

FaSTAR

フォーマット

空へ挑み、宇宙を拓く



平成 27 年 1 月



Copyright © 2012 Japan Aerospace Exploration Agency (JAXA)

All rights reserved

本マニュアルの著作権を含む全ての権利は宇宙航空研究開発機構（JAXA）に帰属します。
本マニュアルの全部または一部を，事前の承諾なしに，目的外使用，転載，配布，出版することを禁止します。

宇宙航空研究開発機構（JAXA）

航空本部 数値解析技術研究グループ

〒182-8522 東京都調布市深大寺東町 7-44-1

TEL: 0422-40-3000（代表）

URL: <http://www.aero.jaxa.jp/>



目次

1. 格子データフォーマット.....	1
1.1. fsgrid.....	1
1.2. hnface	6
2. 境界条件フォーマット	7
2.1. PARAMDAT.BClist.....	7

1. 格子データフォーマット

1.1. fsgrid

内容	FaSTAR 格子データ
ファイルタイプ	Unformatted(big-endian)

- ファイルフォーマット

n_p_f n_el_f n_pyr n_prism n_hex n_bd_f n_rect	・・・各エレメントの数
ds_nd(1,1) ds_nd(2,1) ds_nd(3,1)	・・・ノード番号：1
ds_nd(1,2) ds_nd(2,2) ds_nd(3,2)	・・・ノード番号：2
...	
ds_nd(1,n_p_f) ds_nd(2,n_p_f) ds_nd(3,n_p_f)	・・・ノード番号：n_p_f
itet(1,1) itet(2,1) itet(3,1) itet(4,1)	・・・テトラを構成する4つのノード番号
itet(1,2) itet(2,2) itet(3,2) itet(4,2)	
...	
itet(1,n_el_f) itet(2,n_el_f) itet(3,n_el_f) itet(4,n_el_f)	
ipyr(1,1) ipyr(2,1) ipyr(3,1) ipyr(4,1) ipyr(5,1)	・・・ピラミッドを構成する5つのノード番号
ipyr(1,2) ipyr(2,2) ipyr(3,2) ipyr(4,2) ipyr(5,2)	
...	
ipyr(1,n_pyr) ipyr(2,n_pyr) ipyr(3,n_pyr) ipyr(4,n_pyr) ipyr(5,n_pyr)	
ipri(1,1) ipri(2,1) ipri(3,1) ipri(4,1) ipri(5,1) ipri(6,1)	・・・プリズムを構成する6つのノード番号
ipri(1,2) ipri(2,2) ipri(3,2) ipri(4,2) ipri(5,2) ipri(6,2)	
...	
ipri(1,n_prism) ipri(2,n_prism) ipri(3,n_prism) ipri(4,n_prism) ipri(5,n_prism) ipri(6,n_prism)	
ihex(1,1) ihex(2,1) ihex(3,1) ihex(4,1) ihex(5,1) ihex(6,1) ihex(7,1) ihex(8,1)	ヘキサを構成する8つのノード番号
ihex(1,2) ihex(2,2) ihex(3,2) ihex(4,2) ihex(5,2) ihex(6,2) ihex(7,2) ihex(8,2)	
...	
ihex(1,n_hex) ihex(2,n_hex) ihex(3,n_hex) ihex(4,n_hex) ihex(5,n_hex) ihex(6,n_hex) ihex(7,n_hex) ihex(8,n_hex)	
ids_bd(1,1) ids_bd(2,1) ids_bd(3,1) ids_bd(4,1)	・・・三角形を構成する3つのノード番号+境界ゾーン番号
ids_bd(1,2) ids_bd(2,2) ids_bd(3,2) ids_bd(4,2)	
...	
ids_bd(1,n_bd_f) ids_bd(2,n_bd_f) ids_bd(3,n_bd_f) ids_bd(4,n_bd_f)	
ids_rect(1,1) ids_rect(2,1) ids_rect(3,1) ids_rect(4,1) ids_rect(5,1)	四角形を構成する4つのノード番号+境界ゾーン番号
ids_rect(1,2) ids_rect(2,2) ids_rect(3,2) ids_rect(4,2) ids_rect(5,2)	
...	
ids_rect(1,n_rect) ids_rect(2,n_rect) ids_rect(3,n_rect) ids_rect(4,n_rect) ids_rect(5,n_rect)	



● Fortran 読み込み用フォーマット

```

read(io_unit) n_p_f, n_el_f, n_pyr, n_prism, n_hex, n_bd_f, n_rect
read(io_unit) ((ds_nd(n,i), n=1,3), i=1, n_p_f)
read(io_unit) ((itet(n,i), n=1,4), i=1, n_el_f)
read(io_unit) ((ipyr(n,i), n=1,5), i=1, n_pyr)
read(io_unit) ((ipri(n,i), n=1,6), i=1, n_prism)
read(io_unit) ((ihex(n,i), n=1,8), i=1, n_hex)
read(io_unit) ((ids_bd(n,i), n=1,4), i=1, n_bd_f)
read(io_unit) ((ids_rect(n,i),n=1,5), i=1, n_rect)

```

integer :: n_p_f	!ノード数
integer :: n_el_f	!テトラ数
integer :: n_pyr	!ピラミッド数
integer :: n_prism	!プリズム数
integer :: n_hex	!ヘキサ数
integer :: n_bd_f	!三角形境界数
integer :: n_rect	!四角形境界数
real(8) :: ds_nd(1, i)	!X座標
real(8) :: ds_nd(2, i)	!Y座標
real(8) :: ds_nd(3, i)	!Z座標
integer :: itet(n,i), ipyr(n,i), ipri(n,i), ihex(n,i)	!ノード番号
integer :: ids_bd(1~3, i)	!ノード番号
integer :: ids_bd(4, i)	!境界のゾーン番号
integer :: ids_rect(1~4, i)	!ノード番号
integer :: ids_rect(5, i)	!境界のゾーン番号

※ 大規模格子だと Unformatted の 2G 制限を超えてしまうことがあるので、実際には次項のようになっている

- Fortran 読み込み用フォーマット(Unformatted の 2G 制限を考慮する場合)

```

read(io_unit) n_p_f, n_el_f, n_pyr, n_prism, n_hex, n_bd_f, n_rect
! 格子節点 (セル頂点) 座標
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ds_nd) * 3 )
array_loop      = n_p_f/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_p_f )
  read(io_unit) ((ds_nd(n,i), n=1,3), i=array_begin,array_end)
end do
! セル頂点情報
! TETRA
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(itet) * 4 )
array_loop = n_el_f/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_el_f )
  read(io_unit) ((itet(n,i), n=1,4), i=array_begin,array_end)
end do
! PYRAMID
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ipyr) * 5 )
array_loop = n_pyr/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_pyr )
  read(io_unit) ((ipyr(n,i), n=1,5), i=array_begin,array_end)
end do
! PRISM
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ipri) * 6 )
array_loop = n_prism/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_prism )
  read(io_unit) ((ipri(n,i), n=1,6), i=array_begin,array_end)
end do

```

```

! HEXA
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ihex) * 8 )
array_loop = n_hex/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_hex )
  read(io_unit) ((ihex(n,i), n=1,8), i=array_begin,array_end)
end do
! 境界面頂点情報
! TRIANGLE
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ids_bd) * 4 )
array_loop = n_bd_f/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_bd_f )
  read(io_unit) ((ids_bd(n,i), n=1,4), i=array_begin,array_end)
end do
! RECTANGLE
array_allowable = FORT_UNIFORM_IOSIZE / ( kind(ids_rect) * 5 )
array_loop = n_rect/array_allowable + 1
do i_loop = 1, array_loop
  array_begin = (i_loop-1)*array_allowable + 1
  array_end   = min( i_loop*array_allowable, n_rect )
  read(io_unit) ((ids_rect(n,i), n=1,5), i=array_begin,array_end)
end do

```

integer,parameter :: FORT_UNIFORM_IOSIZE = 2147483647

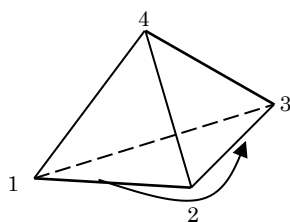
! Fortran unformatted data の record marker が4バイトと想定 = 2**31-1

integer :: array_allowable	!一度の読み書きで許される配列寸法
integer :: array_loop	!分割読み書きするために必要なループ回数
integer :: i_loop	!ループカウンタ
integer :: array_begin	!分割読み書き時の開始番号
integer :: array_end	! 分割読み書き時の終了番号

※ ノード番号の並びについては次項参照

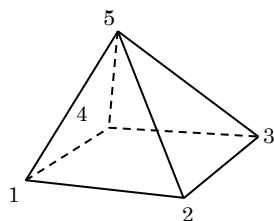
※ 各エレメントのノード番号の並びについて

✓ テトラ



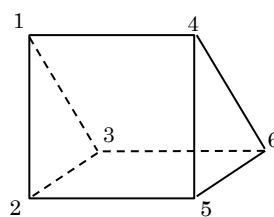
辺 12 と辺 34 はねじれの位置

✓ ピラミッド



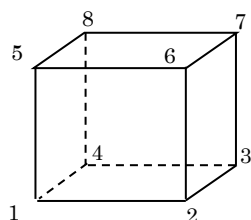
1~4 が下面(四角形)になる。並びに注意
頂点が N₅

✓ プリズム



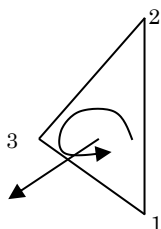
1~3 で一つの三角形面
4~6 でもう一つの三角形面を構成

✓ ヘキサ



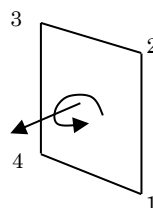
1~4 で一つの四角形面
5~8 で対面の四角形面を構成

✓ 三角形境界



ノード番号順から決まる右ねじの方向が
領域の内側を向く
4 つめの値は境界のゾーン番号

✓ 四角形境界



ノード番号順から決まる右ねじの方向が
領域の内側を向く
5 つめの値は境界のゾーン番号

1.2. hnface

内容	FaSTAR 格子ハンギングノードデータ
ファイルタイプ	ASCII

● ファイルフォーマット

nfh	・・・ハンギングノードを持つ四角面セットの数
n_p_h(1)	・・・1つめの四角面セットの親面と子面の合計数
node(1,1,1) node(1,1,2) node(1,1,3) node(1,1,4)	・・・親面
node(1,2,1) node(1,2,2) node(1,2,3) node(1,2,4)	・・・子面 1
...	
node(1,n_p_h(1),1) node(1, n_p_h(1),2) node(1, n_p_h(1),3) node(1, n_p_h(1),4)	
n_p_h(2)	・・・2つめの四角面セットの親面と子面の合計数
node(2,1,1) node(2,1,2) node(2,1,3) node(2,1,4)	
node(2,2,1) node(2,2,2) node(2,2,3) node(2,2,4)	
...	
node(2,n_p_h(2),1) node(2, n_p_h(2),2) node(2, n_p_h(2),3) node(2, n_p_h(2),4)	
...	
n_p_h(nfh)	・・・nfh めの四角面セットの親面と子面の合計数
node(n_f_h,1,1) node(n_f_h,1,2) node(n_f_h,1,3) node(n_f_h,1,4)	
node(n_f_h,2,1) node(n_f_h,2,2) node(n_f_h,2,3) node(n_f_h,2,4)	
...	
node(nfh,n_p_h(nfh),1) node(nfh, n_p_h(nfh),2) node(nfh, n_p_h(nfh),3) node(nfh, n_p_h(nfh),4)	

● Fortran 読み込み用フォーマット

read(io_unit) nfh
do i = 1,nfh
read(io_unit) n_p_f(i)
do j = 1, n_p_f(i)
read(io_unit) (node(i,j,k), k=1,4)
enddo
enddo

integer :: nfh !ハンギングノードを持つ四角面セットの数
integer :: n_p_f(i) !セット内の親面と子面の合計数
integer :: node(i,1,k) !親面のノード番号
integer :: node(i,j(>1),k) !子面のノード番号

2. 境界条件フォーマット

2.1. PARAMDAT.BClist

- ファイルフォーマット例

9			・・・境界面集合の個数		
1	0	001	Farfield	1	・・・左から、境界属性
10	0	002	Symmetry	2	空力係数を考慮するフラグ
1	0	003	Farfield	3	境界グループ番号
1	0	004	Farfield	4	境界名
1	0	005	Farfield	5	目視用順番
1	0	006	Farfield	6	
20	1	101	Body	7	
20	1	201	Wing	8	
20	1	301	Tail	9	

- 境界属性について

1 : 一様流
10 : 対称面または滑り壁
20 : 滑り無し壁面

- 空力係数を考慮するフラグについて

0 : 空力係数を取得しない
1 : 空力係数を取得する

- 境界グループ番号について

同じ番号に設定した境界の空力係数は合計値が出力される

- 境界名について

任意の名前を設定

- 目視用順番について

FaSTAR では使用しないが、上から順に境界のゾーン番号(fsgrid 内の番号)が割り振られる