

# EEM-RTM データ作成ライブラリ

## 取扱説明書(C版)

株式会社 EEM 2014年12月

### 1. 概要

本ライブラリはEEM-RTM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-RTMデータが出力されます。  
対応するEEM-RTMのバージョンは3.1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・リンクして実行します。  
ソースコードのファイル名をsample\_rtm.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\_rtm.c rtm\_datalib.c コンパイル・リンク(Microsoft Visual C++の場合)  
> sample\_rtm 実行、EEM-RTMデータ出力  
出力されたEEM-RTMデータはEEM-RTMで開くことができます。EEM-RTM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rtm\_datalib.c, rtm\_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(rtm\_datalib.c)の各関数の仕様は以下の通りです。  
関数名はすべて"rtm\_"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。  
最初に(1)のrtm\_init、最後に(14)のrtm\_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 rtm_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 rtm_title(const char *title); タイトル(オプション)
```

#### (3)周波数

void rtm\_freq(double freq);                    周波数[Hz]

#### (4) 物性値

void rtm\_material(double epsr,                    比誘電率  
                  double esgm,                    導電率[S/m]  
                  double thick,                  厚さ[m]、0 のときは透過波が存在しません  
                  const char \*name);              コメント( “” も可)

void rtm\_material\_file(const char \*file,        反射等価係数ファイル  
                          const char \*name);      コメント( “” も可)

※複数使用可能、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)

#### (5) 多角柱

断面が多角形で底面と上面が水平な多角柱を指定します。

##### (5.1) 一般の多角柱

void rtm\_pillar(int n,                            頂点の数(2以上30以下)  
                  const double \*x,                頂点のX座標[m]  
                  const double \*y,                頂点のY座標[m]  
                  double z1,                        底面の高さ[m]  
                  double z2,                        上面の高さ[m]  
                  int m);                          物性値番号(1以上)

##### (5.2) 長方形の柱

void rtm\_pillar\_rect(double x1, double y1,      左下のX, Y座標[m]  
                      double x2, double y2;      右上のX, Y座標[m]  
                      double z1,                    底面の高さ[m]  
                      double z2,                    上面の高さ[m]  
                      int m);                      物性値番号(1以上)

##### (5.3) 鉛直の壁

void rtm\_pillar\_wall(double x1, double y1,      始点のX, Y座標[m]  
                      double x2, double y2;      終点のX, Y座標[m]  
                      double z1,                    底面の高さ[m]  
                      double z2,                    上面の高さ[m]  
                      int m);                      物性値番号(1以上)

#### (6) 地面

四角形を指定します。四頂点のX, Y, Z座標は独立です。

##### (6.1) 一般の地面

void rtm\_plane(const double x[4],                四頂点のX座標[m]  
                  const double y[4],                四頂点のY座標[m]  
                  const double z[4],                四頂点のZ座標[m]  
                  int m);                          物性値番号(1以上)

## (6.2) 長方形の水平面

```
void rtm_plane_rect(double x1, double y1,  左下の X, Y 座標 [m]
                   double x2, double y2;  右上の X, Y 座標 [m]
                   double z0,             高さ [m]
                   int m);               物性値番号 (1 以上)
```

## (7) アンテナ特性

アンテナ特性を指定します。以後変更されるまで、送信点・観測点・観測線・観測面に適用されます。初期値は無指向性、垂直偏波です。

### (7.1) 無指向性

```
void rtm_antenna_iso(int pol);          偏波 (1:垂直, 2:水平, 3:右旋円偏波, 4:左旋円偏波)
```

### (7.2) ダイポール

```
void rtm_antenna_dipole(double theta,  軸方向の  $\theta$  [度]
                        double phi,    軸方向の  $\phi$  [度]
                        double wbeam,  3dB ビーム幅 [度]
                        int pol);      偏波 (1:垂直, 2:水平, 3:右旋円偏波, 4:左旋円偏波)
```

### (7.3) ビーム

```
void rtm_antenna_beam(double theta,  ビーム中心軸の  $\theta$  [度]
                      double phi,    ビーム中心軸の  $\phi$  [度]
                      double wtheta,  $\theta$  方向の 3dB 幅 [度]
                      double wphi,    $\phi$  方向の 3dB 幅 [度]
                      int pol);      偏波 (1:垂直, 2:水平, 3:右旋円偏波, 4:左旋円偏波)
```

### (7.4) ファイル指定

```
void rtm_antenna_file(double theta,  回転中心軸の  $\theta$  [度]
                      double phi,    回転中心軸の  $\phi$  [度]
                      double rot,    回転角 [度]
                      int raw,       0:正規化する、1:正規化しない
                      const char *file);  アンテナ指向性ファイル名
```

## (8) 送信点

```
void rtm_tx(double x,    X 座標 [m]
            double y,    Y 座標 [m]
            double z,    Z 座標 [m]
            double power, 送信電力 [W]
            double phase); 送信位相 [度]
```

## (9) 観測点

```
void rtm_rx0(double x,    X 座標 [m]
             double y,    Y 座標 [m]
             double z);   Z 座標 [m]
```

#### (10) 観測線

```
void rtm_rx1(double x[2],  始点と終点のX座標[m]
             double y[2],  始点と終点のY座標[m]
             double z[2],  始点と終点のZ座標[m]
             int div);     線分の分割数
```

#### (11) 観測面

```
void rtm_rx2(double x[4],  4頂点のX座標[m]
             double y[4],  4頂点のY座標[m]
             double z[4],  4頂点のZ座標[m]
             int div12,    頂点1-2方向の分割数
             int div14);   頂点1-4方向の分割数
```

#### (12) 計算条件

```
void rtm_solver(int maxref,  最大反射回数
                int raydiv,  緯度方向分割数
                int ndiffr,  回折(0:なし、1:回折のみ、2:回折+反射まで)
                int trans,  透過波を計算するか(Y/N:1/0)
                int beam,   レイ放射方向を絞るか(Y/N:1/0)
                int pathlog, path.logを出力するか(Y/N:1/0)
                int adiffr,  回折波を近似計算するか(Y/N:1/0)
                double att,  減衰定数[dB/m]
                int maxpath); 各受信点への最大伝搬経路数
```

#### (13) その他データ

```
void rtm_misc(int size,    画面上の大きさ(ピクセル)、縦・横の大きい方
              int margin); 画面上の周囲の余白(ピクセル)
    ※既定値は size=400, margin=0
```

#### (14) ファイル出力

```
void rtm_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。
```

## 4. サンプルプログラム

```
/*
   sample_rtm.c

   make EEM-RTM data
*/

#include "rtm_datalib.h"

int main(void)
```

```

{
  double xp[4], yp[4], zp[4];

  // ==== initialize ====

  rtm_init();

  // ==== data ====

  // (2) title

  rtm_title("sample");

  // (3) frequency

  rtm_freq(2e9);

  // (4) material

  rtm_material(3.0, 0.2, 0, "");
  rtm_material(5.0, 0.1, 0.03, "");

  // (5/6) geometry

  rtm_pillar_rect(+10, +10, +50, +30, 0, 10, 2);
  rtm_pillar_rect(+10, -10, +50, -30, 0, 15, 2);
  rtm_pillar_rect(-10, +10, -50, +30, 0, 20, 2);
  rtm_pillar_rect(-10, -10, -50, -30, 0, 25, 2);

  rtm_plane_rect(-50, -30, +50, +30, 0, 3);

  // (8) TX

  rtm_antenna_dipole(0, 0, 90, 1);
  rtm_tx(12, 12, 15, 1, 0);

  // (9) Rx0

  rtm_antenna_iso(1);
  rtm_rx0(-20, 0, 1.5);

  // (10) Rx1

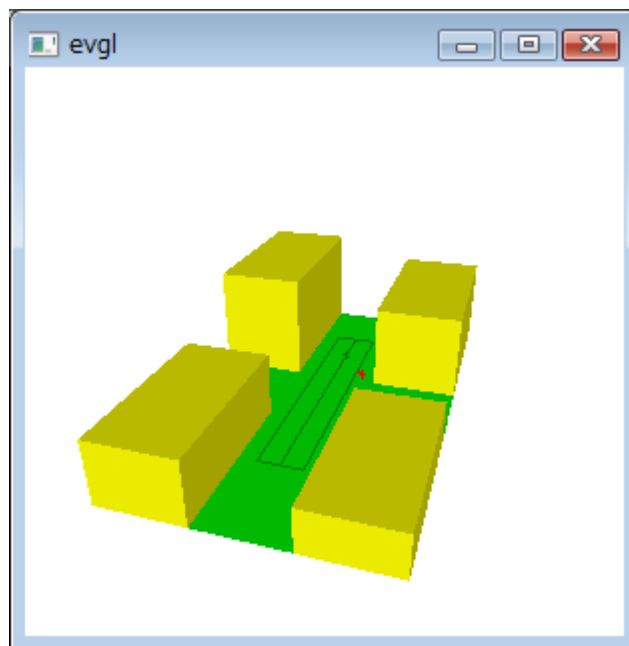
  xp[0] = -30; xp[1] = 30;
  yp[0] = 0; yp[1] = 0;
  zp[0] = 1.5; zp[1] = 1.5;
  rtm_rx1(xp, yp, zp, 60);

  // (11) Rx2

  xp[0] = -30; xp[1] = -30; xp[2] = +30; xp[3] = +30;
  yp[0] = -5; yp[1] = +5; yp[2] = +5; yp[3] = -5;
  zp[0] = 1.5; zp[1] = 1.5; zp[2] = 1.5; zp[3] = 1.5;

```

```
rtm_rx2(xp, yp, zp, 10, 60);  
  
// (12) solver  
rtm_solver(2, 90, 1, 0, 0, 0, 1, 0.0, 10);  
  
// (13) misc  
rtm_misc(500, 20);  
  
// ==== output ====  
rtm_outdat("sample.rtm");  
  
return 0;  
}
```



サンプルプログラムの出力データを EEM-RTM で開いた図