

ジャイロ運動論的シミュレーションコードGKVを用いた微視的不安定性・乱流輸送解析 —ポスト処理編—

前山伸也

名大理

GKV講習会
2017年12月15日

今日の実習の問題設定(再掲)

円形トカマクモデル磁場の下で、微視的不安定性の線形解析をする。

1. 物理パラメータを `run/gkvp_f0.48_namelist` に入力する。
2. `src/gkvp_f0.48_header.f90` に計算格子数およびMPI並列数を入力する。
3. バッチジョブスクリプト `sub.q` を設定する。
4. ジョブ投入スクリプト `shoot` にディレクトリの設定をする。
5. コンパイルし、計算を実行。
6. **出力データを解析する。(ポスト処理ツールの利用)**

※非線形解析も計算タイプ"nonlinear"とし高解像度化する位で同様の手順。

6. 出力データを解析する。(再掲)

出力データを解析するには、

6-a) 自力で何とかする。

- GKVの出力データは一覧にまとめてあるので、後は適当にポスト処理する。
- アスキー形式の標準出力くらいなら簡単。
- MPI領域分割されたバイナリデータを読み込むのは結構手間。

コードのオープン化にあたり、ポスト処理ツールとして以下の2つを提供。

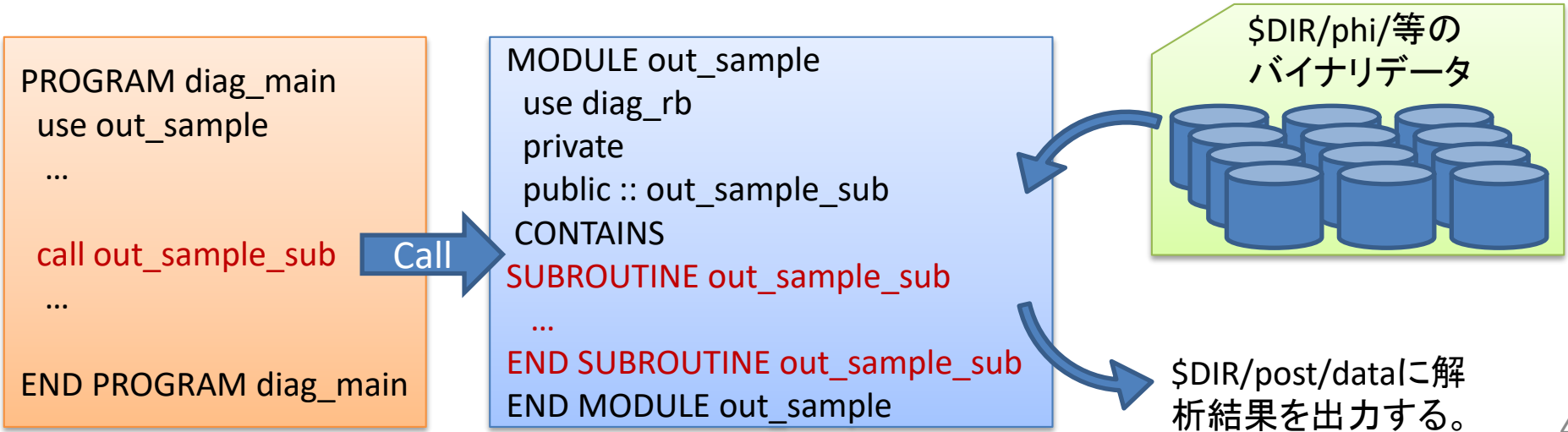
6-b) hst/のアスキー標準出力を一括でPDF化するためのスクリプト `fig_stdout`

6-c) **phi/などのバイナリデータを解析するためのポスト処理プログラム `diag`**

6-c) phi/などのバイナリデータを解析するためのポスト処理プログラム diag

diagで何ができるのか。

- diag_rbモジュールを利用することで、MPI分割されたGKVの出力データから、「解析したい時間ステップの解析したい物理量」を読み込める。(Appendix Bも参照)
- 読み込まれた変数は、領域分割されていない変数として再構成されるので、GKVのMPI分割に習熟していないユーザでも比較的簡便に扱える。
- 何か解析したいと思ったときは、その解析を行うためのモジュールout_*****.f90を作成し、変数などは外から見えないようにする。主プログラムdiag_main.f90はモジュールを呼び出すだけなので、複数人が独立にモジュール作成しても競合しない。



6-c) phi/などのバイナリデータを解析するためのポスト処理プログラム diag

extra_tools/v29diag.tar.gz を適当な場所に展開すると、

v29diag/

Makefile

go.diag_ps_fx バッチジョブ用スクリプト

backup/

plotfile/ gnuplot用スクリプトのサンプル

src/ ソースコード

diag_header.f90	解像度等の設定
diag_main.f90	解析モジュールの呼び出し
diag_rb.f90	GKV出力バイナリデータ読み込みモジュール
diag_*****.f90	その他各種設定
...	
out_*****.f90	特定の図を描くための専用モジュール群
...	

6-c) phi/などのバイナリデータを解析するためのポスト処理プログラム diag

【diagの使い方】

1. v29diag/src/diag_header.f90で、!%%% DIAG parameters %%% と !%%% GKV parameters %%% 内のパラメータを設定する。
2. src/diag_main.f90で解析したいモジュールを呼び出す。
3. go.diag_ps_fxで、解析したいデータのあるディレクトリを DIR に設定する。
4. コンパイル。(make)
5. 実行。(pjsub go.diag_ps_fx)
6. $\${DIR}/post$ 下にポスト処理されたデータが書き出される。

解析例① あるモードk の静電ポテンシャルの磁力線方向z分布

解析例② 静電ポテンシャルのx,y平面分布(悪い曲率領域 $z=0$ での断面図)

diagの使い方1. src/diag_header.f90にパラメータを設定する。

```
!%%% DIAG parameters %%%
```

```
integer, parameter :: snm = 1 ! Beginning of simulation runs
```

```
integer, parameter :: enm = 1 ! end of simulation runs
```

```
!%%%%%%%%%
```

※解析したいシミュレーションのラン数の範囲を指定する。

```
!%%% GKV parameters %%%
```

```
integer, parameter :: nxw = 2, nyw = 20
```

```
integer, parameter :: nx = 0, global_ny = 12 ! 2/3 de-aliasing rule
```

```
integer, parameter :: global_nz = 48, global_nv = 24, global_nm = 15
```

```
integer, parameter :: nzb = 3, & ! the number of ghost grids in z
```

```
nvb = 3 ! the number of ghost grids in v and m
```

```
integer, parameter :: nprocw = 2, nprocz = 4, nprocv = 4, nprocm = 2, nprocs = 1
```

```
!%%%%%%%%%
```

※gkvp_f0.48_header.f90と同じ格子数・MPI数を設定する。

v29diag/src/diag_header.f90

diagの使い方2. src/diag_main.f90で解析したいモジュールを呼び出す。

例として、静電ポテンシャル揺動の x, y 平面分布データ $\tilde{\phi}(x, y)$ を出力する。

```
PROGRAM diag
```

```
v29diag/src/diag_main.f90
```

```
...
```

```
use out_mominxy, only : phiinxy 解析モジュールの呼び出し
```

```
implicit none
```

```
integer :: giz, loop
```

```
...
```

```
giz = 0
```

```
loop = 100
```

```
call phiinxy( giz, loop )
```

z 座標の格子番号 ($-\text{global_nz} \leq \text{giz} \leq \text{global_nz}-1$) を指定。

出力時刻の番号 ($\text{time} = \text{dtout_ptn} * \text{loop}$) を指定。

$\text{giz}=0, \text{loop}=100$ における $\tilde{\phi}(x, y)$ が出力される。

```
...
```

```
END PROGRAM diag
```

※この3行をdiag_main.f90に追記してください。

diagの使い方2. src/diag_main.f90で解析したいモジュールを呼び出す。

別の例として、ある波数 k_{\perp} の静電ポテンシャルの磁力線方向分布 $\hat{\phi}_{k_{\perp}}(z)$ を出力する。

PROGRAM diag

v29diag/src/diag_main.f90

...

use out_mominz, only : phiinz 解析モジュールの呼び出し

implicit none

integer :: mx, gmy, loop

...

mx = 0

x座標の格子番号 ($-nx \leq mx \leq nx$) を指定。

gmy = 6

y座標の格子番号 ($0 \leq gmy \leq \text{global_ny}$) を指定。

loop = 100

出力時刻の番号 ($\text{time} = \text{dtout_ptn} * \text{loop}$) を指定。

call phiinz(mx, gmy, loop) $mx=0, gmy=6, loop=100$ における $\hat{\phi}_{k_{\perp}}(z)$ が出力される。

...

END PROGRAM diag

※この4行をdiag_main.f90に追記してください。

diagの使い方3. go.diag_ps_fxでディレクトリを設定する。

```
#PJM -L "rscunit=fx"  
#PJM -L "rscgrp=X24"  
#PJM -L "node=1"  
#PJM -L "elapse=00:15:00"  
#PJM -j  
#PJM -g 17293
```

v29diag/go.diag_ps_fx

```
setenv PARALLEL 8      # Thread number for automatic parallelization  
setenv OMP_NUM_THREADS 8 # Thread number for Open MP
```

```
### Working directory  
set DIR = /data/Ing/maeyama/gkv_training/linear_test  
set LDM = v29diag.exe
```

※解析したいデータのある
ディレクトリを DIR に設定する。

...

以下は不変

diagの使い方4&5. コンパイル&実行。

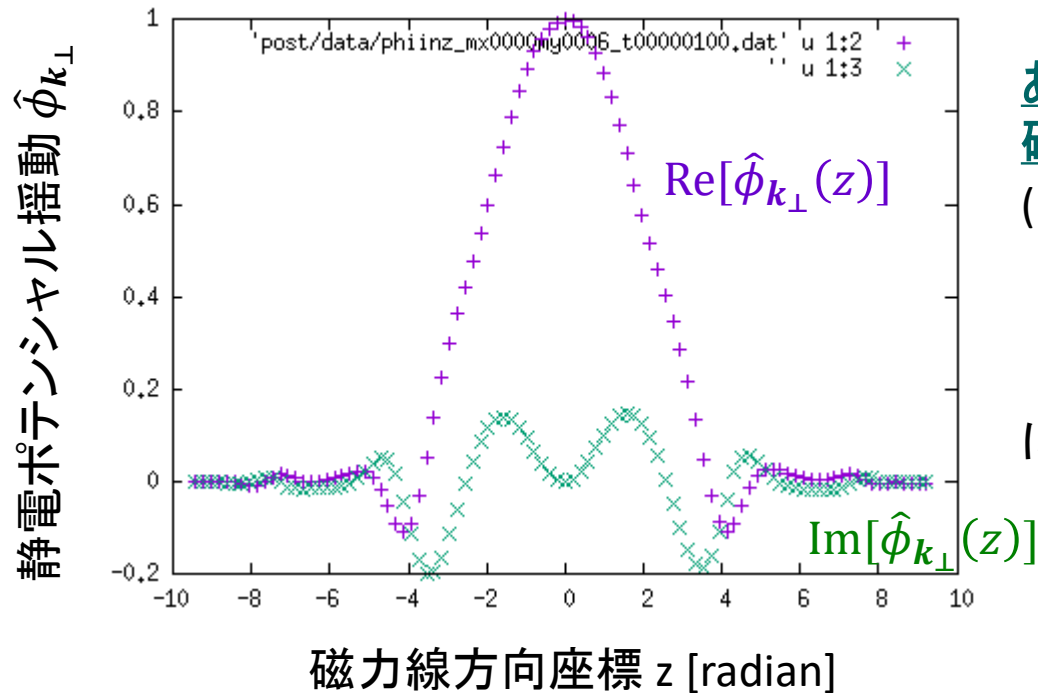
コンパイルする。v29diag/にて、
module purge
module load tcsuite-fx fftw-fx
make clean
make

計算を実行する。v29diag/にて、
pjsub go.diag_ps_fx

diagの使い方6. 解析結果を見る。

解析結果は $\{\text{DIR}\}/\text{post}$ (今例では、`/data/In/maeyama/gkv_training/linear_test/post`) に格納される。gnuplotで図にしてみよう。

```
plot 'post/data/phiinz_mx0000my0006_t00000100.dat' u 1:2, '' u 1:3
```



あるモードの静電ポテンシャル揺動の 磁力線z方向分布 $\hat{\phi}_{k_{\perp}}(z)$

($m_x=0, g_{m_y}=6, \text{loop}=100$)は、今例では、
 $k_x \rho_{ti} = k_{x,\min} * m_x = 0,$
 $k_y \rho_{ti} = k_{y,\min} * g_{m_y} = 0.3,$
 $t = dt_{\text{out_ptn}} * \text{loop} = 10R/v_{ti}$
に対応する。

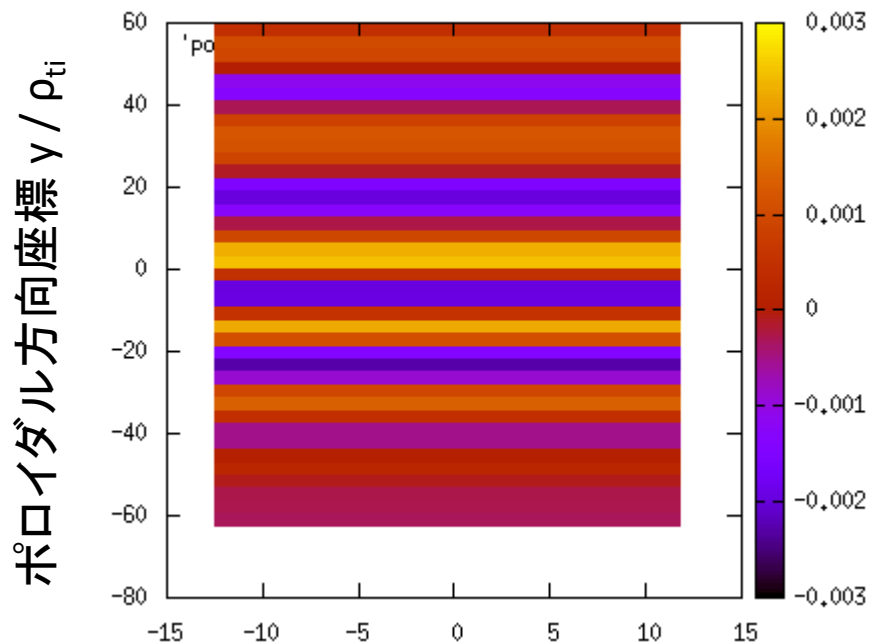
diagの使い方6. 解析結果を見る。

```
set pm3d map
```

```
splot 'post/data/phiinxy_z0000_t00000100.dat' u 1:2:3
```

静電ポテンシャル揺動のx,y断面分布 $\tilde{\phi}(x, y)$

(giz=0は、 $\theta = 0$ の悪い曲率領域に相当。)



カラー: 静電ポテンシャル揺動振幅 $\tilde{\phi}$

まとめ

GKVの利用方法をハンズオン形式で説明した。

要約

GKVの物理モデル・数値モデル
を踏まえた上で、

実験→GKVパラメータ換算

MHD平衡データの加工

の準備をしてから、

src/gkvp_f0.48_header.f90

run/gkvp_f0.48_namelist

run/sub.q

run/shoot

解像度、MPIの設定

物理・数値パラメータの設定

MPI・OpenMPの設定

平衡データ・出力ディレクトリの設定

の後にコンパイル、実行。

hst/のアスキー標準出力を一括でPDF化するためのスクリプト `fig_stdout`

phi/などのバイナリデータを解析するためのポスト処理プログラム `diag`

などを利用して、結果を解析する。

線形解析ができた人用の発展課題

非線形乱流シミュレーションを行い、fig_stdoutやdiagを利用してデータを解析する。
`nonlinear_test.tar.gz` を解凍し、パラメータ・ディレクトリを指定し、コンパイル&実行。

※実習時間では終わらないと思うので、今回は、計算結果サンプルをダウンロード。
`cp -r /tmp/GKV_Sample_Data/nonlinear_test/ /data/Ing/maeyama/gkv_training/
cd /data/Ing/maeyama/gkv_training/nonlinear_test/`

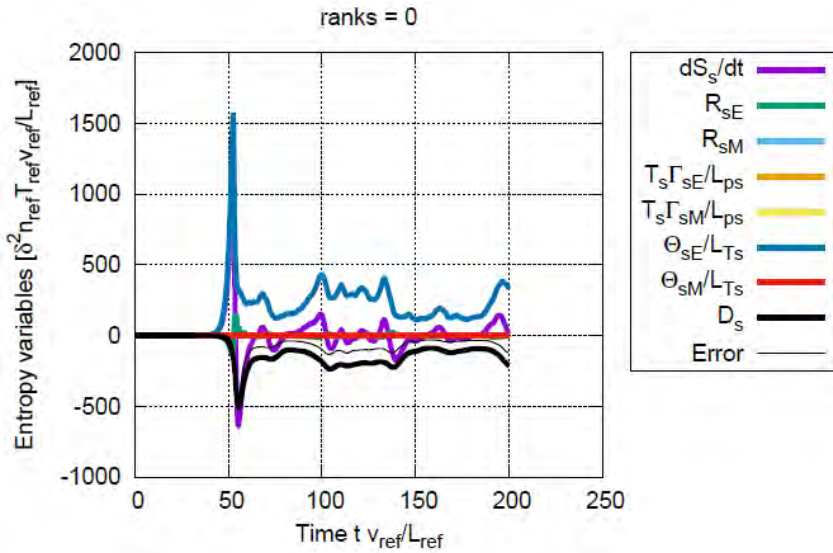
- fig_stdoutで結果を見てみましょう。
- diagで静電ポテンシャルのxy平面分布を書き出してみましょう。(diag_header.f90やdiag_main.f90やgo.diag_ps_fxの設定を忘れないこと。)
- ある時間帯について時間平均した静電ポテンシャル揺動のky波数スペクトルを作ってみましょう。例として、hst/gkvp_f0.48.eng.001などのアスキー標準出力を時間平均するawkスクリプトの例を/tmp/GKV_Sample_Data/time_average.awkに置きました。

使い方:

```
awk -f time_average.awk -v tsta=開始時刻 tend=終了時刻 ファイル名
```

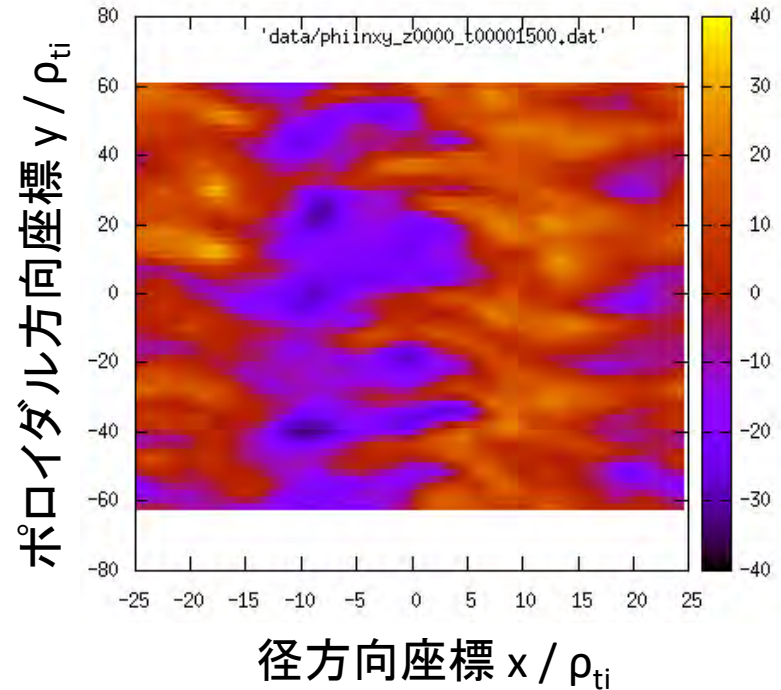
線形解析ができた人用の発展課題

エントロピーバランス方程式の確認 (fig_stdoutの出力PDFより抜粋)



イオン熱輸送項 Θ_{iE} / L_{Ti} と衝突散逸項 D_i が釣り合って定常に至る。
(低解像度のため若干誤差あり)

静電ポテンシャル揺動の2次元分布 (diagを用いて、 $z=0, t=150R/v_{ti}$ における $\tilde{\phi}(x, y)$ を描画)



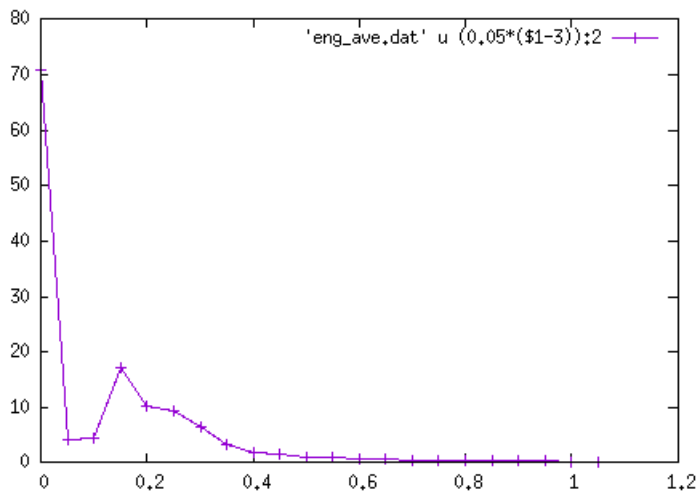
帯状流(ポロイダル方向に一様)が存在。16

線形解析ができた人用の発展課題

静電ポテンシャル揺動のポロイダル方向波数スペクトル

(time_average.awkを利用して、gkvp_f0.48.eng.001を $100 < tv_{ti}/R < 200$ の範囲で時間平均した。)

静電ポテンシャル
揺動振幅 $\langle |\hat{\phi}_{k_{\perp}}|^2 \rangle$



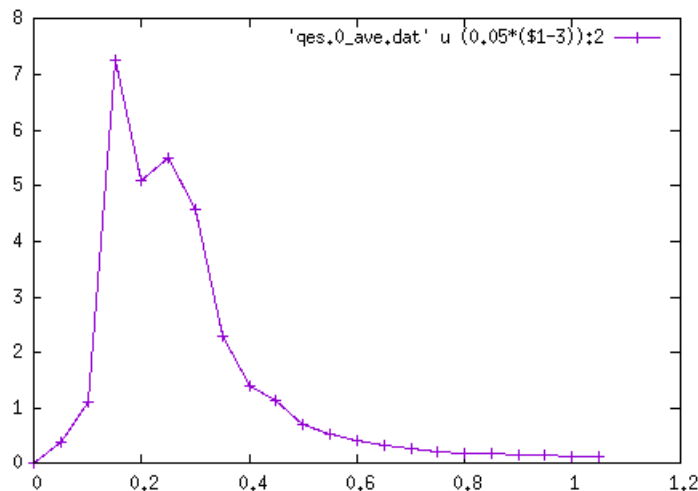
ポロイダル方向波数 $k_y \rho_{ti}$

帯状中($k_y=0$)の振幅が卓越。

イオン熱輸送フラックスのポロイダル方向波数スペクトル

(time_average.awkを利用して、gkvp_f0.48.qes.0.001を $100 < tv_{ti}/R < 200$ の範囲で時間平均した。)

イオン熱輸送
フラックス $Q_{k_{\perp}}$



ポロイダル方向波数 $k_y \rho_{ti}$

ピークは線形よりもやや低波数側にシフト。

Appendix B. ポスト処理プログラムdiag におけるバイナリデータの読み込み

diag_rbモジュールを利用してバイナリデータを読み込む

ポスト処理プログラムdiag中でGKV出力バイナリデータを読み込むには、diag_rbモジュールで用意されているサブルーチンをcallする。

```
例) use diag_rb, only : rb_phi_loop  
    complex(kind=DP) :: phi(-nx:nx,0:global_ny,-global_nz:global_nz-1)  
    integer :: loop = 100  
    call rb_phi_loop(loop, phi)
```

※出力レコード loop = 100 番目の静電ポテンシャルphiを読み込む。

※phiの場合は time = dtout_ptn * loop の時刻のデータに相当。

以下、diag_rbの構造と、主要なサブルーチンについて一覧を示す。

v29diag/src/diag_rb.f90の構造

```
MODULE diag_rb
```

```
  public
```

```
  integer, dimension(1:enum) :: loop_phi_sta, loop_phi_end
```

※phiのバイナリデータのレコード数を
格納しておく変数

```
CONTAINS
```

```
SUBROUTINE rb_phi_gettime( loop, time )
```

※レコード数に対応するtimeを
読み込むサブルーチン

```
  ...
```

```
END SUBROUTINE rb_phi_gettime
```

```
SUBROUTINE rb_phi_loop( loop, phi )
```

※レコード数に対応するphiを
読み込むサブルーチン

```
  ...
```

```
END SUBROUTINE rb_phi_loop
```

```
...
```

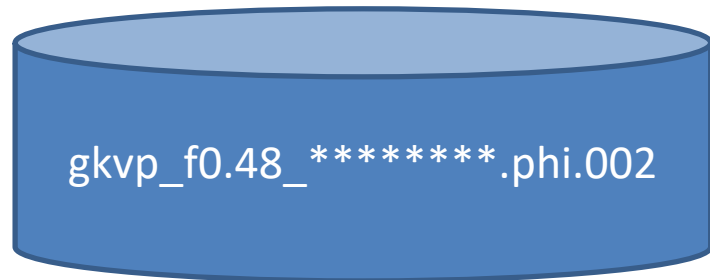
```
END MODULE
```

バイナリデータのレコード数

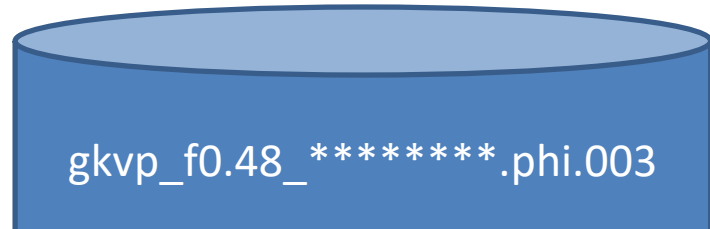
integer, dimension(1:enum) :: loop_phi_sta, loop_phi_end



loop_phi_sta(001)
...
...
Loop_phi_end(001)



Loop_phi_sta(002)
...
...
Loop_phi_end(002)



Loop_phi_sta(003)
...
...

バイナリデータのサイズから各ランにおけるレコード数を算出できる。

全レコード数は loop_phi_sta(001) から loop_phi_end(enum) まで。

※ラン数 snum から enum までしか解析しないとしても、データはラン数 001 から置いておく必要あり。

rb_phi_gettime(loop, time)

【引数の型、入出力】

integer, intent(in) :: loop

real(kind=DP), intent(out) :: time

【役割】

静電ポテンシャルのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).0.phi.(ラン数3桁)

について、

レコード数 loop の時刻に対応する時刻を time に読み込む。

rb_AI_gettime(loop, time)

【引数の型、入出力】

integer, intent(in) :: loop

real(kind=DP), intent(out) :: time

【役割】

ベクトルポテンシャルのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).0.AI.(ラン数3桁)

について、

レコード数 loop の時刻に対応する時刻を time に読み込む。

rb_mom_gettime(loop, time)

【引数の型、入出力】

integer, intent(in) :: loop

real(kind=DP), intent(out) :: time

【役割】

流体モーメントのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).(粒子種1桁).mom.(ラン数3桁)
について、

レコード数 loop の時刻に対応する時刻を time に読み込む。

rb_trn_gettime(loop, time)

【引数の型、入出力】

integer, intent(in) :: loop

real(kind=DP), intent(out) :: time

【役割】

エントロピーバランスに関する諸量のバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).(粒子種1桁).trn.(ラン数3桁)
について、

レコード数 loop の時刻に対応する時刻を time に読み込む。

rb_phi_loop(loop, phi)

【引数の型、入出力】

integer, intent(in) :: loop

complex(kind=DP), dimension(-nx:nx,0:global_ny,-global_nz:global_nz-1), intent(out) :: phi

【役割】

静電ポテンシャルのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).0.phi.(ラン数3桁)

について、

レコード数 loop の時刻に対応する静電ポテンシャルを phi に読み込む。

rb_AI_loop(loop, AI)

【引数の型、入出力】

integer, intent(in) :: loop

complex(kind=DP), dimension(-nx:nx,0:global_ny,-global_nz:global_nz-1), intent(out) :: AI

【役割】

ベクトルポテンシャルのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).0.AI.(ラン数3桁)

について、

レコード数 loop の時刻に対応するベクトルポテンシャルを AI に読み込む。

rb_mom_imomisloop(imom, is, loop, mom)

【引数の型、入出力】

integer, intent(in) :: imom, is, loop

complex(kind=DP), dimension(-nx:nx,0:global_ny,-global_nz:global_nz-1), intent(out) :: mom

【役割】

流体モーメントのバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).(粒子種1桁).mom.(ラン数3桁)
について、

レコード数 loop の時刻に対応する流体モーメントを mom に読み込む。

ここで、isは粒子種番号で、imom=0~5はそれぞれ順に以下の流体モーメントに対応する。

$$\tilde{n}_{sk} = \int dv^3 J_{0sk} \tilde{f}_{sk}, \quad \tilde{u}_{\parallel sk} = \int dv^3 v_{\parallel} J_{0sk} \tilde{f}_{sk}, \quad \tilde{p}_{\parallel sk} = \int dv^3 \frac{v_{\parallel}^2}{2} J_{0sk} \tilde{f}_{sk},$$

$$\tilde{p}_{\perp sk} = \int dv^3 \mu B J_{0sk} \tilde{f}_{sk}, \quad \tilde{q}_{\parallel\parallel sk} = \int dv^3 v_{\parallel} \frac{v_{\parallel}^2}{2} J_{0sk} \tilde{f}_{sk}, \quad \tilde{q}_{\parallel\perp sk} = \int dv^3 v_{\parallel} \mu B J_{0sk} \tilde{f}_{sk}$$

rb_trn_itrnisloop(itrn, is, loop, mom)

【引数の型、入出力】

integer, intent(in) :: itrn, is, loop

complex(kind=DP), dimension(-nx:nx,0:global_ny), intent(out) :: trn

【役割】

エントロピーバランスに関する諸量のバイナリデータ

phi/gkvp_f0.48_(MPIランク6桁).(粒子種1桁).trn.(ラン数3桁)

について、

レコード数 `loop` の時刻に対応するエントロピーバランスに関する諸量を `trn` に読み込む。

ここで、`is`は粒子種番号で、`itrn=0~11`はそれぞれ順に、

ジャイロ中心揺動エントロピー、静電揺動エネルギー（イオン分極項含む）、磁場揺動エネルギー、波粒子相互作用 ($W_E \rightarrow S_s$)、波粒子相互作用 ($W_M \rightarrow S_s$)、 $E \times B$ 流による非線形エントロピー伝達、磁場揺動による非線形エントロピー伝達、衝突散逸、 $E \times B$ 流による粒子輸送フラックス、磁場揺動による粒子輸送フラックス、 $E \times B$ 流によるエネルギー輸送フラックス、磁場揺動によるエネルギー輸送フラックス。

※繰り返しになるが、`itrn=8,9,10,11`が粒子・熱輸送束の波数空間スペクトルに対応。

Appendix C. ポスト処理プログラムdiag の解析モジュールの例

解析モジュール名	含まれる解析サブルーチン名
out_mominxy	phiinxy, Alinxy, mominxy
out_mominz	phiinz, Alinz, mominz, phiinz_connect
out_mominkxky	phiinkxky, Alinkxky, mominkxky
out_trninkxky	trninkxky
out_triinkxky	triinkxky

out_mominxy: phiinxy(giz, loop)

【引数の型、入出力】

integer, intent(in) :: giz, loop

【役割】

レコード数 loop の時刻に対応するz方向座標番号 giz の位置での静電ポテンシャル揺動の x,y 平面分布 $\tilde{\phi}(x, y)$ を

post/data/phiinxy_z(giz 4桁)_t(loop 8桁).dat
に書き出す。

out_mominxy: Alinxy(giz, loop)

【引数の型、入出力】

integer, intent(in) :: giz, loop

【役割】

レコード数 loop の時刻に対応するz方向座標番号 giz の位置でのベクトルポテンシャル揺動のx,y 平面分布 $\tilde{A}_{\parallel}(x, y)$ を

post/data/Alinxy_z(giz 4桁)_t(loop 8桁).dat
に書き出す。

out_mominxy: mominxy(giz, is, loop)

【引数の型、入出力】

integer, intent(in) :: giz, is, loop

【役割】

レコード数 loop の時刻に対応するz方向座標番号 giz の位置での粒子種 is の流体モーメント量のx,y 平面分布 \tilde{n} , \tilde{u}_{\parallel} , \tilde{p}_{\parallel} , \tilde{p}_{\perp} , $\tilde{q}_{\parallel\parallel}$, $\tilde{q}_{\parallel\perp}$ を

post/data/mominxy_z(giz 4桁)s(is 1桁)_t(loop 8桁).dat
に書き出す。

out_mominz: phiinz(mx, gmy, loop)

【引数の型、入出力】

integer, intent(in) :: mx, gmy, loop

【役割】

レコード数 loop の時刻に対応する k_x, k_y モード数 mx, gmy について、静電ポテンシャル揺動のフーリエ係数の磁力線方向 z 分布 $\tilde{\phi}_k(z)$ を

post/data/phiinz_mx(mx 4桁)my(gmy 4桁)_t(loop 8桁).dat
に書き出す。

out_mominz: Alinz(mx, gmy, loop)

【引数の型、入出力】

integer, intent(in) :: mx, gmy, loop

【役割】

レコード数 loop の時刻に対応する k_x, k_y モード数 mx, gmy について、ベクトルポテンシャル揺動のフーリエ係数の磁力線方向 z 分布 $\tilde{A}_{\parallel k}(z)$ を

post/data/Alinz_mx(mx 4桁)my(gmy 4桁)_t(loop 8桁).dat
に書き出す。

out_mominz: mominz(mx, gmy, is, loop)

【引数の型、入出力】

integer, intent(in) :: mx, gmy, is, loop

【役割】

レコード数 loop の時刻に対応する k_x, k_y モード数 mx, gmy について、粒子種 is の流体モーメント量のフーリエ係数の磁力線方向 z 分布 $\tilde{n}, \tilde{u}_{\parallel}, \tilde{p}_{\parallel}, \tilde{p}_{\perp}, \tilde{q}_{\parallel\parallel}, \tilde{q}_{\parallel\perp}$ を
post/data/mominz_mx(mx 4桁)my(gmy 4桁)s(is 1桁)_t(loop 8桁).dat
に書き出す。

out_mominz: phiinz_connect(mx, gmy, loop)

【引数の型、入出力】

integer, intent(in) :: mx, gmy, loop

【役割】

レコード数 loop の時刻に対応する k_x, k_y モード数 mx, gmy について、静電ポテンシャル揺動のフーリエ係数の磁力線方向 z 分布 $\tilde{\phi}_k(z)$ を

post/data/phiinz_connect_mx(mx 4桁)my(gmy 4桁)_t(loop 8桁).dat

に書き出す。この際、フラックスチューブモデルにおける異なるモード間の磁力線方向への接続境界を考慮して、磁力線方向に伸展した分布を出力する。

out_mominkxky: phiinkxky(loop)

【引数の型、入出力】

integer, intent(in) :: loop

【役割】

レコード数 loop の時刻に対応する静電ポテンシャル揺動の2次元(kx, ky)波数スペクトル

$\langle |\tilde{\phi}_k|^2 \rangle / 2$ を

post/data/phiinkxky_t(loop 8桁).dat
に書き出す。

out_mominkxky: Alinkxky(loop)

【引数の型、入出力】

integer, intent(in) :: loop

【役割】

レコード数 loop の時刻に対応するベクトルポテンシャル揺動の2次元(kx, ky)波数スペクトル $\langle |\tilde{A}_{\parallel k}|^2 \rangle / 2$ を
post/data/Alinkxky_t(loop 8桁).dat
に書き出す。

out_mominkxky: mominkxky(is, loop)

【引数の型、入出力】

integer, intent(in) :: is, loop

【役割】

レコード数 loop の時刻に対応する粒子種 is の流体モーメントの2次元(kx, ky)波数スペクトル

$\frac{1}{2}\langle|\tilde{n}_k|^2\rangle$, $\frac{1}{2}\langle|\tilde{u}_{\parallel k}|^2\rangle$, $\frac{1}{2}\langle|\tilde{p}_{\parallel k}|^2\rangle$, $\frac{1}{2}\langle|\tilde{p}_{\perp k}|^2\rangle$, $\frac{1}{2}\langle|\tilde{q}_{\parallel\parallel k}|^2\rangle$, $\frac{1}{2}\langle|\tilde{q}_{\parallel\perp k}|^2\rangle$ を

post/data/mominkxky_s(is 1桁)_t(loop 8桁).dat

に書き出す。

out_trninkxky: trninkxky(is, loop)

【引数の型、入出力】

integer, intent(in) :: is, loop

【役割】

レコード数 loop の時刻に対応する粒子種 is のエントロピーバランスに関する諸量の2次元 (kx, ky) 波数スペクトルを

post/data/trninkxky_s(is 1桁)_t(loop 8桁).dat
に書き出す。

out_triinkxky: triinkxky(mxt, myt, is, loop)

【引数の型、入出力】

integer, intent(in) :: mxt, myt, is, loop

【役割】

レコード数 loop の時刻に対応する粒子種 is、モード $k = kx(mxt), ky(myt)$ の三波結合伝達関数について、2次元(px, py)波数スペクトル $J_{SEk}^{p,q}, J_{SEp}^{q,k}, J_{SEq}^{k,p}, J_{SMk}^{p,q}, J_{SMp}^{q,k}, J_{SMq}^{k,p}$ を post/data/trninkxky_s(is 1桁)mx(mxt 4桁)my(myt 4桁)_t(loop 8桁).dat に書き出す。