

EEM-MOM データ作成ライブラリ

取扱説明書(C版)

株式会社 EEM 2012年3月

1. 概要

本ライブラリは EEM-MOM の入力データを作成するものです。
対応言語は C です。
必要な関数を呼ぶことによって EEM-MOM データが出力されます。
対応する EEM-MOM のバージョンは 2.2 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・リンクして実行します。
ソースコードのファイル名を `sample_mom.c` とすると、コマンドラインで以下の操作を行います。
> `cl sample_mom.c mom_datalib.c` コンパイル・リンク (Microsoft Visual C++ の場合)
> `sample_mom` 実行、EEM-MOM データ出力
出力された EEM-MOM データは EEM-MOM で開くことができます。EEM-MOM で小さい修正を行うことも可能です。
なお、上記は同じフォルダに `mom_datalib.c`, `mom_datalib.h` の二つのファイルがあることを仮定しています。
これらのファイルが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 (`mom_datalib.c`) の各関数の仕様は以下の通りです。
関数名はすべて "mom_" で始まります。
実数の引数はすべて倍精度 (double) で単位は MKSA, 度です。
XYZ 座標は、円筒座標のときは r, ϕ, Z 、極座標のときは r, θ, ϕ に読みかえて下さい。
最初に `mom_init`、最後に `mom_outdat` を呼びます。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。

(1) 初期化

`void mom_init(void);` 初期化します。
※最初に一度呼び出す必要があります。

(2) タイトル

`void mom_title(const char *title);` タイトル (オプション)

(3) 周波数

```
void mom_freq(double fstart,      開始周波数[Hz]
              double fend,       終了周波数[Hz]
              int fdiv);        周波数分割数
```

※必須データです。

※単一周波数のときは、開始周波数と終了周波数に同じ値を代入し、周波数分割数を0にします。

(4) グランド板

```
void mom_ground(void);
```

※Z=0面に完全導体のグラウンド板を置きます。本関数を呼び出さなければグラウンド板はありません

(5) 分割数

```
void mom_division(int division);  波長あたりの分割数 (波長は終了周波数に対応)
```

※分割数が指定されないすべてのユニットに適用されます。オプション。既定値は10。

(6) 導線半径

```
void mom_radius(double radius);  導線半径[m]
```

※導線半径が指定されないすべてのユニットに適用されます。オプション。既定値は1mm。

※負の値のときは[導線半径/要素長]の符号を反転したものになります。

(7) 平面波入射

```
void mom_wave(double theta,      入射方向の  $\theta$  [度]
```

```
              double phi,       入射方向の  $\phi$  [度]
```

```
              int pol);         偏波方向 0:垂直偏波 1:水平偏波 2:右旋円偏波 3:左旋円偏波
```

※本関数を呼び出さなければ平面波入射はありません。

(8) 計算条件

```
void mom_solver_cg(int maxiter,  最大反復回数
```

```
                  double tol,   収束判定条件
```

```
                  int nout);    出力間隔
```

※共役勾配法を指定するとき使用します。

※本関数を呼び出さなければ修正コレスキー法が指定されます。

(9.1) 線状ユニット

```
void mom_wire6(int coord,          座標系(0:XYZ座標, 1:円筒座標, 2:極座標)
               double x1, double y1, double z1,  頂点1のXYZ座標[m]
               double x2, double y2, double z2,  頂点2のXYZ座標[m]
               int div);          分割数
```

※分割数=0のときは(5)が適用されます。

(9.2) 線状ユニット

```
void mom_wire(int coord,          座標系(0:XYZ座標, 1:円筒座標, 2:極座標)
              double p[2][3],     頂点1,2のXYZ座標[m]
              int div);          分割数
```

※分割数=0のときは(5)が適用されます。

(10.1) 四角形ユニット

```
void mom_plane12(int coord,
                 double x1, double y1, double z1,
                 double x2, double y2, double z2,
                 double x3, double y3, double z3,
                 double x4, double y4, double z4,
                 int div12,
                 int div14);
```

座標系 (0:XYZ 座標, 1:円筒座標, 2:極座標)
頂点 1 の XYZ 座標 [m]
頂点 2 の XYZ 座標 [m]
頂点 3 の XYZ 座標 [m]
頂点 4 の XYZ 座標 [m]
頂点 1-2 方向の分割数
頂点 1-4 方向の分割数

※分割数=0 のときは (5) が適用されます。

(10.2) 四角形ユニット

```
void mom_plane(int coord,
               double p[4][3],
               int div12,
               int div14);
```

座標系 (0:XYZ 座標, 1:円筒座標, 2:極座標)
頂点 1, 2, 3, 4 の XYZ 座標 [m]
頂点 1-2 方向の分割数
頂点 1-4 方向の分割数

※分割数=0 のときは (5) が適用されます。

(11.1) 直方体ユニット

```
void mom_box6(int coord,
              double x1, double y1, double z1,
              double x2, double y2, double z2,
              int divx,
              int divy,
              int divz);
```

座標系 (0:XYZ 座標, 1:円筒座標, 2:極座標)
下端の XYZ 座標 [m]
上端の XYZ 座標 [m]
X 方向の分割数
Y 方向の分割数
Z 方向の分割数

※分割数=0 のときは (5) が適用されます。

(11.2) 直方体ユニット

```
void mom_box(int coord,
             double p[2][3],
             int divx,
             int divy,
             int divz);
```

座標系 (0:XYZ 座標, 1:円筒座標, 2:極座標)
下端と上端の XYZ 座標 [m]
X 方向の分割数
Y 方向の分割数
Z 方向の分割数

(12) 給電点

```
void mom_wire_feed(double amp,      振幅 [V]
                   double phase);   位相 [度]
```

※最後に指定された線状ユニットに適用されます。

(13) 負荷

```
void mom_wire_load(double r,      抵抗 R [Ω]
                   double l,      インダクタンス L [H]
                   double c);     キャパシタンス C [F]
```

※最後に指定された線状ユニットに適用されます。

(14) 導線半径

```
void mom_unit_radius(double radius);  導線半径 [m]
```

※最後に指定されたユニットに適用されます。指定しないユニットには(6)が適用されます。

(15) 平行移動

```
void mom_unit_origin(double x,      X 方向平行移動[m]
                    double y,      Y 方向平行移動[m]
                    double z);      Z 方向平行移動[m]
```

※最後に指定されたユニットに適用されます。

(16) ファイル出力

```
void mom_outdat(const char *file);    データをファイル"file"に出力します。
```

※最後に一度呼び出す必要があります。