

Sphere

Skeleton for PHysical and Engineering REsearch
Ver 2.0.0

カプラクラスマニュアル

2010 年 8 月 01 日

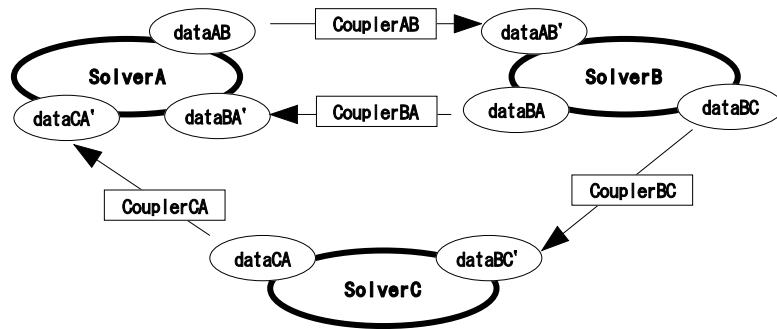
目次

| | | |
|---------|----------------------|----|
| 1. | 概要 | 3 |
| 1. 1 | クラス構成 | 3 |
| 1. 2 | カプラフロー | 4 |
| 1. 3 | カプラの実行タイミング | 7 |
| 2. | コンフィグレーション | 7 |
| 2. 1 | カプラ要素 | 7 |
| 2. 2 | データコピー要素 | 8 |
| 2. 3 | 参照データ要素 | 9 |
| 2. 4 | 更新データ要素 | 9 |
| 2. 5 | 記述例 | 10 |
| 3. | ソルバのカプラ実装 | 10 |
| 3. 1 | カプラの生成クラス・メソッド | 10 |
| 3. 2 | カプラのテンプレート | 11 |
| 3. 2. 1 | コンストラクタ | 13 |
| 3. 2. 2 | デストラクタ | 13 |
| 3. 2. 3 | カプラ初期化 | 13 |
| 3. 2. 4 | カプラ実行 | 14 |
| 3. 2. 5 | カプラ破棄 | 14 |
| 4. | カプラメソッド | 15 |
| 4. 1 | カプラ初期化 | 15 |
| 4. 2 | カプラ実行 | 16 |
| 4. 3 | カプラ破棄 | 17 |
| 4. 4 | 参照データオブジェクトの取得 | 17 |
| 4. 5 | 更新データオブジェクトの取得 | 18 |
| 4. 6 | 参照データのガイドセルの通信 | 18 |
| 4. 7 | 参照データの収集 | 18 |
| 4. 8 | 更新データ補正コピー | 19 |

1. 概要

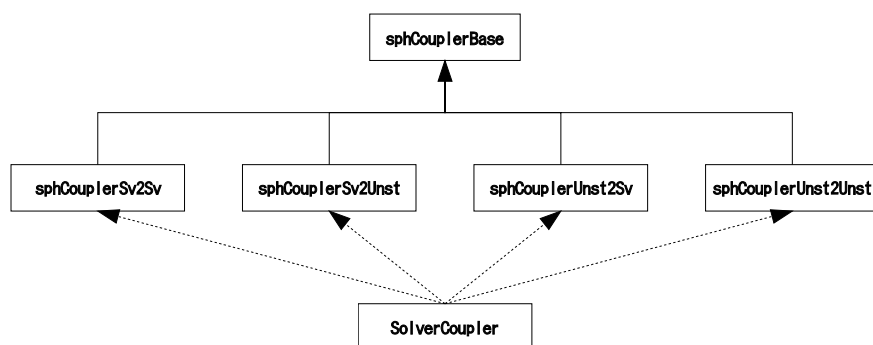
カプラは2つのソルバー間のデータのコピーを行い、連成ソルバーの補助を行います。カプラはソルバーにて組み込みを行います。SPHEE フレームワークでは構造格子の基本的な線形補間のデータコピーメソッドを用意しています。

ラプラス補間等の別の補間方法、及びソルバー独自の連成を行う場合はソルバー側のカプラに実装をすることにより自由にデータ連成を行うことが可能です。



ソルバ側で実装するカプラクラスは、1つのカップリング方式に対して1つのカプラクラスを実装する必要があります。補間方法、連成方法が同一であれば複数のデータを1つのカプラクラスで行うことができます。

1. 1 クラス構成

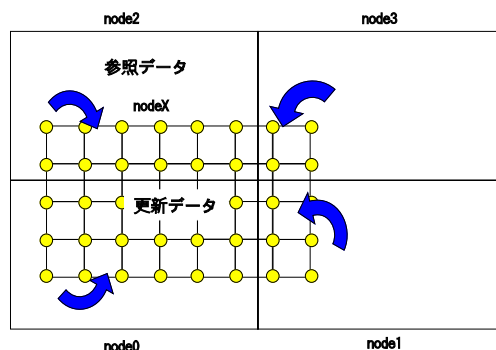


| クラス名 | 説明 |
|-------------------|--------------------------------|
| sphCouplerBase | カプラクラスの基底クラスです。 |
| sphCouplerSv2Sv | 構造格子データから構造格子データへのデータ連成を行います。 |
| sphCouplerSv2Unst | 構造格子データから非構造格子データへのデータ連成を行います。 |

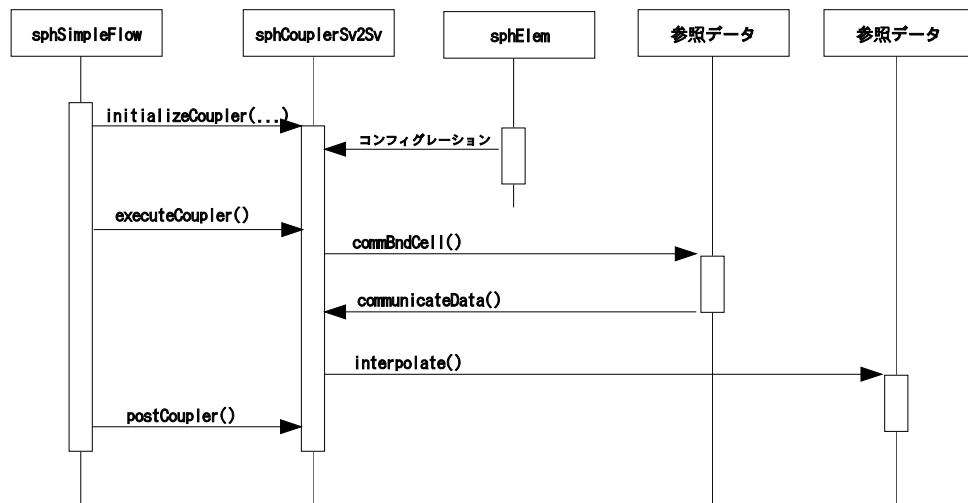
| | |
|---------------------|--|
| sphCouplerUnst2Sv | 非構造格子データから構造格子データへのデータ連成を行います。 |
| sphCouplerUnst2Unst | 非構造格子データから非構造格子データへのデータ連成を行います。 |
| SolverCoupler | <p>ソルバ側で実装するカプラクラスです。</p> <p>sphCouplerSv2Sv, sphCouplerSv2Unst, sphCouplerUnst2Sv, sphCouplerUnst2Unst のいずれかのカプラクラスを基底クラスの持ちます。</p> |

1. 2 カプラフロー

カプラは更新データ領域と重複する参照データを全ノードから収集して更新データの値を更新します。



カプラのデータ収集、データ更新のフローは以下となります。



(1) initializeCoupler(...)

フロー制御(sphFlowBase)から呼び出されます。

カプラの初期化を行います。

コンフィグレーションから参照データ、更新データの情報を取得します。

参照データを持つノードから通信を行うテーブルを作成します。

(2) execteCouler()

フロー制御(sphFlowBase)から呼び出されます。

カプラの実行を行います。

以下の3つのメソッドを呼び出します。

- a) commBndCell()
- b) communicateData()
- c) interpolate()

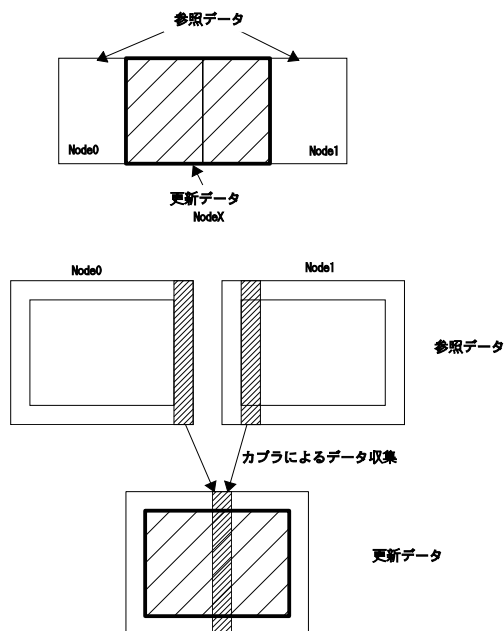
ソルバによって commBndCell()が必要ない場合、データ補間方法を変更する場合は、ソルバ側でインスタンスしたカプラクラスの execteCouler()に独自のメソッドを実装してください。

(3) commBndCell()

参照データの comm()を呼び出し、ガイドセル領域の通信を行います。

カプラによる連成を行う前にガイドセル領域の通信を行い、ガイドセル領域をノード間で同期を行わないとガイドセル領域のデータが不定となります。

(ガイドセル領域の通信の必要性)



複数のノードが重なっている参照データ領域（ガイドセル部分）からデータを更新する場合、複数のノードからデータ収集を行います。

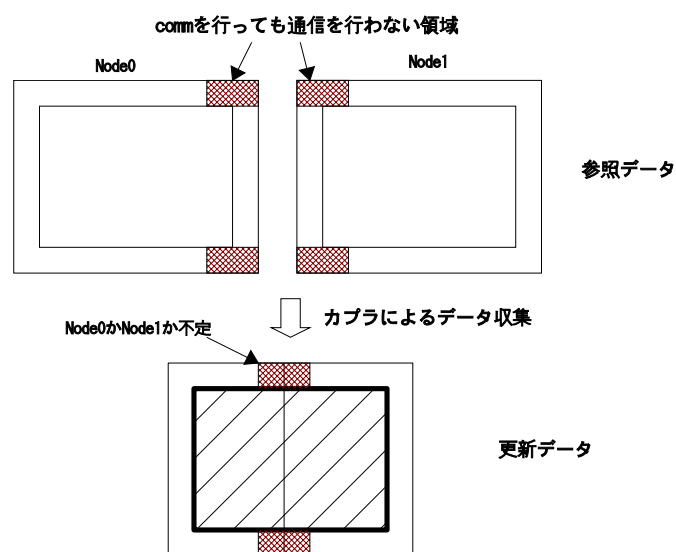
その時、ガイドセル領域（Node0 のガイドセル領域）と計算領域（Node1 の計算領域）の両方からデータを収集します。

連成を行う前にガイドセル領域の通信(commBndCell)を行わないと、更新されるデータはガイドセル領域 (Node0 のガイドセル領域) のデータによって更新される可能性があります。

よって、カプラによる連成を行う場合、事前にガイドセル領域の通信(commBndCell)を行う必要があります。

また、ガイドセル領域の通信(commBndCell)を行ってもマルチボックス分割においては最外郭の斜め方向のガイドセル領域は不定となります。

(マルチボックス分割によるデータ不定)



マルチボックス分割においては、最外郭の斜め方向の通信は行いません。

よって、ガイドセル領域の通信(commBndCell)を行っても複数のノードが重なっている領域 (通信を行っていない領域) はどのノードのデータが更新されるか不定となります。

(4) communicateData()

参照データを収集して受信バッファに格納します。

(5) interpolate()

受信バッファから更新データにデータをコピーします。

データコピーは、参照データと更新データのデータ定義位置、座標から線形補間を行い更新データにコピーします。

線形補間ではない他の補間方法、計算を行う場合はソルバのカプラクラスにて実装してください。

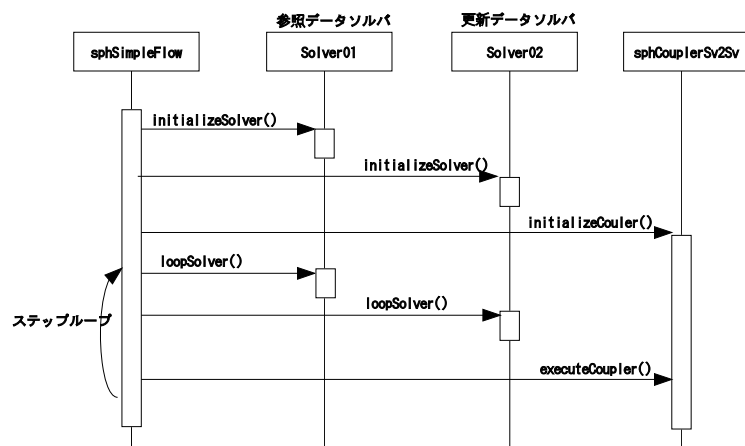
(6) postCoupler()

カプラクラスの破棄処理を行います。

1. 3 カプラの実行タイミング

カプラの初期化(initializeCoupler)、実行(executeCouper)はフロー制御クラス(sphFlowBase)によって呼び出されるタイミングが決定されます。

デフォルトのフロー制御クラスは sphSimpleFlow です。



(1) カプラの初期化(initializeCoupler)

カプラの初期化はすべてのソルバの初期化(initializeSolver)後、呼び出されます。

よって、連成を行う参照データ、更新データはソルバの初期化にて生成、データマネージャへの登録されている必要があります。

(2) 実行(executeCouper)

カプラの実行はすべてのソルバのステップ実行(loopSolver)後に呼び出されます。

よって、最初のステップ実行においては更新データに参照データはコピーされません。

2. コンフィグレーション

SPHERE ではコンフィグレーションにカプラ要素を定義することにより、データ連成をフレームワーク側で自動的に実行します。

以下、コンフィグレーションのカプラ要素について説明します。

詳細については「コンフィグレーション文法マニュアル」を参照してください。

2. 1 カプラ要素

| | | |
|-----------------------|---|----|
| 要素名 | SphCoupler | |
| 用 途 | カブラを指定する | |
| 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 | インスタンスカブラクラス名 | 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 | カブラでデータコピーを行うデータオブジェクトを指定する。 | 必須 |

2. 2 データコピー要素

| | | |
|-----------------------|--|----|
| 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 | 必須 |

| | | |
|-----------|---------------------------------|----|
| | ジェクトを指定する | |
| 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指定する | 必須 |

2. 3 参照データ要素

| | | |
|-----------------------|---------------------------------|----|
| 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 | 参照用データの識別ラベルを指定する。 | 必須 |

2. 4 更新データ要素

| | | |
|-----------------------|---------------------------------|----|
| 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 | 更新用データの識別ラベルを指定する。 | 必須 |

参照、更新データの識別ラベルは、コンフィグレーションのデータリスト要素 (SphDataList) に定義されたデータ識別ラベルである必要はありません。

ソルバ側でアロケートするデータオブジェクトの登録ラベルでも可能です。

但し、ソルバの初期化処理にて必ず同一の登録ラベルにてアロケートを行い、データマネージャに登録済みでなければなりません。

2. 5 記述例

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

3. ソルバのカプラ実装

3. 1 カプラの生成クラス・メソッド

カプラはソルバ側に実装された `sphCouplerFactory` クラスによって生成されます。

| | | |
|--|------------|----------|
| sphCouplerBase* | | |
| sphCouplerFactory::factoryCoupler(const char* class_name, const char* label, int id) const | | |
| カプラの生成を行う | | |
| 引数 | class_name | カプラクラス名 |
| | label | カプラ識別ラベル |
| | id | カプラ識別 ID |
| 戻り値 | 生成カプラ | |

カプラクラス名、カプラ識別ラベル、カプラ識別 ID はコンフィグレーションに定義されたインスタンスカプラクラス名、識別名（文字列）、識別 ID（整数）です。

カプラクラス名によってソルバ側に実装しているカプラクラスを判断し、カプラクラスの生成を行います。

（使用例）

```

sphCouplerBase*
sphCouplerFactory::factoryCoupler(const char* class_name, const char* label, int id) const
{
    sphCouplerBase* couplerObj = NULL;
    if (class_name != NULL) {
        if (strcasemp(class_name, "COUPLER01") == 0) couplerObj = new sphCoupler01(label, id);
        if (strcasemp(class_name, "COUPLER02") == 0) couplerObj = new sphCoupler02(label, id);
    }
    return couplerObj;
}

```

3. 2 カプラのテンプレート

カプラは以下のテンプレートをソルバ側に実装する必要があります。

(1) カプラヘッダーファイル (ファイル名 : sphCoupler01.h)

```

#ifndef _SPHCOUPLER01_H_
#define _SPHCOUPLER01_H_

#include "solverbase/sphCouplerSv2Sv.h"

class sphCouplerFsi: public sphCouplerSv2Sv
{
public:
    sphCoupler01();
    sphCoupler01(const char* label);
    sphCoupler01(const char* label, int id);
    virtual ~sphCoupler01();

    virtual int initializeCoupler(const char* ref_reg_label = NULL, const char* upd_reg_label = NULL);
    virtual int executeCoupler();
    virtual int postCoupler();
};
#endif /* _SPHCOUPLER01_H_ */

```

(2) カプラソースファイル (ファイル名 : sphCoupler01.C)

```
#include "sphCoupler01.h"

sphCoupler01::sphCoupler01()
{
}

sphCoupler01::sphCoupler01(const char* label) : sphCouplerSv2Sv(label, 0)
{
}

sphCoupler01::sphCoupler01(const char* label, int id) : sphCouplerSv2Sv(label, id)
{
}

sphCoupler01::~sphCoupler01()
{
}

int
sphCoupler01::initializeCoupler(const char* ref_reg_label, const char* upd_reg_label)
{
    if (sphCouplerSv2Sv::initializeCoupler(ref_reg_label, upd_reg_label) != SPHERE_SUCCESS)
        return SPHERE_TERMINATE;
    return SPHERE_SUCCESS;
}

int
sphCoupler01::executeCoupler()
{
    if (sphCouplerSv2Sv::executeCoupler() != SPHERE_SUCCESS) ¥
        return SPHERE_TERMINATE;
    return SPHERE_SUCCESS;
}

int
```

```
sphCoupler01::postCoupler()
{
    return SPHERE_SUCCESS;
}
```

以下、テンプレートに実装されたメソッドについて説明します。

3. 2. 1 コンストラクタ

| | | |
|---|-------|----------|
| sphCoupler01(); sphCoupler01(const char* label); sphCoupler01(const char* label, int id); | | |
| カプラクラスの初期化を行う | | |
| 引数 | label | カプラ識別ラベル |
| | id | カプラ識別 ID |
| 戻り値 | なし | |

カプラクラスの初期化を行います。
コンストラクタにて必ず基底カプラクラスのコンストラクタを呼び出してください。

3. 2. 2 デストラクタ

| | | |
|--------------------------|----|--|
| virtual ~sphCoupler01(); | | |
| カプラクラスの破棄処理を行う | | |
| 引数 | なし | |
| 戻り値 | なし | |

カプラクラスの破棄処理を行います。

3. 2. 3 カプラ初期化

| | | |
|--|--|--|
| int initializeCoupler(const char* ref_reg_label = NULL, const char* upd_reg_label = NULL); | | |
| カプラの初期化を行う | | |

| | | |
|-----|--|------------|
| 引数 | ref_reg_label | 参照データ登録ラベル |
| | upd_reg_label | 更新データ登録ラベル |
| 戻り値 | -1:error, 0:forced terminated, 1:normality | |

カプラの初期化を行います。

基底カプラクラスの補間、データコピーを実行する場合は、基底カプラクラスの

"initializeCoupler"を呼び出してください。

ソルバ独自のカプラの実行を行う場合は、基底カプラクラスの"initializeCoupler"の呼び出しは削除してください。

詳細は後述のカプラメソッドを参照してください。

3. 2. 4 カプラ実行

| | | |
|----------------------|--|--|
| int executeCoupler() | | |
| カプラの実行を行う | | |
| 引数 | なし | |
| 戻り値 | -1:error, 0:forced terminated, 1:normality | |

カプラの実行を行います。

基底カプラクラスの補間、データコピーを実行する場合は、基底カプラクラスの

"executeCoupler"を呼び出してください。

ソルバ独自のカプラの実行を行う場合は、基底カプラクラスの"executeCoupler"の呼び出しは削除してください。

詳細は後述のカプラメソッドを参照してください。

3. 2. 5 カプラ破棄

| | | |
|-------------------|--|--|
| int postCoupler() | | |
| カプラの破棄を行う | | |
| 引数 | なし | |
| 戻り値 | -1:error, 0:forced terminated, 1:normality | |

カプラの破棄処理を行います。
詳細は後述のカプラメソッドを参照してください。

4. カプラメソッド

カプラをソルバにて実装するに為に以下のメソッドを用意しています。

| No | 実装クラス | メソッド名 | 説明 |
|----|-----------------|--|--------------------------------|
| 1 | ソルバカプラ | int initializeCoupler(const char* ref_reg_label= NULL,, const char* upd_reg_label = NULL); | カプラの初期化を行います。 |
| 2 | ソルバカプラ | int executeCoupler() | カプラの実行を行います。 |
| 3 | ソルバカプラ | int postCoupler() | カプラの破棄を行います。 |
| 4 | sphCouplerBase | sphData* getReferData(); | 参照データオブジェクトを取得します。 |
| 5 | sphCouplerBase | sphData* getUpdateData(); | 更新データオブジェクトを取得します。 |
| 6 | sphCouplerSv | bool commBndCell(); | 参照データのガイドセル領域の通信を行います。 |
| 7 | sphCouplerSv2Sv | bool communicateData(); | 参照データを受信バッファに収集します。 |
| 8 | sphCouplerSv2Sv | bool interpolate(); | 受信バッファから更新データにデータ補間を行い、コピーします。 |

4. 1 カプラ初期化

| | | |
|--|--|------------|
| int initializeCoupler(const char* ref_reg_label = NULL, const char* upd_reg_label = NULL); | | |
| カプラの初期化を行う | | |
| 引数 | ref_reg_label | 参照データ登録ラベル |
| | upd_reg_label | 更新データ登録ラベル |
| 戻り値 | -1:error, 0:forced terminated, 1:normality | |

カプラの初期化を行います。

参照データ登録ラベル、更新データ登録ラベルは SPHERE フレームワークから呼び出された場合は、コンフィグレーションに定義されたデータ識別ラベルが渡されます。

ソルバ側から呼び出す場合は、データマネージャに登録済みの登録ラベルを指定する必要があります。

識別ラベル、登録ラベルによって指定されたデータオブジェクトはソルバの初期化処理にて生成、データマネージャへの登録済みである必要があります。

"sphCouplerSv2Sv::initializeCoupler(...)"では、更新データのデータ定義位置、領域座標から重なりを持つ参照データのノード番号、送受信領域、通信タイプのテーブルを作成します。

送受信テーブルにしたがってカプラの実行（executeCoupler）にてデータの収集、データ補間、コピーを行います。

（使用例）

```
int
sphCouplerSolv01::initializeCoupler(const char* ref_reg_label, const char* upd_reg_label)
{
    if (sphCouplerSv2Sv::initializeCoupler(ref_reg_label, upd_reg_label) != SPHERE_SUCCESS)
        return SPHERE_TERMINATE;

    return SPHERE_SUCCESS;
}
```

ソルバ側にてカプラの初期化処理を実装する場合は、
"sphCouplerSv2Sv::initializeCoupler(...)"の呼び出しを削除して、初期化処理を記述してください。

4. 2 カプラ実行

| | |
|----------------------|--|
| int executeCoupler() | |
| カプラの実行を行う | |
| 引数 | なし |
| 戻り値 | -1:error, 0:forced terminated, 1:normality |

カプラの実行を行います。

参照データを収集して、更新データにコピーします。
"sphCouplerSv2Sv::executeCoupler0"では、線形補間を行い更新データにコピーを行います。

(使用例)

```
int
sphCouplerSolv01::executeCoupler0
{
    if (sphCouplerSv2Sv::executeCoupler0 != SPHERE_SUCCESS)
        return SPHERE_TERMINATE;
    return SPHERE_SUCCESS;
}
```

ソルバ側にてカブラの実行処理を実装する場合は、"sphCouplerSv2Sv::executeCoupler0"の呼び出しを削除して、カブラ実行処理を記述してください。

4. 3 カブラ破棄

| | |
|------------------|--|
| int postCoupler0 | |
| カブラの破棄を行う | |
| 引数 | なし |
| 戻り値 | -1:error, 0:forced terminated, 1:normality |

カブラの破棄処理を行います。

(使用例)

```
int
sphCouplerSolv01::postCoupler0
{
    return SPHERE_SUCCESS;
}
```

4. 4 参照データオブジェクトの取得

| | |
|--------------------------|-------------|
| sphData* getReferData(); | |
| 参照データオブジェクトの取得を行う | |
| 引数 | なし |
| 戻り値 | 参照データオブジェクト |

参照データオブジェクトを取得します。データ収集を行った参照データではなく各ノードに登録されている参照データオブジェクトです。

実行ノードに参照データのソルバが実行されていない場合は、NULL が返ります。

カプラの実行をソルバ側で実装する時に使用します。

4. 5 更新データオブジェクトの取得

| | |
|----------------------------|-------------|
| sphData* getUpdateData (); | |
| 更新データオブジェクトの取得を行う | |
| 引数 | なし |
| 戻り値 | 更新データオブジェクト |

更新データオブジェクトを取得します。

実行ノードに更新データのソルバが実行されていない場合は、NULL が返ります。

カプラの実行をソルバ側で実装する時に使用します。

4. 6 参照データのガイドセルの通信

| | |
|------------------------|----|
| bool commBndCell(); | |
| 参照データのガイドセル領域の通信を行います。 | |
| 引数 | なし |
| 戻り値 | 成否 |

参照データのガイドセル領域の通信を行います。

4. 7 参照データの収集

| |
|-------------------------|
| bool communicateData(); |
|-------------------------|

| | |
|----------------|----|
| 参照データの収集を行います。 | |
| 引数 | なし |
| 戻り値 | 成否 |

自ノードの更新データの領域と重なっている座標の参照データを持つすべてのノードから収集を行います。

"initializeCoupler"にて作成した送受信テーブルに従って参照データを持つノードからデータを収集します。

収集データは、受信バッファにコピーされます。

4. 8 更新データ補正コピー

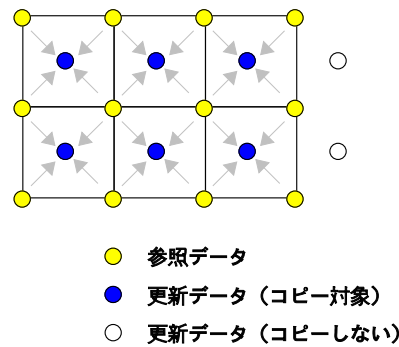
| | |
|--|----|
| bool interpolate(); | |
| 収集を行った受信バッファからデータ補間を行い、更新データにコピーを行います。 | |
| 引数 | なし |
| 戻り値 | 成否 |

収集を行った参照データは受信バッファにコピーされます。

受信データからデータ補間を行い、更新データにコピーします。

"sphCouplerSv2Sv::interpolate()"では参照データのデータ定義位置の座標と参照データの定義位置座標から線形補間を行い、データを更新データにコピーします。

線形補間を行う為には参照データに内包する更新データである必要があります。参照データに内包していない更新データはデータを更新しません。



(更新対象とならない更新データ)

- a) 参照データの領域外 (内包していない)
- b) データ定義位置 Stagerd の最外郭データ
u-J,K 方向のプラス方向最外郭

$v\text{-}I, K$ 方向のプラス方向最外郭
 $w\text{-}I, J$ 方向のプラス方向最外郭