

# BPSD 標準データセット仕様書

## 1 BPSD 標準データセットとは

トラスプラズマ統合シミュレーションにおいては、さまざまなコードが同期をとりながら、プラズマや電磁界、装置等の状態を示す情報を交換することによって、プラズマの時間発展を自己無撞着に記述する。コード間でやり取りされる情報は多様な形態をとることができるが、その形態を標準化することによって、コード間のインターフェース作成を容易にすることができる。

BPSD 標準データセットの目的は、プラズマや電磁界、装置等の状態を記述する最小データセットを標準化するとともに、データ交換のためのプログラムインターフェースを開発し、データ処理に必要な機能になるべく取り入れていくことにある。

## 2 外部データ仕様

プログラムインターフェースを通して交換されるデータには、任意のデータに利用でき、データ名称をもつ汎用データ型と、データ内容が定義されている既定データ型がある。

### 2.1 汎用データ型

#### 2.1.1 0次元汎用データ型

```
type bpsd_0ddata_type
  character(len=32) :: dataName
  real(8) :: time
  integer :: ndmax      ! Number of data
  real(8), dimension(:), allocatable :: data
  character(len=32), dimension(:), allocatable :: kid
end type bpsd_0ddata_type
```

`dataName` はデータを識別する文字列、`time` は秒単位の時刻、`ndmax` はデータの個数を指定し、`data` はデータ値を格納する `allocatable` な 1次元配列 (`ndmax`)、`kid` はデータ名称を与える文字列を格納する `allocatable` な 1次元配列 (`ndmax`) である..

#### 2.1.2 1次元汎用データ型

```
type bpsd_1ddata_type
  character(len=32) :: dataName
  real(8) :: time
  integer :: nrmax      ! Number of radial points
  integer :: ndmax      ! Number of data
  real(8), dimension(:), allocatable :: s
  real(8), dimension(:, :), allocatable :: data
  character(len=32), dimension(:), allocatable :: kid
end type bpsd_1ddata_type
```

`nrmax` は径方向の分割数, `s` は磁気軸で 0, プラズマ表面で 1 に規格化されたトロイダル磁束として  $[0, 1]$  で定義される径方向の座標を格納する allocatable な 1 次元配列 (`nrmax`) である. 規格化小半径  $\rho$  の 2 乗に比例し, 両端点を含む. `data` はデータ値を格納する allocatable な 2 次元配列 (`nrmax, ndmax`) である.

### 2.1.3 2次元汎用データ型

```
type bpsd_2ddata_type
  character(len=32) :: dataName
  real(8) :: time
  integer :: nthmax      ! Number of poloidal points
  integer :: nrmax       ! Number of radial points
  integer :: ndmax       ! Number of data
  real(8), dimension(:), allocatable :: th
  real(8), dimension(:), allocatable :: s
  real(8), dimension(:,:,:), allocatable :: data
  character(len=32), dimension(:), allocatable :: kid
end type bpsd_2ddata_type
```

`nthmax` はポロイダル方向の分割数, `th` は区間  $[0, 2\pi)$  で定義されたポロイダル座標  $\theta$  を格納する allocatable な 1 次元配列 (`nthmax`) である.  $\theta$  に関する周期性を仮定し,  $\theta = 2\pi$  の値は  $\theta = 0$  での値に等しいので省略する. `data` はデータ値を格納する allocatable な 3 次元配列 (`nthmax, nrmax, ndmax`) である.

### 2.1.4 3次元汎用データ型

```
type bpsd_3ddata_type
  character(len=32) :: dataName
  real(8) :: time
  integer :: nphmax      ! Number of toroidal points
  integer :: nthmax      ! Number of poloidal points
  integer :: nrmax       ! Number of radial points
  integer :: ndmax       ! Number of data
  real(8), dimension(:), allocatable :: ph
  real(8), dimension(:), allocatable :: th
  real(8), dimension(:), allocatable :: s
  real(8), dimension(:,:,:,:), allocatable :: data
  character(len=32), dimension(:), allocatable :: kid
end type bpsd_3ddata_type
```

`nphmax` はトロイダル方向の分割数, `ph` は区間  $[0, 2\pi)$  で定義されたトロイダル座標  $\phi$  を格納する allocatable な 1 次元配列 (`nphmax`) である.  $\phi$  に関する周期性を仮定し,  $\phi = 2\pi$  の値は  $\phi = 0$  での値に等しいので省略する. `data` はデータ値を格納する allocatable な 4 次元配列 (`nphmax, nthmax, nrmax, ndmax`) である.

## 2.2 既定データ型

あらかじめ、内容が定義されたデータである。一覧表を予定。現在のところ、第5節標準データセットを参照

## 3 データ交換インターフェース

### 3.1 初期化

Subroutine: `bpsd_init`

既定データ型に対する内部データを初期化する。要素として含まれる `allocatable` 配列は割り付けない。

### 3.2 書き込み

Subroutine: `bpsd_set_data(var, ierr)`

汎用データ型あるいは既存データ型の変数 `var` を格納する。`ierr` は出力変数で、その値は以下の通り（現在のところ、割付エラーには対応していない）

---

<code>ierr= 0</code>	正常終了
----------------------	------

---

この subroutine を使用するにあたっては、変数 `var` を定義し、`var%ndmax`, `var%nrmax` 等によって指定された大きさのデータ配列 `var%data` を割り付け、値を代入しておく必要がある。

### 3.3 読み取り

Subroutine: `bpsd_get_data(var, ierr)`

汎用データ型あるいは既存データ型の変数 `var` に値を代入する。`ierr` は出力変数で、その値は以下の通り（現在のところ、割付エラーのすべてには対応していない）

---

<code>ierr= 0</code>	正常終了
<code>1</code>	変数が定義されていない
<code>2</code>	変数に値が代入されていない
<code>3</code>	<code>data</code> 配列の大きさが足りない（割付エラー）

---

この subroutine には2つの使い方がある。`var%nrmax = 0` の場合には、内部データをそのまま `var%data` に代入して戻る。`var%nrmax ≠ 0` の場合には、与えられた `s`, `th`, `ph` 変数に対するデータ値をスプライン補間により求め代入して戻る。`s` に対しては、両端で2階微分が0となる境界条件で補間する。`th`, `ph` に対しては、周期境界条件で補間する。データ配列 `var%data` が割り付けられていなければ、内部で割り付ける。

## 4 内部データ仕様

`bpsd` 内部で格納されるデータ型 `bpsd_0ddatax_type` 等には、次の変数が付加されている。`var%status` はデータ状態を表す。

---

status=	0	変数未定義
	1	データ値未定義
	2	データ変数未割付
	3	データ変数割付済み spline 変数未割付
	4	データ変数割付済み spline 変数割付済み

---

var%spline は spline 補間の係数を保管する配列である.

## 5 標準データセット

c

```
type bpsd_shot_type
  character(len=32) :: deviceID
  integer :: shotID
  integer :: modelID
end type bpsd_shot_type
```

c

```
type bpsd_device_type
  real(8) :: rr      ! Geometrical major radius [m]
  real(8) :: zz      ! Geometrical vertical position [m]
  real(8) :: ra      ! Typical minor radius (Rmax-Rmin)/2 [m]
  real(8) :: rb      ! Typical wall radius [m]
  real(8) :: bb      ! Vacuum toroidal magnetic field at rr [T]
  real(8) :: ip      ! Typical plasma current [A]
  real(8) :: elip    ! Typical ellipticity
  real(8) :: trig    ! Typical triangularity
end type bpsd_device_type
```

c

```
type bpsd_species_data
  real(8) :: pa      ! Mass number (n. of protons + n. of neutrons)
  real(8) :: pz      ! Charge number (n. of protons - n. of electrons)
  real(8) :: pz0     ! Atomic number (n. of protons)
end type bpsd_species_data
type bpsd_species_type
  integer :: nsmax    ! Number of particle species
  type(bpsd_species_data), dimension(:), allocatable :: data
end type bpsd_species_type
```

c

```
type bpsd_equiD_data
  real(8) :: psit    ! Toroidal magnetic flux [Wb]  $\sim \pi r^2 B$ 
  real(8) :: psip    ! Poloidal magnetic flux [Wb]  $\sim 2\pi R r B_p$ 
  real(8) :: ppp     ! Plasma pressure [Pa]
  real(8) :: piq     ! Inverse of safety factor, iota
  real(8) :: pip     ! Poloidal current [A]  $\sim 2\pi R B / \mu_0$ 
  real(8) :: pit     ! Toroidal current [A]  $\sim 2\pi r B_p / \mu_0$ 
```

```

end type bpsd_equ1D_data
type bpsd_equ1D_type
  real(8) :: time
  integer :: nrmax      ! Number of radial points
  real(8), dimension(:), allocatable :: s
                        ! (rho^2) normarized toroidal magnetic flux
  type(bpsd_equ1D_data), dimension(:), allocatable :: data
end type bpsd_equ1D_type

```

c

```

type bpsd_metric1D_data
  real(8) :: pvol      ! Plasma volume [m^3] ~2*pi*R*pi*r^2
  real(8) :: psur      ! Plasma surface [m^2] ~pi*r^2
  real(8) :: dvpsit    ! dV/dPsit
  real(8) :: dvpsip    ! dV/dPsip
  real(8) :: aver2     ! <R^2>
  real(8) :: aver2i    ! <1/R^2>
  real(8) :: aveb2     ! <B^2>
  real(8) :: aveb2i    ! <1/B^2>
  real(8) :: avegv2    ! <|gradV|^2>
  real(8) :: avegvr2   ! <|gradV|^2/R^2>
  real(8) :: avegpp2   ! <|gradPsip|^2>
  real(8) :: averr     ! <R>
  real(8) :: avera     ! <a>
  real(8) :: aveelip   ! <elip>
  real(8) :: avetrig   ! <trig>
end type bpsd_metric1D_data
type bpsd_metric1D_type
  real(8) :: time
  integer :: nrmax      ! Number of radial points
  real(8), dimension(:), allocatable :: s
                        ! (rho^2) normarized toroidal magnetic flux
  type(bpsd_metric1D_data), dimension(:), allocatable :: data
end type bpsd_metric1D_type

```

c

```

type bpsd_plasmaf_data
  real(8) :: pn        ! Number density [m^-3]
  real(8) :: pt        ! Temperature [eV]
  real(8) :: ptp      ! Parallel temperature [eV]
  real(8) :: ptp      ! Perpendicular temperature [eV]
  real(8) :: pu        ! Parallel flow velocity [m/s]
end type bpsd_plasmaf_data
type bpsd_plasmaf_type
  real(8) :: time
  integer :: nrmax      ! Number of radial points

```

```

integer :: nsmax      ! Number of particle species
real(8), dimension(:), allocatable :: s
                        ! (rho^2) : normarized toroidal magnetic flux
real(8), dimension(:), allocatable :: qinv
                        ! 1/q : inverse of safety factor
type(bpsd_plasmaf_data), dimension(:,:), allocatable :: data
end type bpsd_plasmaf_type

```