

EEM-FDM 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

本ライブラリは EEM-FDM 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-FDM データが出力されます。
対応する EEM-FDM のバージョンは 2.1 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_fdm.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_fdm.java FdmData.java` コンパイル
> `java sample_fdm` 実行、EEM-FDM データ出力
出力された EEM-FDM データは EEM-FDM で開くことができます。EEM-FDM で小さい修正を行うことも可能です。
なお、上記は `FdmData.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 (`FdmData`) 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、度です。
最初に (1) でインスタンスを作成し、最後に (11) 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材質が使用されます。

(1) コンストラクタ

```
public FdmData()                                           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) 計算方法

public void setDomain(int domain) 計算方法を指定します
 ※引数：0=時間領域(パルス)、1=時間領域(正弦波)、2=周波数領域
 ※既定値は時間領域(パルス)、周波数領域のときはE法で計算されます

(4) 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)

(5) 物性値

複数使用可能です。物性値番号は入力した順に2, 3, ... になります。(1:PECは不要です)

public void addMaterial(double epsr, 比誘電率
 double esgm, 導電率[S/m]
 double amur, 比透磁率
 double msgm) 導磁率[1/Sm]

public void addMaterial(String filename) 分散ファイル名
 ※分散性媒質に使用します。

public void addMaterial(double epsr0, ϵ_{∞}
 double epsr1, double epsr2, double epsr3, a_e, b_e, c_e
 double amur0, μ_{∞}
 double amur1, double amur2, double amur3) a_h, b_h, c_h

※分散性媒質に使用します。

(6) 形状

public void addUnit(int shape, 形状の種類 1:直方体、2:球、その他はEEM-FDM取説付録B参考
 int material, 物性値番号 1:PEC、2, 3, ... :誘電体
 double p[]) 座標データ[m](6個、四角柱/四角錘台/円錐台のときは10個)
 ※座標を配列に格納したとき使用します、複数使用可能

public void addUnit(int shape, 形状の種類 1:直方体、2:球、その他はEEM-FDM取説付録B参考
 int material, 物性値番号 1:PEC、2, ... :誘電体
 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、複数使用可能

```
public void addLoad(int type,          集中定数の種類 1:抵抗、2:キャパシタ、3:インダクタ
                    int direction,     向き 1:X方向、2:Y方向、3:Z方向
                    double x, double y, double z, XYZ座標[m]
                    double parameter) 集中定数値(Ω、Henry, Farad)
```

※集中定数があるとき使用します、複数使用可能

```
public void addLoad(int direction,     向き 1:X方向、2:Y方向、3:Z方向
                    double x, double y, double z, XYZ座標[m]
                    String filename)   V-I特性ファイル名
```

※集中定数があるとき使用します、複数使用可能

(7)波源

```
public void addFeed(int direction,     向き 1:X方向、2:Y方向、3:Z方向
                    double x, double y, double z, XYZ座標[m]
                    double amp)        振幅[V]
```

※時間領域(パルス)のときの給電点を指定します、複数使用可能

```
public void addFeed(int direction,     向き 1:X方向、2:Y方向、3:Z方向
                    double x, double y, double z, XYZ座標[m]
                    double amp,        振幅[V]
                    double phase)     位相[度]
```

※時間領域(パルス)以外のときの給電点を指定します、複数使用可能

```
public void setFeedR(double r)         仮想抵抗値[Ω]、波源が給電点のとき有効
```

※本関数を呼ばないときは仮想抵抗を使用しません

```
public void setPulseWidth(double w)    パルス幅/Δt、時間領域(パルス)のとき有効
```

※本関数を呼ばないときは既定値が使用されます(推奨)

```
public void setWave(double theta,     入射方向のθ[度]
                    double phi,      入射方向のφ[度]
                    int pol,         偏波方向 0:垂直偏波 1:水平偏波
                    double amp)     振幅[V/m]
```

※平面波入射のとき使用します、複数使用したときは最後が有効

(8)観測点

```
public void addPoint(int direction,     観測点の向き 1:X方向、2:Y方向、3:Z方向
                    double x, double y, double z) XYZ座標[m]
```

※複数使用可能、ポート番号は入力した順に1,2,...になります

```
public void setPropagation(int propagation) 観測点1を通る伝搬方向
```

※引数=11/12/13/14/15/16 : +X/-X/+Y/-Y/+Z/-Z方向

※観測点があるとき必要

(9)周波数

```
public void setFreq(double f_start,    開始周波数[Hz]
                    double f_end,      終了周波数[Hz]
                    int f_div)         周波数分割数
```

※近傍界・遠方界の計算に使用します

```
public void setFreq2(double f_start,      開始周波数[Hz]
                    double f_end,        終了周波数[Hz]
                    int f_div)           周波数分割数
    ※時間領域(パルス)のとき、給電点、集中定数、観測点の周波数特性に使用します
```

(10) 計算条件

```
public void setSolver(double convergence, 収束判定条件
                      int maxstep,       最大タイムステップ数、最大反復回数
                      int nout)         表示出力間隔
    ※本関数を呼ばないときは既定値が使用されます
```

```
public void setPML(void)                 PMLを使用します(PMLパラメータは既定値)
    ※本関数を呼ばないときはMur一次が使用されます(推奨)
```

```
public void setPBC(char c)              周期境界条件を指定します(c='x'/'y'/'z':X/Y/Z方向)
    ※本関数を呼ばないときはすべての方向にMur一次が指定されます
    ※複数の方向が設定できます
```

```
public void setRect1st(void)             [重複部は直方体ユニットを優先]をONにします
    ※ユニット数が非常に多いときに前処理時間が短縮されます。通常は使用しないで下さい。
```

(11) 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
```