R2D2

リリース *1.1*

Hideyuki Hotta

2019 年 **12** 月 **25** 日

目次

このページは太陽のための輻射磁気流体コード R2D2(RSST and Radiation for Deep Dynamics) のマニュアルである。

PDF 版はこちら

第 1 章

R2D2 を使い始めるには

第 2 章

R2D2 python でのキーワードの説明

R2D2**.p**

- datadir (str) -- データの保存場所
- nd (int) -- 現在までのアウトプット時間ステップ数 (3 次元データ)
- ni (int) -- 現在までのアウトプット時間ステップ数 (光学的厚さ一定のデータ)
- xdcheck (int) -- x 軸方向に解いているか。解いていたら 2、解いていなかったら 1
- ydcheck (int) -- y 軸方向に解いているか。解いていたら 2、解いていなかったら 1
- zdcheck (int) -- z 軸方向に解いているか。解いていたら 2、解いていなかったら 1
- margin (int) --
- nx (int) --
- ny (int) --
- nz (int) --
- npe (int) --
- ix0 (int) --
- jx0 (int) --
- kx0 (int) --
- mtype (int) --
- xmax (float) --
- xmin (float) --
- ymax (float) --

- ymin (float) --
- zmax (float) --
- zmin (float) --
- dtout (float) --
- tend (float) --
- swap (int) --
- ix_e (int) --
- jx_e (int) --
- ixr (int) --
- jxr (int) --
- m_in (int) --
- m_tu (int) --
- dtoui (float) --
- ifac (int) --
- jc (int) --
- kc (int) --
- deep_top_flag (int) --
- ib_excluded_top (int) --
- endian (char) --
- ix (int) --
- jx (int) --
- kx (int) --
- x (float) [ix] --
- y (float) [jx] --
- z (float) [kx] --
- pr0 (float) [ix] --
- te0 (float) [ix] --

- ro0 (float) [ix] --
- se0 (float) [ix] --
- en0 (float) [ix]
- op0 (float) [ix]
- tu0 (float) [ix]
- dsedr0 (float) [ix]
- dtedr0 (float) [ix]
- dprdro (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- dprdse (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- dtedro (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- dtedse (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- dendro (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- dendse (float) [ix] -- 背景場の $(\partial p/\partial \rho)_s$
- gx (float) [ix] --
- kp (float) [ix] --
- cp (float) [ix] --
- fa (float) [ix] --
- sa (float) [ix] --
- xi (float) [ix] --
- rsun (float) [ix] --
- xr (float) [ix] -- x/rsun
- xn (float) [ix] -- (x-rsun)*1.e-8
- m2da (int) --
- cl (char) [m2da] --
- iss (int) [npe] --
- iee (int) [npe] --
- jss (int) [npe] --

- jee (int) [npe] --
- iixl (int) [npe] --
- jjxl (int) [npe] --
- np_ijr (int) [npe] --
- ir (int) [npe] --
- jr (int) [npe] --
- i2ir (int) [ix] --
- j2jr (int) [jx] --

a

aaa

第 3 章

数値スキーム

3.1 MHD スキーム

3.1.1 空間微分

R2D2 では、4 次の中央差分を用いている。格子間隔が一様な場合には中央差分では微分は

$$\left(\frac{\partial q}{\partial x}\right)_i = \frac{-q_{i+2} + 8q_{i+1} - 8q_{i-1} + q_{i-2}}{12\Delta x_i}$$

となる。R2D2 では、非一様な格子間隔にも対応しており、

3.1.2 時間積分

R2D2 では、

第 4 章

パラメータ

第 5 章

方程式

第 6 章

コード構造

第 7 章

座標生成

R2D2 では中央差分法を用いているが、そのほとんどは数値フラックスを用いて書き直すことでき、提供される x , y , z などは **セル中心** で定義される。よって計算領域内の最初のグリッドは、計算境界から半グリッド進んだところにある。

また、R2D2 では一様グリッドと非一様グリッドどちらでも計算できるようにしている。

7.1 一様グリッド

一様グリッドを用いるときは

- 格子間隔を計算する

$$\Delta x = \frac{x_{\max} - x_{\min}}{N_x}$$

ここで、コードでは、配列の要素数には margin も含むので N_x を計算するには margin の部分を引く必要があることに注意。

- x_1 を設定。margin の分も考慮して計算する。
- do loop で順次足していく

コードは以下のようになる

```
dx_unif = (xmax-xmin)/real(ix00-2*marginx)
x00(1) = xmin + (0.5d0-dble(marginx))*dx_unif
if(xdcheck == 2) then
    do i = 1+i1,ix00
        x00(i) = x00(i-i1) + dx_unif
    enddo
endif
```

7.2 非一様グリッド

非一様グリッドを用いるときは、太陽光球付近は、輻射輸送のために一様なグリッド、ある程度の深さから格子間隔が線形に増加する非一様グリッドを使うことにしている。

よって、ある初期の格子間隔 Δx_0

第 8 章

境界条件

論文を書くときは

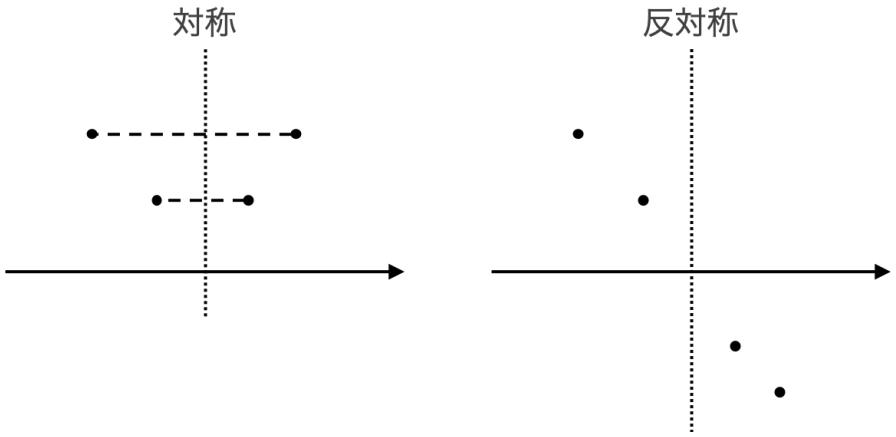
- x , y : 水平方向
- z : 鉛直方向

となっているが、R2D2 のコード内では

- x: 鉛直方向
- y, z: 水平方向

となっている。この取扱説明書では、コードに合わせた表記を用いる。

また、対称・反対称とは以下のような状況を表す。



8.1 上部境界

8.1.1 ポテンシャル磁場

磁場があるときは、上部ではポテンシャル磁場境界条件を使う。

8.2 下部境界

開く時どの質量フラックスも対称にする。 計算領域内での質量を一定に保つために、水平に平均した密度 $\langle\rho_1\rangle$ は反対称。 そこからのずれ $\rho_1 - \langle\rho_1\rangle$ は対称な境界条件をとる。

一方、エントロピー s_1 は上昇流で反対称、下降流で反対称な境界条件をとる。この心は、開く境界条件を取るときは計算をしている領域の結果は信用するが、外から入ってくる物理量は、計算領域に寄らないというものである。 下降流は現在計算している領域内部での情報を持って計算領域の外に出ていくので、対称な境界条件を用いる。一方、上昇流は、計算している領域の外からの情報を持って計算領域に入ってくるので、反対称な境界条件を用いて擾乱をゼロにする。これは元々の Model S での量を上昇流のエントロピーに用いるということである。

第 9 章

人工粘性

第 10 章

出力と読込

10.1 出力

10.1.1 Fortran コード

10.2 読込

読み込みについては、Python コードと IDL コードを用意しているが、開発の頻度が高い Python コードの利用を推奨している。

10.2.1 Python コード

Python で R2D2 で定義された関数を使うには

```
import R2D2
```

として、ライブラリを読み込む。R2D2 にはグローバル変数、関数が定義してある。

以下にそれぞれの関数を示すが、

```
R2D2.help()
```

とすると実行環境で簡単な説明を見ることができる。

グローバル変数

Python では一般に推奨されないが、R2D2 ではメモリ節約の目的のためにいくつかのグローバル変数を用意している。R2D2 をインポートした後は、`R2D2.*` とすることで変数にアクセスできる。

p
: 基本的なパラメタ。格子点数 `ix` や領域サイズ `xmax` など。 `read_init` で読み込んだ結果。

q2
: 2 次元の numpy array。ある高さのデータ。`read_qq_select` で読み込んだ結果。

q3
: 3 次元の numpy array。計算領域全体のデータ。`read_qq` で読み込んだ結果。

qi
: 2 次元の numpy array。ある光学的厚さの面でのデータ。現在は光学的厚さ 1, 0.1, 0.01 でのデータを出力している。 `read_qq` で読み込んだ結果。

vc
: Fortran の計算の中で解析した結果。 `read_vc` で読み込んだ結果。

qc
: 3 次元の numpy array。計算領域全体のデータ。Fortran の計算でチェックポイントのために出力しているデータを読み込む。主に解像度をあげたいときのために使う `read_qq_check` で読み込んだ結果。

`p` については、`init.py` などで

```
for key in R2D2.p:
    exec('%s = %s%s%s' % (key, 'R2D2.p["',key,'"]'))
```

としているために、辞書型の `key` を名前にする変数に値が代入されている。例えば、 `R2D2.p['ix']` と `ix` には同じ値が入っている。

関数

関数で指定する `dir` はデータの場所を示す変数。R2D2 の計算を実行すると `data` ディレクトリが生成されて、その中にデータが保存される。この場所を指定すれば良い。

read_init(*dir*)
: R2D2 でデータを解析するときに、一番はじめに実行すべき関数。計算設定などのパラメタが読み込まれる。 `R2D2.p` にデータが保存される。

  パラメータ　**dir**(*str*) -- データの場所

  戻り値　None

read_qq_select (*xs*, *n*, *silent*, *out*)

ある高さのデータのスライスを読み込む。戻り値を返さない時も R2D2.q2 にデータが保存される。

パラメータ

- **xs** (*float*) -- 読み込みたいデータの高さ
- **n** (*int*) -- 読み込みたい時間ステップ

戻り値　out=True が指定されているとデータが返される。

read_qq(*n*)

3 次元のデータを読み込む。戻り値を返さない時も R2D2.q3 にデータが保存される。

パラメータ

- **xs** (*float*) -- 読み込みたいデータの高さ
- **n** (*int*) -- 読み込みたい時間ステップ

戻り値　out=True が指定されているとデータが返される。

read_time(*n*)

時間を読み込む

パラメータ　**n** (*int*) -- 読み込みたい時間ステップ

戻り値　時間ステップでの時間が返される

read_vc(*n*)

Fortran コードの中で解析した計算結果を読み込む。戻り値を返さない時も R2D2.vc にデータが保存される。

パラメータ　**n** (*int*) -- 読み込みたい時間ステップ

戻り値　out=True が指定されているとデータが返される。

10.2.2 IDL コード

GitHub の公開レポジトリ に簡単な説明あり

10.3 バージョン履歴

- ver. 1.0: バージョン制を導入
- ver. 1.1: 光学的厚さが 0.1, 0.01 の部分も出力することにした。qq_in, vc を config のグローバル変数として取扱うことにした。

第 11 章

Sphinx 使用の覚書

11.1 はじめに

Sphinx は、reStructuredText から HTML や Latex などの文章を生成するソフトウェアである。Sphinx の公式サイト最近では Markdown でも記述できるが、結局最後のところは reStructuredText で記述することになるので、現状では、Markdown は使用していない。このウェブサイトも Sphinx で生成しているので、覚書をここに記す。

11.2 インストール

ここでは Anaconda がすでにインストールしてある Mac に Sphinx をインストールすることを考える。基本的には以下のコマンドを実行するのみである。

```
pip install sphinx
```

Markdown を使いたい時は以下のようにする。

```
pip install commonmark recommonmark
```

11.3 HTML ファイルの生成

適当なディレクトリを作成 (ここでは `test`) とする。そこで、 `sphinx-quickstart` コマンドにより Sphinx で作るドキュメントの初期設定を行う。

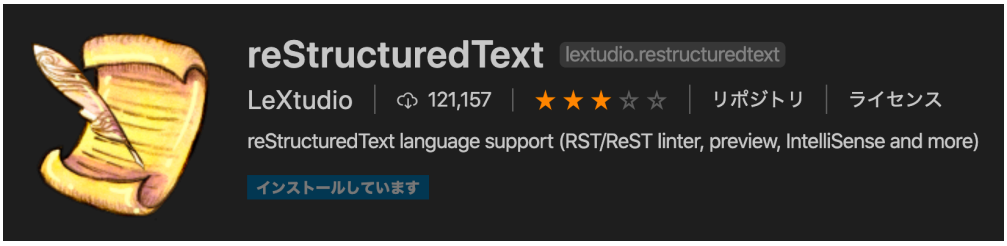
```
mkdir test # ディレクトリ作成
cd test    # ディレクトリに移動
sphinx-quickstart
```

いくつか質問をされる。基本的には読めばわかる質問であるが少し戸惑う質問を以下にあげる。

- プロジェクトのリリース: 1.0 などと version を答える。後に `conf.py` を編集すれば変更可能
- プロジェクトの言語: デフォルトは英語の `en` であるが、日本語を使いたい時は `ja` とする

11.4 VS code の利用

VS code を利用すると快適に reStructuredText を作成することができる。`*.rst` ファイルを VS code で開くと自動で確認されるが、以下のプラグインをインストールする。



`Cmd+k Cmd+r` で画面を分割してプレビューできる。正しい `conf.py` の場所を設定する必要がある。

11.5 環境設定

デフォルトの設定では、数式を書く時に Mathjax を使用するようで、数式の太字が意図するように表示されなかったので svg で出力することにした。以下のように `conf.py` に追記する。

```
extensions += ['sphinx.ext.imgmath']
imgmath_image_format = 'svg'
imgmath_font_size = 14
pngmath_latex='platex'
```

また、ウェブサイトのテーマを変更することもできる。どのようなテーマがあるかは Sphinx のテーマを参照。好きなテーマを選んで `conf.py` に以下のように設定。

```
html_theme = 'bizstyle'
html_theme_options = {'maincolor' : "#696969"}
```

今後変更の余地あり。

11.6 記法

11.6.1 リンク

- 外部ウェブサイト

```
`Twitter <https://twitter.com>`_
```

などとすると

Twitter

とリンクが生成される

- 内部サイト

自分で作成しているドキュメントをリンクするには

```
:doc:`index`
```

などとすると

R2D2 マニュアル

とリンクが生成される。

11.6.2 コード

Sphinx では、コードを直接記載することができる。また、言語に合わせてハイライトも可能。コードの表記に選択できる言語は Pygments にまとめてある。

```
.. code:: fortran

    implicit none
    real(KIND=0.d0) :: a,b,c

    a = 1.d0
    b = 2.d0
    c = a + b
```

このようにすると、以下のように表示される

```
implicit none
real(KIND=0.d0) :: a,b,c

a = 1.d0
```

(次のページに続く)

(前のページからの続き)

```
b = 2.d0
c = a + b
```

11.6.3 画像

画像の挿入には image ディレクティブを使う。オプションで、画像サイズなどを調整できる。堀田はだいたい width で調整している。

```
.. image:: source/figs/R2D2_logo.png
    :width: 350 px
```

とすると下記のように画像が挿入される。

11.6.4 数式

Sphinx では Latex を用いて数式を記述することができる。1 行の独立した数式を取り扱うときは

```
..  math::

    \frac{\partial \rho}{\partial t} = -\nabla\cdot \left(\rho {\boldsymbol v}\right)
```

とすると以下のように表示される。

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{v})$$

インラインの数式では

```
ここで :math:`\rho_1=x^2` とする
```

とすると

ここで $\rho_1 = x^2$ とする

と表示される。

第 12 章

ライセンス

R2D2 は現状、公開ソフトウェアではなく再配布も禁じている。これは今後、変更される可能性はあるが、開発者 (堀田) の利益を守るためである。共同研究者のみが使って良いというルールになっており、R2D2 の使用には、以下の規約を守る必要がある。

- 再配布しない
- 改変は許されるが、その時の実行結果について堀田は責任を持たない
- R2D2 で実行する計算は、堀田と議論する必要がある。パラメタ変更などの細かい変更には相談する必要はないが、新しいプロジェクトを開始するときはその都度相談すること。堀田自身のプロジェクト、堀田の指導学生のプロジェクトとの重複を避けるためである。
- R2D2 を用いた論文を出版するときは Hotta et al., 2019 を引用すること。より詳しく説明した論文も出版予定である。

第 13 章

索引と検索ページ

- genindex
- search

Ω

索引