

Sphere

Skeleton for PHysical and Engineering REsearch

Ver 2.0.0

コンフィグレーションファイル文法マニュアル

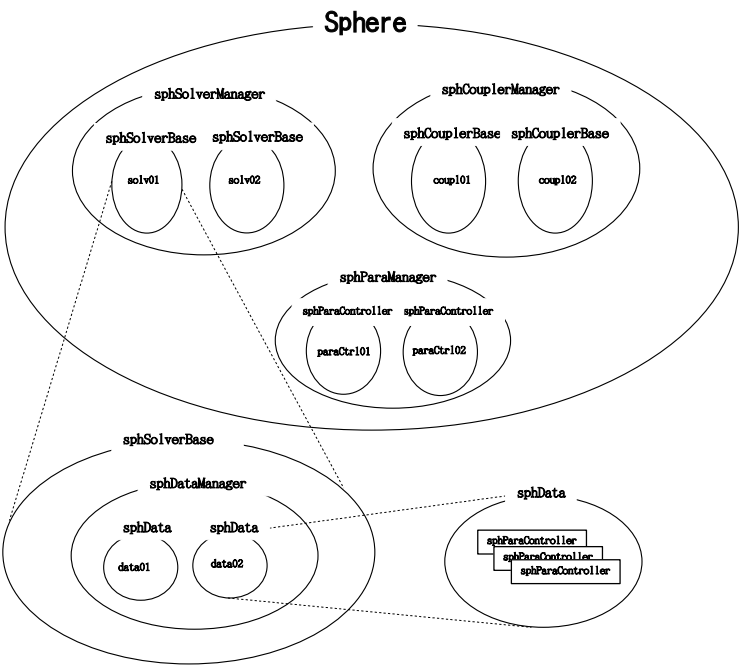
2010 年 8 月 01 日

目次

1.	概要	4
2.	要素一覧	4
2. 1	SphereConfig要素 (トップ要素)	7
2. 2	SphFlow要素 (プログラムフローの指定)	7
2. 3	SphSolverList要素 (ソルバリスト)	9
2. 4	SphCouplerList要素 (カプラリスト)	9
2. 5	SphFileList要素 (ファイルリスト)	10
2. 6	SphStepTime要素 (計算ステップ、時刻設定)	11
2. 7	SphSolver要素 (ソルバ)	12
2. 8	SphDomainInfo要素 (領域情報)	13
2. 9	SphSteer要素 (ソルバの実行パラメータ)	14
2. 10	SphDataList要素 (データリスト)	15
2. 11	SphUserDefine要素 (ユーザ設定パラメータ)	15
2. 12	SphVoxelOrigin要素 (原点座標)	16
2. 13	SphVoxelPitch要素 (ピッチサイズ)	17
2. 14	SphVoxelSize要素 (計算領域のボクセルサイズ)	17
2. 15	SphStartIndex要素 (始点インデックス)	18
2. 16	SphVoxelWidth要素 (計算空間の幅の記述)	19
2. 17	SphDir要素 (ソルバの入出力フォルダ)	19
2. 18	SphNodeProcess要素 (ノードプロセス数)	20
2. 19	SphVoxelDivision要素 (ノード分割方法)	20
2. 20	SphNodeList要素 (ノードリスト)	21
2. 21	SphStartCondition要素 (ソルバの開始条件の記述)	22
2. 22	SphNode要素 (ノード情報)	22
2. 23	SphSize要素 (ボクセルサイズ)	23
2. 24	SphDataObj要素 (データオブジェクト)	23
2. 25	SphCoupler要素 (カプラ)	38
2. 26	SphDataCopy要素 (コピーデータ)	38
2. 27	SphRefObj要素 (カプラ用参照データオブジェクト)	40
2. 28	SphUdtObj要素 (カプラ用更新データオブジェクト)	40
2. 29	SphFile要素 (入出力ファイル指定)	41
2. 30	SphFileNameFormat要素 (入出力ファイル名の書式情報)	44
2. 31	SphFileOption要素 (ファイル入出力オプション情報)	45
2. 32	SphMultiFile要素 (入力ファイル複数指定)	48
2. 33	sph_xml_config_file要素 (外部XMLファイル要素)	49

3. 未定義要素.....	51
---------------	----

1. 概要



2. 要素一覧

No	親要素名	子要素名	説明・用途	必須
1	(トップ要素)	SphereConfig	SPHERE フレームワークで作成されたアプリケーションを実行するためのコンフィグレーションファイルのトップ要素。	必須
2	SphereConfig	SphFlow	プログラムフローを定義する。 この要素がない場合、シンプルフローとなる	任意
3		SphSolverList	フレームワーク内でインスタンスされるソルバを定義する。	必須
4		SphCouplerList	フレームワーク内でインスタンスされるカプラを記述する。	任意
5		SphFileList	アプリケーションで読み込む（もしくは書き出す）ファイルのリストを定義する。	任意
6	SphFlow	SphStepTime	ループステップ数、時間を定義する。	任意
7	SphSolverList	SphSolver	実行ソルバを定義する。	必須

No	親要素名	子要素名	説明・用途	必須
8	SphSolver	SphFlow	プログラムフローを定義する。。 この要素がない場合、シンプルフローとなる	任意
9		SphDomainInfo	計算領域の情報を定義する。	必須
10		SphSteer	ソルバの実行パラメータを定義する。	必須
11		SphDataList	子要素でデータオブジェクトを定義する。	任意
12		SphUserDefine	ユーザ設定のパラメータを定義する。	任意
13	SphDomainInfo	SphVoxelOrigin	原点座標を定義する。	任意
14		SphVoxelPitch	ピッチを定義する。	任意
15		SphVoxelSize	計算領域全体のボクセルサイズを定義する。(ガイドセルは含まない)	必須
		SphStartIndex	始点インデックスを定義する。	任意
16		SphVoxelWidth	計算空間の幅を定義する。	任意
17	SphSteer	SphDir	ソルバの入出力フォルダを定義する。	任意
18		SphNodeProcess	ソルバの実行プロセス数を定義する	任意
19		SphVoxelDivision	ノード分割数を定義する。	任意
20		SphNodeList	ノードリストを定義する。	任意
21		ShpStartCondition	ソルバの開始条件を定義する。	任意
22	SphNodeList	SphNode	ノードの情報を定義する。	任意
23	SphNode	SphSize	ノードのボクセルサイズ、読み込みファイルのサイズを定義する。(ガイドセルは含まない)	任意
24		SphStartIndex	始点グローバルインデックスを定義する。	任意
25	SphDataList	SphDataObj	アプリケーションで使用するデータクラスオブジェクトを定義する。	任意
26	SphDataObj	SphFile	読込データファイルを定義する。	任意
27		SphFileMulti	読込データファイルリストを定義する。	任意
28	SphUserDefine	(任意)	ユーザ設定のパラメータを定義する。	任意
29	SphCouplerList	SphCoupler	カプラを定義する。	任意
30	SphCoupler	SphRefObj	カプラで参照用データとして使用するデータオブジェクトを定義する。	任意
31		SphUdtObj	カプラで更新用データとして使用するデータオブジェクトを定義する。	任意
32	SphFileList	SphFile	入出力ファイルを定義する。	任意
33		SphFileMulti	読込データファイルリストを定義する。	任意

No	親要素名	子要素名	説明・用途	必須
34	SphFile	SphNumberingFormat	入出力ファイル名書式を定義する。	任意
35		SphFileOption	ファイルの入出力のオプション情報を定義する。	任意
36		SphStartIndex	ファイルの始点インデックスを指定する。	任意
37	sph_xml_config_file	(任意)	外部 XML ファイル要素	任意

2. 1 SphereConfig 要素（トップ要素）

記述ルール

要素名	SphereConfig	
用 途	SPHERE フレームワークで作成されたアプリケーションを実行するためのコンフィグレーションファイルのトップ要素。	
属 性		
属性名	値	種別
version	SPHERE バージョン	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFlow	複数ソルバの実行メソッドの記述	必須
SpnSolverList	複数ソルバの設定	必須
SphCouplerList	カブラの設定	任意
SphFileList	入出力ファイルの設定	必須

記述例

```
<?xml version="1.0"?>
<SphereConfig version="2.0">
  <SphFlow></SphFlow>
  <SphSolverList> </SphSolverList>
  <SphCouplerList></SphCouplerList>
  <SphFileList></SphFileList>`
</SphereConfig>
```

2. 2 SphFlow 要素（プログラムフローの指定）

記述ルール

要素名	SphFlow		
用 途	プログラムフローを指定する。 この要素がない場合、シンプルフローとなる		
属 性			
属性名	値		種別

label	プログラムフローの識別ラベル	必須
id	プログラムフローの識別番号	必須
class	インスタンスフロークラス名 以下のクラス名は、デフォルトの予約クラス名とする。 "sph_simple_voxel" : SimpleVoxel タイプのデフォルトフローを行う。	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphStepTime	計算ステップ数を指定する。	任意

記述例

```
<SphFlow label=" sph_simple_voxel " id=1>
  < SphStepTime  max_step="10000"/>
</SphFlow>
```


2. 3 SphSolverList 要素 (ソルバリスト)

記述ルール

要素名	SphSolverList	
用 途	フレームワーク内でインスタンスされるソルバを記述する。 連成を考慮して複数記述可能。	
属 性		
属性名	値	種別
なし		
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphSolver	ソルバ定義	必須

記述例

<pre> <SphSolverList> <SphSolver id="1" label="c3d" class="c3d"> <SphDataList> </SphDataList> <SphUserDefine></SphUserDefine> </SphSolver> <SphSolver id="2" label="mtl"> <SphDataList> </SphDataList> <SphUserDefine></SphUserDefine> </SphSolver> </SphSolverList> </pre>

2. 4 SphCouplerList 要素 (カプラリスト)

記述ルール

要素名	SphCouplerList		
用 途	フレームワーク内でインスタンスされるカプラを記述する。		
属 性			
属性名	値		種別
なし			

内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphCoupler	カプラを指定する	必須

記述例

```

<SphCouplerList>
  <SphCoupler id="1" label="coupler01" >
    <SphFlow label=" sph_simple_voxel " id=1>
    <SphNodeProcess solverID=1>
    <SphDataCopy>
      <SphRefObj solverID="1" dataobjID="1"/>
      <SphUdtObj solverID="2" dataobjID="1"/>
    </ SphDataCopy >
    <SphDataCopy>
      <SphRefObj solverID="1" dataobjID="2"/>
      <SphUdtObj solverID="2" dataobjID="2"/>
    </ SphDataCopy >
  </SphCoupler>
  <SphCoupler id="1" label="coupler01" >
    <SphFlow label="couplerC3D " id=1>
    <SphNodeProcess solverID=2>
    <SphDataCopy>
      <SphRefObj solverID="2" dataobjID="1"/>
      <SphUdtObj solverID="1" dataobjID="1"/>
    </ SphDataCopy >
  </SphCoupler>
</SphCouplerList>

```

2. 5 SphFileList 要素（ファイルリスト）

記述ルール

要素名	SphFileList
用 途	アプリケーションで読み込む（もしくは書き出す）ファイルのリスト
属 性	

属性名	値	種別
id	識別 ID（整数）	任意
label	識別名（文字列）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFile	入出力ファイル指定	任意
SphMultiFile	複数の入力ファイル指定	任意

記述例

```

<SphFileList>
  <SphNumberingSize value="10"/>
  <SphFile id="3" format="spx" name="uvw.spx" commant="コメント"/>
  <SphFile id="4" format="spx" name="tmp.spx" commant ="コメント"/>
  < SphFileMulti>
    <SphFile id="1" format="spx" name="mat00.sbx" commant="コメント"/>
    <SphFile id="2" format="spx" name=" mat01.sbx" commant ="コメント"/>
  </SphFileMulti>
</SphFileList>

```

2. 6 SphStepTime 要素（計算ステップ、時刻設定）

記述ルール

要素名	SphStepTime	
用 途	計算ステップ数、時刻を定義する。	
属 性		
属性名	値	種別
max_step	計算ステップ数（整数） デフォルト＝1000	任意
base_step	基準ステップ デフォルト＝0	任意
base_time	基準時刻 デフォルト＝0.0	任意
delta_time	1ステップ時刻 デフォルト＝0.0	任意
内包可能な要素（子要素として取り得る要素）		

要素名	意味	種別
なし		

記述例

```
< SphStepTime  max_step="1000"  base_step="10
                base_time="5.0"  delta_time = "0.5" />
```

2. 7 SphSolver 要素 (ソルバ)

記述ルール

要素名	SphSolver	
用 途	実行ソルバを定義する。	
属 性		
属性名	値	種別
id	識別 ID（整数）	必須
label	識別名（文字列）	必須
class	インスタンスソルバクラス名	必須
mode	実行モードを設定する。 "check":チェックモード （MainLoop は実行しない。）	任意
dims	次元数を定義する。 "1": 1 次元 "2": 2 次元 "3": 3 次元（デフォルト）	任意
enable	実行可能／不可フラグ "true":実行可能 （デフォルト） "false":実行不可	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFlow	ソルバ独自のプログラムフローを定義する。	任意
SphDomainInfo	ソルバのボクセル定義情報の記述	必須
SphSteer	ソルバの実行パラメータ	必須

SphDataList	データリスト	必須
SphUserDefine	ユーザパラメータ	必須

記述例

```
<SphSolver id="1" lavel="c3d">
  <SphFlow></ SphFlow >
  <SphDomainInfo></SphDomainInfo>
  <SphSteer></SphSteer>
  <SphDataList></SphDataList>
  <SphUserDefine></SphUserDefine>
</SphSolver>
```

2. 8 SphDomainInfo 要素（領域情報）

記述ルール

要素名	SphDomainInfo	
用 途	計算領域の情報を定義する。	
属 性		
属性名	値	種別
id	識別 ID（整数）	任意
label	識別名（文字列）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphVoxelOrigin	原点座標を定義する。	必須
SphVoxelPitch	ピッチを定義する。	任意 (※1)
SphVoxelSize	計算領域のボクセルサイズを定義する。 （ガイドセルは含まない）	必須
SphStartIndex	始点インデックスを定義する。 記述がない場合：{ 0, 0, 0 }	任意
SphVoxelWidth	計算空間の幅を指定する。	任意 (※1)

記述例

```

<ShpDomainInfo>
  <SphVoxelOrigin ox="-0.5" oy="-0.5" oz="-0.5" />
  <SphVoxelSize ix="64" jx="64" kx="64" />
  <SphVoxelPitch dx="1.5625e-02" dy="1.5625e-02" dz="1.5625e-02" />
</SphDomainInfo>

```

(1) <SphVoxelDivision>要素、<SphNodeList>要素が存在しない場合は、分割数から均等分割を自動計算する。

(2) (※1) <SphVoxelPitch>, <SphVoxelWidth>要素のどちらか一方は必ず記述しなければなりません。

2. 9 SphSteer 要素 (ソルバの実行パラメータ)

記述ルール

要素名	SphSteer	
用 途	ソルバの実行パラメータを定義する。	
属 性		
属性名	値	種別
id	識別 ID（整数）	必須
label	識別名（文字列）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphDir	ソルバの入出力フォルダの記述	任意
SphNodeList	ノード別のサイズ、始点グローバルインデックスを定義する。	任意
SphVoxelDivision	ノードの分割方法を定義する。	任意
SphNodeProcess	プロセス数	任意
ShpStartCondition	ソルバの開始条件の記述	任意

(注意点)

(1) <SphNodeList>, <SphVoxelDivision>, <SphNodeProcess>によって、プロセス数、分割方法は決定されるが、優先順位は以下とする。

SphNodeList > SphVoxelDivision > SphNodeProcess

上位の優先順位の要素が記述された場合、下位の要素は無視する。

(2) <SphVoxelDivision>要素、<SphNodeList>要素、<SphNodeProcess>要素が存在しない場合は、ソルバに割当てられたプロセス数から均等分割を自動計算する。

記述例

```
< SphSteer >
  < ShpStartCondition type="initialize"/>
  <SphDir  input="./input"  output="./result"/>
</SphSteer>
```

2. 1 0 SphDataList 要素 (データリスト)

記述ルール

要素名	SphDataList	
用 途	子要素でデータオブジェクトを記述する データオブジェクトは複数記述可能	
属 性		
属性名	値	種別
なし		
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphDataObj	データリスト	必須

記述例

```
<SphDataList>
  <SphDataObj id="1" label="pressure" cmt="必要に応じて他の属性を追加">
    <SphFile id="2"/>
  </SphDataObj>
  <SphDataObj id="2" label="velocity" cmt="必要に応じて他の属性を追加" />
</SphDataList>
```

2. 1 1 SphUserDefine 要素 (ユーザ設定パラメータ)

記述ルール

要素名	SphUserDefine		
用 途	ユーザ設定のパラメータ		
属 性			
属性名	値		種別

なし		
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
任意	ユーザ設定のパラメータ記述	必須

記述例

```

< SphUserDefine >
  <Steer>
    <SphElem name="VoxelDivisionMethod">
      <SphParam name="i" dtype="INT" value="2"/>
      <SphParam name="j" dtype="INT" value="2"/>
      <SphParam name="k" dtype="INT" value="2"/>
    </SphElem>
    <SphParam name="Acceleration"      dtype="REAL"   value="1.0" />
    <SphParam name="Algorithm_Flow"    dtype="STRING" value="EE_FS"/>
  </Steer>
</SphUserDefine >

```

<SphElem>, <SphParam>の場合のみ、name 検索、id 検索メソッドを用意するか？

2. 1 2 SphVoxelOrigin 要素（原点座標）

記述ルール

要素名	SphVoxelOrigin	
用 途	原点座標を定義する。	
属 性		
属性名	値	種別
ox	X 方向の原点座標	必須
oy	Y 方向の原点座標 (2 次元モデルの場合は必須)	任意
oz	Z 方向の原点座標 (3 次元モデルの場合は必須)	任意

内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
<SphVoxelOrigin ox="-0.5" oy="-0.5" oz="-0.5" />
```

2. 1 3 SphVoxelPitch 要素（ピッチサイズ）

記述ルール

要素名	SphVoxelPitch	
用 途	ピッチを定義する。	
属 性		
属性名	値	種別
dx	X 方向のピッチ	必須
dy	Y 方向のピッチ (2 次元モデルの場合は必須)	任意
dz	Z 方向のピッチ (3 次元モデルの場合は必須)	任意
内包可能な要素 (子要素として取り得る要素)		
要素名	意味	種別
なし		

記述例

```
<SphVoxelPitch dx="1.5625e-02" dy="1.5625e-02" dz="1.5625e-02" />
```

2. 1 4 SphVoxelSize 要素（計算領域のボクセルサイズ）

記述ルール

要素名	SphVoxelSize
用 途	計算領域全体のボクセルサイズを定義する。 (ガイドセルは含まない)
属 性	

属性名	値	種別
ix	I 方向のボクセルサイズ	必須
jx	J 方向のボクセルサイズ (2次元モデルの場合は必須)	任意
kx	K 方向のボクセルサイズ (3次元モデルの場合は必須)	任意
内包可能な要素 (子要素として取り得る要素)		
要素名	意味	種別
なし		

記述例

```
< SphVoxelSize ix="64" jx="64" kx="64" />
```

2. 1 5 SphStartIndex 要素 (始点インデックス)

記述ルール

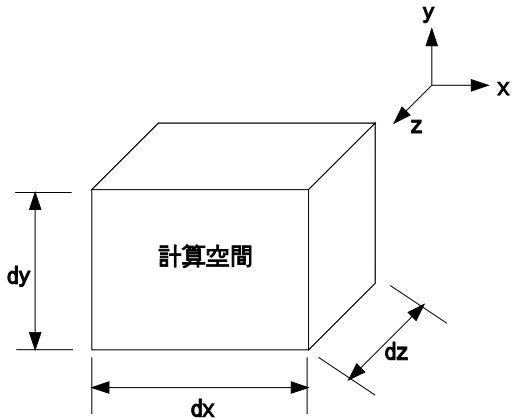
要素名	SphStartIndex	
用 途	始点グローバルインデックスを定義する。	
属 性		
属性名	値	種別
i	I 方向の始点グローバルインデックス	必須
j	J 方向の始点グローバルインデックス (2 次元モデルの場合は必須)	任意
k	K 方向の始点グローバルインデックス (3 次元モデルの場合は必須)	任意
内包可能な要素 (子要素として取り得る要素)		
要素名	意味	種別
なし		

記述例

```
< SphStartIndex i="5" j="4" k="6" />
```

2. 1 6 SphVoxelWidth 要素（計算空間の幅の記述）

計算空間の幅を記述する要素として SPHERE は“SphVoxelWidth”要素を用意しています。”SphVoxelWidth”要素は属性として、“dx”、“dy”及び“dz”を持ち、それぞれ X 軸方法、Y 軸方向及び Z 軸方向の計算空間の幅を実数で指定します。これら属性“dx”、“dy”及び“dz”は省略することはできません。



記述ルール

要素名	SphVoxelWidth	
用 途	計算空間の幅を指定する。	
属 性		
属性名	値	種別
dx	(実数) X 軸方向の幅	必須
dy	(実数) Y 軸方向の幅	必須
dz	(実数) Z 軸方向の幅	必須

記述例【計算空間の幅が (wx, wy, wz) = (128.0, 256.0, 64.0) の場合】

<SphVoxelWidth dx=" 128.0" dy=" 256.0" dz=" 64.0" />

2. 1 7 SphDir 要素（ソルバの入出力フォルダ）

記述ルール

要素名	SphDir
-----	--------

用 途	ソルバの入出力フォルダを記述する。	
属 性		
属性名	値	種別
input	入力フォルダ名 デフォルト : "."	任意
output	出力フォルダ名 デフォルト : "."	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
<SphDir input="c3d/input" output="c3d/result"/>
```

2. 1 8 SphNodeProcess 要素（ノードプロセス数）

記述ルール

要素名	SphNodeProcess	
用 途	ソルバの実行プロセス数を定義する。	
属 性		
属性名	値	種別
value	ソルバの実行プロセス数	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
< SphNodeProcess value="8" />
```

2. 1 9 SphVoxelDivision 要素（ノード分割方法）

記述ルール

要素名	SphVoxelDivision
-----	------------------

用 途	ノード分割数を定義する。	
属 性		
属性名	値	種別
i	I 方向の分割数	必須
j	J 方向の分割数 (2 次元モデルの場合は必須)	任意
k	K 方向の分割数 (2 次元モデルの場合は必須)	任意
内包可能な要素 (子要素として取り得る要素)		
要素名	意味	種別
なし		

記述例

```
< SphVoxelDivision  i="2" j="1" k="1" />
```

2. 2 0 SphNodeList 要素 (ノードリスト)

記述ルール

要素名	SphNodeList	
用 途	ノードリストを定義する。	
属 性		
属性名	値	種別
id	識別 ID（整数）	必須
label	識別名（文字列）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphNode	ノードの情報を定義する。	任意

記述例

```
<SphNodeList>
  <SphNode nodeid="0">
    <ShpSize ix="64" iy="32" iz="32"/>
    <SphStartIndex i="0" j="32" k="32"/>
  </SphNode>
</SphNodeList>
```

```

</SphNode nodeid>
<SphNode nodeid="1">
  <ShpSize ix="64" iy="32" iz="32"/>
  <SphStartIndex i="32" j="0" k="0"/>
</SphNode nodeid>
</SphNodeList>

```

2. 2 1 SphStartCondition 要素（ソルバの開始条件の記述）

記述ルール

要素名	SphStartCondition	
用 途	ソルバの開始条件を記述する。	
属 性		
属性名	値	種別
type	開始条件（Initial もしくは Restart）	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
<SphStartCondition type="initial" />
```

2. 2 2 SphNode 要素（ノード情報）

記述ルール

要素名	SphNode	
用 途	ノードの情報を定義する。	
属 性		
属性名	値	種別
nodeid	ノード番号	必須
label	識別名（文字列）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphSize	ノード別のボクセルサイズを定義する。	任意

SphStartIndex	ノード別の始点グローバルインデックスを定義する。	任意
---------------	--------------------------	----

記述例

```
<SphNode nodeid="0">
  <SphSize ix="10" jx="20" kx="15"/>
  <SphStartIndex i="5" j="10" k="0"/>
</SphNode>
```

2. 2 3 SphSize 要素 (ボクセルサイズ)

記述ルール

要素名	SphSize	
用 途	ノードのボクセルサイズ、読み込みファイルのサイズを定義する。（ガイドセルは含まない）	
属 性		
属性名	値	種別
ix	I 方向のボクセルサイズ	必須
jx	J 方向のボクセルサイズ （２次元モデルの場合は必須）	任意
kx	K 方向のボクセルサイズ （３次元モデルの場合は必須）	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
< SphSize ix="64" jx="64" kx="64" />
```

2. 2 4 SphDataObj 要素 (データオブジェクト)

記述ルール

要素名	SphDataObj
-----	------------

用 途	アプリケーションで使用するデータクラスオブジェクトを記述する。 フラグによってアプリケーション起動時にフレームワーク側でインスタンスするか、ユーザが明示的にインスタンスするかを決定できる。	
属 性		
属性名	値	種別
label	識別名（文字列）	必須
format	インスタンスするデータクラスの種類 scalar, scalar2d, scalar3d, vector2d, ...	必須
data_type	データタイプの種類 char, short, int, long, longlong, unsigned_char, unsigned_short, unsigned_int, unsigned_long, float, double	必須
instance	インスタンスフラグ "initialize": 起動時にフレームワーク側でインスタンスする（デフォルト） "user": ソルバ側でインスタンスする。	任意
domain	作成エリア 実行 Solver の SphDomainInfo, SphSteer から算出 "node": 分割された自ノード分の領域サイズ（デフォルト） "solver": ソルバ領域サイズ	任意
crddef	データ定義タイプ "regular": 格子上データ定義（デフォルト） "collocate": 格子中心データ定義 "staggered_1": 格子中心マイナス方向データ定義 "staggered_2": 格子中心プラスデータ定義	任意
gc	ガイドセル数（デフォルト：0）	任意
data_num	1 格子当りのデータ数 データクラスの種類がベクトルデータのみ有効 デフォルト：次元数 （例） 3D の場合：デフォルトデータ数＝3	任意

comment	コメント	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFile	読込データファイル	任意
SphMultiFile	読込データファイルリスト	任意

記述例

```
<SphDataObj label="pressure" >
  <SphFile id="2"/>
</SphDataObj>
```

(1) <SphFile>の id=2 を読み込む

redid が記述されている場合は<SphFileList>から同一"id"の SphFile 要素を参照する。

(2) 読込データを属性設定されている情報にて変換を行う。

（データ定義タイプ：crddef 属性）

データ定義タイプは以下の4つをサポートします。

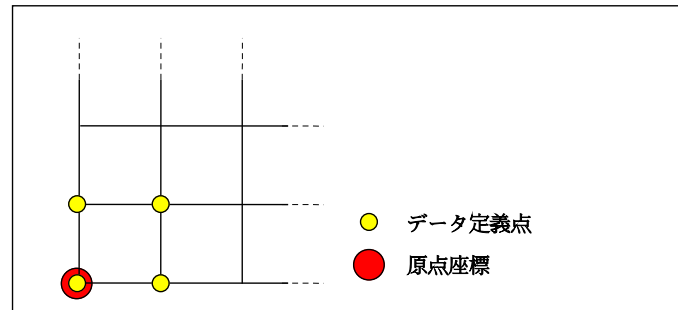
"staggered_1", "staggered_2"については、データクラス（format 属性）、データ数（data_num 属性）はベクトルデータクラスのみサポートします。

No	crddef 属性値	定義説明	サポート対象	
			データクラス	データ数
1	"regular"	格子上データ定義（デフォルト）	すべて	無制限
2	"collocate"	格子中心データ定義	すべて	無制限
3	"staggered_1"	格子中心マイナス方向データ定義	"vector2d" "vector2drv" "vector3d" "vector3drv"	次元数
4	"staggered_2"	格子中心プラスデータ定義	"vector2d" "vector2drv" "vector3d" "vector3drv"	次元数

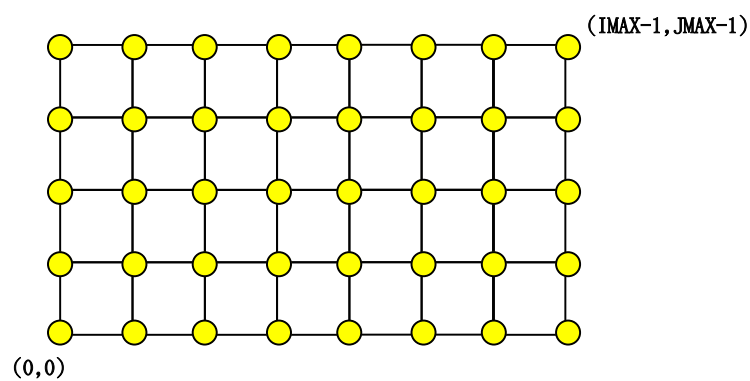
(1) regular

原点座標から格子ピッチのグリッドの交わる点にデータを定義する。

よって、データ座標は原点座標と格子ピッチに等間隔で配置される。



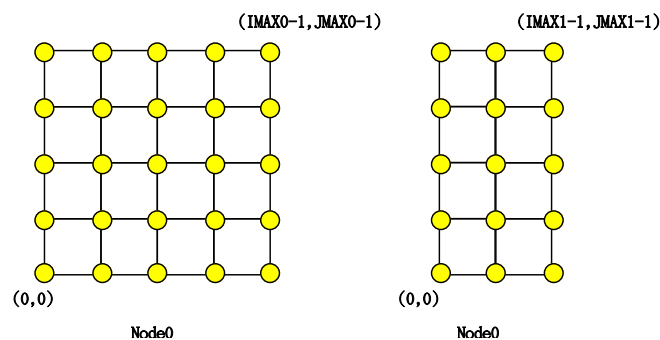
SphSize 要素の{ix, iy, iz}属性の設定値を{IMAX, JMAX, KMAX}のボクセルサイズにて定義した場合、データ領域は以下のサイズの確保を行う。



アロケートサイズ = {IMAX, JMAX, KMAX}

領域内データ範囲 = {0,0,0} ~ {IMAX-1, JMAX-1, KMAX-1}

並列実行においてもすべてのノードで分割ボクセルサイズとデータサイズは等しい。



計算領域全体 $\{IMAX, JMAX, KMAX\}$ を $Node0=\{IMAX0, JMAX0, KMAX0\}$,
 $Node1=\{IMAX1, JMAX1, KMAX1\}$ で分割した場合

Node0

アロケートサイズ $= \{IMAX0, JMAX0, KMAX0\}$

領域内データ範囲 $= \{0,0,0\} \sim \{IMAX0-1, JMAX0-1, KMAX0-1\}$

Node1

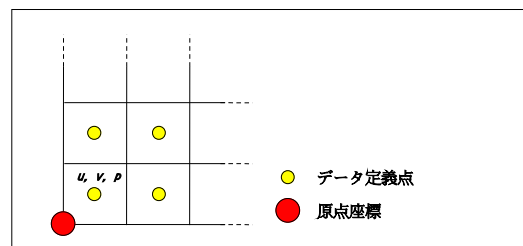
アロケートサイズ $= \{IMAX1, JMAX1, KMAX1\}$

領域内データ範囲 $= \{0,0,0\} \sim \{IMAX1-1, JMAX1-1, KMAX1-1\}$

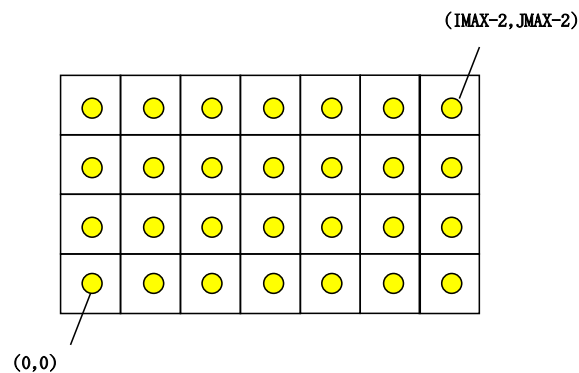
(2) collocate

格子の中央にデータを定義する。

よって、データ座標は原点座標に対して格子ピッチ／2の位置に配置される。



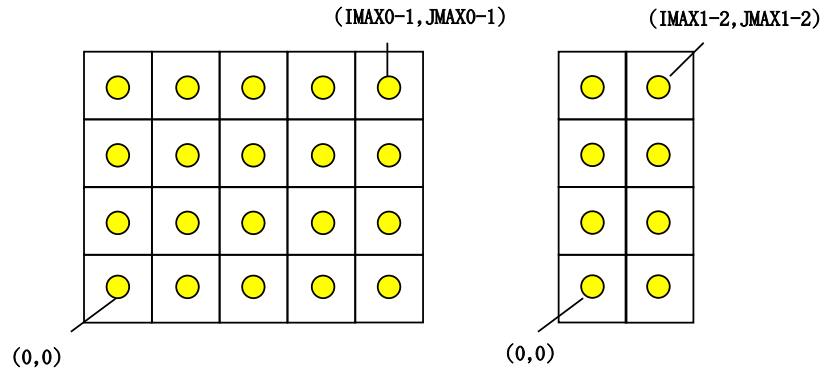
SphSize 要素の $\{ix, iy, iz\}$ 属性の設定値を $\{IMAX, JMAX, KMAX\}$ のボックスサイズにて定義した場合、データ領域は以下のサイズの確保を行う。



アロケートサイズ $= \{IMAX-1, JMAX-1, KMAX-1\}$

領域内データ範囲 $= \{0,0,0\} \sim \{IMAX-2, JMAX-2, KMAX-2\}$

並列実行においては最外郭ノードのみボクセルサイズに対してデータサイズは-1 少なくなる。



計算領域全体 $\{IMAX, JMAX, KMAX\}$ を $Node0=\{IMAX0, JMAX0, KMAX0\}$,
 $Node1=\{IMAX1, JMAX1, KMAX1\}$ で分割した場合

Node0

アロケートサイズ $=\{IMAX0, JMAX0, KMAX0\}$

領域内データ範囲 $= \{0,0,0\} \sim \{IMAX0-1, JMAX0-1, KMAX0-1\}$

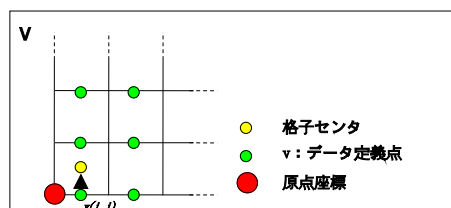
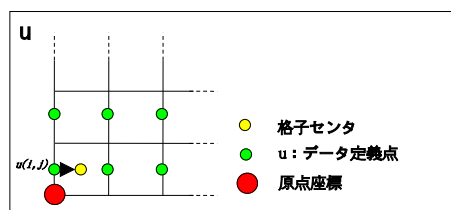
Node1

アロケートサイズ $=\{IMAX1-1, JMAX1-1, KMAX1-1\}$

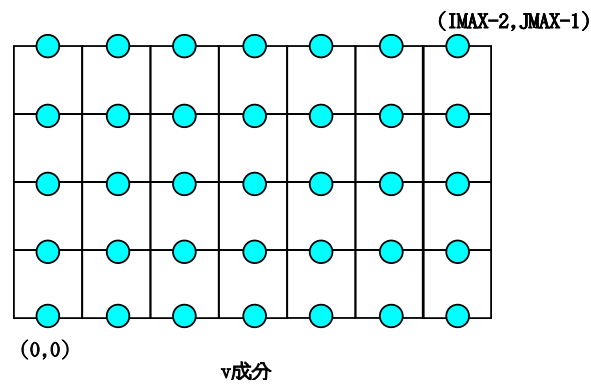
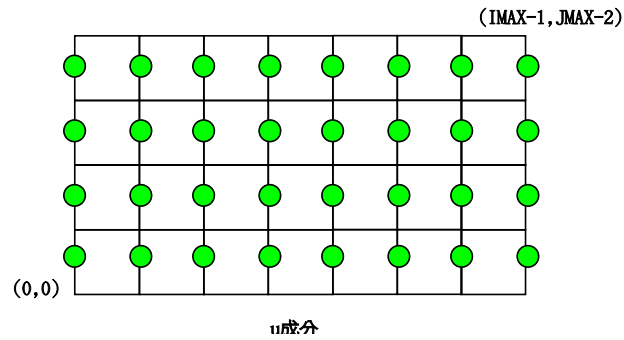
領域内データ範囲 $= \{0,0,0\} \sim \{IMAX1-2, JMAX1-2, KMAX1-2\}$

(3) staggered_1

u, v, w のベクトルデータであり、格子センタから u は-I 方向、 v は-J 方向、 w は-K 方向に格子ピッチ/2 の位置に配置される。



SphSize 要素の{ix, iy, iz}属性の設定値を{IMAX, JMAX, KMAX}のボクセルサイズにて定義した場合、データ領域は以下のサイズの確保を行う。



アロケートサイズ = {IMAX, JMAX, KMAX}

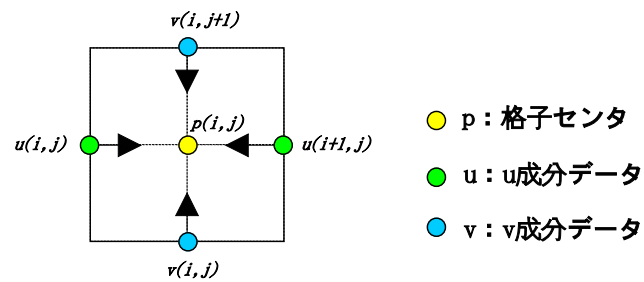
領域内データ範囲 =

u 成分: {0,0,0} ~ {IMAX-1, JMAX-2, KMAX-2}

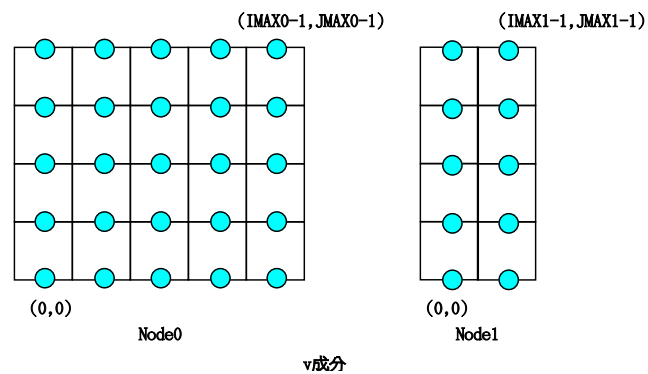
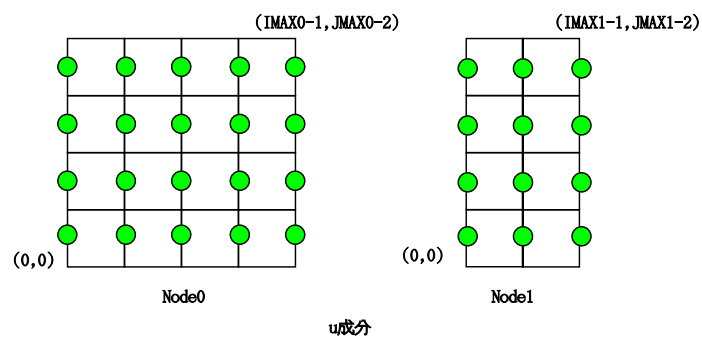
v 成分: {0,0,0} ~ {IMAX-2, JMAX-1, KMAX-2}

w 成分: {0,0,0} ~ {IMAX-2, JMAX-2, KMAX-1}

格子センタに p(圧力)データが存在する場合、p と u,v,w 成分の関係は以下となる。



並列実行においてもすべてのノードで分割ボクセルサイズとデータサイズは等しい。



計算領域全体 $\{IMAX, JMAX, KMAX\}$ を $Node0=\{IMAX0, JMAX0, KMAX0\}$,
 $Node1=\{IMAX1, JMAX1, KMAX1\}$ で分割した場合

Node0

アロケートサイズ $=\{IMAX0, JMAX0, KMAX0\}$

領域内データ範囲 =

u 成分: $\{0,0,0\} \sim \{IMAX0-1, JMAX0-2, KMAX0-2\}$

v 成分: $\{0,0,0\} \sim \{\text{IMAX0-2}, \text{JMAX0-1}, \text{KMAX0-2}\}$

w 成分: $\{0,0,0\} \sim \{\text{IMAX0-2}, \text{JMAX0-2}, \text{KMAX0-1}\}$

Node1

アロケートサイズ = $\{\text{IMAX1}, \text{JMAX1}, \text{KMAX1}\}$

領域内データ範囲 =

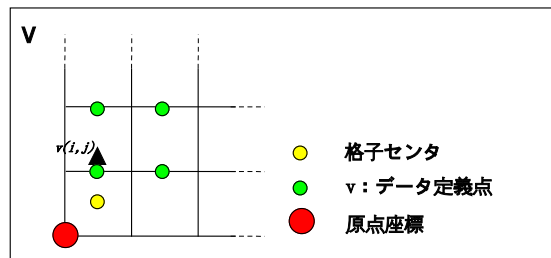
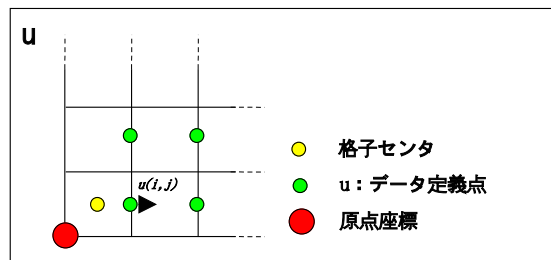
u 成分: $\{0,0,0\} \sim \{\text{IMAX1-1}, \text{JMAX1-2}, \text{KMAX1-2}\}$

v 成分: $\{0,0,0\} \sim \{\text{IMAX1-2}, \text{JMAX1-1}, \text{KMAX1-2}\}$

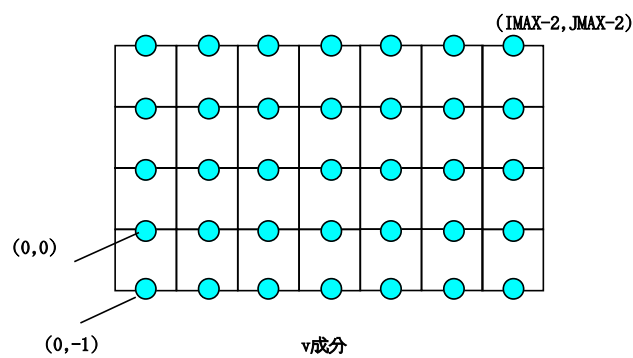
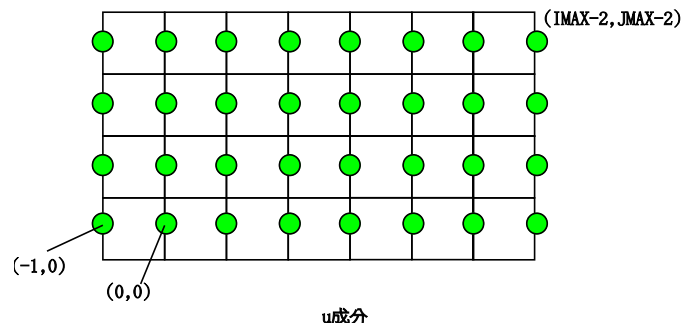
w 成分: $\{0,0,0\} \sim \{\text{IMAX1-2}, \text{JMAX1-2}, \text{KMAX1-1}\}$

(4) staggered_2

u, v, w のベクトルデータであり、格子センタから u は+I 方向、v は+J 方向、w は+K 方向に格子ピッチ／2 の位置に配置される。



SphSize 要素の $\{ix, iy, iz\}$ 属性の設定値を $\{\text{IMAX}, \text{JMAX}, \text{KMAX}\}$ のボックスサイズにて定義した場合、データ領域は以下のサイズの確保を行う。



アロケートサイズ = {IMAX, JMAX, KMAX}

領域内データ範囲 =

u 成分: {-1,0,0} ~ {IMAX-2, JMAX-2, KMAX-2}

v 成分: {0,-1,0} ~ {IMAX-2, JMAX-2, KMAX-2}

w 成分: {0,0,-1} ~ {IMAX-2, JMAX-2, KMAX-2}

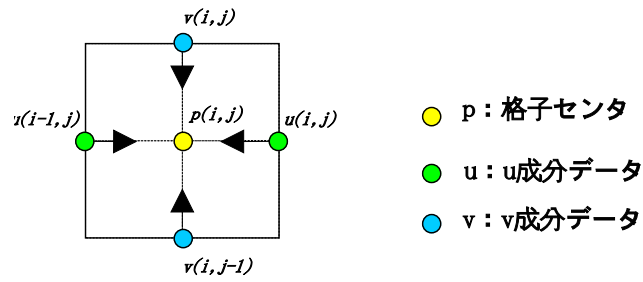
実際のデータ配列上のデータ範囲 =

u 成分: {0,0,0} ~ {IMAX-1, JMAX-2, KMAX-2}

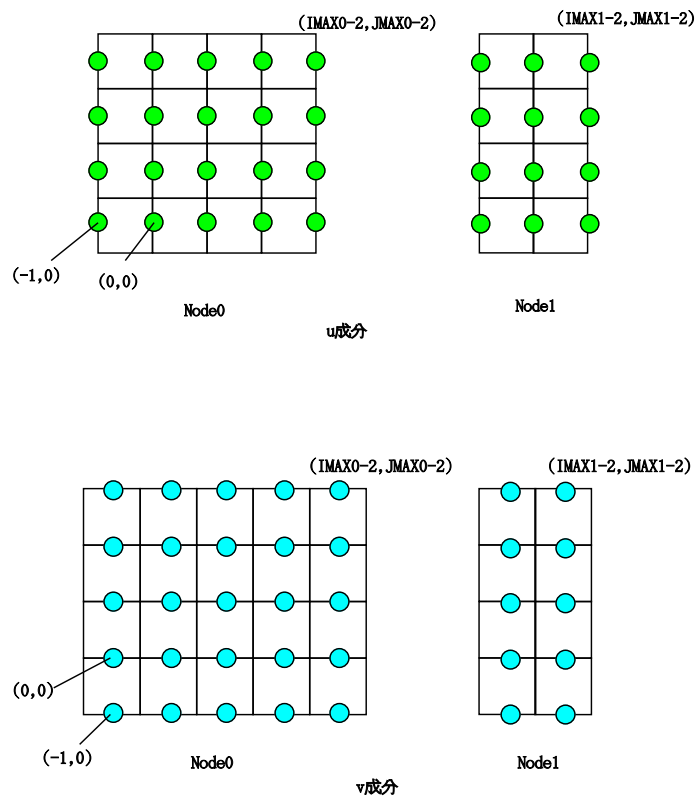
v 成分: {0,0,0} ~ {IMAX-2, JMAX-1, KMAX-2}

w 成分: {0,0,0} ~ {IMAX-2, JMAX-2, KMAX-1}

格子センタに p(圧力)データが存在する場合、p と u,v,w 成分の関係は以下となる。



並列実行においてもすべてのノードで分割ボックスサイズとデータサイズは等しい。



計算領域全体 $\{IMAX, JMAX, KMAX\}$ を $Node0=\{IMAX0, JMAX0, KMAX0\}$,
 $Node1=\{IMAX1, JMAX1, KMAX1\}$ で分割した場合

Node0

アロケートサイズ $=\{IMAX0, JMAX0, KMAX0\}$

領域内データ範囲 =

u 成分: $\{-1, 0, 1\} \sim \{IMAX0-2, JMAX0-2, KMAX0-2\}$

v 成分: $\{0, -1, 1\} \sim \{IMAX0-2, JMAX0-2, KMAX0-2\}$

w 成分: {0,0,-1} ~ {IMAX0-2, JMAX0-2, KMAX0-2}

実際のデータ配列上のデータ範囲 =

u 成分: {0,0,0} ~ {IMAX0-1, JMAX0-2, KMAX0-2}

v 成分: {0,0,0} ~ {IMAX0-2, JMAX0-1, KMAX0-2}

w 成分: {0,0,0} ~ {IMAX0-2, JMAX0-2, KMAX0-1}

Node1

アロケートサイズ = {IMAX1, JMAX1, KMAX1}

領域内データ範囲 =

u 成分: {-1,0,0} ~ {IMAX1-2, JMAX1-2, KMAX1-2}

v 成分: {0,-1,0} ~ {IMAX1-2, JMAX1-2, KMAX1-2}

w 成分: {0,0,-1} ~ {IMAX1-2, JMAX1-2, KMAX1-2}

実際のデータ配列上のデータ範囲 =

u 成分: {0,0,0} ~ {IMAX1-1, JMAX1-2, KMAX1-2}

v 成分: {0,0,0} ~ {IMAX1-2, JMAX1-1, KMAX1-2}

w 成分: {0,0,0} ~ {IMAX1-2, JMAX1-2, KMAX1-1}

(データクラスの種類 : format 属性)

生成するデータのスカラ、ベクトル、次元数を定義する。

サポートするデータクラスの種類は以下である。

No	format 属性値	次元 数	スカラ ベクト ル	データクラス説 明	SphDCType 値
1	"scalar1d"	1	スカラ	1D データクラス データ配列 {i}	SPH_DC_ARRAY_1D
2	"scalar2d"	2	スカラ	2D データクラス データ配列 {i, j}	SPH_DC_ARRAY_2D
3	"scalar3d"	3	スカラ	3D データクラス データ配列 {i, j, k}	SPH_DC_ARRAY_3D
4	"vector1d"	1	ベクト	1D データクラス	SPH_DC_ARRAY_1DN

No	format 属性値	次元 数	スカラ ベクト ル	データクラス説 明	SphDCType 値
			ル	ス データ配列{i, l}	
5	"vector1drv"	1	ベクト ル	1D データクラ ス データ配列{l, i}	SPH_DC_ARRAY_1DNRV
6	"vector2d"	2	ベクト ル	2D データクラ ス データ配列{i, j, l}	SPH_DC_ARRAY_2DN
7	"vector2drv"	2	ベクト ル	2D データクラ ス データ配列{l, i, j}	SPH_DC_ARRAY_2DNRV
8	"vector3d"	3	ベクト ル	3D データクラ ス データ配列{i, j, k,l}	SPH_DC_ARRAY_3DN
9	"vector3drv"	3	ベクト ル	3D データクラ ス データ配列{l,i, j, k}	SPH_DC_ARRAY_3DNRV
10	"unst_node"	3	スカラ	非構造データク ラス（節点、ス カラ）	SPH_DC_UNST_NODE
11	"unst_node_n"	3	ベクト ル	非構造データク ラス（節点、ベ クトル）	SPH_DC_UNST_NODE_N
12	"unst_node_nrv"	3	ベクト ル	非構造データク ラス（節点、ベ クトル、反転）	SPH_DC_UNST_NODE_NRV
13	"unst_elem"	3	スカラ	非構造データク ラス（要素、ス カラ）	SPH_DC_UNST_ELEM

No	format 属性値	次元 数	スカラ ベクトル	データクラス説明	SphDCType 値
14	"unst_elem_n"	3	ベクトル	非構造データクラス（要素、ベクトル）	SPH_DC_UNST_ELEM_N
15	"unst_elem_nrv"	3	ベクトル	非構造データクラス（要素、ベクトル、反転）	SPH_DC_UNST_ELEM_NRV

（データタイプの種類：data_type 属性）

生成するデータのデータ型を定義する。

サポートするデータタイプは以下である。

No	data_type 属性	データタイプ説明	SPL_Datatype
1	"char"	char 型データ	SPL_CHAR
2	"unsigned_char"	unsigned char データ型	SPL_UNSIGNED_CHAR
3	"short"	short 型データ型	SPL_SHORT
4	"unsigned_short"	unsigned short データ型	SPL_UNSIGNED_SHORT
5	"int"	int 型データ型	SPL_INT
6	"unsigned_int"	unsigned int データ型	SPL_UNSIGNED
7	"long"	long 型データ型	SPL_LONG
8	"unsigned_long"	unsigned long データ型	SPL_UNSIGNED_LONG
9	"longlong"	long long 型データ型	SPL_LONG_LONG
10	"float"	float 型データ型	SPL_FLOAT
11	"double"	double 型データ型	SPL_DOUBLE

（1 格子当りのデータ数：data_num 属性）

1 格子当りのデータ数（data_num 属性）が定義されていない場合のデフォルト値はデータクラスの種類（format 属性）により異なる。

ベクトルデータは次元数とします。

スカラデータの場合は data_num 属性を定義することはできず、1 固定となります。

No	format 属性値	次元 数	スカラ ベクトル	data_num 属性 デフォルト値
1	"scalar1d"	1	スカラ	1（固定）

No	format 属性値	次元 数	スカラ ベクトル	data_num 属性 デフォルト値
2	"scalar2d"	2	スカラ	1 (固定)
3	"scalar3d"	3	スカラ	1 (固定)
4	"vector1d"	1	ベクトル	1
5	"vector1drv"	1	ベクトル	1
6	"vector2d"	2	ベクトル	2
7	"vector2drv"	2	ベクトル	2
8	"vector3d"	3	ベクトル	3
9	"vector3drv"	3	ベクトル	3

(読込データファイル)

<SphDataObj>要素の子要素に<SphFile>,<SphMultiFile>要素が記述された場合、<SphFile>,<SphMultiFile>要素に記述されたファイルから自動的に読み込み、データオブジェクトを生成します。

しかし、読込ファイルとデータクラス、データタイプが一致していないとエラーとなります。

サポートを行っているデータオブジェクトとファイルフォーマットの関係は以下となります。

No	データオブジェクト要素		ファイル要素
	データクラス (format 属性)	データタイプの種類 (data_type 属性)	ファイルフォーマット (format 属性)
1	"scalar1d" "scalar2d" "scalar3d"	"unsigned_char" "unsigned_short" "unsigned_int"	"sbx"
2	"scalar1d" "scalar2d" "scalar3d" "vector1drv" "vector2drv" "vector3drv"	"float" "double"	"spx"

2. 2 5 SphCoupler 要素（カプラ）

カプラは、参照用データから更新用データにデータをコピーする。

記述ルール

要素名	SphCoupler	
用 途	カプラを指定する	
属 性		
属性名	値	種別
id	識別 ID（整数）	任意
label	識別名（文字列）	必須
class	インスタンスカプラクラス名	必須
enable	実行可能／不可フラグ "true":実行可能 （デフォルト） "false":実行不可	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphDataCopy	カプラでデータコピーを行うデータオブジェクトをを指定する。	必須

記述例

```
<SphCoupler id="1" label="coupler01" >
  <SphDataCopy>
    <SphRefObj solverID="1" >
      <SphDataObj label="prs" />
    </SphRefObj>
    <SphUdtObj solverID="2">
      <SphDataObj label="prs" />
    </SphUdtObj>
  </ SphDataCopy >
</SphCoupler>
```

2. 2 6 SphDataCopy 要素（コピーデータ）

参照用データから更新用データにデータをコピーする。

記述ルール

要素名	SphDataCopy	
用 途	参照用データ要素と更新用データ要素を指定する	
属 性		
属性名	値	種別
id	識別 ID（整数）	任意
label	識別名（文字列）	任意
ext_gc	拡張ガイドセル データコピー（連成）を行う時に余分にコピーを行う領域セル値である。 データコピー領域は以下のサイズとなる。 data_size + 2*data_gc + 2*ext_gc data_size： データのサイズ data_gc： データのガイドセル デフォルト＝0	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphRefObj	カブラで参照用データとして使用するデータオブジェクトを指定する	必須
SphUdtObj	カブラで更新用データとして使用するデータオブジェクトを指定する	必須

記述例

```

<SphCoupler id="1" label="coupler01">
  < SphDataCopy >
    <SphRefObj solverID="1">
      <SphDataObj label="prs" />
    </SphRefObj>
    <SphUdtObj solverID="2">
      <SphDataObj label="prs" />
    </SphUdtObj>
  </ SphDataCopy >
</SphCoupler>

```

2. 2 7 SphRefObj 要素（カプラ用参照データオブジェクト）

記述ルール

要素名	SphRefObj	
用 途	カプラで参照用データとして使用するデータオブジェクトを指定する	
属 性		
属性名	値	種別
solverId	参照用データのソルバ識別 ID（整数）	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphDataObj	参照用データを指定する。	必須

記述例

```
<SphRefObj solverID="1" >
  <SphDataObj label="pressure"/>
</SphRefObj>
```

2. 2 8 SphUdtObj 要素（カプラ用更新データオブジェクト）

記述ルール

要素名	SphUdtObj	
用 途	カプラで更新用データとして使用するデータオブジェクトを指定する	
属 性		
属性名	値	種別
solverId	更新用データのソルバ識別 ID（整数）	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphDataObj	更新用データを指定する。	必須

記述例

```
<SphUdtObj solverID="2" />
  <SphDataObj label="pressure"/>
```


</SphUdtObj>

2. 2 9 SphFile 要素（入出力ファイル指定）

記述ルール

要素名	SphFile	
用 途	入出力ファイルの指定に使用する	
属 性		
属性名	値	種別
id	識別 ID（整数）	必須(※1)
label	識別名（文字列）	必須(※1)
format	ファイルフォーマット "sbx"：SBX ファイル "spx"：SPX ファイル "sph"：SPH ファイル "text"：テキストファイル	必須(※1)
file_name	ファイル名 拡張子は自動的に付加しませんので、 拡張子まで記述してください。	必須(※2)
prefix	ファイル名接頭文字 ファイル名に付加する接頭文字 ステップ数、ランク番号、拡張子は命名規約に従ってファイル名を自動生成します。	任意(※2)
type	入出力タイプ in：入力ファイル out：出力ファイル	必須(※1)
mode	入出力モード multi：各ノードが読込・書込を行う。 cent：指定ノードが読込。書込を行う。 デフォルト= multi(※3)	任意
io_nodeid	ファイルの読み書きを行うノード番号 入出力モード（mode）が"cent"の時のみ有効。	任意

	デフォルト=0	
comment	コメント	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFileNameFormat	ファイル名の書式を定義する。	任意(※2)
SphFileOption	ファイルの入出力のオプション情報を定義する。	任意
SphStartIndex	ファイルの始点インデックスを指定する。	任意

記述例

```

<SphFile id="3" format="spx" file_name="uvw.spx" comment="コメント"/>
<SphFile id="4" format="spx" file_name="tmp.spx" comment="コメント"/>
<SphFile id="5" format="spx" comment="必要に応じて他の属性を追加">
  <SphFileNameFormat format="prs_id%04d_%08d.spx" value="step_rank"/>
</SphFile>

```

(1) ファイルパス

"file_name" 属性に記述のファイルのパスは、実行ソルバの SphSolver/SphSteer/SphDir[@basedir]に定義されたパスが基準となる。

"basedir"属性の記述がない場合、ファイルパスは記述されている XML ファイルからの相対パス、又は絶対パスとする。

よって、外部 XML ファイル（<sph_xml_config_file>要素）の場合は、外部 XML ファイルのディレクトリからの相対パスとなる。

(2) (※1) <SphFile>要素が<SphMultiFile>要素の子要素である場合の記述規則

属性名 要素名	値	<SphFile> 単独要素	<SphMultiFile> 要素の子要素
id	識別 ID（整数）	必須	任意
label	識別名（文字列）	必須	任意
format	ファイルフォーマット	必須	不要
type	入出力タイプ	必須	不要

SphStartIndex	始 点 イ ン デ ッ ク ス	任意	必須
---------------	--------------------	----	----

(3) (※2) ファイル名関連の属性、子要素の記述規則

"file_name", "prefix"属性、<SphFileNameFormat>要素はファイル名を定義する要素であり、ファイル入出力により以下の記述規則とします。

No	入出力 ("type"属性)	項目	記述 有無	ファイル名
1	入力("in")	"file_name"属性	○	"file_name"属性に記述されたファイル名から入力を行う。
		"prefix"属性	×	
		<SphFileNameFormat>要素	×	
2	入力("in")	"file_name"属性	×	<SphFileNameFormat>要素にて定義されたファイル名から入力を行う。
		"prefix"属性	×	
		<SphFileNameFormat>要素	○	
3	出力("out")	"file_name"属性	○	"file_name"属性に記述されたファイル名から出力を行う。
		"prefix"属性	×	
		<SphFileNameFormat>要素	×	
4	出力("out")	"file_name"属性	×	"prefix"属性の接頭文字からファイル名を自動生成して出力を行う。
		"prefix"属性	○	
		<SphFileNameFormat>要素	×	
5	出力("out")	"file_name"属性	×	<SphFileNameFormat>要素にて定義されたファイル名にて出力を行う。
		"prefix"属性	×	
		<SphFileNameFormat>要素	○	

入力ファイルに<SphFileNameFormat>要素を許可するか？
拡張レコードが必要。

複数の項目が記述された場合の優先順位は以下となります。

(低) "file_name"属性 < "prefix"属性 < "SphFileNameFormat"要素 (高)

(4) "prefix"属性による出力ファイル名の自動生成 (ファイル名命名規約)

"prefix"属性のファイル接頭文字による出力ファイル名は以下の書式になります。

単独実行又は入出力モード (mode 属性) ="cent"の場合

["prefix"属性]_[ステップ数(8桁)].["format"属性]

(例)

出力ファイル要素=<SphFile format="spx" prefix="prs" type="out">

実行ループステップ= 1 0 0

出力ファイル名="prs_00000100.spx"

並列実行の場合

["prefix"属性]_[ステップ数(8桁)]_id[ランク番号(6桁)].["format"属性]
(例)

出力ファイル要素=<SphFile format="spx" prefix="prs" type="out">

実行ループステップ= 1 0 0

グローバルランク番号= 1 0

出力ファイル名="prs_00000100_id000010.spx"

(5) 入出力モード

入出力モードはテキストファイル出力時のみデフォルトのモードが"cent"となり、異なりますので注意してください。

No	入力／出力	フォーマット	入出力モードのデフォルト値
1	入力	"sbx" : SBX ファイル "spx" : SPX ファイル "sph" : SPH ファイル	"multi" すべてのノードが入力を行う。
2	出力	"sbx" : SBX ファイル "spx" : SPX ファイル "sph" : SPH ファイル	"multi" すべてのノードが出力を行う。
3	出力	"text" : テキストファイル	"cent" 指定ノードのみ出力を行う。

2. 3 0 SphFileNameFormat 要素 (入出力ファイル名の書式情報)

記述ルール

要素名	SphFileNameFormat		
用 途	入出力ファイル名をランク番号とソルバ実行ステップ数から生成する書式を設定する。		
属 性			
属性名	値		種別
value	出力値種別		必須

	<p>作成するファイルに出力フォーマットに従い付加する値、</p> <p>"rank" ランク番号</p> <p>"step" ステップ数</p> <p>"step_rank" ステップ数+ランク番号</p> <p>"rank_step" ランク番号+ステップ数</p>	
format	入出力ファイル名書式	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

```
< SphFileNameFormat format="prs_id%04d_%08d.spx" value="step_rank"/>
```

(入出力ファイル名書式)

C 言語"printf"文の書式に準じますが、以下の書式にて記述を行ってください。

書式文字列: %0[出力桁数]d [出力桁数]: 0 埋め出力する桁数

(例) foramt="prs %04d %08d.spx" value="step_rank"

書式=実行ステップを4桁、ランク番号を8桁で出力する。

(出力例)

実行ステップ=10, ランク番号=3 の場合

ファイル名=prs_0010_00000003.spx

format 属性（入出力ファイル名書式）に記述する書式文字列（"%0[出力桁数]d"）は、value 属性（出力値種別）に定義された値が設定されます。よって、format 属性には value 属性により設定される値と一致した数、順序の書式文字列を記述してください。

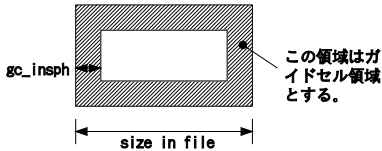
<SphFileNameFormat>要素にてファイル名を記述する場合は、ファイル名接頭文字、拡張子は自動的に付加されませんので、format 属性（入出力ファイル名書式）に出力を行う接頭文字、拡張子も記述してください。

2. 3 1 SphFileOption 要素 (ファイル入出力オプション情報)

記述ルール

要素名	SphFileOption
-----	---------------

用 途	ファイルの入出力のオプション情報を設定する。	
属 性		
属性名	値	種別
block_size	圧縮ブロックサイズ デフォルト=I 方向サイズ x J 方向サイズをブロックサイズとする。 block_size="0"の場合、圧縮を行わない。	任意 (出力)
extrec_flag	拡張レコードの出力フラグ デフォルト="false" "true": 拡張レコードを出力する。 "false": 拡張レコードを出力しない。	任意 (出力)
rlen	実数データタイプ デフォルト="4" (float) "4": float (4バイト) "8": double (8バイト) SBX ファイルの書き込み時のみ有効	任意 (出力)
sbx_validnum	SBX ファイルデータの占有ビット数 占有ビット数: 上位データのビット数 AUX レコードの以下のビット値として出力する。 AUX: 下位 4 ビット 0000XXXX デフォルト=0	任意 (出力)
sbx_validflag	SBX ファイルデータのデータ存在情報 "0":上位、下位ビット共に有効データ "1":下位のみ有効データ "2":上位のみ有効データ AUX レコードの以下のビット値として出力する。 AUX: 上位 4 ビット YYYY0000 デフォルト=0	任意 (出力)
sbx_bitshandling	SBX ファイル読み込み後のビット操作を行う。 "maskupper":上位データをマスクし、下位ビットのみのデータとする。 "shiftlow":下位ビットをシフトし、上位ビットのみのデータとする。	任意 (入力)

guide_out	ファイル出力時にガイドセルを出力するかどうかのフラグ "with" : 出力するデータオブジェクトのガイドセル値に従いガイドセルを出力する。 "without" : ガイドセルを出力しない。 デフォルト="without"	任意 (出力)
interval	ファイル出力ステップ間隔 デフォルト= 1	任意 (出力)
gc_insph	SPH ファイルフォーマット時のファイルに内包するガイドセル値 デフォルト= 0 	任意 (入力)
comment	コメント	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし		

記述例

<pre>< SphFileOption block_size="1024" extrec_flag ="true" rlen="8"/></pre>
--

（有効ファイル属性）

ファイル入出力オプション情報は、ファイルの入出力、ファイルフォーマットによって有効となる属性が異なります。

属性名	説明	有効 入出力	有効 Format
block_size	圧縮ブロックサイズ	出力	SBX SPX
extrec_flag	拡張レコードの出力フラグ	出力	SBX SPX

属性名	説明	有効 入出力	有効 Format
rlen	実数データタイプ	出力	SBX
sbx_validnum	SBX ファイルデータの占有ビット数	出力	SBX
sbx_validflag	SBX ファイルデータのデータ存在情報	出力	SBX
sbx_bitshandling	SBX ファイル読み込み後のビット操作を行う。	入力	SBX
guide_out	ガイドセル出力フラグ	出力	任意
interval	ファイル出力ステップ間隔	出力	任意

2. 3 2 SphMultiFile 要素（入力ファイル複数指定）

記述ルール

要素名	SphMultiFile	
用 途	複数の入力ファイルの指定に使用する	
属 性		
属性名	値	種別
id	識別 ID（整数）	必須
label	識別名（文字列）	必須
format	ファイルフォーマット	必須
mode	入出力モード multi：各ノードが読込・書込を行う。 （デフォルト） cent：指定ノードが読込。書込を行う。	任意
comment	コメント	任意
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
SphFile	入力ファイル指定	必須

記述例


```

<SphMultiFile>
  <SphFile id="1" format="sbx" name="mat_01.spx" commant="コメント"/>
  <SphFile id="2" format="sbx" name="mat_02.spx" commant="コメント"/>
  <SphFile id="3" format="sbx" name="mat_03.spx" commant="コメント"/>
  <SphFile id="4" format="sbx" name="mat_04.spx" commant="コメント"/>
</SphMultiFile>

```

2. 3.3 sph_xml_config_file 要素（外部 XML ファイル要素）

外部 XML ファイルに記述されているコンフィグレーション定義をインポートする。

記述ルール

要素名	sph_xml_config_file	
用 途	外部 XML ファイルをインポートする。	
属 性		
属性名	値	種別
id	識別 ID（整数）	任意
label	識別名（文字列）	任意
file_name	外部 XML ファイル名	必須
内包可能な要素（子要素として取り得る要素）		
要素名	意味	種別
なし	なし	なし

記述例

```
<sph_xml_config_file file_name="sub/a.xml"/>
```

（1）file_name 属性は相対パス、絶対パスの記述ができます。

相対パスの場合は記述されているファイルのパスに対する相対パスとなります。

（例）"b.xml" ファイルパスは"xml/sub/sub2/b.xml"となる。

```
/* ***/ xml/config.xml ***/
```

```
<SphereConfig>
```

```
  <sph_xml_config_file file_name="sub/a.xml"/>
```

```
</ SphereConfig>
```

```
/* ***/ sub/a.xml ファイル ***/
```

```
<file_config_a>  
  <sph_xml_config_file file_name="./sub2/b.xml"/>  <!--  xml/sub/sub2/b.xml  
</file_config_a>
```

3. 未定義要素

Vsphere に存在は存在するが、Sphere では定義していない要素。

- (1) "Monitor" 要素 (モニタ出力)
- (2) "History" 要素 (ヒストリーファイルの指定)
- (3) サブ要素 "Parameter", "InnerBoundary", "OuterBoundary", "Material_Table"