

1 Module axis_z_mod

- Developers: Morikawa Yasuhiro
- Version: \$Id: axis_z.f90,v 1.9 2005/01/19 08:52:24 morikawa Exp \$
- Tag Name: \$Name: \$
- Change History:

1.1 Overview

This module set axis Z or axis Altitude.

Z 軸または高度軸を設定する。

1.2 Error Handling

1.3 Known Bugs

- netCDF データから Z 軸を入力する `axis_z_netcdf` にて、および `axis_z_half_netcdf` にて、元データの `units` によらずデータを入力しているが、本来は判定すべき。(特に圧力座標を扱う場合には必須となるはず)。
- 現在は半整数レベルもこのモジュールで読み込んでいるが、別モジュールにすべきかも知れない。(「バグ情報」と違うが)。

1.4 Note

1.5 Future Plans

1.6 Dependency

```
use type_mod,      only : INTKIND, STRING
use axis_type_mod, only : AXISINFO
```

1.7 Public Interface

```
private
public :: axis_z_init, axis_z_end      ! subroutines
public :: axis_z_manual, axis_z_netcdf ! subroutines
public :: axis_z_sigmahalf_manual     ! subroutines
public :: axis_z_sigmahalf_netcdf     ! subroutines
```

```

public :: axis_z_half_manual          ! subroutines
public :: axis_z_half_netcdf         ! subroutines

```

1.8 Procedure Interface

1.8.1 Initialize module and acquire NAMELIST

モジュールを初期化し、NAMELIST から値を取得する。NAMELIST から値が取得できないものに関しては上記のデフォルト値が用いられる。

NAMELIST ファイルは、メインプログラムにて nmlfile_mod の nmlfile_init で指定されることが想定されているが、もしもこの初期化ルーチンより以前に指定されていないならば、nmlfile_init のデフォルトで指定される NAMELIST ファイルを読む。

```

subroutine axis_z_init

```

Dependency

```

use type_mod      ,only: STRING, TOKEN, INTKIND, REKIND, DBKIND, NMLARRAY
use nmlfile_mod,only: nmlfile_init, nmlfile_open, nmlfile_close
use grid_3d_mod,only: km, grid_3d_init
use axis_type_mod, only : axis_attrs_copy, axis_attrs_init
use constants_mod,only: constants_init
use gt4_history,  only: GT_HISTORY_ATTR
use dc_types,    only: GT_TOKEN => TOKEN, GT_STRING => STRING
use dc_url       ,only: GT_ATMARK, GT_QUESTION
use dc_string    ,only: toChar
use dc_trace     ,only: DbgMessage, BeginSub, EndSub
use dc_message  ,only: MessageNotify

```

NAMELIST axis_z_nml

axis_z_nml には、Z 軸の次元変数に関する情報を与える。値を与えないものに関しては以下のデフォルトの値が用いられる。

変数 decision には Z 軸のデータをどのように与えるかを指定する。

- 'manual'
 - Data 配列に格納したデータをそのまま Z 軸として与える。
- 'sigmahalf'

- Z 軸が整数 レベルであると想定し、axis_z_half_nml で与えられる半整数 レベルの値から、自動的に Z 軸のデータを定める。よって、axis_z_half_nml に有効なデータが与えられない場合にはデータは定まらない。
- gtool4 変数 (例えば 'foo.nc@lon' など)
 - 該当する変数から Z 軸のデータを取得する。
- 上記以外
 - 現在は 'sigmahalf' の場合と同様に設定される

変数 length には、grid_3d_mod の公開要素 km と同じ値を与えなければならない。

```
character(TOKEN)  :: name      = 'sigma'          ! 次元変数名
integer(INTKIND)  :: length    = 12              ! 次元長 (配列サイズ)
character(STRING) :: longname  = 'sigma at full level' ! 次元変数の記述的名称
character(STRING) :: units     = 'sigma_level'    ! 次元変数の単位
character(TOKEN)  :: xtype     = 'float'         ! 次元変数の型
character(STRING) :: decision  = 'sigmahalf'     ! 次元データの取得方法
real(REKIND)      :: Data(NMLARRAY) = 0.0! 次元データ入力用
```

```
namelist /axis_z_nml/ &
  & name      , & ! 次元変数名
  & length    , & ! 次元長 (配列サイズ)
  & longname  , & ! 次元変数の記述的名称
  & units     , & ! 次元変数の単位
  & xtype     , & ! 次元変数の型
  & decision  , & ! 次元データの取得方法
  & Data      ! 次元データ
```

NAMELIST axis_z_half_nml

axis_z_half_nml には、Z 軸の半整数レベルの次元変数に関する情報を与える。値を与えないものに関しては以下のデフォルトの値が用いられる。

変数 decision には Z 軸のデータをどのように与えるかを指定する。

- 'manual'

- Data 配列に格納したデータをそのまま Z 軸半整数レベルとして与える。
- gtool4 変数 (例えば 'foo.nc@lon' など)
 - 該当する変数から Z 軸のデータを取得する。
- 上記以外
 - 現在は 'manual' の場合と同様に設定される

変数 length には、grid_3d_mod の公開要素 km にプラス 1 した値を与なければならない。

```

name          = 'sigmahalf'    ! 次元変数名
length        = 13             ! 次元長 (配列サイズ)
longname      = 'sigma at half level' ! 次元変数の記述的名称
units         = 'sigma_level'  ! 次元変数の単位
xtype         = 'float'        ! 次元変数の型
decision      = 'manual'       ! 次元データの取得方法
Data(1:13)    = (/1, 0.99, 0.97, 0.93, 0.85, 0.75, 0.63, 0.5, &
&                0.36, 0.22, 0.1, 0.05, 0/) ! 次元データ入力
用

```

```

namelist /axis_z_half_nml/ &
& name          , & ! 次元変数名
& length        , & ! 次元長 (配列サイズ)
& longname      , & ! 次元変数の記述的名称
& units         , & ! 次元変数の単位
& xtype         , & ! 次元変数の型
& decision      , & ! 次元データの取得方法
& Data          ! 次元データ

```

NAMELIST axis_z_attr_nml

Z 軸の次元変数の属性に関する情報を与える。NAMELIST に複数の axis_z_attr_nml を用意しておく事で複数の情報を与える事が可能である。与えない場合には属性情報は付加されない。

attrtype には与える属性値の種類を設定する。<URL:http://www.gfd-dennou.org/arch/gtool4/gt4f90io-current/doc/gt_history.htm#derived_gthistoryattr> を参照せよ。なお、arraysize に 1 以上の値を設定すると、配列データが優先されて属性値に設定される。

```

character(GT_TOKEN)  :: attrname = '' ! 属性名
character(GT_TOKEN)  :: attrtype = '' ! 属性値の型
character(GT_STRING) :: cvalue   = '' ! 属性の値 (文字)
integer(INTKIND)     :: ivalue   = 0   ! 属性の値 (整数)
real(REKIND)         :: rvalue   = 0.0 ! 属性の値 (単精度実数)
real(DBKIND)         :: dvalue   = 0.0d0 ! 属性の値 (倍精度実数)
logical              :: lvalue   = .false.! 属性の値 (論理)
integer(INTKIND)     :: arraysize= 0   ! 配列のサイズ
integer(INTKIND)    :: iarray(NMLARRAY) = 0 ! 属性の値 (整数)
real(REKIND)        :: rarray(NMLARRAY) = 0.0 ! 属性の値 (単精度実数)
real(DBKIND)        :: darray(NMLARRAY) = 0.0d0! 属性の値 (倍精度実数)

```

```

namelist /axis_z_attr_nml/ &
  & attrname      , & ! 属性名
  & attrtype      , & ! 属性値の型
  & cvalue        , & ! 属性の値 (文字)
  & ivalue        , & ! 属性の値 (整数)
  & rvalue        , & ! 属性の値 (単精度実数)
  & dvalue        , & ! 属性の値 (倍精度実数)
  & lvalue        , & ! 属性の値 (論理)
  & arraysize     , & ! 配列のサイズ
  & iarray        , & ! 属性の値 (整数)
  & rarray        , & ! 属性の値 (単精度実数)
  & darray        , & ! 属性の値 (倍精度実数)

```

NAMELIST axis_z_half_attr_nml

Z 軸の半整数レベルの次元変数の属性に関する情報を与える。NAMELIST に複数の axis_z_half_attr_nml を用意しておく事で複数の情報を与える事が可能である。与えない場合には属性情報は付加されない。

attrtype には与える属性値の種類を設定する。<URL:http://www.gfd-dennou.org/arch/gtool4/gt4f90io-current/doc/gt_history.htm#derived_gthistoryattr> を参照せよ。なお、arraysize に 1 以上の値を設定すると、配列データが優先されて属性値に設定される。

```

namelist /axis_z_half_attr_nml/ &
  & attrname      , & ! 属性名
  & attrtype      , & ! 属性値の型
  & cvalue        , & ! 属性の値 (文字)

```

```

& ivalue      , & ! 属性の値 (整数)
& rvalue      , & ! 属性の値 (単精度実数)
& dvalue      , & ! 属性の値 (倍精度実数)
& lvalue      , & ! 属性の値 (論理)
& arraysize   , & ! 配列のサイズ
& iarray      , & ! 属性の値 (整数)
& rarray      , & ! 属性の値 (単精度実数)
& darray      , & ! 属性の値 (倍精度実数)

```

1.8.2 Return Sigma Full Level and Sigma Half Level (from NAMELIST) as Data of axis Z

NAMELIST から取得した半整数 レベルのデータと、その半整数 レベルから生成した整数 レベルデータを返す。

axis_z_init の NAMELIST axis_z_nml の decision 変数で 'sigmahalf' が与えられ、且つ NAMELIST axis_z_half_nml の decision 変数で 'manual' が与えられている場合以外は、値に何も代入せず返す。

```
subroutine axis_z_sigmahalf_manual(Dim, DimHalf)
```

Dependency

```

use axis_type_mod, only: axis_type_copy
use dc_trace,    only: DbgMessage, BeginSub, EndSub

```

In/Out

type(AxisInfo), intent(inout) :: Dim, DimHalf ! 次元情報を包括する変数

1.8.3 Return Sigma Full Level and Sigma Half Level (from netCDF Data) as Data of axis Z

netCDF データから取得した半整数 レベルのデータと、その半整数 レベルから生成した整数 レベルデータを返す。

axis_z_init の NAMELIST axis_z_nml の decision 変数で 'sigmahalf' が与えられ、且つ NAMELIST axis_z_half_nml の decision 変数で gtool4 変数が与えられている場合以外は、値に何も代入せず返す。

```
subroutine axis_z_sigmahalf_netcdf(Dim, DimHalf)
```


Dependency

```
use axis_type_mod, only: axis_type_copy
use gt4_history, only: HistoryGet
use dc_url ,    only: UrlSplit
use dc_trace,   only: DbgMessage, BeginSub, EndSub
```

In/Out

type(AxisInfo), intent(inout) :: Dim, DimHalf ! 次元情報を包括する変数

1.8.4 Return Data of axis Z from NAMELIST

NAMELIST から代入されたデータを Z 軸データとして返す。
axis_z_init の NAMELIST axis_z_nml の decision 変数で 'manual' 以外が与えられた場合は値を代入しないで返す。

```
subroutine axis_z_manual(Dim)
  use axis_type_mod, only: axis_type_copy
  use dc_trace,    only: DbgMessage, BeginSub, EndSub
```

In/Out

type(AxisInfo), intent(inout) :: Dim ! 次元情報を包括する変数

1.8.5 Return Data of axis Z from netCDF file

netCDF データから取得したデータを Z 軸のデータとして返す。
axis_z_init の NAMELIST axis_z_nml の decision 変数で gtool4 変数以外が与えられた場合は値を代入しないで返す。

```
subroutine axis_z_netcdf(Dim)
```

Dependency

```
use type_mod,      only: STRING
use axis_type_mod, only: axis_type_copy
use gt4_history, only: HistoryGet
use dc_url ,    only: UrlSplit
use dc_trace,   only: DbgMessage, BeginSub, EndSub
```

In/Out

```
type(AXISINFO), intent(inout) :: Dim          ! 次元情報を包括する  
変数
```

1.8.6 Return Half Level Data of axis Z from NAMELIST

NAMELIST から代入された半整数レベルのデータを半整数レベルの Z 軸データとして返す。

axis_z_init の NAMELIST axis_z_half_nml の decision 変数で 'manual' 以外が与えられた場合は値を代入しないで返す。

```
subroutine axis_z_half_manual(DimHalf)  
  use axis_type_mod, only: axis_type_copy  
  use dc_trace,    only: DbgMessage, BeginSub, EndSub
```

In/Out

```
type(AXISINFO), intent(inout) :: DimHalf      ! 次元情報を包括  
する変数
```

1.8.7 Return Half Level Data of axis Z from netCDF file

netCDF データから取得した半整数レベルデータを半整数レベルの Z 軸のデータとして返す。

axis_z_init の NAMELIST axis_z_half_nml の decision 変数で gtool4 変数以外が与えられた場合は値を代入しないで返す。

```
subroutine axis_z_half_netcdf(DimHalf)
```

Dependency

```
use type_mod,      only: STRING  
use axis_type_mod, only: axis_type_copy  
use gt4_history, only: HistoryGet  
use dc_url ,      only: UrlSplit  
use dc_trace,    only: DbgMessage, BeginSub, EndSub
```

In/Out

```
type(AXISINFO), intent(inout) :: DimHalf      ! 次元情報を包括  
する変数
```

1.8.8 Generate Sigma Level from Sigma Half Level.

axis_z_mod の内部サブルーチン。

引数 DimHalf を半整数 レベルとして、その値から整数 レベルを生成し、引数 Dim として返す。

```
subroutine gen_sigma_from_half(DimHalf, Dim)
```

Dependency

```
use type_mod,    only: DBKIND, STRING, REKIND
use constants_mod,only: RAir, Cp
use dc_string,   only: toChar
use dc_trace,    only: DbgMessage, BeginSub, EndSub
```

Input

```
type(AXISINFO), intent(in)    :: DimHalf ! 半整数 レベル
```

In/Out

```
type(AXISINFO), intent(inout):: Dim      ! 整数 レベル
```

1.8.9 Terminate module

axis_z_init で設定された値を破棄し、デフォルトに戻す。

```
subroutine axis_z_end
```

Dependency

```
use dc_trace,    only: DbgMessage, BeginSub, EndSub
```