

EEM-FDM データ作成ライブラリ

取扱説明書(C版)

株式会社 EEM 2015年7月

1. 概要

本ライブラリーは EEM-FDM 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-FDM データが出力されます。
対応する EEM-FDM のバージョンは 3.2 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_fdm.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_fdm.c fdm_datalib.c コンパイル・リンク (Microsoft Visual C++ の場合)
> 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、上記は同じフォルダに fdm_datalib.c, fdm_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 (fdm_datalib.c) の各関数の仕様は以下の通りです。
関数名はすべて "fdm_" で始まります。
実数の引数はすべて倍精度 (double) で単位は MKSA です。
最初に (1) の fdm_init、最後に (11) の fdm_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。
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) 初期化

void fdm_init(void); 初期化します。
※最初に一度呼び出すことが必要です。

void fdm_section_size(size_t size); 区間数の上限を設定する (既定値は 100)
void fdm_material_size(size_t size); 物性値数の上限を設定する (既定値は 100)
void fdm_unit_size(size_t size); ユニット数の上限を設定する (既定値は 10000)
void fdm_feed_size(size_t size); 給電点数の上限を設定する (既定値は 100)
void fdm_point_size(size_t size); 観測点数の上限を設定する (既定値は 100)
※以上 5 つはオプションで fdm_init の直後に呼び出すことが必要です。

(2) タイトル

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

(3) 計算方法

void fdm_domain(int domain); 計算方法 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。
 メッシュの指定方法には、一括指定と個別指定の二種類があります。一括指定を行うと、個別指定したデータは消去されます。
 区間区切り座標は小さい順に入力して下さい。
 一括指定のときは引数が可変個です。2番目以降の引数は、座標はdouble型、分割数はint型であることを明示して下さい。(暗黙の型変換を使用しないで下さい)
 個別指定のときは、区間区切り座標の数は区間分割数の数+1である必要があります。

(4.1) 一括指定

```
void fdm_xsection(int mx,          X方向の区間数+1
                  double x, ...); X方向の区間区切り座標[m] (区間数+1個必要)

void fdm_ysection(int my,          Y方向の区間数+1
                  double y, ...); Y方向の区間区切り座標[m] (区間数+1個必要)

void fdm_zsection(int mz,          Z方向の区間数+1
                  double z, ...); Z方向の区間区切り座標[m] (区間数+1個必要)

void fdm_xdivision(int nx,         X方向の区間数
                   int n, ...);   X方向の区間分割数 (区間数個必要)

void fdm_ydivision(int ny,         Y方向の区間数
                   int n, ...);   Y方向の区間分割数 (区間数個必要)

void fdm_zdivision(int nz,         Z方向の区間数
                   int n, ...);   Z方向の区間分割数 (区間数個必要)
```

(4.2) 個別指定

```
void fdm_xsection1(double x);      X方向の区間区切り座標[m]
void fdm_ysection1(double y);      Y方向の区間区切り座標[m]
void fdm_zsection1(double z);      Z方向の区間区切り座標[m]
void fdm_xdivision1(int n);        X方向の区間分割数
void fdm_ydivision1(int n);        Y方向の区間分割数
void fdm_zdivision1(int n);        Z方向の区間分割数
```

(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)

```
void fdm_material(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]

void fdm_material_name(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]
                       const char *name); 名称・コメント

void fdm_material_file(const char *filename)  分散ファイル名

void fdm_material_dispersion(double epsr0,    $\epsilon_\infty$ 
                              double epsr1, double epsr2, double epsr3,    $a_e, b_e, c_e$ 
                              double mur0,    $\mu_\infty$ 
                              double mur1, double mur2, double mur3)    $a_h, b_h, c_h$ 
```

※分散性媒質に使用します。

(6) 形状

void fdm_unit(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/四角錐台/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

void fdm_unit6(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

void fdm_load(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 p); 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

(7) 波源

void fdm_feed(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)のときはダミー)
 ※給電点(アンテナ)のとき使用します、複数使用可能

void fdm_pulsewidth(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)

void fdm_rvirtual(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

void fdm_incidence(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

void fdm_incidence_beam(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]
 double wtheta, θ 方向ビーム幅[m]
 double wphi); ϕ 方向ビーム幅[m]
 ※ビーム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

void fdm_point(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

void fdm_propagation(int propagation); 観測点伝搬方向 +1/-1/+2/-2/+3/-3:+X/-X/+Y/-Y/+Z/-Z方向
 ※観測点があるとき必要

(9) 周波数

void fdm_freq(double freq1, 開始周波数[Hz]
 double freq2, 終了周波数[Hz]
 int fdiv); 周波数分割数
 ※近傍界・遠方界の計算に使用します

void fdm_freq2(double freq1, 開始周波数[Hz]
 double freq2, 終了周波数[Hz]
 int fdiv); 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

void fdm_iteration(double convergence, 収束判定条件

int maxiter, 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

void fdm_courant(double courant); 安定性条件(負の値のときはその絶対値が dt[sec])
※本関数を呼ばないときは既定値(=1)が使用されます
※時間領域差分法のととき有効です

void fdm_pml(void); PML を使用します(PML パラメータは既定値)
void fdm_pml2(int l, double m, double r0); PML を使用します(PML パラメータ l, m, r0 指定)
※上記二つの関数を呼ばないときは Mur 一次が使用されます
※時間領域差分法のととき有効です

void fdm_pbc(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

(11) ファイル出力

void fdm_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ことが必要です