EEM-STF データ作成ライブラリ

取扱説明書(Java 版)

株式会社 EEM　2011 年 12 月

1. 概要

本ライブラリーは EEM-STF の入力データを作成するものです。
対応言語は Java です。
必要な関数を呼ぶことによって EEM-STF データが出力されます。
対応する EEM-STF のバージョンは 2.0 以降です。

2. 使用法

下記の関数仕様に従ってソースコードを作成し、コンパイルして実行します。
ソースコードのファイル名を sample_stf.java とすると、コマンドラインで以下の操作を行います。

```
> javac sample_stf.java StfData.java        コンパイル
> java sample_stf                           実行、EEM-STF データ出力
```

出力された EEM-STF データは EEM-STF で開くことができます。EEM-STF で小さい修正を行うことも可能です。
なお、上記は StfData.java ファイルが同じフォルダにあることを仮定しています。違うフォルダにあるときは必要な変更を行って下さい。

3. 関数仕様

本ライブラリ(StfData)の各関数の仕様は以下の通りです。
実数の引数はすべて倍精度(double)で単位は MKSA です。
最初に(1)でインスタンスを作成し、最後に(9)でデータを保存します。その間の関数の呼び出し順は任意ですが、下記の順を推奨します。
同じ関数名で引数の異なるものがありますので注意して下さい。
なお、形状データは後優先です。すなわち、座標が重複するところでは後に指定されたデータの属性が使用されます。

(1)コンストラクタ

```
public StfData()                          インスタンスを作成します。
```

※最初に一度呼び出すことが必要です。

(2)タイトル

```
public void setTitle(String title)        タイトル(オプション)
```

(3)メッシュ

メッシュは X,Y,Z 方向で独立です。
区間区切り座標は小さい順に入力して下さい。

区間区切り座標は大きさが区間数+1 個の配列、区間分割数は大きさが区間数個の配列です。

```
public void setXMesh(int region,     X 方向の区間数
                     double mesh[], X 方向の区間区切り座標(区間数+1 個の配列)
                     int div[])     X 方向の区間分割数(区間数個の配列)

public void setYMesh(int region,     Y 方向の区間数
                     double mesh[], Y 方向の区間区切り座標(区間数+1 個の配列)
                     int div[])     Y 方向の区間分割数(区間数個の配列)

public void setZMesh(int region,     Z 方向の区間数
                     double mesh[], Z 方向の区間区切り座標(区間数+1 個の配列)
                     int div[])     Z 方向の区間分割数(区間数個の配列)
```

(4)誘電率

複数使用可能です。誘電率番号は入力した順に 1,2,...になります。

```
public void addMaterial(double epsr)         比誘電率
```

(5)電圧

複数使用可能です。電圧番号は入力した順に 1,2,...になります。

```
public void addVolt(double volt)             電圧[V]
```

(6)形状

複数使用可能です。誘電率番号は(4)、電圧番号は(5)で指定されたものを使用します。

```
public void addUnit(int type,        種別、1:誘電体、2:電極
                    int id,          誘電体のときは誘電率番号、電極のときは電圧番号
                    int shape,       形状の種類 1:直方体、2:球、その他は EEM-STF 取説付録 B 参考
                    double pos[])    座標データ[m](6 個、四角柱のときは 10 個)
```

※座標を配列に格納したとき使用します

```
public void addUnit(int type,        種別、1:誘電体、2:電極
                    int id,          誘電体のときは誘電率番号、電極のときは電圧番号
                    int shape,       形状の種類 1:直方体、2:球、その他は EEM-STF 取説付録 B 参考
                    double x1, double y1, double z1,     XYZ 座標の下限[m]
                    double x2, double y2, double z2)     XYZ 座標の上限[m]
```

※座標を引数で直接指定するとき使用します

(7)点電荷

```
public void addCharge(double x, double y, double z, double q)    XYZ 座標[m]、電荷[Coulomb]
```

※複数使用可能

(8)計算条件

```
public void setSolver(double convergence,    収束判定条件
                      int maxstep,           最大反復回数
                      int nout,              表示出力間隔
```

double g)　　加速係数ｇ
※本関数を呼ばないときは既定値が使用されます

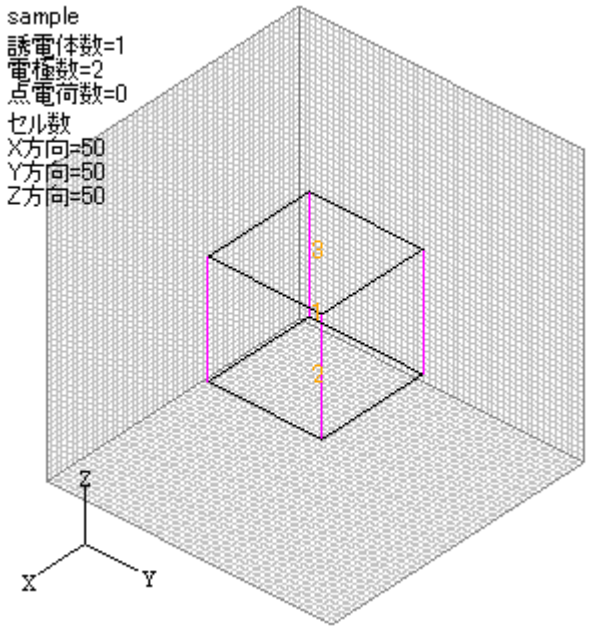
(9)ファイル出力

public void save(String filename)　　データをファイル"filename"に出力します
※最後に一度呼び出すことが必要です

4. サンプルプログラム

```
/*
    sample_stf.java

    EEM-STFデータ作成サンプルプログラム

    使い方：
    > javac sample_stf.java StfData.java
    > java sample_stf
*/

class sample_stf
{
    public static void main(String args[])
    {
        // new

        StfData d = new StfData();

        // title

        d.setTitle("sample");

        // mesh

        double x_mesh[] = new double[2];
        int x_div[] = new int[1];
        x_mesh[0] = -5;
        x_mesh[1] = +5;
        x_div[0] = 50;
        d.setXMesh(1, x_mesh, x_div);

        double y_mesh[] = new double[2];
        int y_div[] = new int[1];
        y_mesh[0] = -5;
        y_mesh[1] = +5;
        y_div[0] = 50;
        d.setYMesh(1, y_mesh, y_div);

        double z_mesh[] = new double[2];
        int z_div[] = new int[1];
        z_mesh[0] = -5;
        z_mesh[1] = +5;
```

```
        z_div[0] = 50;
        d.setZMesh(1, z_mesh, z_div);

        // material

        d.addMaterial(2.0);      // Er = 2

        // volt

        d.addVolt(-1.0);         // -1V
        d.addVolt(+1.0);         // +1V

        // geometry

        d.addUnit(1, 1, 1, -2, -2, -2, +2, +2, +2);
        d.addUnit(2, 1, 1, -2, -2, -2, +2, +2, -2);
        d.addUnit(2, 2, 1, -2, -2, +2, +2, +2, +2);

        // charge

        // solver

        // output

        d.save("sample.stf");
    }
}
```



サンプルプログラムの出力データを EEM-STF で開いた図