

大村研インターンレポート

RayTracing プログラムの UI 化

学生番号:1030-30-1802

中本 悠太

平成 30 年 7 月 20 日

1 概要

地球磁気研では様々なプラズマ波動が生成・放射されているが、その発生メカニズムなどは解明されていない部分も多くある。その中でも、磁力線に沿って伝播し、周波数が数 kHz のホイッスラーモードコーラス波は物理的に重要な意味をもつ。レイトレーシングプログラムとはそのホイッスラーモードコーラス波の軌跡をシミュレーション上で確認するプログラムである。本インターンでは、レイトレーシングの初期値を UI 画面で操作できるようにすることで、レイトレーシングをより容易に確認できるようにプログラムを作成した。

2 内容

1. Raytracing の Matlab プログラムの改良

前年のインターンで金沢大学が公開している RayTracing の Fortran プログラムを Matlab に書き換えられていたので、そのプログラムを引き継いで改良を行った。

2. RayTracing の UI 画面作成

ホイッスラー波の初期位置や周波数、プラズマポーズなどの初期設定を UI 画面を用いて設定できるように Matlab の GUIDE 機能を用いてプログラムを作成した。

3 原理

3.1 ホイッスラー波の軌道

ホイッスラー波の軌道は次の微分方程式を満たす.

$$\frac{dr}{dt} = \frac{1}{\mu^2} \left(\rho_r - \mu \frac{\partial \mu}{\partial \rho_r} \right) \quad (1)$$

$$\frac{d\theta}{dt} = \frac{1}{r\mu^2} \left(\rho_\theta - \mu \frac{\partial \mu}{\partial \rho_\theta} \right) \quad (2)$$

$$\frac{d\phi}{dt} = \frac{1}{r\mu^2 \sin \theta} \left(\rho_\phi - \mu \frac{\partial \mu}{\partial \rho_\phi} \right) \quad (3)$$

$$\frac{d\rho_r}{dt} = \frac{1}{\mu} \frac{\partial \mu}{\partial r} + \rho_\theta \frac{d\theta}{dt} + \rho_\phi \frac{d\phi}{dt} \sin \theta \quad (4)$$

$$\frac{d\rho_\theta}{dt} = \frac{1}{r} \left(\frac{1}{\mu} \frac{\partial \mu}{\partial \theta} - \rho_\theta \frac{dr}{dt} + r\rho_\phi \frac{d\phi}{dt} \cos \theta \right) \quad (5)$$

$$\frac{d\rho_\phi}{dt} = \frac{1}{r \sin \theta} \left(\frac{1}{\mu} \frac{\partial \mu}{\partial \phi} \rho_\phi \frac{dr}{dt} \sin \theta - r\rho_\phi \frac{d\theta}{dt} \cos \theta \right) \quad (6)$$

r, θ, ϕ はホイッスラー波の極座標系での位置, $\rho_r, \rho_\theta, \rho_\phi$ は波数ベクトル, μ は屈折率である. 屈折率 μ は次の式を満たす.

$$A\mu^4 + B\mu^2 + C = 0 \quad (7)$$

A,B,C は次のように表される.

$$A = K_1 \cos^2 \Phi + K_2 \sin^2 \varphi \quad (8)$$

$$B = -[K_1 K_2 (1 + \cos^2 \varphi) + (K_2^2 + K_3^2) \sin^2 \varphi] \quad (9)$$

$$C = (K_2^2 + K_3^2) K_1 \quad (10)$$

$$K_1 = 1 - \sum_i X_i \quad (11)$$

$$K_2 = 1 + \sum_i \frac{X_i}{Y_i^2 - 1} \quad (12)$$

$$K_3 = j \sum_i \frac{X_i Y_i}{Y_i^2 - 1} \quad (13)$$

$$X_i = \frac{f_{Oi}^2}{f^2} \quad (14)$$

$$Y_i = \frac{f_{Hi}}{f} \quad (15)$$

A,B,C は f_{Oi}, f_{Hi} によって決まっており, f_{Oi}, f_{Hi} は i 番目の粒子 (電子またはイオン) のプラズマ周波数とサイクロトロン周波数である.

$$f_{Oi} = \frac{N_i Z_i e^2}{M_i \epsilon_0}, f_{Hi} = \frac{B_0 Z_i e}{M_i} \quad (16)$$

3.2 プラズマ密度モデル

拡散平衡状態の理論的なモデルでは, 電子の密度は次のように表される.

$$N_r(z) = \left[\sum_i \eta_i \exp \left(-\frac{z}{H_i} \right) \right]^{-\frac{1}{2}} \quad (17)$$

この時, z はジオポテンシャルで, H_i はスケールハイトで, i 番目のイオンの温度 T_i と質量 M_i , 距離 r_0 での重力 $g(r_0)$ を用いて次の式で表される.

$$z = \frac{r_0}{r}(r - r_0) \quad (18)$$

$$H_i = \frac{k_B T_i}{M_i g(r_0)} \quad (19)$$

また, i 番目の電子またはイオンの相対的な密度は

$$Q_i(z) = \frac{\eta_i \exp(-\frac{z}{H_i})}{N_r^2(z)} \quad (20)$$

で表され, 絶対的な電子の密度は

$$N_e = N_\theta \left[\sum_i \eta_i \exp(-\frac{z}{H_i}) \right]^{\frac{1}{2}} \quad (21)$$

$$N_i = N_e Q_i \quad (22)$$

電離圏などの高度が低い領域 (高度約 70km より低い領域) では LI モデルを用いる.

$$L_i(r) = 1 - \exp\left(-\frac{r - \text{RRLI}}{\text{HWLI}}\right)^2 \quad (23)$$

4 プログラム

4.1 ファイル構造

- raytracing.m : 「実行」を行うファイル UI 画面で入力した値の処理を行う.
- raytracing.fig : GUIDE を用いて UI 画面の配置を設定する.
- raytracing_main.m : Raytracing の解析を行う.
 - 初期化
INIT.m : 変数の初期化, UI 画面で入力した値を変数に代入する.
 - 軌道データの表示
GRAPH1.m : ホイッスラー波の軌道データをグラフで表示する.
 - 軌道に関する微分方程式の解
ADAMS.m : Adams 法によって軌道の微分方程式を解く
ADAM.B.m : アダムス・バッシュフォース法
ADAM.M.m : アダムス・ムルトン法
ADAM.UPDATE.m, ADAM.RETRY.m
 - 軌道データの座標変換
OUT0.m : ホイッスラー波の軌道の初期化をする. 極座標形式から直交座標へ.
OUT1.m : ホイッスラー波の軌道を出力する. 極座標形式から直交座標へ.
 - 屈折率の計算
REF.m : 屈折率を計算する.
 - 軌道決定用の関数の計算
FUNCT.m : アダムス法に必要な軌道決定用の関数の計算
 - プラズマ密度モデル
DENS.m : 下の 4 つのモデルを足し合わせ, プラズマ密度を求める.
DENSAO.m : Aikyo-Ondo モデル プラズマポーズを定義
DENSDE.m : 拡散平衡モデルを定義
DENSLI.m : 低い高度 (電離圏) のプラズマ密度
DENSOP.m
 - 地磁気モデル
DIPOLE.m : ダイポール磁場モデルを定義する.

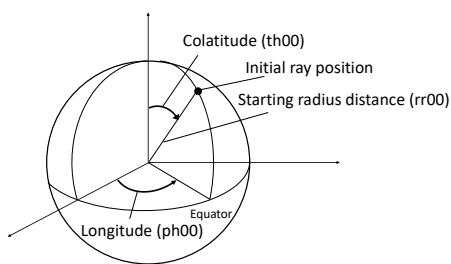
4.2 UI画面の説明

最初のパラメータ設定のUI画面のパラメータは図1のようにになっている。

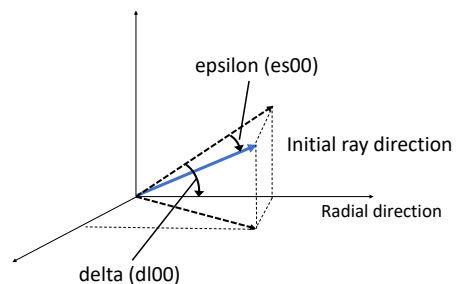
図 1: UI 画面

この時、初期位置と初期速度ベクトルについては図2を参考にする。

- Starting radius distance(from the earth center)：初期位置の地球中心からの半径 (N,rr00)
- Colatitude：余緯度 (th00)
- Longitude：経度 (ph00)
- delta：速度ベクトルを地磁気子午線平面に投影した方向と半径方向とのなす角 (dl00)
- epsilon：速度ベクトルと地磁気子午線平面とのなす角 (es00)
- Frequency：ホイッスラー波の周波数 ([kHz] か [Efreq(サイクロトロン周波数)] を単位とする)(M,fq00)
- Plasmopause：プラズマポーズの位置 (XLPP)
- Max of loops：最大ループ回数 (NLPMAX)



(a) 初期位置



(b) 初速度方向

図 2: 初期パラメータ説明

4.3 UI 画面の変更の仕方

UI 画面にパラメーターを追加する場合や既存のパラメーターを変更する場合の手順を説明する。基本的な流れとして、「GUIDE」で UI 画面を編集し、ボタンや入力枠を配置する。その後、raytracing.m でボタンや入力枠の処理を Callback 関数に書き込むという流れである。下に詳しく流れを示す。

1. matlab コマンドウィンドウで「guide」と入力し、UI の編集画面を立ち上げる

図 3 が「guide」と入力した場合の画面である。この時に、編集する UI 画面 (今回は raytracing.fig) を選択する。

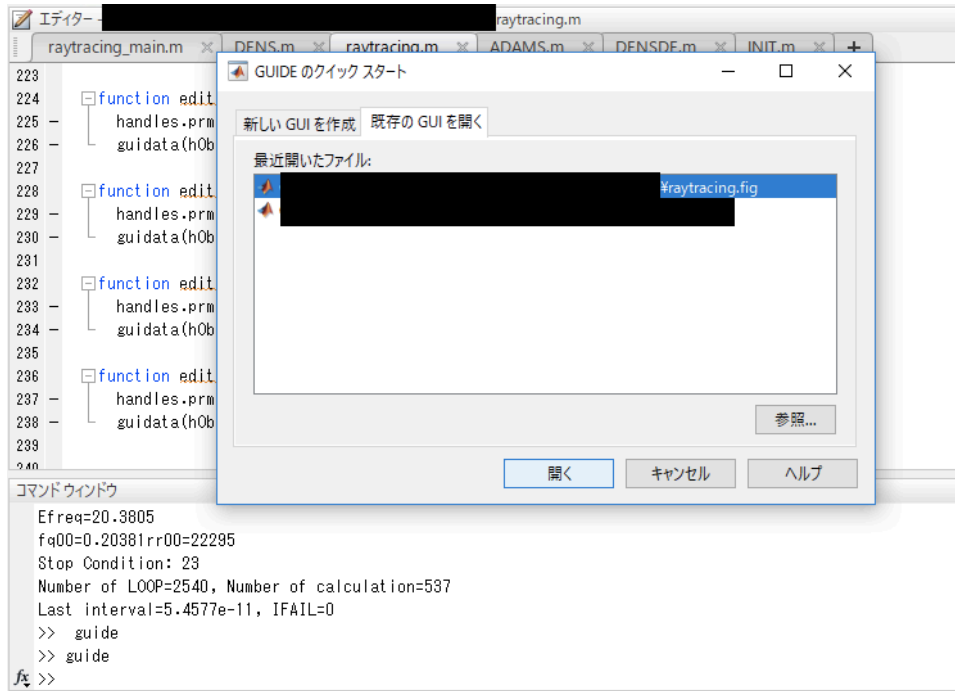


図 3: guide を開いた際の画面

2. 図4のような編集画面が開かれる。
新しく追加する場合は左のボタン群の中から選び、編集画面にドロップ&ドロワーで配置する。

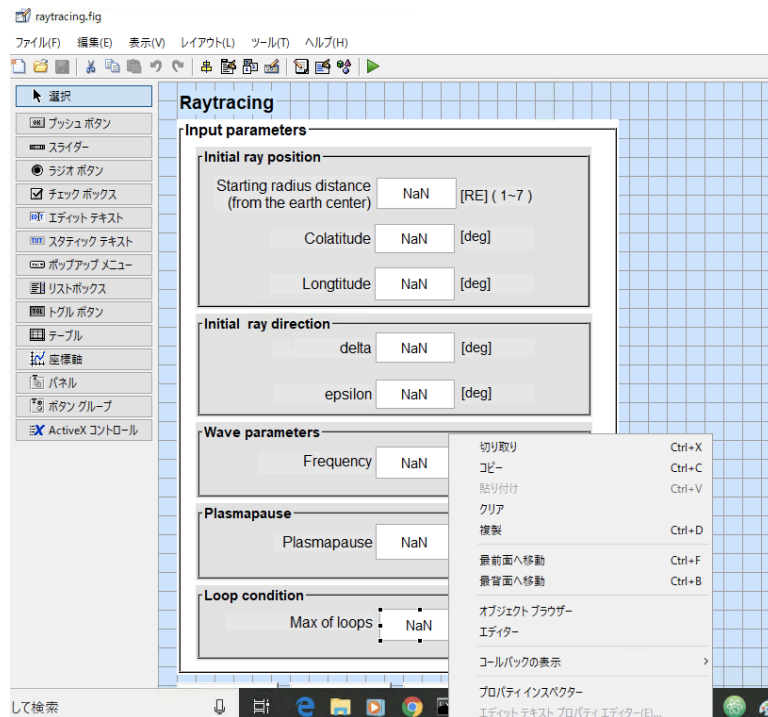


図 4: UI 編集画面

図5のように「右クリックでプロパティインスペクター」もしくは「ダブルクリック」でボタンの詳細を設定できる。この時、ボタンや入力枠は「Tag」の名前に基づいて Callback 関数などが作られるため、「Tag」の名前をわかりやすいものにする。

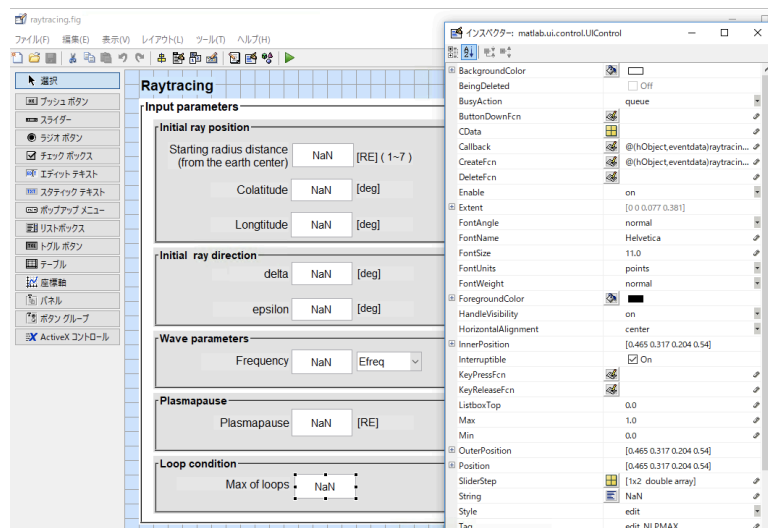


図 5: UI 編集画面のプロパティインスペクター

3. 新しくボタンを追加した後,「保存」ボタンを押すと m ファイルに「Callback」「CreateFcn」などの関数が付け足される.

これはボタンや入力枠の種類によって変わるので,使いたいものに合わせて調べてください.

今回は特に入力を受け付ける「エディットテキスト」について説明を行う.

「エディットテキスト」を保存すると「Callback」「CreateFcn」の2つの関数が付け足される.

```
function edit_NLPMAX_Callback(hObject, eventdata, handles)
% hObject    handle to edit58 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit58 as text
%        str2double(get(hObject,'String')) returns contents of edit58 as a double

% --- Executes during object creation, after setting all properties.
function edit_NLPMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit58 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

図 6: 新しくボタンを追加した際に m ファイルに追加される関数

この時,「CreateFcn」は最初にこのボックスが作られる際にする処理であり,特に変更しない.「Callback」が入力枠に数字などが入れられた時に行う処理である.入力された文字列を数値にして,prm に保存する.

```
function edit_NLPMAX_Callback(hObject, eventdata, handles)
    handles.prm.NLPMAX = str2double(get(hObject,'String'));
    guidata(hObject,handles);
```

図 7: Callback 関数の例

4. input_param.m にパラメータを追加し、パラメータをファイルから読み取れるようにする

```
switch prmname
case 'N'
    prm.N = value;
case 'M'
    prm.M = value;
case 'th00'
    prm.th00 = value;
case 'ph00'
    prm.ph00 = value;
case 'dl00'
    prm.dl00 = value;
case 'es00'
    prm.es00 = value;
case 'XLPP'
    prm.XLPP = value;
case 'NLPMAX'
    prm.NLPMAX = value;
case 'unit_fq00'
    prm.unit_fq00 = value;
end
```

図 8: input_param.m の例

5. set_param.m にパラメータを追加し、パラメータを GUI にセットする際にセットできるようにする

```
function handles = param_set(handles,prm)

try;set(handles.edit_N,'String',num2str(prm.N));catch;end;
try;set(handles.edit_M,'String',num2str(prm.M));catch;end;
try;set(handles.edit_th00,'String',num2str(prm.th00));catch;end;
try;set(handles.edit_ph00,'String',num2str(prm.ph00));catch;end;
try;set(handles.edit_dl00,'String',num2str(prm.dl00));catch;end;
try;set(handles.edit_es00,'String',num2str(prm.es00));catch;end;
try;set(handles.edit_XLPP,'String',num2str(prm.XLPP));catch;end;
try;set(handles.edit_NLPMAX,'String',num2str(prm.NLPMAX));catch;end;

try;set(handles.popup_unit_fq00,'Value',prm.unit_fq00);catch;end;

handles.prm.N = prm.N;
handles.prm.M = prm.M;
handles.prm.th00 = prm.th00;
handles.prm.ph00 = prm.ph00;
handles.prm.dl00 = prm.dl00;
handles.prm.es00 = prm.es00;
handles.prm.XLPP = prm.XLPP;
handles.prm.NLPMAX = prm.NLPMAX;
handles.prm.popup_unit_fq00 = prm.unit_fq00;

guidata(handles.figure1, handles);
```

図 9: set_param.m の例

6. save_param.m にパラメータを追加し、パラメータを保存しているファイルに保存できるようにする

```
function save_param(filename, prm)

    fid = fopen(filename, 'w');

    fprintf(fid, '%s Raytracing / input parameters\n\n');

    fprintf(fid, '%s -- --\n');
    fprintf(fid, 'N = %f;\n', prm.N);
    fprintf(fid, 'M = %f;\n', prm.M);
    fprintf(fid, 'unit_freq00 = %f;\n', prm.unit_freq00);
    fprintf(fid, 'th00 = %f;\n', prm.th00);
    fprintf(fid, 'ph00 = %f;\n', prm.ph00);
    fprintf(fid, 'dl00 = %f;\n', prm.dl00);
    fprintf(fid, 'es00 = %f;\n', prm.es00);
    fprintf(fid, 'XLPP = %f;\n', prm.XLPP);
    fprintf(fid, 'NLPMAX = %f;\n', prm.NLPMAX);
    fprintf(fid, '\n');

    fclose(fid);
    return;
```

図 10: save_param.m の例

7. INIT.m などにあるパラメータを変更する

```
% Following RRMAX & RRMIN: Upper & lower limit of ray tracing(Km???)
% Following ERRR and ERRA; Initial values for the limiting error
% and Hmin: Limit of integration increment
RRMAX = 7.0*RE; %!reset Max
RRMIN = 6370.0;
ERRR = 1.0 * 10^(-5);
ERRA = 1.0 * 10^(-5);
HMIN = 1.0 * 10^(-10);
% Following DISOUT: Every this count, the output data to be printed
DISOUT = 50;
% Following NLPMAX: Maximum number of integration loops
NLPMAX = prm.NLPMAX;
% INDOUT = 20;
% INDMSH = 51;
% IEOF = -1;
```

図 11: init.m の例

8. パラメータを追加した際、input_tmp.dat を読み込むので、初期値は入れておく。

参考文献

1. 金沢大学 RayTracing プログラム (Fortran) と説明
<http://waves.is.t.kanazawa-u.ac.jp/index.php>
2. Matlab GUIDE を使用した簡単なアプリの作成
https://jp.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html
3. GUI プログラムの作成 http://www.math.twcu.ac.jp/ogita/lec/sim_gui_ball.pdf