宇宙磁気流体・プラズマシミュレーションテクニカルマニュアル

平成２４年度

目 次

第 1 章　CANS を使った磁気流体力学シミュレーション **7**
1.1 CANS とは 7
1.1.1 天文数値ソフトウェア 7
1.1.2 CANS で何ができるか？ 7
1.1.3 CANS の特徴 7
1.2 CANS を使ってみよう 8
1.2.1 動作環境 8
1.2.2 インストールと IDL 設定 8
1.2.3 準備コンパイル 9
1.2.4 プログラムの実行 10
1.2.5 IDL による可視化 11
1.3 計算プログラムの解説 12
1.3.1 メインルーチン main.f 12
1.3.2 サブルーチン model.f：メッシュ座標・初期条件設定 19
1.3.3 サブルーチン bnd.f：境界条件 21

第 2 章　CANS ホームページの JavaScript ムービー解説 **23**
2.1 用意すべき画像ファイル 23
2.2 設定 23
2.3 コントロール 25
2.4 ソースコード 26
2.4.1 JavaScript プログラム 26
2.4.2 HTML 部分 32

第 3 章　モーメント法を用いた相対論的輻射流体の数値解法 **33**
3.1 輻射場の支配方程式 33
3.1.1 輻射のモーメント式の導入 33
3.1.2 輻射場の状態方程式　　クロージャー関係式 37
3.2 相対論的輻射流体方程式 39
3.3 輻射モーメント方程式の数値解法 41
3.3.1 離散化 41
3.3.2 数値解法 42
3.3.3 クーラン条件と陰解法への拡張 44

3.4 計算例 46
3.4.1 熱平衡状態への遷移 46
3.4.2 日陰問題 47
3.4.3 光の伝播と衝突 47
3.5 まとめ 49

第 4 章 相対論的散逸流体の数値計算法 53
4.1 背景 53
4.2 基礎方程式 53
4.2.1 理想流体 54
4.2.2 Causal dissipative fluid 54
4.2.3 緩和時間の表式 57
4.3 Numerical scheme 58
4.3.1 Strang splitting method 58
4.3.2 Stiff equation 60
4.3.3 Primitive recovery 61
4.3.4 多次元への拡張 62
4.4 Causality and Stability 62
4.4.1 Stability of the telegrapher equation 62
4.5 テスト計算 64
4.5.1 1D shear flow 65
4.5.2 shock tube test 66

第 5 章 太陽表面対流のための輻射磁気流体方程式の数値解法 71
5.1 はじめに 71
5.2 基礎方程式 71
5.3 流体部分 72
5.3.1 空間差分 72
5.3.2 時間積分 73
5.3.3 テスト問題 74
5.4 輻射輸送部分 75
5.4.1 Short Characteristic 法 75
5.4.2 テスト問題 77
5.5 計算例 77

第 6 章 ブラソフシミュレーション技法 83
6.1 イントロダクション 83
6.2 ブラソフ方程式の数値解法 84
6.2.1 マルチモーメント移流法 85
6.3 マクスウェル方程式の数値解法 86
6.4 変数と方程式の規格化 88

6.5 ブラソフ-マクスウェル系の時間更新 89
6.6 ブラソフシミュレーション 90
6.6.1 垂直伝播波動 90
6.6.2 垂直衝撃波 91
6.7 まとめと今後 92

第1章　CANSを使った磁気流体力学シミュレーション

横山央明 (東京大学)
（ver.1: 花山秀和 (国立天文台) ）

1.1　CANSとは

1.1.1　天文数値ソフトウェア

物理法則にしたがって時間発展する系の進化を計算機で追及するという数値実験が、従来の理論・観測（または室内実験）に加わる第3の研究手法として確立してきた。とくに天体現象は、地上での実験が困難なのでこの手法が有効である。そのような目的で開発される計算コードは、研究者個人が作成し、個人的に使用されるか、あるいは人づてなど狭い範囲での受け渡しが多い。またある特定の研究目的を達成するために作成されているため、可読性が高くない、柔軟性がない、などの難点をもつ。そこで（ある程度の）汎用性をもつコードがいくつか公開されている。ZEUSやFlashなどはその代表例である。これらのコードに対抗して、国産のパッケージとして開発されたのがCANS（Coordinated Astronomical Numerical Softwares）である。

1.1.2　CANSで何ができるか？

CANSを用いると、さまざまな流体現象の数値シミュレーションを実施し可視化することができる。とくに天体現象に適用することをめざしており、それに対応した問題設定（後述）を用意している。なかでも磁気流体力学方程式を解くことを重要視しており、プラズマ流体現象を扱うことができる。

CANSでは、圧縮性流体・磁気流体を基礎として、熱伝導・粘性・磁気拡散などの散逸現象や、光学的に薄い放射による冷却などの物理過程をモジュールとして追加することができる。1次元から3次元までの問題に対応しており、MPIライブラリによる並列化が施されている。

1.1.3　CANSの特徴

CANSの最大の特徴は、方程式を解く計算コードそのものに加えて、「問題設定」が多数用意されていることである。これは、初期条件・境界条件・推奨パラメータ・可視化プログラム・解説をひとまとめにしたもので、（環境さえあれば）すぐにでも結果を可視化するところまでたどりつける。「問題設定」にはたとえば、衝撃波管問題・Rayleigh-Taylor不安定・Sedov解・磁気リコネクションなどの基本的な

問題以外に、天体への応用で恒星風・ジェット伝播・磁気回転不安定・フレア・Parker 不安定・自己重力収縮なども用意されている。これらはシミュレーションに慣れるという目的以外に、それぞれの「問題設定」を改造して実際の研究に使うことができるよう配慮されている。

また、おなじ方程式に対し複数の数値解法が用意されており、比較しながら計算法の特徴を知ることができる。たとえば磁気流体方程式に対しては、改良 Lax-Wendroff 法・Roe 流束による高精度風上差分法・CIP-MOCCT 法など（HLLD 法をまもなく実装予定）がある。拡散方程式に対しては、中心差分陽解法・SOR 陰解法・biCG 陰解法などがある。

1.2　CANS を使ってみよう

CANS のインストールから計算実行・可視化までを概観する。

1.2.1　動作環境

計算コードは、Fortran77（プラス多少の拡張仕様）で記述されている。したがって Fortran コンパイラが必要である。（GCC に含まれている）GNU Fortran で開発されているが、機種依存機能は極力使われておらず、普及しているコンパイラではほぼ問題なく動作する。コマンドライン（つまり非 GUI 環境）で利用されることを前提に作られていて、make コマンドが使えることが必須である。Windows の場合、Cygwin などで使うことを勧める。可視化には、IDL を使うことが前提でプログラムが用意されている。

1.2.2　インストールと IDL 設定

CANS のホームページから配布アーカイブをダウンロードして解凍する。結果以下のようなディレクトリ（フォルダ）構成になる。

```
# ls
cans/
# cd cans/
# ls
Makefile          cans3d/            idl-nosupport/  cans1d/
cans3d-nosupport/  index.html      cans1d-nosupport/  cansio/
cans2d/            cansio-nosupport/  cans2d-nosupport/  idl/
```

ディレクトリ cans1d/、cans2d/、cans3d/、cansio/が、計算コードと問題設定とが含まれる CANS の主たる構成要素である。idl/には可視化にもちいるプログラムの一部が含まれる。*-nosupport/となっているディレクトリに含まれるコードは、動作保証がない開発途中のコードである。ここにある index.html をブラウザで表示すると、CANS 問題設定の計算結果例の画像や動画をみることができる。

以下、cans2d/を例に説明を続ける。

```
# cd cans2d/
# ls
Makefile*     hdcip/       md_corjet/      md_mhdgwave/    md_rt/
README.txt*   hdglr/       md_diskjet/     md_mhdkh/       md_sedov/
bc/           hdmlw/       md_efr/         md_mhdshktb/    md_shkref/
              hdroe/        md_flare/       md_mhdsn/       md_shktb/
cndbicg/      htcl/        md_itmhdshktb/  md_mhdwave/     md_sndwave/
cndbicgmpi/   md_advect/   md_itshktb/     md_mri/         md_sninteraction/
cndsor/       md_awdecay/  md_jetprop/     md_mricyl/      md_thinst/
cndsormpi/    md_cme/      md_kh/          md_parkerinst/
common/       md_cndsp/    md_mhd3shktb/   md_reccnd/
commonmpi/    md_cndtb/    md_mhdcndtb/    md_recon/
```

ディレクトリ名の先頭に「md_」と付いているのが「問題設定」（MoDel）で、それ以外が方程式を解くサブルーチン群である。たとえば hdmlw/は、改良 Lax-Wendroff 法で流体方程式を解くルーチン群が入っている。cndbicg/は、熱伝導方程式を biCG 法で解くルーチンなどといった具合である。詳しくは「README.txt」に記述されている。

md_*について一部紹介すると、md_shktb/は衝撃波管問題（Sod の問題）、md_rt/は Rayleigh-Taylor 不安定、md_mri/は磁気回転不安定、md_efr/は太陽表面の浮上磁場現象、md_recon/は磁気リコネクションなどなど。詳しくは各ディレクトリ下の Readme.pdf ファイルを参照してほしい。

可視化に IDL を使う場合、CANS 用 IDL プロシジャが使えるよう、IDL の探索パスを設定する必要がある。Linux の場合シェル設定ファイル（.cshrc や.bashrc などユーザー依存）で環境変数 IDL_PATH に「（CANS をインストールしたディレクトリ）/cans/idl/」を追加する。よくわからない場合は、IDL に詳しいひとに尋ねてほしい。

1.2.3 準備コンパイル

計算コードをコンパイルする。まずサブルーチン群の「アーカイブファイル」を作る。CANS のルートディレクトリ（cans/）に戻って、以下のように make する。

```
# pwd
/（インストールした元ディレクトリ）/cans/
# cd cans/
# make
...
# ls
....
libcans1d.a libcans2d.a libcans3d.a libcansio.a
```

無事に成功する（MPI コードは除く）と、上記のように libcans*.a というファイルが複数つくられる。

libcans1d.a は、1 次元計算用のルーチン群のオブジェクトファイル（コンパイル済みのサブルーチンプログラム）をまとめたアーカイブである。libcansio.a は、各次元で共通に使うファイル入出力用ルーチン群。

なお上記の方法では通常、MPI 用のルーチン群はまだコンパイルされていない。そこで、必要に応じて同じディレクトリで

```
# pwd
/（インストールした元ディレクトリ）/cans/
# cd cans/
# make mpi FC=mpif90
```

などとして、MPI 用のコンパイラを使って再度同じ作業を繰り返す。

以上の作業で、方程式を解くなどの CANS 基本部分のサブルーチン群（以後「計算エンジン」と呼ぶ）のコンパイルは終了しており、結果がアーカイブファイルにまとめられた。こうしておけば、計算エンジンのルーチンについて再度コンパイルする必要がなくなり、作業効率があがる。また異なった問題を解く際に、共通のエンジン部分のコードをコピーする必要がなくなる。（いっぽう短所もあり、「問題設定」ディレクトリ（次節で説明）に完全な形でのプログラムファイルが存在せず、全体を見通すことが難しくなる。これについては近々対策する予定である。）

1.2.4　プログラムの実行

2 次元問題設定のひとつである md_mhdshktb/ 「磁気流体衝撃波管」（Brio-Wu 問題）で説明する。まず該当のディレクトリに入る。

```
# cd cans2d/md_efr/
# ls
Readme.pdf  Readme.tex  hdcip/  hdmlw/  hdroe/  index.html  mpi-hdmlw/
```

複数ディレクトリがあるのは、それぞれ使っている計算エンジンが異なるためである。ここでは hdmlw/ すなわち改良 Lax-Wendroff 法エンジンを使った場合で説明する。

```
# cd hdmlw/
# ls
Makefile   bnd.f   model.f  moviedt.pro  pldt.pro   pldt1d.pro
batch.pro  main.f  moviedt/  pldt.png      pldt1d.png  rddt.pro
```

各ファイルの説明は次のとおり。重要なものだけ示す。

- Makefile make コマンド用設定ファイル
- main.f メインルーチン
- model.f 初期条件設定ルーチン

- bnd.f 境界条件設定ルーチン
- *.pro 可視化用 IDL プログラム

コンパイル・実行するには make コマンドを使う。

```
# make
# ls
a.out ...
# make run
./a.out
  write    step=       0 time= 0.000E+00 nd =  1
...
  write    step=      49 time= 0.816E-01 nd =  5
  write    step=      61 time= 0.102E+00 nd =  6
  stop     step=      61 time= 0.102E+00
   ### normal stop ###
# ls
Makefile   bnd.f*  main.f*   moviedt/      pldt.png     pr.dac     vx.dac
a.out*     bnd.o   main.o    moviedt.pro*  pldt.pro*    rddt.pro*  vy.dac
az.dac     bx.dac  model.f*  out.txt       pldt1d.png   ro.dac     x.dac
batch.pro  by.dac  model.o   params.txt    pldt1d.pro*  t.dac      y.dac
```

上記のように「normal stop」メッセージで終了すれば、無事計算できたことになる。新しく次のファイル群が生成されるはずである。*.dac が計算結果データ（バイナリ形式）、out.txt が画面出力と同じメッセージ（テキスト形式）、params.txt が計算パラメータ（テキスト形式）。

1.2.5 IDL による可視化

IDL を立ち上げ、計算結果データを読み込み、可視化する。

```
# idl
IDL> .r rddt ; データ読み込み
IDL> .r pldt
 Plot columns & rows ? : 3 2  ; パネル数が聞かれる。この例では「3 行 2 列」を選択
 Variable for color-maps ? (ro,pr,te) : ro ; 表示物理量。この例では「密度」を選択
  start step ? : 0 ; 開始ステップ。この例では「0 番目」すなわち「初期条件から」を選択
```

うまくいけば新しくウィンドウが立ち上がり、上の例では「3 行 2 列に、密度の分布が、初期条件から」表示されるはずである。成功例として同ディレクトリに pldt.png というファイルが用意されているので、一致を確かめてほしい。なお pldt.pro では、指定した物理量以外に、速度が矢印で、磁力線が実線（実際はベクトルポテンシャル A_z の等高線）で示される。

また moviedt.pro を使えば、動画の材料となる画像ファイルを作れる。上と同じように実行すると、あらたに moviedt/というディレクトリが作成され（デフォルトでは作成済み）、その下に連番で画像ファイルが作られる。これをつなぎ合わせれば動画となる。注意 : プレビューの動画は moviedt/以下のファイルを表示しているので、moviedt.pro を実行すると上書きされてしまう。

1.3　計算プログラムの解説

この節では、問題設定 cans1d/md_mhdshktb/hdmlw、磁気流体衝撃波管問題を改良 Lax-Wendroff 法で解く場合を例にとって実際のプログラムについて解説する。ここでは計算法の詳細には踏み込まない。改良 Lax-Wendroff 法についての詳しい説明は、「数値天文学テクニカルマニュアル 2004 年版」の第 1 章に掲載されているのでそちらを参照にしてほしい。

1.3.1　メインルーチン main.f

メインルーチン（ファイル main.f）は、大きく分けて「開始処理部（エピローグ）」「時間発展部」「終了処理部（エピローグ）」の 3 部構成からなる。さらにその前に、Fortran プログラムで必要な「配列定義部」がある。開始処理部では、サブルーチン model を呼び、メッシュ座標や初期条件を設定するのが主な動作である。時間発展部が主たる計算エンジンで、磁気流体方程式を解いて物理変数の時間発展を求める。与えられた終了条件をみたすまでこの部分が繰り返される。そして終了処理部では、終了メッセージを出力してプログラム実行を止める。以下、実際のプログラムにコメントを入れながら解説する。

```
c=======================================================================|
c     array definitions
c=======================================================================|
      implicit double precision (a-h,o-z)
! 倍精度浮動小数点データを基本データ型とする。
      parameter (margin=1)
      parameter (ix=1024+2*margin)
      dimension x(ix),xm(ix),dx(ix),dxm(ix)
! x(*)がメッシュ座標、xm(*)はメッシュ境界の座標、dx(*)がメッシュ幅、dxm(*)はメッシュ
中心間の距離。
! ixはx軸のメッシュ数
! marginは、興味ある領域の外に準備する、計算上必要な袖メッシュ数。計算エンジンに依存して
数が異なる。ここで用いるmlw_mでは、1個。

      dimension ro(ix),pr(ix)
      dimension vx(ix),vy(ix),by(ix)
! 時間変化する基本変数の配列。roは密度、prは圧力、vx、vyは速度、byは磁場
      dimension bx(ix),bxm(ix)
! x軸に沿った1次元MHDでは、磁場のx成分は時間変動しない（divB=0なので)。
! サブルーチンmodel.fで固定値として与えられる。

913   format (1x,' write    ','step=',i8,' time=',e10.3,' nd =',i3)
915   format (1x,' stop     ','step=',i8,' time=',e10.3)
c=======================================================================|
c     prologue
c=======================================================================|
      mcont=0
! mcont=1とすると、過去の計算結果データ（ディレクトリin/以下に置く）を読み込んで継続計
算を実行する。継続計算では、下のtendの値設定に注意すること。前回の計算の最終時刻がtend
より大きいと、継続計算がただちに終了してしまう。
c-----------------------------------------------------------------------|
c     set parameters controling finalization and data-output
      tend=0.1d0
! tendは終了させたい時刻。
      nstop=100000
! nstopは終了させたいステージ数。
! t > tend または ns>nstopが満たされたら計算終了。
      dtout=0.01d0
! dtoutは、データ出力間隔。tの値がdtoutの（ほぼ）整数倍になったら出力。
c     dtout=1.d-10
c     nstop=1
c-----------------------------------------------------------------------|
c  initialize counters
```

```
      merr  = 0
      ns    = 0
      t  = 0.0d0
      tp = 0.0d0
      nd=1
      mwflag=0
! 以上、カウンタの初期化。
! merr はエラーが起きたときゼロ以外の値が代入される。
! ns は計算ステージ数。
! t は時刻。tp は、ひとつ前のステージでの時刻。データ出力タイミングの判断に使う。
! nd は、次に出すデータが何番目か。
! mwflag は、「いま保持している計算結果を出力しました」というフラグ。重複して同じデータを
出力するのを防ぐための仕掛け。
c-------------------------------------------------------------------------|
c  file open for "standart output"
      mf_out=7
      open(mf_out,file='out.txt',status='replace',iostat=merr)
      if (merr.ne.0) then
        merr=10001
        goto 9999
      endif
      close(mf_out)
! 標準出力（コンソール出力）と同じ内容を残すファイル（out.txt）のオープン
! out.txt は出力のたびに開け閉めする。こうすると長時間の計算途中でも中身を確認できる。
c-------------------------------------------------------------------------|
c  file open
      mf_params=9
      call dacdefparam(mf_params,'params.txt')
! 計算パラメータを出力するファイル（params.txt）をオープン
      mf_t =10
      call dacdef0s(mf_t,'t.dac',6)
      mf_ro=20
      call dacdef1s(mf_ro,'ro.dac',6,ix)
      mf_pr=21
      call dacdef1s(mf_pr,'pr.dac',6,ix)
      mf_vx=22
      call dacdef1s(mf_vx,'vx.dac',6,ix)
      mf_vy=23
      call dacdef1s(mf_vy,'vy.dac',6,ix)
      mf_by=24
      call dacdef1s(mf_by,'by.dac',6,ix)
! 計算結果を出力するファイル（t.dac、ro.dac、pr.dac など）をオープン
! 引数「6」は、データ型が倍精度であることを示す。
```

```
      call dacputparami(mf_params,'ix',ix)
      call dacputparami(mf_params,'margin',margin)
! 計算パラメータを出力
c-----------------------------------------------------------------------|
c   setup numerical model (grid, initial conditions, etc.)
      call model(ro,pr,vx,vy,by,bx,bxm,gm,margin,x,ix,mf_params)
! メッシュ座標・初期条件を設定（後で詳述）
      call grdrdy(dx,xm,dxm,x,ix)
! メッシュ座標（x）から、メッシュ幅（dx）やメッシュ境界での座標など（xm、dxm）を計算
      call bnd(margin,ro,pr,vx,vy,by,ix)
! 境界条件の適用
c     floor=1.d-9
c     call chkdav(n_floor,ro,vx,floor,ix)
c     call chkdav(n_floor,pr,vx,floor,ix)
c-----------------------------------------------------------------------|
c  read-data
      ndi=1000
      if (mcont.eq.1) then
      mfi_t=60
      call dacopnr0s(mfi_t,'in/t.dac',mtype,nx0)
      mfi_ro=70
      call dacopnr1s(mfi_ro,'in/ro.dac',mtype,ix0,nx0)
      mfi_pr=71
      call dacopnr1s(mfi_pr,'in/pr.dac',mtype,ix0,nx0)
      mfi_vx=72
      call dacopnr1s(mfi_vx,'in/vx.dac',mtype,ix0,nx0)
      mfi_vy=73
      call dacopnr1s(mfi_vy,'in/vy.dac',mtype,ix0,nx0)
      mfi_by=74
      call dacopnr1s(mfi_by,'in/by.dac',mtype,ix0,nx0)
      do n=1,ndi
        read(mfi_t,end=9900) t
        read(mfi_ro) ro
        read(mfi_pr) pr
        read(mfi_vx) vx
        read(mfi_vy) vy
        read(mfi_by) by
      enddo
9900  continue
! 以上、継続計算（mcont=1）のとき、過去の計算結果（ディレクトリ in/の下に置く）を読み込
む。
      endif
```

```
c-----------------------------------------------------------------------|
c     data output
      mf_x=11
      call dacdef1d(mf_x,'x.dac',6,ix)
      write(mf_x) x
      close(mf_x)
      mf_bx=12
      call dacdef1d(mf_bx,'bx.dac',6,ix)
      write(mf_bx) bx
      close(mf_bx)
      call dacputparamd(mf_params,'gm',gm)
      write(mf_t) t
      write(mf_ro) ro
      write(mf_pr) pr
      write(mf_vx) vx
      write(mf_vy) vy
      write(mf_by) by
      write(6,913) ns,t,nd
      open(mf_out,file='out.txt',status='old',form='formatted'
     &    ,position='append')
            write(mf_out,913) ns,t,nd
      close(mf_out)
      nd=nd+1
! 初期条件を出力する
c=======================================================================|
c     time integration
c=======================================================================|
1000  continue
         ns = ns+1
         mwflag=0
c-----------------------------------------------------------------------|
c     obtain time spacing

         safety=1.0d0
         dtmin=1.d-10
         call cfl_m(dt,safety,dtmin,merr,gm,bx,ro,pr,vx,vy,by,dx,ix)
         if (merr.ne.0) goto 9999
! CFL 条件にもとづいて、時間刻み幅 dt を各ステージで決定する。
! safety は、CFL 条件の臨界値から実際の dt を求めるときの安全係数。通常 1 以下の値。
! dtmin は、dt の値に対する下限値。dt<dtmin になるとエラーとみなして計算を止める。
         tp = t
         t = t+dt
! 時刻の更新
```

```
c-------------------------------------------------------------------------|
c     solve hydrodynamic equations
         qav=1.d0
         call mlw_m(ro,pr,vx,vy,by,bx,bxm,dt,qav,gm,dx,dxm,ix)
! 主要エンジン部。ここでは改良 Lax-Wendroff（MLW）（と Lapidus 人工粘性）で MHD 方程式を解
く。
! qav は人工粘性の強さ。
         call bnd(margin,ro,pr,vx,vy,by,ix)
! 境界条件の適用
c        floor=1.d-9
c        call chkdav(n_floor,ro,vx,floor,ix)
! 計算結果をチェックして、floor の値より ro が小さいメッシュでは値を floor と置き換え、vx
をゼロとする。この問題設定では不要。
c        call chkdav(n_floor,pr,vx,floor,ix)
! 同様。ただし pr をチェック。
c-------------------------------------------------------------------------|
c     data output
         mw=0
         nt1=int(tp/dtout)
         nt2=int(t/dtout)
         if (nt1.lt.nt2) mw=1
! 「dtout の整数倍」の判断をする部分
         if (mw.ne.0) then
           write(mf_t) t
           write(mf_ro) ro
           write(mf_pr) pr
           write(mf_vx) vx
           write(mf_vy) vy
           write(mf_by) by
           write(6,913) ns,t,nd
           open(mf_out,file='out.txt',status='old',form='formatted'
     &          ,position='append')
             write(mf_out,913) ns,t,nd
           close(mf_out)
            nd=nd+1
            mwflag=1
         endif
! データ出力
c-------------------------------------------------------------------------|
c     loop test
      if (ns .lt. nstop .and. t .lt. tend) goto 1000
! 主要計算部を繰り返すかどうかの判断文
! 「t<tend かつ ns<nstop」であれば繰り返す
```

```
c=======================================================================|
c     epilogue
c=======================================================================|
9999  continue
c-----------------------------------------------------------------------|
c  data output
      if (mwflag.eq.0) then
           write(mf_t) t
           write(mf_ro) ro
           write(mf_pr) pr
           write(mf_vx) vx
           write(mf_vy) vy
           write(mf_by) by
           write(6,913) ns,t,nd
           open(mf_out,file='out.txt',status='old',form='formatted'
     &          ,position='append')
             write(mf_out,913) ns,t,nd
           close(mf_out)
      endif
! データ出力
c-----------------------------------------------------------------------|
c  ending message
      write(6,915) ns,t
      if (merr.eq.0) then
        write(6,*) '  ### normal stop ###'
      else
        write(6,*) '  ### abnormal stop ###'
        write(6,*) '  merr = ',merr
      endif
      open(mf_out,file='out.txt',status='old',form='formatted'
     &    ,position='append')
      write(mf_out,915) ns,t
      if (merr.eq.0) then
        write(mf_out,*) '  ### normal stop ###'
      else
        write(mf_out,*) '  ### abnormal stop ###'
        write(mf_out,*) '  merr = ',merr
      endif
      close(mf_out)
! 終了メッセージ
      stop
! プログラム終了
      end
```

1.3.2 サブルーチン model.f：メッシュ座標・初期条件設定

サブルーチン model.f では、計算に必要なパラメータ（比熱比γなど）や、メッシュ座標（配列 x(ix)）、初期条件を設定する。

```
c=======================================================================|
      subroutine model(ro,pr,vx,vy,by,bx,bxm,gm,margin,x,ix
     &     ,mf_params)
c=======================================================================|
      implicit double precision (a-h,o-z)
c-----------------------------------------------------------------------|
      dimension dxm(ix),x(ix)
      dimension ro(ix),pr(ix),vx(ix)
      dimension vy(ix),by(ix)
      dimension bx(ix),bxm(ix)

c-----------------------------------------------------------------------|
c   parameters
c-----------------------------------------------------------------------|
      pi = acos(-1.0d0)
! pi は、円周率。逆三角関数を使って値を出している。
      gm=2.d0
! gm は、比熱比

c-----------------------------------------------------------------------|
c   grid
c-----------------------------------------------------------------------|
      dx0=1.d0/real(ix-margin*2)
! この問題設定では、一様幅のメッシュを配置するので、計算領域サイズ（=1 としている）とメッシュ数（ix）とからメッシュ幅を求める。袖メッシュの分をだけ引くのを忘れずに。

      do i=1,ix
         dxm(i)=dx0
      enddo

      izero=ix/2
        x(izero)=-dxm(izero)/2
! x=0 となる座標を、ix/2 番目と（ix/2+1）番目とのメッシュ境界に置いた。こうすればメッシュ数が偶数になり、dx0 がキリのよい数値になる。
```

```
        do i=izero+1,ix
           x(i) = x(i-1)+dxm(i-1)
        enddo
        do i=izero-1,1,-1
           x(i) = x(i+1)-dxm(i)
        enddo
! メッシュ座標値の設定
c-----------------------------------------------------------------------|
c     initial condition
c-----------------------------------------------------------------------|
      b0=sqrt(4.d0*pi)

      do i=1,ix
         if (x(i).le.0) then
           ro(i)  = 1.d0
           pr(i)  = 1.d0
           by(i)  = b0
         else
           ro(i)  = 0.125d0
           pr(i)  = 0.1d0
           by(i)  = -b0
         endif
         vx(i) = 0.0d0
         vy(i) = 0.0d0
         bx(i) = b0*0.75d0
         bxm(i) = b0*0.75d0
      enddo
! 初期条件を設定。基本 Brio & Wu (1988) にならった。
! x 軸に沿った 1 次元 MHD では、磁場の x 成分（bx(*)、bxm(*)）は時間変動しない（divB=0 なので）。
! サブルーチン model.f で固定値として与えられる。

c-----------------------------------------------------------------------|
c     write parameters to file
c-----------------------------------------------------------------------|



      return
      end
```

1.3.3 サブルーチン bnd.f : 境界条件

サブルーチン bnd.f では、境界条件を適用する。この問題設定 md_mhdshktb では、対称境界条件を与えている。

```
c=======================================================================|
      subroutine bnd(margin,ro,pr,vx,vy,by,ix)
c=======================================================================|
c     apply boundary condition
c-----------------------------------------------------------------------|
      implicit double precision (a-h,o-z)
      dimension ro(ix),pr(ix),vx(ix),vy(ix),by(ix)
c-----------------------------------------------------------------------|
      call bdsppx(0,margin,ro,ix)
      call bdsppx(0,margin,pr,ix)
      call bdspnx(0,margin,vx,ix)
      call bdsppx(0,margin,vy,ix)
      call bdsppx(0,margin,by,ix)
! 「bd（境界）s（対称）p(メッシュ境界）p（正対称）x（x 座標)」あるいは
! 「bd（境界）s（対称）p(メッシュ境界）m（反対称）x（x 座標)」の意味。
! 最初の引数「0（1)」は、x 座標の負（正）サイドの境界という意味。

      call bdsppx(1,margin,ro,ix)
      call bdsppx(1,margin,pr,ix)
      call bdspnx(1,margin,vx,ix)
      call bdsppx(1,margin,vy,ix)
      call bdsppx(1,margin,by,ix)



      return
      end
```

第2章　CANSホームページのJavaScriptムービー解説

福田尚也 (岡山理科大学)

2.1　用意すべき画像ファイル

シミュレーションの結果で用意すべきファイルは、連番のファイル名のついた画像ファイルである。CANS の基本課題 Movie の Web ページでは、000000.png から 000XXX.png までの画像ファイルを用意し、1 枚ずつ画像を表示することでコントロールを行なっている。

連番ファイルのコントロールは以下のようにしている。7 桁の数値に拡張子 4 文字を加えた文字列配列 teststr[i] から、先頭の数値 1 文字を取り除く substring をとることで、6 桁の数値と拡張子 4 文字の文字列配列を作成する。画像形式は PNG からの変更も可能である。

```
//ムービーファイルの設定
for(i=0;i<movKazu;i++){
  movimg[i] = new Image()
  teststr[i]=(i*stepNo+1000000)+'.png'
  movimg[i].src = teststr[i].substring(1,11)
```

連番画像の入ったフォルダーに、CANS のホームページから HTML ファイルのソース
(http://www.astro.phys.s.chiba-u.ac.jp/netlab/cans/cans1d/movie/md_shktb/index.html)
をダウンロードすれば、JavaScript ムービーにすることが可能である。アイコンの画像ファイル 16 種類もダウンロードし、icon フォルダを並列に作成してコピーしておくと良い。

2.2　設定

基本的な設定としては、画像枚数を movKazu、開始画像の番号を imgNo、連番の刻みを stepNo としており、作成した画像枚数や刻みに応じて変更できる。

```
//変数の初期設定
var movKazu = 101 //ムービーに使う画像枚数
var imgNo   = 0  //0 番目の画像からムービーをスタートする
var timeID  = 0  //タイマー
var movSpeed     // x/1000 秒のスピードで動かす
var stepNo  = 1
```

2.3 コントロール

ムービーのコントロールは、左から順に、最初の表示、巻き戻し、コマ戻し、停止、再生、コマ送り、早送り、最後の表示の 8 通りの処理を行なえるように設計した。処理は function で記述される。

処理番号	動作	function	初期アイコン	起動時アイコン
0	停止	B01ClickST(bt)	BSTOF.GIF	BSTON.GIF
1	再生	B01ClickPL(bt), animation1()	BPLOF.GIF	BPLON.GIF
2	早送り	B01ClickFW(bt), animation1()	BFWOF.GIF	BFWON.GIF
3	巻き戻し	B01ClickRW(bt), animation2()	BRWOF.GIF	BRWON.GIF
4	最後へ	B01ClickED(bt), animation3()	BEDOF.GIF	BEDON.GIF
5	最初へ	B01ClickFD(bt), animation4()	BFDOF.GIF	BFDON.GIF
6	コマ送り	B01ClickFS(bt)	B1OF.GIF	B1ON.GIF
7	コマ戻し	B01ClickRS(bt)	B2OF.GIF	B2ON.GIF

2.4　ソースコード

JavaScript のプログラムは HTML ファイルの head 部に記述される。画面表示は body 部に記述される。

2.4.1　JavaScript プログラム

```
<script language="JavaScript">

//変数の初期設定
var movKazu = 101 //ムービーに使う画像枚数
var imgNo   = 0  //0 番目の画像からムービーをスタートする
var timeID  = 0  //タイマー
var movSpeed     // x/1000 秒のスピードで動かす
var stepNo  = 1

//ボタン処理ここから----------------------
/************************************
  ボタン用イメージの事前読み込み
************************************/
// 0：停止
// 1：再生
// 2：早送り
// 3：巻き戻し
// 4：最後へ
// 5：最初へ
if(document.images)
{
   //クリックされた時のボタン (カラーボタン) を読み込む
  btClickimg = new Array()
  btClickimg[0] = new Image();btClickimg[0].src ="../icon/BSTON.GIF";
  btClickimg[1] = new Image();btClickimg[1].src ="../icon/BPLON.GIF";
  btClickimg[2] = new Image();btClickimg[2].src ="../icon/BFWON.GIF";
  btClickimg[3] = new Image();btClickimg[3].src ="../icon/BRWON.GIF";
  btClickimg[4] = new Image();btClickimg[4].src ="../icon/BFINON.GIF";
  btClickimg[5] = new Image();btClickimg[5].src ="../icon/BINION.GIF";
  btClickimg[6] = new Image();btClickimg[6].src ="../icon/B1ON.GIF";
  btClickimg[7] = new Image();btClickimg[7].src ="../icon/B2ON.GIF";

   //マウスが触った時のボタン (グレーボタン) を読み込む
  btOverimg = new Array();
```

```
  btOverimg[0] = new Image();btOverimg[0].src ="../icon/BSTOF.GIF";
  btOverimg[1] = new Image();btOverimg[1].src ="../icon/BPLOF.GIF";
  btOverimg[2] = new Image();btOverimg[2].src ="../icon/BFWOF.GIF";
  btOverimg[3] = new Image();btOverimg[3].src ="../icon/BRWOF.GIF";
  btOverimg[4] = new Image();btOverimg[4].src ="../icon/BFINOF.GIF";
  btOverimg[5] = new Image();btOverimg[5].src ="../icon/BINIOF.GIF";
  btOverimg[6] = new Image();btOverimg[6].src ="../icon/B1OF.GIF";
  btOverimg[7] = new Image();btOverimg[7].src ="../icon/B2OF.GIF";

   //マウスが離れた時のボタン (グレーボタン) を読み込む
  btOutimg = new Array();
  btOutimg[0] = new Image();btOutimg[0].src ="../icon/BSTOF.GIF"
  btOutimg[1] = new Image();btOutimg[1].src ="../icon/BPLOF.GIF";
  btOutimg[2] = new Image();btOutimg[2].src ="../icon/BFWOF.GIF";
  btOutimg[3] = new Image();btOutimg[3].src ="../icon/BRWOF.GIF";
  btOutimg[4] = new Image();btOutimg[4].src ="../icon/BFINOF.GIF";
  btOutimg[5] = new Image();btOutimg[5].src ="../icon/BINIOF.GIF";
  btOutimg[6] = new Image();btOutimg[6].src ="../icon/B1OF.GIF";
  btOutimg[7] = new Image();btOutimg[7].src ="../icon/B2OF.GIF";
}

/*****************************************
  ボタンを Over/Out/Click 時の処理
*****************************************/

fl=0;//ONしている画像 NO
offsetB=5;//1 個目のボタンの index 番号

   //ボタンの初期化 (グレーボタンにする)
function B01Ini(bt){
  if(document.images)
    document.images['BTN'+bt].src= btOutimg[bt].src
}

   //ストップボタンをクリックされた時 (onClick) の処理
function B01ClickST(bt){
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(カラーボタンにする)
    fl=bt;
```

```
    clearTimeout(timeID);//(ムービーを止める)
  }
}

   //スタートボタンをクリックした時 (onClick) の処理
function B01ClickPL(bt){
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(オレンジボタンにする)
    fl=bt;
    movSpeed=120;
    animation1();
  }
}

   //早送りボタンをクリックした時 (onClick) の処理
function B01ClickFW(bt){
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(オレンジボタンにする)
    fl=bt;
    movSpeed=10;
    animation1();
  }
}

   //巻きもどしボタンをクリックした時 (onClick) の処理
function B01ClickRW(bt){
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(紫ボタンにする)
    fl=bt;
    movSpeed=30;//巻きもどしスピード
    if(imgNo==0)imgNo=movKazu;//巻きもどしが終了してれば最後に送る
    animation2();
  }
}

   //エンドボタンをクリックした時 (onClick) の処理
function B01ClickED(bt){
```

```
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(カラーボタンにする)
    fl=bt;
    imgNo=movKazu-1;
    animation3();
  }
}

   //ファーストボタンをクリックされた時 (onClick) の処理
function B01ClickFD(bt){
  if(document.images){
    B01Ini(fl);//ボタンの初期化
    document.images['BTN'+bt].src= btClickimg[bt].src;//(カラーボタンにする)
    fl=bt;
    imgNo=1;
    animation4();
  }
}

   //コマ送りをクリックした時 (onClick) の処理
function B01ClickFS(bt){
  if(document.images){
    B01Ini(fl);
    document.images['BTN'+bt].src= btClickimg[bt].src;
    fl=bt;
    movSpeed=10;
    imgNo++;
    if(imgNo>movKazu-1)imgNo=movKazu-1;
    document.images['imgA'].src = movimg[imgNo].src;
    clearTimeout(timeID);
  }
}

   //コマ戻しをクリックした時 (onClick) の処理
function B01ClickRS(bt){
  if(document.images){
    B01Ini(fl);
    document.images['BTN'+bt].src= btClickimg[bt].src;
    fl=bt;
```

```
    movSpeed=10;
    imgNo--;
    if(imgNo<0)imgNo=0
    document.images['imgA'].src = movimg[imgNo].src;
    clearTimeout(timeID);
  }
}
//ボタン処理ここまで----------------------

//ムービー処理ここから--------------------

//イメージの事前読み込み
if(document.images){

  movimg = new Array()
  teststr = new Array()

  //ムービーファイルの設定
  for(i=0;i<movKazu;i++){
    movimg[i] = new Image()
    teststr[i]=(i*stepNo+1000000)+'.png'
    movimg[i].src = teststr[i].substring(1,11)
  }
}

//アニメーション
function animation1(){
  if( imgNo < movKazu ){
    document.images['imgA'].src = movimg[imgNo].src
    timeID = setTimeout("animation1()",movSpeed)
    imgNo++
  } else {
    clearTimeout(timeID)
  }
}

//巻きもどしアニメーション
function animation2(){
  if( imgNo > 0 ){
    imgNo--
```

```
    document.images['imgA'].src = movimg[imgNo].src
    timeID = setTimeout("animation2()",movSpeed)
  } else {
    clearTimeout(timeID)
  }
}

//最終ステージを表示
function animation3(){
  if( imgNo < movKazu ){
    document.images['imgA'].src = movimg[imgNo].src
    timeID = setTimeout("animation3()",movSpeed)
    imgNo++
  } else {
    clearTimeout(timeID)
    imgNo=movKazu-1
  }
}

//初期ステージを表示
function animation4(){
  if( imgNo > 0 ){
    imgNo--
    document.images['imgA'].src = movimg[imgNo].src
    timeID = setTimeout("animation4()",movSpeed)
  } else {
    clearTimeout(timeID)
    imgNo=0
  }
}
//ムービー処理ここまで----------------------
</script>
```

2.4.2　HTML 部分

初期に表示する画像は”000000.png”としている。画像サイズは width=640、height=512 とした。

```
<body BGCOLOR="#000000">

<CENTER>
<IMG src="title.gif" border="0"><BR>
<img src="000000.png" name="imgA" width=640 height=512 border='0'><br>
</CENTER>

<CENTER>
<a href="javascript:B01ClickFD(5)" name='BTST'>
<img src='../icon/BINIOF.GIF' NAME='BTN5' width="36" height="18" border='0'></a>
<a href="javascript:B01ClickRW(3)" name='BTST'>
<img src='../icon/BRWOF.GIF' name="BTN3" width="36" height="18" border='0'></a>
<a href="javascript:B01ClickRS(7)" name='BTST'>
<img src='../icon/B2OF.GIF' NAME='BTN7' width="36" height="18" border='0'></a>
<a href="javascript:B01ClickST(0)">
<img src='../icon/BSTOF.GIF' NAME='BTN0' width="36" height="24" border='0'></a>
<a href="javascript:B01ClickPL(1)" name='BTST'>
<img src='../icon/BPLOF.GIF' NAME='BTN1' width="36" height="24" border='0'></a>
<a href="javascript:B01ClickFS(6)" name='BTST'>
<img src='../icon/B1OF.GIF' NAME='BTN6' width="36" height="18" border='0'></a>
<a href="javascript:B01ClickFW(2)" name='BTST'>
<img src='../icon/BFWOF.GIF' NAME='BTN2' width="36" height="18" border='0'></a>
<a href="javascript:B01ClickED(4)" name='BTST'>
<img src='../icon/BFINOF.GIF' NAME='BTN4' width="36" height="18" border='0'></a>
</CENTER>

</body>
```

参考文献

高橋登史郎著「だれでもカンタン JavaScript サンプル集」秀和システム、p163–p184
B01ClickRW(3)

第3章　モーメント法を用いた相対論的輻射流体の数値解法

高橋博之 (CfCA, 国立天文台)

近年のスーパーコンピュータの発達と数値計算コードの高精度化に伴い、より高解像度・長時間の磁気流体計算を行う事が可能になってきた。その一方で、磁気流体に自己重力や、特殊・一般相対性理論、電気抵抗、熱伝導、粘性、輻射輸送など、より多くの物理を含めた現実的な計算も行われている。この章では相対論的 (磁気) 流体に輻射の効果を無矛盾に取り入れた相対論的輻射流体数値計算の方法を紹介する。ここで言う相対論は特殊相対論であり一般相対論的効果は考えない。そのため空間メトリックは平坦であると仮定する (e.g., diag $g_{\mu\nu} = (-1, 1, 1, 1)$、添字のギリシャ文字は時空4次元を表す)。一般相対論への拡張は (9) や (15) 等を参照してほしい。

詳細は後で述べるが、輻射場にモーメント法と呼ばれる簡単化した方程式を扱う。この章では数値計算方法に主題をおくため、輻射場のモーメント方程式の導出には途中式をかなり省く。詳しい導出には (8) を参照してほしい。また、(6) はブラックホール降着円盤の教科書であるが、その巻末にモーメント法を用いた方程式の導出が簡便に書かれており、非常に有用である。その他、輻射に関する有用な教科書として (11)、(1)、(10) を挙げておく。

3.1　輻射場の支配方程式

3.1.1　輻射のモーメント式の導入

プラズマで第一原理計算を行う際もそうであるように、輻射を扱うためには光子のボルツマン方程式を解く必要がある。これは輻射輸送方程式と呼ばれており、以下のように表される。

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + (\boldsymbol{l}\cdot\nabla)I_\nu = \eta_\nu - \rho(\kappa_\nu + \sigma_\nu)I_\nu. \tag{3.1.1}$$

I_ν は輻射強度 [erg s^{-1} cm^{-2} ster^{-1} Hz^{-1}] であり、次元は異なるが光子の分布関数に対応する物理量である。ベクトル $\boldsymbol{l}$ は光の伝搬ベクトルであり、光子の4元ベクトル k^μ と $k^\mu = \nu(1, \boldsymbol{l})$ という関係がある。ただし ν は光の振動数である。η_ν は放射率 (emissivity, [erg cm^{-3} ster^{-1} Hz^{-1}]) と呼ばれる量で、ガスが単位体積、単位立体角あたりに、ある振動数で放射するエネルギーを表している。κ_ν と σ_ν はそれぞれ吸収係数と散乱係数 [cm^2 g^{-1}] で、ρ はガスの質量密度である。この式は観測者系でも共動座標系でも成り立つ、ローレンツ不変な式である。

式 (3.1.1) を眺めてみると、左辺は輻射強度の時間微分と空間微分が含まれており、これらは輻射の移流を表していることがわかる。粒子のボルツマン方程式と異なり、輻射輸送方程式には加速度に比例す

る項 ($\propto \dot{p}$) が含まれていない。これは光は真っ直ぐに進む (null) ため[1]、光の 4 元運動量 $p^\mu = dx^\mu/d\lambda$ (λ はアフィン係数) の微分が $\dot{p}^\mu = d^2x^\mu/d\lambda^2 = 0$ となるためである。次に式 (3.1.1) の右辺を眺めてみる。第 1 項の j_ν に比例する項はガスが自発的に放射した分だけ輻射強度が増えることを表しており、第 2 項は吸収や散乱によって光とガスが相互作用する事を表している。

輻射輸送方程式はボルツマン方程式と同様に、t, x^i, p^i の 7 つの独立変数を持つ (添字のラテン文字は空間 3 次元を表す)。従って (3.1.1) を数値的に解く場合、空間、運動量空間を N グリッドで分割すると必要なメモリ量は N^6 に比例するため、実際に数値計算を行う事は簡単ではない。そこでここでは仮定をして独立変数を減らすことをを考える。非相対論的現象を扱う場合、流体速度は光速に比べて十分に遅いため、輻射場は定常であると仮定して時間依存性を落とすことができる。しかし、相対論的現象では流体速度 (や、音波の伝搬速度) が光速と同程度になるため、時間依存性を落とす事はできない[2]。もし、星のように空間の対称性がよい場合には空間を半径方向の 1 次元に落とすことが可能である。空間の構造が複雑で次元を落とす事が出来ない場合には運動量空間の自由度を減らすことになる。運動量空間の次元を落とすには、輻射強度を運動量空間で積分すればよい。そこで以下のように放射強度を (運動量空間の) 立体角で積分した量を定義する:

$$E_{r\nu}(t, x^k, \nu) = \frac{1}{c}\int d\Omega I_\nu, \tag{3.1.2}$$

$$F^i_{r\nu}(t, x^k, \nu) = \int d\Omega I_\nu l^i, \tag{3.1.3}$$

$$P^{ij}_{r\nu}(t, x^k, \nu) = \frac{1}{c}\int d\Omega I_\nu l^i l^j. \tag{3.1.4}$$

上から順に単位振動数あたりの輻射エネルギー密度、輻射フラックス、輻射ストレスで、それぞれ 0, 1, 2 次モーメントとも呼ばれる。この様に l^i をかけて立体角で積分することを**モーメントをとる**と言う。原理的にはより高次のモーメントを取る事も出来るが、ここでは 2 次までにしておく。輻射ストレスは l^i, l^j について明らかに対称なので、独立な成分は 6 つであることに注意する。

この時点で独立変数は 7 つから 5 つまで減った。さらに式 (3.1.2)-(3.1.4) を振動数で積分する。

$$E_r(t, x^k) = \int d\nu E_r = \frac{1}{c}\int d\Omega d\nu I_\nu, \tag{3.1.5}$$

$$F^i_r(t, x^k) = \int d\nu F^i_r = \int d\Omega d\nu I_\nu l^i, \tag{3.1.6}$$

$$P^{ij}_r(t, x^k) = \int d\nu P^{ij}_r = \frac{1}{c}\int d\Omega d\nu I_\nu l^i l^j. \tag{3.1.7}$$

それぞれ輻射エネルギー密度、輻射フラックス、輻射ストレスと呼ぶ。これらのモーメントの独立変数は時間と空間のみであり、ガスを記述する物理量（密度や圧力等）と同様に N^3 程度のメモリ量で計算

[1] 光の経路が曲がるような一般相対論果を考える際にはこの式は正しくないが、ここではそのような状況は考えない事にする。

[2] 実はこれは利点だったりもする。定常を仮定すると問題は境界値問題になるため、大規模な行列反転が必要になる。これは数値計算上、並列化の観点から好ましくない。時間微分を残すと、適切なクーラン条件の下に初期値問題を解くことになるため、比較的並列化が容易となる。また、そもそも時間微分を落とした所でメモリの節約にはならない。それどころか超並列が難しくなるため、大規模計算を行うためにはノードあたりのメモリサイズが大きくないといけないという制限までかかってくる。

することができる。つまり、**モーメント量を扱う事によって光のエネルギースペクトルは犠牲になるが、空間構造が複雑な系でも輻射とガスの力学進化を同時に (無矛盾に) 追うことが出来る**のである。

では次にこれらモーメントが満たすべき方程式を輻射輸送方程式 (3.1.1) から導出する。輻射エネルギー密度の発展方程式は式 (3.1.1) を運動量空間 (立体角＋振動数) で積分すれば得られる。同様に輻射フラックスの発展方程式は式 (3.1.1) に l^i をかけて運動量空間で積分すればよい。その結果得られる式は以下のようになる[3]

$$\frac{\partial E_r}{\partial t} + \frac{\partial F_r^j}{\partial x^j} = \rho\gamma\kappa_0(4\pi \mathrm{B} - cE_{r0}) - \rho\gamma(\kappa_0+\sigma_0)(\beta_j F_{r0}^j), \tag{3.1.8}$$

$$\frac{1}{c^2}\frac{\partial F_r^i}{\partial t} + \frac{\partial P_r^{ij}}{\partial x^j} = \rho\gamma\kappa_0\frac{\beta^i}{c}(4\pi \mathrm{B} - cE_{r0}) - \rho(\kappa_0+\sigma_0)\frac{\gamma^2(\beta_j F_{r0}^j)}{(\gamma+1)c}\beta^i - \rho(\kappa_0+\sigma_0)\frac{F_{r0}^i}{c}. \tag{3.1.9}$$

ここで $\beta^i = v^i/c$は光速で規格化された流体のバルク速度であり、$\gamma = (1-\boldsymbol{\beta}^2)^{-1/2}$ はローレンツ因子である。式 (3.1.8)-(3.1.9) の右辺に $(\beta_i F_r^i)$ のような項があるがこれは縮約を取っている $(\beta_i F_r^i = \sum_{i=1}^3 \beta_i F_r^i)$。下付添字の'0' は特に明示しない限り共動座標系で見た物理量を表し、添字の無いものは観測者系で見た物理量を表す。ガス密度 ρ については添字'0' は付いていないが、共動座標系で見た固有密度を表す (同様にガス圧やガス温度も慣習的に添字'0' をつけないで共動座標系の物理量を表す)。

ここで式 (3.1.8)-(3.1.9) を得る際に以下の関係式を用いて η_0 を消した:

$$\eta_0 = \kappa_0 \mathrm{B}, \tag{3.1.10}$$

B は黒体輻射強度で Stefan-Boltzmann 係数 σ_{SB} を用いて

$$\mathrm{B} = \frac{\sigma_{\mathrm{SB}} T_g^4}{\pi}, \tag{3.1.11}$$

と表される。T_g はガス温度である。式 (3.1.10) は Kirchhoff-Planck 関係式と呼ばれ、局所熱力学平衡 (Local Thermodynamic Equilibrium) にある場合には正しい。しかし、熱力学的平衡が成り立つかどうかは自明ではなく、光学的に薄い状況では LTE が成り立ちにくい。非局所熱力学平衡の物理を正しく理解するためには式 (3.1.1) に戻らなければならない。

式 (3.1.8) を見ると、左辺は輻射エネルギー密度の時間微分と輻射フラックスの空間微分が含まれており、これは輻射エネルギーがフラックスによって運ばれていることを示している。一方、右辺の第 1 項はガスが放射することによって輻射エネルギーが増える項、第 2 項は輻射がガスと吸収や放射を通してエネルギーをやり取りする効果を表している。第 3 項は輻射フラックスに比例しており、これは輻射がガスに仕事をする事によってエネルギーが増減する効果を表している。この様にモーメントを取って輻射場の運動量空間の情報を減らしたにも関わらず、輻射輸送方程式と同様の性質をきちんと持っている事がわかる。

ところで式 (3.1.8)-(3.1.9) をよく見てみると、左辺の物理量は観測者系、右辺の物理量は共動座標系で定義されていることがわかる。左辺については、通常我々は自分たちが見ている系で定義された時間軸と座標軸で物理を調べたいので、それに対応して輻射エネルギー密度とフラックスも観測者系で表す。

[3] ここで式 (3.1.8)-(3.1.9) の吸収係数と散乱係数の添字 ν が消えていることに注意する。通常 $\kappa_{0\nu}$ や $\sigma_{0\nu}$ は振動数や散乱角に依存するため、式 (3.1.1) を運動量空間で積分する際に $\int d\nu d\Omega(\kappa_{0\nu}+\sigma_{0\nu})I_\nu$ の扱いが問題になる。式 (3.1.8)-(3.1.9) を導く際には散乱に対して振動数が変化しないという仮定をしたり、吸収・散乱係数について振動数空間である種の平均操作 (Rossland mean) を行っている。詳しくは (8) 等、参考文献を参照してほしい。

一方、吸収や散乱といった輻射とガスの相互作用は局所的な (粒子スケールの) 現象であるため、バルク速度を除いた共動座標系で表現した方が式は見通しが良くなる。

これら観測者系と共動座標系の物理量はそれぞれローレンツ変換で結ばれる。具体的には E_r, F_r^i, P_r^{ij} からなる輻射場のエネルギー運動量テンソル $T_r^{\mu\nu}$ をローレンツ変換マトリックス Λ_ν^μ を用いて変換する:

$$T_r^{\mu\nu} = \begin{pmatrix} E_r & F_r^i/c \\ {}^T F_r^i/c & P_r^{ij} \end{pmatrix}, \tag{3.1.12}$$

$$T_{r0}^{\mu\nu} = \Lambda_\alpha^\mu \Lambda_\beta^\nu T_r^{\alpha\beta}, \tag{3.1.13}$$

$$\Lambda_\nu^\mu(\beta) = \begin{pmatrix} \gamma & -\gamma\boldsymbol{\beta} \\ -\gamma\boldsymbol{\beta} & \boldsymbol{\delta} + (\gamma-1)\beta^{-2}\boldsymbol{\beta}\boldsymbol{\beta} \end{pmatrix}, \tag{3.1.14}$$

ここで $\boldsymbol{\delta}$ はクロネッカーデルタである。この変換則を用いると E_r, F_r^i, P_r^{ij} と $E_{r0}, F_{r0}^i, P_{r0}^{ij}$ の間に以下の関係が得られる。

$$E_{r0} = \gamma^2 E_r - 2\gamma \frac{u_k F_r^k}{c^2} + \frac{u_j u_k P_r^{jk}}{c^2}, \tag{3.1.15}$$

$$F_{r0}^i = \gamma \left\{ F_r^i + \left[\frac{2\gamma+1}{\gamma+1} \frac{u_j F_r^j}{c^2} - \gamma E_r - \frac{u_j u_k P_r^{jk}}{(\gamma+1)c^2} \right] v^i - v_k P^{ik} \right\}, \tag{3.1.16}$$

$$\begin{aligned} P_{r0}^{ij} &= P_r^{ij} + \frac{u^i u_k P_r^{jk} + u^j u_k P_r^{ik}}{(\gamma+1)c^2} + \frac{u^i u^j u_k u_l P_r^{kl}}{(\gamma+1)^2 c^4} \\ &+ \frac{u^i u^j}{c^2} E_r - \frac{u^i F_r^j + u^j F_r^i}{c^2} - \frac{2u^i u^j u_k F_r^k}{(\gamma+1)c^4}. \end{aligned} \tag{3.1.17}$$

ここで $u^\mu = (\gamma c, \gamma v^i)$ は 4 元速度である。

式 (3.1.15)-(3.1.17) を式 (3.1.8)-(3.1.9) に代入すると、観測者系で定義された輻射場で表されたモーメント方程式が得られる:

$$\frac{1}{c}\frac{\partial E_r}{\partial t} + \frac{1}{c}\frac{\partial F_r^j}{\partial x^j} = -G^0, \tag{3.1.18}$$

$$\frac{1}{c^2}\frac{\partial F_r^i}{\partial t} + \frac{\partial P_r^{ij}}{\partial x^j} = -G^i, \tag{3.1.19}$$

$$-G^0 = \frac{\rho\gamma\kappa_0}{c}\left(4\pi \mathrm{B} - cE_r + \beta_j F_r^j\right) + \frac{\rho\gamma\sigma_0}{c}\left[\frac{u^2 E_r}{c} + \frac{u_j u_k P_r^{jk}}{c} - \left(2\gamma^2 - 1\right)\beta_j F_r^j\right], \tag{3.1.20}$$

$$\begin{aligned} -G^i &= \frac{\rho\gamma\kappa_0}{c}\left(4\pi \mathrm{B}\beta^i - F_r^i + v_k P_r^{ik}\right) \\ &- \frac{\rho\gamma\sigma_0}{c}\left[F_r^i - \gamma E_r u^i - v_k P_r^{ik} + v^i\left(\frac{2\gamma u_k F_r^k}{c^2} - \frac{u_j u_k P_r^{jk}}{c^2}\right)\right]. \end{aligned} \tag{3.1.21}$$

G^μ は輻射 4 元力密度とよばれる。式 (3.1.18)-(3.1.19) は輻射のエネルギー運動量テンソルを用いて $T_{r,\nu}^{\mu\nu} = -G^\mu$ とも書ける。

式 (3.1.18) と (3.1.19) を見ると輻射場は観測者系で定義されているが、吸収係数 κ_0 と散乱係数 σ_0 は共動座標系で定義されている。吸収・散乱係数は粒子-光子相互作用で決まる量であり、通常は共動座標

系 (バルク速度=0) で調べられている。そのため、これらは共動座標系で定義しておく方が便利である。このような系を混合座標系 (mixed frame) と呼ぶ。

式 (3.1.8)、(3.1.9) に比べると、式 (3.1.18)-(3.1.21) はかなり複雑である。本質的には同じ式なのでどちらの方程式を解いてもよく、実際、式 (3.1.8)-(3.1.9) を解く方法も提案されている (2)。しかし、実は数値計算を行う上では、(3.1.18)-(3.1.21) のように輻射場のモーメント量の座標系をそろえた方が都合がよいことが §3.3 でわかる。そこで、ここでは後者の混合座標系で解くことにする。

3.1.2 輻射場の状態方程式 ~ クロージャー関係式~

§ 3.1.1 では輻射輸送方程式から始まって 0,1 次モーメントの式を導出した。輻射エネルギー密度 E_r は式 (3.1.18) に従って輻射フラックス F_r^i によって移流される。その輻射フラックスは式 (3.1.19) に従って輻射ストレス P_r^{ij} によって運ばれる。では輻射ストレスはどのように決まるのか。実はこの輻射ストレスを決める指導原理はない。この原因は輻射輸送方程式に対してモーメントを取ったために、情報が減ってしまったことに由来する。輻射輸送方程式に 2 次のモーメントを取れば P_r^{ij} の発展方程式を導出する事は出来るが、その場合は 3 次のモーメントが必要になり、幾ら高次のモーメントを取っても永遠に方程式が閉じることはない。この問題を解決する方法は P_r^{ij} が E_r と F_r^i の関数であると仮定して、その関係式を手で与える事である。これはクロージャー関係式と呼ばれており、輻射場の状態方程式を与える。例えば流体の場合はクロージャー関係式として等方圧力とポリトロピック関係 $p_g^{ij} = \delta^{ij} p_g(\rho)$ を仮定する事がある。輻射場の場合、一般には

$$P_r^{ij} = D^{ij}(E_r, F_r^k) E_r, \tag{3.1.22}$$

と書ける。D^{ij} はエディントンテンソル[4]と呼ばれる無次元量で D^{ij} を手で与える事によって方程式を閉じさせる。

最も単純な状態方程式はエディントン近似と呼ばれる近似である。光学的に厚い場合、輻射はガスとよく相互作用するため、輻射場は等方的になる。従ってエディントンテンソルは

$$D_0^{ij} = \frac{\delta^{ij}}{3}, \tag{3.1.23}$$

と書ける。これは massless 粒子 (相対論的粒子) のポリトロピック関係と同じ状態方程式になる。

重用な注意点として、エディントン近似を用いた場合、光は等方的に伝わるために波面の速度は $c/\sqrt{3}$ となり、光速で伝わらない。これは、式 (3.1.23) で与えられるクロージャー関係式が、相対論的流体の状態方程式と同等であり、相対論的流体の音速が $c/\sqrt{3}$ となることからも理解できる。この遅い光の伝播は後で数値計算結果でも紹介する。

式 (3.1.23) の D^{ij} を見ると、添字'0' が付いている。光学的に厚い場合、輻射場が等方的に見えるのは共動座標系だけで成り立つ話であり、流体が動いていれば光は流体とともに動いて見えるはずである。従って一般の観測者系で輻射ストレスを $P_r^{ij} = \delta^{ij} E_r/3$ とするのは間違いである。実際には $P_{r0}^{ij} = \delta^{ij} E_{r0}/3$

[4]正確には相対論の範囲でテンソルではないことは 3×3 行列であることからも明らかである。P_r^{ij} も輻射のストレステンソルとしばしば呼ばれるが、相対論的な意味でテンソルではないため、この章では輻射ストレスと呼ぶ事にする。

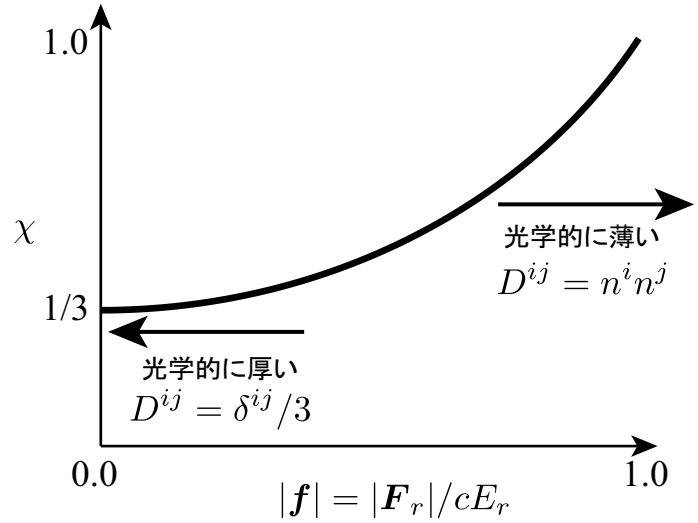


図 3.1: M-1 クロージャーにおける $|\boldsymbol{f}|$ と χ の関係。f が小さい (大きい) とき、光学的に厚い (薄い) 状況のクロージャーに漸近するようになっている。

が成り立つので、これをローレンツ変換して P_r^{ij} を得る。実際にローレンツ変換を行うと

$$\begin{aligned} P_r^{ij} &+ \left[-\frac{\delta^{ij}}{3}\frac{u_k u_l}{c^2} + \frac{u^i u^j u_k u_l}{(1+\gamma)^2 c^4}\right] P_r^{kl} + \frac{u^i u_k P_r^{jk} + u^j u_k P_r^{ik}}{(\gamma+1)c^2} \\ &= \frac{\delta^{ij}}{3}\gamma^2\left(E_r - 2\frac{\beta_j F_r^j}{c}\right) - \frac{u^i u^j E_r}{c^2} + \frac{u^i F_r^j + u^j F_r^i}{c^2} + \frac{2u^i u^j u_k F_r^k}{(\gamma+1)c^4}, \end{aligned} \tag{3.1.24}$$

という非常に複雑な式になる。 E_r, F_r^j, u^i を与えて P_r^{ij} を得るためには、P_r^{ij} は 6 つの独立変数を持っているため、6×6 の線形方程式を解く必要がある。高々 $6^3 = 216$ 回の演算ではあるが、全グリッドで行う必要があるので、意外とこの計算量は馬鹿にならない。数値計算を行う場合には LU 分解などを用いて行列反転を行い、P_r^{ij} を計算する (13)。

もし流体がそれほど速く動いていないのであれば、式 (3.1.24) を (v/c) の 2 次以上のオーダーを無視することで

$$P_r^{ij} = \frac{\delta^{ij}}{3}E_r + \frac{\beta^i F_r^j + \beta^j F_r^i}{c} - \frac{2}{3}\delta^{ij}\frac{\beta_k F_r^k}{c}, \tag{3.1.25}$$

が得られる。この場合は面倒な線形方程式を解く必要がなく、すぐさま P_r^{ij} を得る事が出来る。

エディントン近似は輻射場が等方的であることを仮定していた。これは光学的に厚い場合には正しい。しかし、光学的に薄い状況ではこの仮定は破綻する。光学的に薄い場合、光は指向性を持って真っ直ぐに進むが、エディントン近似を用いると光は等方的に伝わってしまう。この事はエディントンテンソルが相対論的流体と同じ状態方程式に帰着することからも理解できる。また、先に述べたようにエディントンテンソルを用いると光の波面は光速よりもゆっくりと $(c/\sqrt{3})$ 伝わってしまうという問題もあった。

これらの問題を解決するために、ここでは M-1 クロージャーと呼ばれる方法を紹介する (Levermore '84)。具体的にはエディントンテンソルは以下の式で与えられる：

$$D^{ij} = \frac{3(1-\chi)}{2}\frac{\delta^{ij}}{3} + \frac{3\chi-1}{2}n^i n^j, \tag{3.1.26}$$

$$\chi = \frac{3+4|\boldsymbol{f}|^2}{5+2\sqrt{4-3|\boldsymbol{f}|^2}}, \tag{3.1.27}$$

$$n^i = \frac{F_r^i}{|\boldsymbol{F}_r|}, \tag{3.1.28}$$

$$\boldsymbol{f} = \frac{\boldsymbol{F}_r}{cE_r}. \tag{3.1.29}$$

エディントンテンソル (3.1.26) を見ると、右辺第 1 項の $\delta^{ij}/3$ に比例する項は等方圧力に対応する (光学的に厚い) 成分を表し、第 2 項は指向性をもつ、光の動圧に対応する (光学的に薄い) 成分を表している。D^{ij} はそれらの重ね合わせで表されており、その混ざり具合は χ、すなわち $f = |\boldsymbol{f}| = |\boldsymbol{F}_r|/cE_r$ によって決められている。

例えば光学的に厚くてフラックスが 0 の時、$f = 0$ より $D^{ij} = \delta^{ij}/3$ が得られる。一方、光学的に薄く、$F_r = cE_r$ の時は $|\boldsymbol{f}| = 1$ より $D^{ij} = n^i n^j$ となり、動圧がストレスを決める事がわかる (図 3.1 参照)。波面の速度は $f = 0$ ならば $c/\sqrt{3}$、$f = 1$ であれば c で伝わる事はすぐに確かめられる。このように M-1 クロージャーは光学的に薄い状況と厚い状況の両極限をうまく再現できるようになっており、その間は適当な関数 χ が繋いでいる。

E_r, F_r^i が与えられると、輻射ストレスは式 (3.1.26) を用いて $P_r^{ij} = D^{ij}(E_r, F_r^i)E_r$ で計算できる。ここであえて E_r, F_r^i, P_r^{ij} に添字'0' を付けなかった事に注意する。これは M-1 クロージャーが観測者系でも使える事を示している。なぜ観測者系でも使えるのかというのは、χ の関数形に秘密がある。E_r, F_r^i, P_r^{ij} から作られる作られるエネルギー運動量テンソルが共変になるように χ の形が決められている。エディントン近似を用いる場合はローレンツ変換して観測者系のエディントンテンソル (3.1.24) を導出し、6×6 の線形方程式を解いて P_r^{ij} を導出する必要があった。しかし、M-1 クロージャーでは元々クロージャーが共変なためにローレンツ変換を行う必要は無く、E_r と F_r^j からすぐさま P_r^{ij} を得ることが出来る。この様に M-1 は様々な面でエディントン近似よりも優れているため、数値計算では M-1 クロージャーを用いることをお勧めする (14)[5]。

最後に、モーメント方程式＋クロージャー関係で輻射場を記述する事は近似であり、必ずしも正しくない事に注意しておく。このような近似は流体でも行うが、流体はガス同士が衝突し、自分自身で緩和することが出来る。従って衝突系プラズマではモーメント法 (流体近似) はよい近似となる。しかし、光は (量子効果をのぞけば) 光同士で衝突する事はないので、緩和する機構がない。従ってモーメントをとることや、クロージャーを与えて方程式を閉じ させることにどれほどの意味があるのかは不明瞭である。また、式 (3.1.10) を用いることや、吸収、散乱係数の振動数依存性について平均操作することも近似であるため、近い将来、輻射輸送方程式を解いて、モーメント法で得られた結果と比較する事が必要である。

3.2 相対論的輻射流体方程式

ここでは相対論的輻射流体力学方程式を紹介する。磁場を入れたり、さらに電気抵抗を入れる方法はここでは紹介しないが、その方法は非常に単純なので、余力のある方は各自でチャレンジしてほしい。

[5]ただし、M-1 クロージャーをそのまま一般相対論にまで拡張するには注意が必要である (12)。

基礎方程式は
連続の式

$$(\rho u^{\nu})_{,\nu} = 0, \tag{3.2.1}$$

全エネルギー運動量保存

$$(T_r^{\mu\nu} + T_h^{\mu\nu})_{,\nu} = 0, \tag{3.2.2}$$

輻射のモーメント方程式

$$T_{r,\nu}^{\mu\nu} = -G^{\mu}, \tag{3.2.3}$$

となる。ここで下付きの $_{,\nu}$ は微分を表している (以下、ν は振動数ではなく 4 元の添字を表す)。一般には共変微分を考えるが今は特殊相対論の範囲を扱っているので普通の微分である。$T_h^{\mu\nu}$ は流体のエネルギー運動量テンソルで

$$T_h^{\mu\nu} = \rho h u^{\mu} u^{\nu} + p_g g^{\mu\nu}, \tag{3.2.4}$$

である。p_g はガス圧で比エンタルピー h は比熱比 Γ を用いて

$$h = 1 + \frac{e}{\rho c^2} + \frac{p_g}{\rho c^2} = 1 + \frac{\Gamma}{\Gamma - 1}\frac{p_g}{\rho c^2}, \tag{3.2.5}$$

と表される。ただし e は内部エネルギー密度で $p_g = (\Gamma - 1)e$ を仮定した。式 (3.2.2) から (3.2.3) を引くと

$$T_{h,\nu}^{\mu\nu} = G^{\mu}, \tag{3.2.6}$$

が得られる。この式から G^{μ} が 4 元力密度と呼ばれる理由がわかると思う。

数値計算では式 (3.2.1)-(3.2.3) の流体 5 本+輻射 4 本+流体の状態方程式+輻射のクロージャー関係式を同時に解けばよい。しかし、解析に便利なため、流体の運動方程式も導出しておく。まず [式 (3.2.6) の空間成分] - [$hu^i \times$ 式 (3.2.1)] を計算するとすぐに以下の式が得られる

$$\rho\gamma \frac{\mathrm{D}(hu^i)}{\mathrm{D}t} = -\nabla p_g + G^i. \tag{3.2.7}$$

これが相対論的輻射流体の運動方程式である。流体に働く力は圧力勾配力と輻射力である。左辺を見ると非相対論的な場合に比べて γ や h の余計なファクターが入っている。これは相対論的な補正である。流体速度が光速に近い場合や、温度 $k_B T_g$ がイオンの静止質量エネルギー $m_p c^2$ 程度になってくると実効的に慣性が増える事を示している。

式 (3.2.7) が運動方程式を表している事は間違いないが、左辺の時間微分に h が入っているのはなんとも気持ちが悪い。非相対論的な Euler 方程式と対応づけるためにも速度の時間微分の形で表したくなるだろう。その場合には式 (3.2.6) の空間成分から時間成分に β^i を掛けたものを引いて (結構頑張って) 整理すると以下の形の運動方程式が得られる。

$$\rho h\gamma^2 \frac{\mathrm{D}v^i}{\mathrm{D}t} = -\partial^i p_g - \frac{\beta^i}{c}\frac{\partial p_g}{\partial t} + \rho(\kappa_0 + \sigma_0)\left[\gamma\frac{F_r^i}{c} - \frac{E_r u^i + u_j P_r^{ij}}{c} + \frac{u^i u_j F_r^j}{\gamma c^3}\right]. \tag{3.2.8}$$

左辺は速度の時間微分にガスの慣性 $\rho h\gamma^2$ を掛けたもので表されており、非相対論的な運動方程式と対応している。一方、右辺は圧力勾配力に加えて第二項にガス圧の時間微分が入っている。この項はガス圧

が下がると加速されることを示している。相対論的な場合、熱エネルギーが慣性として働くことは先に述べたが、断熱膨張等によってガス圧（熱エネルギー）が下がると慣性も下がるため、ガスはさらに加速されやすくなる、という効果を表している。[ ] 内の第一項目は輻射とガスの間の運動量交換を表す項である。第二項は速度と輻射エネルギー密度に比例して負の力が働いている。これは輻射抵抗と呼ばれる項で、輻射場中をガスが走ると抵抗を受けて減速することを示している。この項は v^i/c の 1 次のオーダーであり、相対論的 (速度が光速に近いか、輻射エネルギー密度がガスエネルギー密度に比べて大きい）になって初めて現れる効果である。

3.3 輻射モーメント方程式の数値解法

前章までで、輻射モーメント方程式 (3.1.18), (3.1.19) とクロージャー関係 (3.1.22) を導入し、さらに流体と合わせた相対論的輻射流体方程式 (3.2.1)-(3.2.3) を導入した。この章ではこれらの方程式の数値解法を紹介する。ただし、流体の数値解法はよく知られているため (例えば数値天文学マニュアル 5 章)、ここでは輻射場の数値解法のみに主題をおく。

3.3.1 離散化

多次元への拡張は容易なため、以下では 1 次元 x 方向のみを考える。方程式を抽象的に書くと 1 次元輻射流体方程式は

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}^x}{\partial x} = \mathcal{S}. \tag{3.3.1}$$

となる。ここで

$$\mathcal{U} = \begin{pmatrix} \mathcal{U}_D \\ \mathcal{U}_{mx} \\ \mathcal{U}_{my} \\ \mathcal{U}_{mz} \\ \mathcal{U}_E \\ \mathcal{U}_{Er} \\ \mathcal{U}_{Fx} \\ \mathcal{U}_{Fy} \\ \mathcal{U}_{Fz} \end{pmatrix} = \begin{pmatrix} \rho\gamma \\ \rho hc^2\gamma u^x + F_r^x \\ \rho hc^2\gamma u^y + F_r^y \\ \rho hc^2\gamma u^z + F_r^z \\ \rho hc^2\gamma^2 - p_g + E_r \\ E_r \\ F_r^x \\ F_r^y \\ F_r^z \end{pmatrix}, \tag{3.3.2}$$

$$\mathcal{F}^x = \begin{pmatrix} \rho u^x \\ \rho hc^2 (u^x)^2 + c^2 p_g + c^2 P_r^{xx} \\ \rho hc^2 u^y u^x + c^2 P_r^{yx} \\ \rho hc^2 u^z u^x + c^2 P_r^{zz} \\ \rho hc^2\gamma u^x + F^x \\ F_r^x \\ c^2 P_r^{xx} \\ c^2 P_r^{yx} \\ c^2 P_r^{zx} \end{pmatrix}, \tag{3.3.3}$$

$$\mathcal{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ S_E \\ S_F^x \\ S_F^y \\ S_F^z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -cG^0 \\ -c^2G^x \\ -c^2G^y \\ -c^2G^z \end{pmatrix}. \tag{3.3.4}$$

上から順に保存量 ($\mathcal{U}$)、フラックス ($\mathcal{F}^x$、輻射フラックスではない)、ソース項 ($\mathcal{S}$) を表している。基本量 $\mathcal{P}$ は

$$^T\mathcal{P} = (\rho, u^x, u^y, u^z, p_g, E_r, F_r^x, F_r^y, F_r^z), \tag{3.3.5}$$

となる。よく見ると、輻射場については基本量と保存量が同じになっていることがわかる。これは式 (3.1.8)-(3.1.9) を解くのではなく、ローレンツ変換して式 (3.1.18)-(3.1.19) に変形する事によって、輻射場の物理量を全て観測者系に揃えたためである。従って後で見るように輻射場については保存量から基本量へ戻す操作が必要ない[6]。

式 (3.3.1) を時間、空間で差分化すると

$$\mathcal{U}_i^{n+1} = \mathcal{U}_i^n - \frac{\Delta t}{\Delta x}\left(f_{i+1/2}^n - f_{i-1/2}^n\right) + \mathcal{S}_i^n \Delta t, \tag{3.3.6}$$

が得られる。上付添字は時間ステップ $t = n\Delta t$、下付添字はメッシュ番号 $x_i = i\Delta x$(Δx はグリッドサイズ) を表している。f は後で述べる数値フラックスを表す。

3.3.2　数値解法

式 (3.3.6) を解く手順を以下に示す。

1. セル中心のソース項を計算する: $\mathcal{P}_i^n \to \mathcal{S}_i^n$
2. セル中心 i で定義された基本量をセル境界 $i \pm 1/2$ に補間し、左状態 ($s = L$) と右状態 ($s = R$) の基本量を得る: $\mathcal{P}_i \to \mathcal{P}_{i\pm1/2,s}$
3. セル境界で定義された基本量から数値フラックスを計算: $\mathcal{P}_{i\pm1/2,s} \to f_{i\pm1/2}$
4. 得られた数値フラックスとソース項を用いて式 (3.3.6) を積分する: $\mathcal{S}_i^n, f_{i\pm1/2}^n \to U_i^{n+1}$
5. 保存量から基本量へ変換する: $\mathcal{U}_i^{n+1} \to \mathcal{P}_i^{n+1}$

ステップ 1 は単純に式 (3.1.20)-(3.1.21) を用いて輻射 4 元力密度を計算すればよいので、難しい事はない。

[6]座標系をそろえない方法も提案されている (2) が、この場合は基本量と保存量の変換が必要になったり、後で述べる陰解法が難しくなるという問題が出てくる

ステップ 2 は流体でよく用いられる reconstruction の方法を用いる。左状態と右状態の基本量は以下の式で与えられる。

$$\mathcal{P}_{i+1/2,L} = \mathcal{P}_i + \frac{\Delta\mathcal{P}}{2}, \tag{3.3.7}$$

$$\mathcal{P}_{i+1/2,R} = \mathcal{P}_{i+1} - \frac{\Delta\mathcal{P}}{2}. \tag{3.3.8}$$

ΔP は slope limiter と呼ばれており様々な関数形が提案されているが、ここでは vanLeer(17) の方法を紹介する:

$$\Delta\mathcal{P} = \frac{2\mathrm{max}(0, \Delta\mathcal{P}_+\Delta\mathcal{P}_-)}{\Delta\mathcal{P}_+ + \Delta\mathcal{P}_-}, \tag{3.3.9}$$

ここで $\Delta\mathcal{P}_\pm = \pm(\mathcal{P}_{i\pm1} - \mathcal{P}_i)$ である。この式からセル境界における基本量が計算できる。

ステップ 3 は数値フラックスの計算方法である。ここでは以下の式を用いて数値フラックスを評価する。

$$f_{i+1/2} = \frac{\lambda_+\mathcal{F}_{i+1/2,L} - \lambda_-\mathcal{F}_{i+1/2,R} + \lambda_+\lambda_-(\mathcal{U}_{i+1/2,R} - \mathcal{U}_{i+1/2,L})}{\lambda_+ - \lambda_-}. \tag{3.3.10}$$

ここで、$\mathcal{U}_{i+1/2,s} = \mathcal{U}(P_{i+1/2,s}), \mathcal{F}_{i+1/2,s} = \mathcal{F}(P_{i+1/2,s})$ はそれぞれ、ステップ 2 で得られた基本量 $\mathcal{P}_{i+1/2,s}$ から作られる保存量とフラックスである。λ_+, λ_- はそれぞれ正（負）の方向に伝搬する最も速い波の特性速度である。式 (3.3.10) を用いて数値フラックスを計算する方法を (4, の頭文字を取って)HLL 法と呼ぶ。波の伝搬速度を計算するには $\mathcal{U}$ と $\mathcal{F}$ からなるヤコビアン行列の固有値を計算すればよい。しかし、この固有値の計算はそれほど簡単ではない (エディントン近似を用いる場合は Farris et al. 2、M-1 クロージャーを用いる場合は González et al. 3参照)。

光の伝わる速度は最大で光速なので、ここでは固有値を計算せずに $\lambda_\pm = \pm c$ と置く。すると式 (3.3.10) は以下のように書ける:

$$\mathcal{F}_{i+1/2} = \frac{\mathcal{F}_{i+1/2,L} + \mathcal{F}_{i+1/2,R} - c(\mathcal{U}_{i+1/2,R} - \mathcal{U}_{i+1/2,L})}{2}. \tag{3.3.11}$$

式 (3.3.11) で数値フラックスを計算する方法を Lax-Friedrich 法と呼ぶ。本来光速より遅い可能性のある波の速度を光速だと思ってフラックスを計算するため、真面目に固有値を計算する HLL フラックスを用いるよりも数値拡散が大きくなる (3)。しかし、多少の犠牲はあるものの、非常に簡単にフラックスを評価できるのは利点である。

ステップ 4 では得られた $\mathcal{S}_i^n$ と $f_{i\pm1/2}^n$ を (3.3.6) に代入し、積分すればよい。

ステップ 5 は保存量から基本量への変換である。ここで、輻射場については基本量と保存量が同じであるという利点が生かされる。つまり、9 つの基本量 (3.3.5) の内、E_r, F_r^i の 4 つは既にわかっている。そこで以下の量を定義する

$$E_h \equiv \mathcal{U}_E - \mathcal{U}_{Er} = \rho hc^2\gamma^2 - p_g, \tag{3.3.12}$$

$$m^i \equiv \mathcal{U}_{m^i} - \mathcal{U}_{Fx} = \rho hc^2\gamma u^x, \tag{3.3.13}$$

$$D \equiv \mathcal{U}_D = \rho\gamma. \tag{3.3.14}$$

物理的にはそれぞれ流体のエネルギー密度、運動量密度、観測者系からみたガス密度を表す。E_h, m^i, D は既知量であるので、5 つの既知量から 5 つの未知量 ρ, u^i, p_g を求めれば良い。これには通常の流体や磁

気流体で用いられる基本量への変換手法を適用することができる。このように輻射場と流体を完全に分離できるのは、式 (3.1.18)-(3.1.19) で輻射場を全て観測者系の物理量に書き下したおかげであり、輻射流体でなく輻射磁気流体や抵抗性輻射磁気流体の場合でも同様に輻射場を分離することができる。相対論的磁気流体における基本量への変換方法については数値天文学マニュアル 6 章を参照してもらえば良い (より洗練された方法も Mignone & Bodo 7 によって提案されている)。

ここでは (18) によって提案された、相対論的流体 (または相対論的抵抗性磁気流体や相対論的 2 流体でも使う事ができる) における基本量変換の方法を紹介する。式 (3.3.14) を用いて式 (3.3.12)-(3.3.13) の ρ を消すと

$$E_h = \left(\frac{Dc^2}{\gamma} + \Gamma_1 p_g\right)\gamma^2 - p_g, \tag{3.3.15}$$

$$m^i = \left(\frac{Dc^2}{\gamma} + \Gamma_1 p_g\right)\gamma u^i, \tag{3.3.16}$$

ここで $\Gamma_1 = \Gamma/(\Gamma-1)$ である。式 (3.3.16) より p_g について解くと

$$p_g = \frac{1}{\Gamma_1}\left(\frac{m}{\gamma u} - \frac{Dc^2}{\gamma}\right), \tag{3.3.17}$$

ここで $m = |\boldsymbol{m}|$, $u = |\boldsymbol{u}|$ である。次に (3.3.17) を式 (3.3.15) に代入して整理すると u についての 4 次方程式が得られる:

$$\begin{aligned}
a_4(u/c)^4 &+ a_3(u/c)^3 + a_2(u/c)^2 + a_1(u/c) + a_0 = 0, && (3.3.18)\\
a_4 &= \Gamma_1^2\left(E_h^2 - (mc)^2\right), && (3.3.19)\\
a_3 &= -2\Gamma_1 mDc^3, && (3.3.20)\\
a_2 &= \Gamma_1^2 E_h^2 - D^2c^4 - 2\Gamma_1(\Gamma_1-1)(mc)^2, && (3.3.21)\\
a_1 &= -2(\Gamma_1-1)mDc^3, && (3.3.22)\\
a_0 &= (\Gamma_1-1)^2(mc)^2. && (3.3.23)
\end{aligned}$$

まず、式 (3.3.18) をニュートン法や、Ferrari の公式等を使って解き、u を求める。すると γ が計算できるので式 (3.3.17), (3.3.14), (3.3.16) より順にそれぞれ p_g, ρ, u^i が求まる。

こうして求まった基本量 $\mathcal{P}^{n+1}$ を用いてステップ 1 に戻れば時間発展させることができる。1-5 ステップを進めた時点で時間を Δt だけ進めると時間 1 次精度の積分になる。時間方向に 2 次精度化するには、式 (3.3.6) を以下のように 2 段階にわけて積分すればよい:

$$\begin{aligned}
\mathcal{U}_i^{n+1/2} &= \mathcal{U}_i^n - \frac{\Delta t}{\Delta x}\left(f_{i+1/2}^n - f_{i-1/2}^n\right) + \mathcal{S}_i^n\Delta t, && (3.3.24)\\
\mathcal{U}_i^{n+1} &= \mathcal{U}_i^n - \frac{\Delta t}{\Delta x}\left(f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n+1/2}\right) + \mathcal{S}_i^{n+1/2}\Delta t, && (3.3.25)
\end{aligned}$$

3.3.3　クーラン条件と陰解法への拡張

この節の最後に、タイムステップの大きさ (クーラン条件) について考える。輻射流体方程式は 3 つのタイムスケールを持っている。一つ目は流体が動くタイムスケールで力学的時間と呼ばれる。典型的な

速度 v、系の大きさを $L(\Delta t$ を決める時は $L=\Delta x$ と置く) とすると力学的時間は $t_{dyn}=L/v$ と表される。この他に、波が系を横切る時間 t_c がある。波は音波や磁気音波等があるが、ここでは輻射を考えているので、波の速さを光速とおくと $t_c=L/c$ となる。これら 2 つのタイムスケールは、相対論的現象 $v\simeq c$ を考える際には同程度である。

この 2 つのタイムスケールに加えて、光がガスと相互作用するタイムスケールがある。これは吸収と散乱のタイムスケールで決まっており、それぞれ $t_{ab}\simeq 1/(\rho\kappa c)$, $t_{ab}\simeq 1/(\rho\sigma c)$ と表される。陽的解法を用いて安定に数値積分を行うためには、タイムステップの大きさを他のどの物理タイムスケールよりも小さく取る必要がある。従って Δt の満たすべき式は[7]

$$\Delta t=\min(t_{dyn},t_c,t_{ab},t_{sc}), \tag{3.3.26}$$

となる。

ところで、吸収のタイムスケールと力学的時間の比をとると

$$\frac{\text{力学的時間}}{\text{吸収にかかる時間}}\simeq\frac{L/c}{1/\rho\kappa c}=\rho\kappa L=\tau. \tag{3.3.27}$$

と書ける。ここで流体速度 $\sim c$ と仮定した。最後の τ は光学的厚みを表す。つまり光学的に厚い状況では、力学的時間に比べて吸収や散乱のタイムスケールが非常に短い。従って、力学的タイムスケールの現象を追いたいと思っても、Δt の大きさは式 (3.3.26) によって律速されてしまうため、光学的に厚い状況では非常にたくさんのステップ数が必要になってしまう。

この問題を解決するために式 (3.3.6) を以下のように 2 つに分ける方法が提案されている (13; 14):

$$\frac{\mathcal{U}_i^*-\mathcal{U}_i^n}{\Delta t}+\frac{f_{i+1/2}^n-f_{i-1/2}^n}{\Delta x} = 0, \tag{3.3.28}$$

$$\frac{\mathcal{U}_i^{n+1}-\mathcal{U}_i^*}{\Delta t} = \mathcal{S}_i^{n+1}. \tag{3.3.29}$$

このように、方程式を移流の項とソース項に分ける方法を operator splitting と呼ぶ。式 (3.3.28) は移流を表すので、この方程式が持つタイムスケールは t_{dyn} と t_c である。式 (3.3.29) はガスと輻射の相互作用を表すので、この式が持つタイムスケールは t_{ab} と t_{sc} である。

ここで式 (3.3.29) の右辺に注目してほしい。ソース項を n ステップでなく $n+1$ ステップ目の値で評価する。このような時間積分方法を陰的時間積分と呼び、$\Delta t>t_{ab},t_{sc}$ でも安定に時間積分する事が可能である。ただし、$n+1$ ステップ目の保存量 $\mathcal{U}^{n+1}$ を得るために $n+1$ ステップ目のソース項 $\mathcal{S}^{n+1}$ を必要とするため、行列反転やイテレーション等の多少の工夫が必要になり、計算量は増える。しかし、その犠牲を払ったとしても Δt は t_{ab},t_{sc} に律速されないため、この方法は非常に威力を発揮する。式 (3.3.28)-(3.3.29) のように陽的解法と陰的解法を合わせた時間積分方法を陽陰的解法 (Explicit-Implicit scheme) と呼ぶ。

最後に並列化について述べておく。陽的解法 (3.3.6) を用いる場合、輻射のモーメント方程式は流体方程式と同じ形をしているため、並列化は流体と同じように行えば良い。つまり、計算領域をノード毎の担当箇所に分けて計算し、計算領域の端のデータだけを他のノードと通信すればよい (領域分割)。陽陰的解法 (3.3.28)-(3.3.29) の場合も全く同じである。「陰解法だから並列化は難しい」という先入観を持っ

[7]正しくは $\Delta t<\min\left(t_{dyn},t_c,|E_r/S_E|,|F_r^i/S_F^i|\right)$ を満たす必要がある。

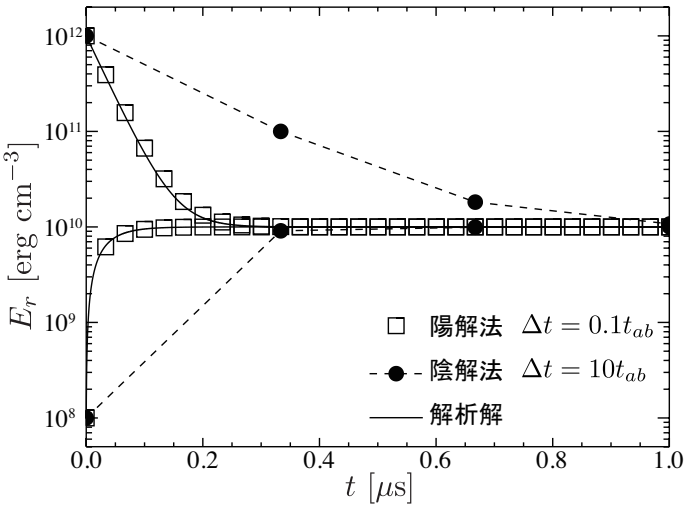


図 3.2: 非熱平衡状態にある輻射が熱平衡へと遷移していく過程を計算した結果。四角は陽解法で解析解 (実線) とよく一致していることがわかる。黒丸は陰解法を用いた結果で、この場合にはタイムステップを大きくしても平衡解へと安定に近づくことがわかる。

てしまうかもしれないが、陰的に時間積分するのは式 (3.3.29) のみである。この式は' 局所的' にガスと輻射がエネルギーのやり取りをする項であり、他の場所の物理量に影響されない。従って他のノードとのデータ通信は必要ないのである。陽的解法にしても陽陰的解法にしても、モーメント法を用いた相対論的輻射流体の並列化は非常に簡単である。ただし、非相対論輻射流体では $t_c \ll t_{dyn}$ であるため、移流項 (3.3.28) も陰的に積分する必要がある。移流項は当然、他の場所の物理量の影響を受けるため、陰的に積分するためには大規模な行列反転が必要になり、ノード間の並列化は至難である。

3.4　計算例

この章では数値計算例を幾つか紹介する。

3.4.1　熱平衡状態への遷移

最初のテストは (16) によって提案されたテスト問題で、定在した一様なガスと初期に非熱平衡にある輻射が、吸収・再放射過程によって熱平衡状態に遷移していく過程の計算である。ここでは簡単のために流体は解かず、ガス温度は常に一定とした (つまりガスは熱浴とする)。この場合、式 (3.1.18), (3.1.20) より、輻射エネルギー密度は以下の式に従って進化する：

$$E_r(t) = \frac{4\pi}{c}\mathrm{B} - \left[\frac{4\pi}{c}\mathrm{B} - E_r(t=0)\right] e^{-\rho\kappa_0 ct}. \tag{3.4.1}$$

図 3.2 に数値計算の結果を示す。初期の輻射エネルギー密度は $10^8, 10^{12}$ erg cm^{-3} とした (それぞれ図中の下側と上側)。他の計算パラメータは $\rho = 0.025$ g cm$^{-3}, a_R T_g^4 = 10^{10}$ erg cm$^{-3}, u^i = 0$, $\Gamma = 5/3$, $\mu = 1, \kappa_0 = 0.04$ cm^2 s^{-1}, $\sigma_0 = 0$ である。ここで $a_R = 4\sigma_{\mathrm{SB}}/c$ は輻射密度定数である。

図からわかるように輻射エネルギーは平衡状態へと遷移していること、さらに解析解 (実線) とよく一致していることがわかる。図中の黒丸は陰的解法を用いて、吸収のタイムスケール $t_{ab} = 1/(\rho\kappa c)$ よりも 10 倍大きなタイムステップを用いた場合の結果である。陽的解法で $\Delta t = 10t_{ab}$ とすると解は振動しすぐに発散してしまうが、陰的解法を用いると大きなタイムステップを用いても解は安定に収束することがわかる。この様に、吸収のタイムスケールより長いタイムスケールの現象を追いたい場合に陰的解法は非常に威力を発揮する。

3.4.2 日陰問題

この問題は (5) によって提案され、(3) によって M-1 クロージャーのテスト問題に応用された。光学的に厚いクランプに向けて光を当てた際、クランプの背面に陰が出来る過程を計算する。2 次元 $x = [0, 0.5]$ cm, $y = [-0.5, 1.5]$ cm の箱の中に、初期に一様なガス ($\rho = \rho_0 = 1$ g cm^{-3}, $e = E_{\rm lte} = a_R T_g^4 = 10^{10}$ erg cm^{-3}) を置き、輻射は熱平衡状態にあるとする。この一様なガスに加えて原点から半径 $r = 0.2$ cm に高密度のクランプ ($\rho = 10^4\rho_0$) を置く。このクランプに向けて境界 $x = -0.5$ cm から光を照射する ($E_{inj} = 10^5 E_{\rm lte}$)。吸収係数は $\kappa_0 = 0.01$ cm^2 g としたので、一様なガスに対しては光学的に薄いが、クランプに対しては光学的に厚い ($\tau = \rho\kappa(2r) = 40$)、という状況である。境界条件は $x = -0.5$ cm で固定境界、$y = 0$ で対称境界とし、それ以外は自由境界とした。

図 3.3 は輻射エネルギー密度のカラーコンターで上はエディントン近似、下は M-1 クロージャーを用いた場合の結果であり、左と右はそれぞれ $t = 33.4,\ 116.8$ ps におけるスナップショットである。まず左の図を見てみると、どちらのクロージャーを用いた場合でもクランプの部分で輻射エネルギー密度が小さくなっていることがわかる。これは吸収によって輻射エネルギーが失われている事を示している。光が吸収されるという振る舞いはどちらのクロージャーを用いても変わらない。しかし光の波面を見てみると、M-1 クロージャーとエディントン近似で大きく異なることがわかる。M-1 クロージャーを用いた場合、光学的に薄いガス中を光は光速で伝わるが、エディントン近似では光は等方的に伝わると仮定したため、光の進む速度は $c/\sqrt{3}$ となる。このために波面の進む速度に両者で大きな隔たりが出来てしまう。

終状態 (図 3.3 右) を見ると、M-1 クロージャーを用いた場合には、クランプの背面に輻射エネルギー密度が低い「日陰」が再現できていることがわかる。しかしエディントン近似を用いると日陰を再現することは出来ない。これも等方分布を仮定したためである。エディントン近似を用いると日陰と日向で輻射圧勾配が生じるために光はクランプを回り込んで日陰を消してしまう。M-1 クロージャーを用いた場合には両者の間で圧力や応力が働かないため (日向では $D^{xx} = 1$、それ以外は $D^{ij} = 0$)、日陰を再現できる。この様に M-1 クロージャーはエディントン近似に比べて、特に光学的に薄い場合に非常によい性質を持つ。

3.4.3 光の伝播と衝突

次に 2 次元的な光の伝播を見てみる。2 次元計算ボックスのサイズを $x = [-1,\ 1]$ cm, $y = [-1,\ 1]$ cm とし、ガスと輻射は初期に熱平衡状態にある。基本的なパラメータは日陰問題と同じである。この状態

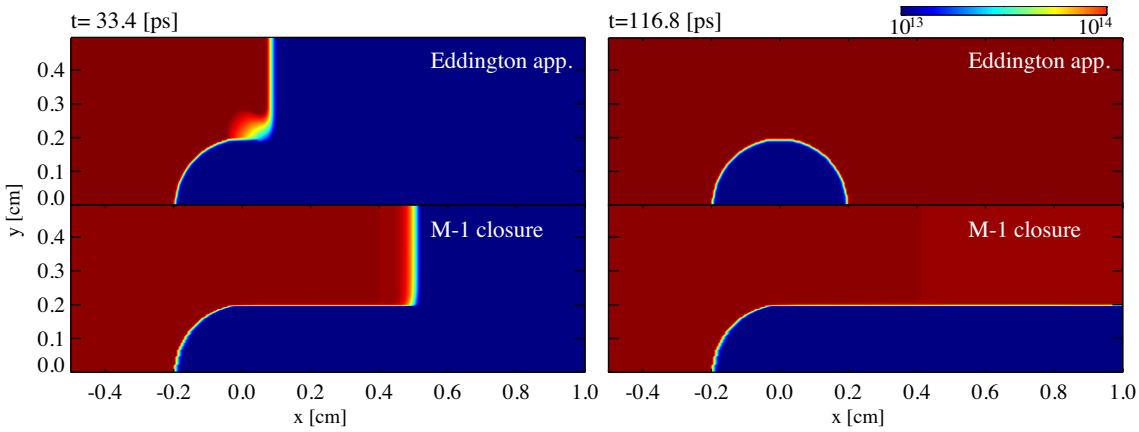


図 3.3: 原点付近にある球形の光学的に厚いクランプに向けて $x = -0.5$ から光りを照らした結果。上はクロージャーにエディントン近似を用いた結果で下は M-1 クロージャーを用いた。左と右はそれぞれ $t = 33.4,\ 116.8$ ps のスナップショット。

で、境界から 2 つの光線を注入する:

$$(E_r, F_r^x, F_r^y) = \begin{cases} \left(E_{inj}, \frac{cE_{inj}}{\sqrt{2}}, \frac{cE_{inj}}{\sqrt{2}}\right), & \text{for } x \le -1 \text{ cm and } -0.9 \text{ cm} \le y \le -0.8 \text{ cm} \\ \left(E_{inj}, -\frac{cE_{inj}}{\sqrt{2}}, \frac{cE_{inj}}{\sqrt{2}}\right), & \text{for } x \le 1 \text{ cm and } -0.9 \text{ cm} \le y \le -0.8 \text{ cm}. \end{cases} \tag{3.4.2}$$

$x = \pm 1$ cm は固定境界、$y = \pm 1$ cm は自由境界とした。

図 3.3 に数値計算の結果を示す。カラーは輻射エネルギー密度、矢印は輻射フラックスを輻射エネルギー密度で規格化した量 ($\boldsymbol{f}$) を示す。

境界から注入された光は光学的に薄いガスの中を真っ直ぐに進む (図 3.4 左)。二つの光線が重なると、通常、光同士は相互作用しないので光はお互いをすり抜けて最終的には X 型の波形になることが期待される。しかし、M-1 クロージャーを用いると図 3.4 右のように光同士が衝突してしまう。

なぜこのような光の衝突が起きてしまうのか。簡単のために流体は静的であるとする。M-1 クロージャーの式 (3.1.26)-(3.1.29) に立ち返ってみる。エディントンテンソルは、光学的に厚い状況を再現する $\delta^{ij}/3$ に比例する成分と、光学的に薄い状況を再現する $n^i n^j$ に比例する成分の重ね合わせで表されており、その割合は χ、つまり $f = |\boldsymbol{F}_r|/cE_r$ によって決められていた。光学的に厚い場合は $f = 0$ が期待されるので、この場合はエディントン近似に帰着する。

では逆に「$f = 0$ ならば光学的に厚い」は成り立つか？物理的にはこれは成り立たない。光学的に薄い状況で二つの光が正面衝突すれば、ネットのフラックスは 0 になるが、光はお互いを通り抜けるはずである。しかし、M-1 クロージャーは $f = 0$ ならば光学的に厚いと盲目的に判断してしまうため、等方的なエディントン近似を採用してしまう。つまり、重要なことは、**M-1 クロージャーにおいてエディントンテンソルは f の値のみで決まっており、光学的厚みには直接は依っていない**' という事である。

図 3.4 右ではまさにそのような状況が起きている。式 (3.4.2) によって注入された 2 つの光線が重なった点では $(E_r, F_r^x, F_r^y) = (2E_{inj}, 0, \sqrt{2}cE_{inj})$ となるため、$f = 1/\sqrt{2} < 1$ となる。対応する χ は 0.61 であるので、エディントンテンソルに等方的な輻射圧成分が 58%も混ざってしまう。その結果、光学的に

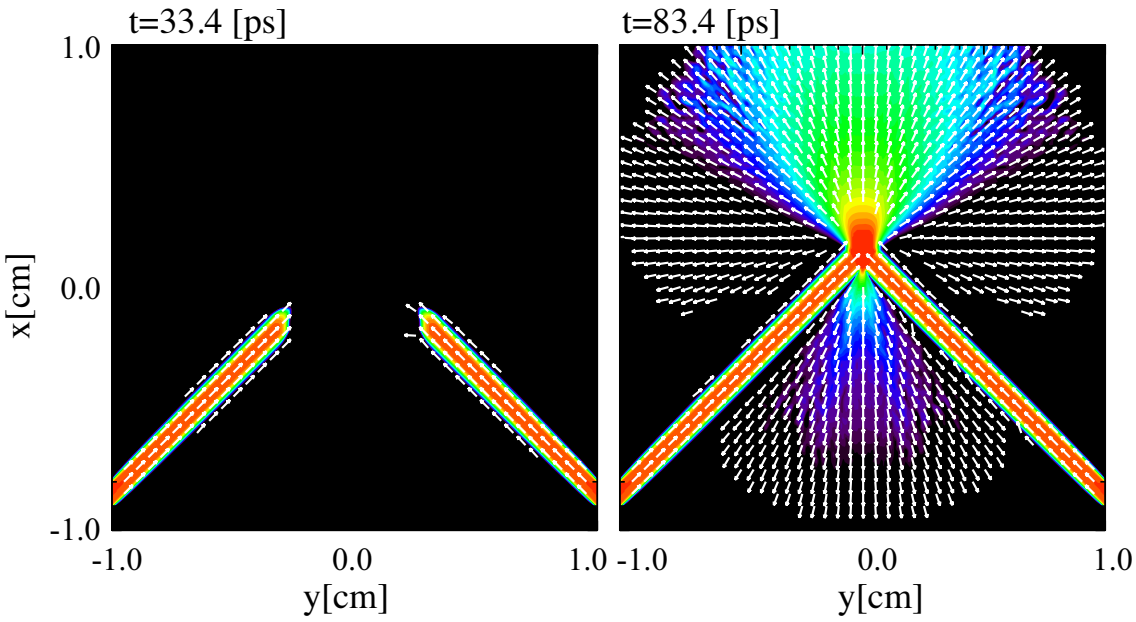


図 3.4: 2 本の光線の伝播。カラーは輻射エネルギー密度、矢印は f^i を表す。

薄いガス中にも関わらず、光は (主に y 方向に伝播するにしても) 等方的に膨張を始めるため、まるで二つの光線が衝突したように見えるのである。この問題は輻射輸送方程式 (3.1.1) のモーメントを取ってクロージャーで方程式を閉じてしまったことに由来する。この問題を解決するためには、輻射のモーメントを取らずに輻射輸送方程式を解く必要がある。

以上で輻射のモーメント方程式の計算例を見せてきたが、これらは基本的には流体は静的であると仮定している。輻射と流体が強くカップルしているような系では静的という仮定はなりたたず、流体と輻射を同時に解く必要がある。そのような場合でもいくつかのテスト問題が提案されている (たとえば非相対論的輻射流体:(16)、相対論的輻射流体:Farris et al. (2)) ので、そちらを参照してほしい。

3.5 まとめ

この章では輻射のモーメント方程式を導出し、相対論的輻射流体方程式の陽的数値解法を紹介した。ここでは単純な相対論的流体を仮定したが、磁気流体や抵抗性磁気流体にする事は簡単である。また、一般相対論化や、粘性や熱伝導といった散逸性流体への拡張、さらに現実的な状態方程式を考慮した、より現実的な流体計算を行うことも現在のスパコンであれば十分に可能である。しかし、いつでも心に留めておかなければいけないのは、輻射場の流体近似 (モーメント方程式) は非常に大きな仮定であるということである。非常にたくさんの近似の上に成り立っているモーメント法を正当化するには、輻射輸送方程式を解くことで確かめなければならない。また、モーメント法を正当化するという目的だけでなく、輻射輸送方程式を解くことで、モーメント法では無視していた効果が現れたり、観測とのスペクトルの直接比較が出来るようになる。近い将来実現可能となるであろう輻射輸送方程式を解く数値計算に備えて、輻射輸送方程式を安定に解く数値計算スキームの開発を今から行うべきであろう。

関連図書

[1] Castor, J. I. 2004, Radiation Hydrodynamics, ed. Castor, J. I.

[2] Farris, B. D., Li, T. K., Liu, Y. T., & Shapiro, S. L. 2008, Phys. Rev. D, 78, 024023

[3] González, M., Audit, E., & Huynh, P. 2007, A&A, 464, 429

[4] Harten, A., Lax, P. D., & van Leer, B. 1983, SIAM Rev., 25, 35

[5] Hayes, J. C. & Norman, M. L. 2003, ApJS, 147, 197

[6] Kato, S., Fukue, J., & Mineshige, S. 2008, Black-Hole Accretion Disks — Towards a New Paradigm —, ed. Kato, S., Fukue, J., & Mineshige, S.

[7] Mignone, A. & Bodo, G. 2006, MNRAS, 368, 1040

[8] Mihalas, D. & Mihalas, B. W. 1984, Foundations of radiation hydrodynamics (New York, Oxford University Press, 1984, 731 p.)

[9] Park, M.-G. 2006, MNRAS, 367, 1739

[10] Pomraning, G. C. 1973, The equations of radiation hydrodynamics, ed. Pomraning, G. C.

[11] Rybicki, G. B. & Lightman, A. P. 1979, Radiative processes in astrophysics, ed. Rybicki, G. B. & Lightman, A. P.

[12] Shibata, M., Kiuchi, K., Sekiguchi, Y., & Suwa, Y. 2011, Progress of Theoretical Physics, 125, 1255

[13] Takahashi, H. R., Ohsug, K., Sekiguchi, Y., Inoue, T., & Tomida, K. 2011, submitted to ApJ

[14] Takahashi, H. R. & Ohsuga, K. 2012, in prep.

[15] Takahashi, R. 2008, MNRAS, 383, 1155

[16] Turner, N. J. & Stone, J. M. 2001, ApJS, 135, 95

[17] van Leer, B. 1977, Journal of Computational Physics, 23, 263

[18] Zenitani, S., Hesse, M., & Klimas, A. 2009, ApJ, 696, 1385

第4章　相対論的散逸流体の数値計算法

高本亮 (京都大学)

4.1　背景

近年観測機器の発展に伴い様々な高エネルギー天体現象の詳細な描像が得られており、それに伴い相対論的流体近似を用いた理論研究が盛んに行われている。現在の観測は主に電磁波を通じて行われており、観測を説明するためには散逸を考慮して流体のエネルギーを熱エネルギーに変換することが必要である。また大局的な計算では多くの場合計算機資源の問題から乱流や輻射の影響を精度よく考慮することは難しく、それらを散逸として現象論的モデル化を行うことも頻繁に行われている。他方核物理での分野では近年重イオン衝突の際に生成されるクォークグルーオンプラズマ (QGP) が相対論的流体で良く近似出来ることが知られており、近年その粘性の影響が非常に活発に議論されている。このように数値シミュレーションにおける相対論的散逸への興味は非常に大きいが、相対論的散逸における因果律の問題や計算効率の問題のために近年まで相対論的散逸を考慮した研究が行われることはなかった。本発表では上記の問題が解決された、発表者が新たに開発した相対論的散逸流体の数値計算法について解説する (1)。

4.2　基礎方程式

この発表では $c = 1$ という単位系を用い、Cartesian coordinate での Minkowski 計量テンソル $\eta_{\mu\nu}$ の符号系として $\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$ を用いる。またギリシャ文字で表された変数は 0 から 3 までをとり、ローマ文字で表された変数は 1 から 3 までを取る。

また共変的な微分として、共変的時間微分 $\hat{D}$ と共変的空間微分 ∇_α を以下のように定義する。

$$\hat{D}A^{\mu_1\cdots\mu_n} \equiv u^\beta A^{\mu_1\cdots\mu_n}_{;\beta}, \tag{4.2.1}$$

$$\nabla_\alpha A^{\mu_1\cdots\mu_n} \equiv \gamma^\beta_\alpha A^{\mu_1\cdots\mu_n}_{;\beta}. \tag{4.2.2}$$

ここでテンソル $\gamma^{\mu\nu}$ は 4 元ベクトル u^μ に垂直な超平面への射影テンソルで、次のように定義される。

$$\gamma^{\mu\nu} = \eta^{\mu\nu} + u^\mu u^\nu \tag{4.2.3}$$

4.2.1　理想流体

相対論的流体方程式は粒子数、運動量、エネルギーの保存則から次のように与えられる。

$$N^{\mu}_{;\mu} = 0, \tag{4.2.4}$$
$$T^{\mu\nu}_{;\mu} = 0, \tag{4.2.5}$$

ここで N^{μ} は粒子数密度流束で、$T^{\mu\nu}$ はエネルギー運動量テンソルである。完全流体を考える場合、これらのテンソルは次のように表される。

$$N^{\mu} = nu^{\mu}, \tag{4.2.6}$$
$$T^{\mu\nu} = \rho h u^{\mu}u^{\nu} + p\eta^{\mu\nu}, \tag{4.2.7}$$

ここで n は粒子数密度を表し、p は圧力、$h = 1 + \epsilon + p/\rho, \epsilon$ は無次元化されたエンタルピーと内部エネルギーを、$\rho \equiv mn$ は質量密度を表す。以上の量は全て流体静止系で定義されている。また m は構成粒子の静止質量で、u^{μ} は流体の 4 元速度を表しており、次の関係を満たす: $u^{\mu}u_{\mu} = -1$。

Cartesian 座標で考える場合、相対論的流体の時間発展方程式は次のように表される。

$$\frac{\partial}{\partial t}\begin{pmatrix} D \\ m^i \\ E \end{pmatrix} + \frac{\partial}{\partial x^j}\begin{pmatrix} Dv^j \\ m^i v^j + pI^{ij} \\ m^j \end{pmatrix} = 0, \tag{4.2.8}$$

ここで $\mathbf{v}$ は流体の 3 次元速度を、$D, \mathbf{m}, E$ はそれぞれ実験室系での質量、運動量、エネルギー密度を表しており、I^{ij} は単位テンソルを表す。実験室系では D, $\mathbf{m}$, E は次のように表される。

$$D = \gamma\rho, \tag{4.2.9}$$
$$\mathbf{m} = \rho h\gamma^2\mathbf{v}, \tag{4.2.10}$$
$$E = \rho h\gamma^2 - p, \tag{4.2.11}$$

ここで $\gamma = (1 - v^2)^{-1/2}$ はロレンツ因子である。上記の形の方程式は相対論的数値流体では一般的に用いられているものである。

4.2.2　Causal dissipative fluid

相対論的散逸流体を考える場合、散逸により考える系は一様でも等方でもなくなるため粒子数密度流束ベクトル N^{μ} とエネルギー運動量テンソル $T^{\mu\nu}$ のテンソル分解は理想流体の場合から変わることになる。また相対論的散逸流体の場合、上記の 2 つのテンソルから 2 つの特徴的な方向として N^{μ} と $T^{0\mu}$ が現れ、流体の 4 元速度 u^{μ} の方向の定義に不定性が現れることになる。つまり流体の 4 元速度は一般に 2 つの独立なベクトルである N^{μ} と $T^{0\mu}$ の線形結合により表される。本発表では非相対論と同様に 4 元速度 u^{μ} の方向を粒子流束ベクトル N^{μ} の方向として定義した Eckart 分解のみを考える。(なおその他の有名な速度の定義としては、4 元速度 u^{μ} の方向をエネルギー $T^{0\mu}$ の方向として定義した Landau-Lifshitz 分解などが知られている (2))。

Eckart 分解を考える場合、粒子流束ベクトル N^μ とエネルギー運動量テンソル $T^{\mu\nu}$ は次のように表される。

$$N^\mu = nu^\mu, \tag{4.2.12}$$

$$T^{\mu\nu} = \rho h u^\mu u^\nu + p\eta^{\mu\nu} + q^\mu u^\nu + q^\nu u^\mu + \tau^{\mu\nu}, \tag{4.2.13}$$

ここで q^μ は熱伝導ベクトルで $\tau^{\mu\nu}$ は粘性テンソルである。

式 (4.2.4)、(4.2.5) より、散逸流体の発展方程式は次のようになる。

$$\frac{\partial}{\partial t}\begin{pmatrix} D \\ m^i + q^0 u^i + q^i u^0 + \tau^{0i} \\ E + 2q^0 u^0 + \tau^{00} \end{pmatrix} + \frac{\partial}{\partial x^j}\begin{pmatrix} Dv^j \\ m^i v^j + pI^{ij} + q^i u^j + q^j u^i + \tau^{ij} \\ m^j + q^0 u^j + q^j u^0 + \tau^{0j} \end{pmatrix} = 0, \tag{4.2.14}$$

非相対論の場合と違い、これらの式では時間微分されている保存量の部分に散逸項が含まれている。特に heat flux vector は流体の静止系に移ったとしても運動量保存の式 (4.2.14) において保存量として残り続ける。これは相対論の場合はエネルギー流束と運動量流束密度が同一視されることに起因している。ここでもし Navier-Stokes 方程式の散逸量を相対論に拡張した表式を用いた場合、流体方程式は放物型になり、無限大の特性速度を含んでしまうことが知られている。これは散逸変数が自身の平衡状態へ無限小の時間で緩和することに起因し、これにより上記の流体方程式に含まれる散逸の時間微分から非物理的な発散が現れることを示している。さらに Hiscock と Lindblom (3; 4) らにより、相対論的 Navier-Stokes 方程式の線形摂動は 4 元速度の定義によらず必ず指数関数的に増大するモードを含むことが示されている。(なお Landau-Lifshitz 分解だけは流体静止系でのみ安定であることが分かっている)

これらの方程式の安定性と因果律の問題を解決するために、Israel と Stewart は相対論的 Boltzman 方程式から以下のような 2 次の散逸理論を求めた。

$$\hat{D}\Pi = \frac{1}{\tau_\Pi}(\Pi_{NS} - \Pi) - I_\Pi, \tag{4.2.15}$$

$$\hat{D}\pi^{\mu\nu} = \frac{1}{\tau_\pi}(\pi^{\mu\nu}_{NS} - \pi^{\mu\nu}) - I^{\mu\nu}_\pi, \tag{4.2.16}$$

$$\hat{D}q^\mu = \frac{1}{\tau_q}(q^\mu_{NS} - q^\mu) - I^\mu_q, \tag{4.2.17}$$

ここで $\pi^{\mu\nu}$ と Π は shear viscosity と bulk viscosity と呼ばれる粘性で、それぞれ粘性テンソル $\tau^{\mu\nu}$ の対称非対角成分と対角成分で定義される。

$$\tau^{\mu\nu} \equiv \Pi\gamma^{\mu\nu} + \pi^{\mu\nu}, \tag{4.2.18}$$

また τ_q、τ_Π、τ_π はそれぞれの散逸量の緩和時間で、I は散逸量や流体変数の勾配の積で表される 2 次のオーダーの大きさの量である。(この 2 次のオーダーの量は勾配が比較的滑らかな場合が多い宇宙物理での応用では通常無視される)

上記の方程式では散逸量の時間発展、つまり散逸量の緩和現象を考慮に入れていることになっている。Israel と Stewart の求めた散逸流体方程式はこれらの項を考えたことにより方程式は双曲型になり、その結果方程式は安定で因果律を守る。なお緩和時間として適切なパラメータを選ばなかった場合は因果律や安定性は必ずしも守られない。これらの適切なパラメータに関しては Sec 4.4 で議論する。

散逸の Navier-Stokes 項は次のように表される。

$$q^{\mu}_{NS} = -\kappa\gamma^{\mu\nu}\left(T_{,\nu} + Tu^{\rho}u_{\nu,\rho}\right), \tag{4.2.19}$$

$$\begin{aligned}\tau^{\mu\nu}_{NS} &= \pi^{\mu\nu}_{NS} + \Pi_{NS}\gamma^{\mu\nu} \\ &= -\gamma^{\mu\rho}\gamma^{\nu\sigma}\left[\eta\left(u_{\rho,\sigma} + u_{\sigma,\rho} - \frac{2}{3}\eta_{\rho\sigma}u^{\lambda}_{,\lambda}\right)\right] - \zeta u^{\lambda}_{,\lambda}\gamma^{\mu\nu},\end{aligned} \tag{4.2.20}$$

ここで κ は熱伝導係数、η は shear viscosity 係数、そして ζ は bulk viscosity 係数を表し、添字 NS は “Navier-Stokes” 項であることを意味している。もし散逸変数が流体変数に比べて十分小さい場合は、熱伝導の表式 (4.2.19) に $u^{\rho}u_{\nu,\rho}$ に理想流体の関係式を代入して次のように書き換えることが出来る。

$$q^{\mu}_{NS} = -\kappa\gamma^{\mu\nu}\left(T_{,\nu} - \frac{T}{\rho h}p_{,\nu}\right). \tag{4.2.21}$$

本発表では Navier-Stokes の熱伝導ベクトルとしては式 (4.2.21) を用いることにする。

なお熱伝導ベクトル、粘性テンソルは見かけ上それぞれ 4 成分、１０成分になるが、これらには次のような直交関係が成り立つ。

$$\tau^{i0}u_0 = -\tau^{ij}u_j, \tag{4.2.22}$$

$$\tau^{00}u_0 = -\tau^{0j}u_j, \tag{4.2.23}$$

$$q^{0}u_0 = -q^{j}u_j. \tag{4.2.24}$$

その結果それぞれ物理的なのは 3 成分、6 成分になる。

式 (4.2.20)、(4.2.21) は共変形式で与えられている。これらの式に対して平坦な時空上の Cartesian 座標を考え、また 1 次元問題を考えると次のようになる。

$$\begin{aligned}\tau^{0x}_{NS} &= -\gamma^{0\rho}\gamma^{x\sigma}\left[\eta(\partial_{\rho}u_{\sigma} + \partial_{\sigma}u_{\rho}) + \left(\zeta - \frac{2}{3}\eta\right)g_{\rho\sigma}\partial_{\lambda}u^{\lambda}\right] \\ &= -\left[\eta\left\{u^{x}u^{t}\partial_{t}u^{t} + (-1 + (u^{t})^{2})\partial_{t}u^{x} + (1 + (u^{x})^{2})\partial_{x}u^{t} + u^{t}u^{x}\partial_{x}u^{x}\right\}\right. \\ &\quad \left. + \left(\zeta - \frac{2}{3}\eta\right)u^{0}u^{x}\theta\right],\end{aligned} \tag{4.2.25}$$

$$\begin{aligned}\tau^{0\perp}_{NS} &= -\gamma^{0\rho}\gamma^{\perp\sigma}\left[\eta(\partial_{\rho}u_{\sigma} + \partial_{\sigma}u_{\rho}) + \left(\zeta - \frac{2}{3}\eta\right)g_{\rho\sigma}\partial_{\lambda}u^{\lambda}\right] \\ &= -\left[\eta\left\{u^{\perp}u^{t}\partial_{t}u^{t} + (-1 + (u^{t})^{2})\partial_{t}u^{\perp} + u^{\perp}u^{x}\partial_{x}u^{t} + u^{t}u^{x}\partial_{x}u^{\perp}\right\}\right. \\ &\quad \left. + \left(\zeta - \frac{2}{3}\eta\right)u^{0}u^{\perp}\theta\right],\end{aligned} \tag{4.2.26}$$

$$\begin{aligned}
\tau_{NS}^{yz} &= -\gamma^{y\rho}\gamma^{z\sigma}\left[\eta(\partial_\rho u_\sigma + \partial_\sigma u_\rho) + \left(\zeta - \frac{2}{3}\eta\right) g_{\rho\sigma}\partial_\lambda u^\lambda\right] \\
&= -\left[\eta\left\{u^y u^t \partial_t u^z + u^z u^t \partial_t u^y + u^y u^x \partial_x u^z + u^z u^x \partial_x u^y\right\}\right. \\
&\quad \left. + \left(\zeta - \frac{2}{3}\eta\right) u^y u^z \theta\right], \qquad (4.2.27) \\
\tau_{NS}^{x\perp} &= -\gamma^{x\rho}\gamma^{\perp\sigma}\left[\eta(\partial_\rho u_\sigma + \partial_\sigma u_\rho) + \left(\zeta - \frac{2}{3}\eta\right) g_{\rho\sigma}\partial_\lambda u^\lambda\right] \\
&= -\left[\eta\left\{u^\perp u^t \partial_t u^x + u^x u^t \partial_t u^\perp + u^\perp u^x \partial_x u^x + (1 + (u^x)^2)\partial_x u^\perp\right\}\right. \\
&\quad \left. + \left(\zeta - \frac{2}{3}\eta\right) u^x u^\perp \theta\right], \qquad (4.2.28) \\
\tau_{NS}^{xx} &= -\gamma^{x\rho}\gamma^{x\sigma}\left[\eta(\partial_\rho u_\sigma + \partial_\sigma u_\rho) + \left(\zeta - \frac{2}{3}\eta\right) g_{\rho\sigma}\partial_\lambda u^\lambda\right] \\
&= -\eta\partial_x u^x - \left(\zeta - \frac{2}{3}\eta\right)(1 + (u^x)^2)\theta, \qquad (4.2.29) \\
q_{NS}^{x} &= -\gamma^{x\mu}\kappa\left[\partial_\mu - \frac{T}{\rho h}\partial_\mu p\right] \\
&= -\kappa\left[u^x u^t\left(\partial_t T - \frac{T}{\rho h}\partial_t p\right) + (1 + (u^x)^2)\left(\partial_x T - \frac{T}{\rho h}\partial_x p\right)\right], \qquad (4.2.30) \\
q_{NS}^{\perp} &= -\gamma^{\perp\mu}\kappa\left[\partial_\mu - \frac{T}{\rho h}\partial_\mu p\right] \\
&= -\kappa u^\perp\left[u^t\left(\partial_t T - \frac{T}{\rho h}\partial_t p\right) + u^x\left(\partial_x T - \frac{T}{\rho h}\partial_x p\right)\right], \qquad (4.2.31)
\end{aligned}$$

ここで θ は流体の expansion で次のように定義される。

$$\theta \equiv \nabla_\mu u^\mu = \partial_\mu u^\mu + \Gamma^\mu_{\alpha\mu} u^\alpha. \qquad (4.2.32)$$

なお上の式では粘性テンソルは shear viscosity と bulk viscosity が混在して扱われている。これは我々が用いる directional split のための対称性を考慮した結果である。unsplit 法を用いて計算する場合は必ずしも上の形式を使う必要はない。

4.2.3 緩和時間の表式

Israel-Stewart 理論はいくつかの運動論的係数を含んでいる。つまり散逸係数と緩和時間である。これらは一般には分布関数と粒子同士の散乱断面積に依存している。ここでは流体近似に興味があるため、平衡分布に非常に近い場合についてのみ取り扱う (5; 6)。

この場合、緩和時間の具体的な表式は次のようになる。

$$\tau_\Pi = \frac{\zeta\beta_0}{3}, \quad \tau_q = \kappa T\beta_1, \quad \tau_\pi = 2\eta\beta_2 \qquad (4.2.33)$$

常識に含まれる運動論的パラメータは次のように表される。

$$\Gamma/(\Gamma-1) = \beta^2(1+5h/\beta-h^2), \tag{4.2.34}$$

$$\alpha_0 = (\Gamma-1)\Omega^{**}/\Gamma\Omega p, \quad \alpha_1 = -(\Gamma-1)\Gamma p, \tag{4.2.35}$$

$$\beta_0 = \frac{3\Omega^*}{h^2\Omega^2 p}, \qquad \beta_1 = \left(\frac{\Gamma-1}{\Gamma}\right)^2 \frac{\beta}{hp}\left(5h^2 - \frac{\Gamma}{\Gamma-1}\right), \tag{4.2.36}$$

$$\beta_2 = \frac{1+6h/\beta}{2h^2 p}, \qquad a_1 = -\frac{1+h\beta(\Gamma-1)/\Gamma}{h^2 p}, \tag{4.2.37}$$

$$\Omega = 3\Gamma - 5 + 3\Gamma/h\beta, \quad \Omega^* = 5 - 3\Gamma + 3(10-7\Gamma)h/\beta, \tag{4.2.38}$$

$$\Omega^{**} = 5 - 3\Gamma + 3\Gamma^2/(\Gamma-1)h^2\beta^2, \tag{4.2.39}$$

ここで $\beta = m/T$ で h はエンタルピーである。

緩和時間の非相対論、相対論的極限での漸近形は次のようになる。

1. 非相対論的極限　$(\beta \to \infty, \quad \Gamma = 5/3)$

$$\tau_\Pi = \frac{2}{5}\frac{\zeta\beta^2}{p}, \quad \tau_q = \frac{2\kappa T\beta}{5p}, \quad \tau_\pi = \frac{\eta}{p} \tag{4.2.40}$$

2. 相対論的極限　$(\beta \to 0, \quad \Gamma = 4/3)$

$$\tau_\Pi = \frac{72\zeta}{\beta^4 p}, \quad \tau_q = \frac{5\kappa T}{4p}, \quad \tau_\pi = \frac{3\eta}{2p} \tag{4.2.41}$$

散逸係数 κ, η, ζ の具体的表式は粒子間の相互作用に依存しており、考える問題に適した表式を用いればよい。

4.3 Numerical scheme

この章では新しい数値計算法を 1 次元問題で説明を行う。多次元への拡張については Sec 4.3.4 で説明をする。

4.3.1 Strang splitting method

相対論的散逸流体方程式は双曲型緩和方程式であり、理想流体の部分と散逸部分の各々に特有の数値的困難がある。方程式の理想流体部分の困難は流体方程式の非線形性に起因しており、これは非相対論の場合にも存在している。一方散逸部分の困難は、散逸の発展方程式 (4.2.15)、(4.2.16)、(4.2.17) が stiff 方程式になっていることにある。もし stiff 方程式を陽的差分法で差分近似する場合、安定に解くためには時間刻み幅 Δt は緩和時間よりも短く取る必要がある。しかし散逸の緩和時間 $\tau_\Pi, \tau_\pi, \tau_q$ は一般に流体の特性時間に比べ極めて短いため、数値的に解く場合大きな計算コストが必要になる。

これらの問題を別々に扱うため、ここでは Strang splitting 法を適用し、相対論的散逸流体方程式を次のように分離する。

$$\frac{\partial}{\partial t}\begin{pmatrix} D \\ m^i \\ E \end{pmatrix} + \frac{\partial}{\partial x^j}\begin{pmatrix} Dv^j \\ m^i v^j + pI^{ij} \\ m^j \end{pmatrix} = 0, \tag{4.3.1}$$

$$\frac{\partial}{\partial t}\begin{pmatrix} D \\ m^i + q^0 u^i + q^i u^0 + \tau^{0i} \\ E + 2q^0 u^0 + \tau^{00} \end{pmatrix} + \frac{\partial}{\partial x^j}\begin{pmatrix} 0 \\ q^i u^j + q^j u^i + \tau^{ij} \\ q^0 u^j + q^j u^0 + \tau^{0j} \end{pmatrix} = 0, \tag{4.3.2}$$

まず理想流体部分の式 (4.3.1) は Riemann solver を用いれば非常に精度よく解くことが出来る。Riemann solver とはセル境界での数値流束を Riemann 問題の解を用いて計算するという手法で、極めて安定に精度よく理想流体方程式を解くことが出来ることが知られている。

次に散逸部分の式 (4.3.2) について考える。式 (4.3.2) の数値流束部分は式 (4.2.14) の散逸部分であり、散逸変数と 4 元速度 u^μ を含んでいる。ここで時間の 2 次精度を考える場合は、この数値流束として時間が $\Delta t/2$ 進んだ値を用いなければならない。

式 (4.3.2) の第一項は時間の保存量であり、非相対論の場合と異なり散逸変数を含んでいる。これらを安定に時間発展させるためには、これらに含まれる散逸変数として Navier-Stokes 項ではなく、式 (4.2.15), (4.2.16), (4.2.17) で時間発展させた値を代入しなければならない。さらに式 (4.3.2) の第一項には 4 元速度 u^μ も含まれていることに注意が必要である。これらの理由により、散逸変数を含んだ保存量は理想流体部分の時間発展の前に計算を行い、散逸部分の時間発展までメモリ上に保存させておく必要がある。

まとめると、式 (4.3.2) の保存変数 U を次のように分離する。

$$U = U_{ideal} + U_{dissip}, \tag{4.3.3}$$

ここで

$$U_{ideal} = \begin{pmatrix} D \\ m^i \\ E \end{pmatrix}, \quad U_{dissip} = \begin{pmatrix} 0 \\ q^0 u^i + q^i u^0 + \tau^{0i} \\ 2q^0 u^0 + \tau^{00} \end{pmatrix}. \tag{4.3.4}$$

まず私たちは $U = U_{ideal} + U_{dissip}$ を最初に計算する。そして理想流体部分 U_{ideal} を Riemann solver を用いて時間発展させる。次に散逸部分で用いる保存変数 U の初期値を $\tilde{U}_{ideal} + U_{dissip}$ で求め、式 (4.3.2) を Δt 時間発展させる。ここで $\tilde{U}_{ideal}$ は Riemann solver を用いて時間発展させた流体部分の保存変数である。式 (4.3.1) と (4.3.2) に対して保存形式を用いることで、この手法は質量、運動量、エネルギーを計算機の丸め誤差の範囲内で厳密に保存させることが出来る。

これらに加え、私たちは散逸の独立変数として $q^x, q^y, q^z, \tau^{0x}, \tau^{0y}, \tau^{0z}, \tau^{yz}, \tau^{zx}, \tau^{xy}$ を採用した。この選択は空間方向に対して対等にするという要請に基づいており、directional split を用いる場合に便利なようになっている。その他の散逸量は直交関係 (4.2.22)、(4.2.23)、(4.2.24) を用いることで導出が可能である。また以上の独立変数として採用した散逸変数は primitive 変数を求める際にも用いるため、これらの量はセルの中心に定義される。散逸の数値流速を計算する場合は、必要な空間精度に応じてセル境界へ補完を行い、セル境界で $\Delta t/2$ 時間発展させるようにする。

4.3.2 Stiff equation

相対論的散逸流体を Israel-Stewart 理論を用いて解析する場合、散逸変数の時間発展を常に考える必要がある。しかしこれらの方程式は stiff 方程式、つまり次のような形の式になっている。

$$\partial_t U = \frac{S(U)}{\tau_{relax}}, \tag{4.3.5}$$

ここで τ_{relax} は緩和時間で、U の時間発展における特徴的時間になっている。この式においてもしこの特徴的時間 τ が流体の特徴的時間 τ_{fluid} よりも極めて短い場合に式 (4.3.5) は “stiff 方程式” と呼ばれ、陽的差分法による近似は時間刻み幅を $\Delta t \lesssim \tau_{relax} \ll \tau_{fluid}$ のように取った場合にのみ安定になる。流体近似を考える場合は一般にこの制限は流体の Courant-Friedrichs-Lewy (CFL) 条件 $\Delta t \leq \Delta x / c_{charact}$ よりもはるかに厳しい。ここで $c_{charact}$ は流体の特徴的速度、例えば音速や Alfvén 速度である。この制限は非常に大きな計算コストを要求することになるため、単純な陽的差分による近似を使用することは事実上不可能である。この時間刻み幅の問題を回避するために、私たちは Strang-splitting 法を用いる。

まずは式 (4.2.15)、(4.2.16)、(4.2.17) を座標依存の形に書き直すと次のようになる。

$$\gamma \left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) \Pi = \frac{1}{\tau_\Pi} (\Pi_{NS} - \Pi) - I_\Pi, \tag{4.3.6}$$

$$\gamma \left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) \pi^{\mu\nu} = \frac{1}{\tau_\pi} (\pi^{\mu\nu}_{NS} - \pi^{\mu\nu}) - I_\pi, \tag{4.3.7}$$

$$\gamma \left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) q^\mu = \frac{1}{\tau_q} (q^\mu_{NS} - q^\mu) - I_q. \tag{4.3.8}$$

この式を次のように split する。

$$\left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) \Pi = -\frac{I_\Pi}{\gamma}, \tag{4.3.9}$$

$$\left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) \pi^{\mu\nu} = -\frac{I_\pi}{\gamma}, \tag{4.3.10}$$

$$\left(\frac{\partial}{\partial t} + v^j \frac{\partial}{\partial x^j} \right) q^\mu = -\frac{I_q}{\gamma}, \tag{4.3.11}$$

$$\frac{\partial}{\partial t} \Pi = \frac{1}{\gamma \tau_\Pi} (\Pi_{NS} - \Pi), \tag{4.3.12}$$

$$\frac{\partial}{\partial t} \pi^{\mu\nu} = \frac{1}{\gamma \tau_\pi} (\pi^{\mu\nu}_{NS} - \pi^{\mu\nu}), \tag{4.3.13}$$

$$\frac{\partial}{\partial t} q^\mu = \frac{1}{\gamma \tau_q} (q^\mu_{NS} - q^\mu). \tag{4.3.14}$$

式 (4.3.9)、(4.3.10)、(4.3.11) は source 項付きの移流方程式になっており、通常の風上差分を用いれば安定に精度よく計算が可能である。一方式 (4.3.12)、(4.3.13)、(4.3.14) は stiff 方程式になっており、特別な計算法が必要になる。私たちの方法では、これらの stiff 方程式を Piecewise-Exact-Solution(PES) 法

を用いて計算を行う (7)。上式で Strang-splitting 法を用いたため、式 (4.3.12)、(4.3.13)、(4.3.14) の形式解を次のように求めることが出来る。

$$\Pi = (\Pi_0 - \Pi_{NS}) \exp\left[-\frac{t-t_0}{\tau_\Pi}\right] + \Pi_{NS}, \tag{4.3.15}$$

$$\pi^{\mu\nu} = (\pi_0^{\mu\nu} - \pi_{NS}^{\mu\nu}) \exp\left[-\frac{t-t_0}{\tau_\pi}\right] + \pi_{NS}^{\mu\nu}, \tag{4.3.16}$$

$$q^\mu = (q_0^\mu - q_{NS}^\mu) \exp\left[-\frac{t-t_0}{\tau_q}\right] + q_{NS}^\mu, \tag{4.3.17}$$

ここで添字 0 は初期値を表している。このように形式解を用いて時間発展をさせることで、時間刻み幅に関係なく安定に数値計算を行うことが可能になる。

ところで Navier-Stokes 項 (4.2.25) - (4.2.31) は非相対論の場合と異なり空間微分だけではなく時間微分も含んでいる。私たちはこの時間微分について次のような 1 次の陽的差分を用いて計算する。

$$\partial_t U^n = \frac{\hat{U}_{ideal}^{n+1} - U^n}{\Delta t}. \tag{4.3.18}$$

ここで $\hat{U}_{ideal}^{n+1}$ は理想流体の時間発展によって得られた変数を表しており、U^n は n ステップ目の初期値を表している。このように我々の方法では各ステップの流体変数の初期値を散逸の計算ステップまで保存しておく必要がある。以上に加え、Navier-Stokes 項の中の空間微分は中心差分で近似するべきである。これは散逸変数は物理的に拡散を表すためである。

4.3.3 Primitive recovery

流体方程式の数値計算において保存形式を使う場合、時間発展の結果得られるのは保存量であり primitive 変数ではない。このため保存量から primitive 変数への変換が各時間ステップで必要になる。非相対論の場合と違い相対論的流体の場合、この変換は理想流体の場合でも非線形な代数方程式を解くことが必要になる。相対論的散逸流体の場合、保存量の中で散逸量は一般に 4 元速度との積の形で現れるため、この primitive 変数への変換はさらに複雑になる。この章ではこの primitive 変数への変換をある程度簡略化出来る方法について説明する。

まず保存量に含まれる散逸変数は、緩和方程式 (4.2.15)、(4.2.16)、(4.2.17) から既に求まっていることに注意する。すると primitive 変数への変換が複雑になる主な原因は、上記の散逸変数に対して 4 元速度 u^μ が掛かっていることにあることが分かる。ここでもし散逸変数による寄与が十分小さい場合、散逸ステップにおける 4 元速度 u^μ の変化は理想流体ステップによる変化に対して小さいことが期待出来る。このような理由により、私たちは u^μ の初期値として、理想流体ステップで求まった u^μ を用いる。すると保存変数 U_{dissip} に現れる散逸変数に関係する部分は計算することが出来てしまう。そして保存変数 U から U_{dissip} を差し引くことで理想流体部分 U_{ideal} を求めることが出来、これは理想流体で一般的に用いられる primitive 変数の変換法を用いて primitive 変数の計算が可能になる。次に U_{dissip} の中の 4 元速度 u^μ の初期推定値として求まった 4 元速度を用いて、上記と同じ計算を再度行う。このようにして計算が必要な精度まで収束するまで iterative に primitive 変数を計算することで、consistent に primitive 変数を求めることが出来る。

以上をまとめると次のようにすれば良い。

1. まず U_{dissip} を理想流体ステップで求まった u^μ を初期推定値として用いて計算する。
2. $U_{ideal} = U - U_{dissip}$ を計算し、primitive 変数を理想流体の primitive 変数への変換法を用いて計算する。
3. U_{dissip} の中の u^μ の初期推定値として以上の計算で求めた u^μ を用い、再度 primitive 変数への変換を行う。
4. 以上の計算を primitive 変数が十分収束するまで繰り返す。

4.3.4　多次元への拡張

ここまでは 1 次元問題の場合の数値計算法について説明をした。多次元計算の場合は directional splitting 法を用いることで、1 次元での計算法を多次元へ拡張することが出来る。この手法では多次元計算は、1 次元計算を各空間方向に続けて行うことで時間発展が可能になる。なお時間の 2 次精度を達成する場合は、1 次元計算を次のような順番で行えば良い。まず 2 次元の場合は、

$$U^{n+1} = L_x^{1/2} L_y L_x^{1/2} U^n, \tag{4.3.19}$$

そして 3 次元の場合は、

$$U^{n+1} = L_x^{1/6} L_y^{1/6} L_z^{1/3} L_y^{1/6} L_x^{1/3} L_z^{1/6} L_y^{1/3} L_x^{1/6} L_z^{1/3} L_x^{1/6} L_y^{1/3} L_z^{1/6} L_x^{1/6} U^n. \tag{4.3.20}$$

4.4　Causality and Stability

Israel-Stewart 理論は散逸の時間発展を考慮することで散逸流体方程式を因果律を守り安定にしているが、正確に言うと適切なパラメータを考えないとこれらの条件は満たされない。この章では適切なパラメータの条件について議論を行う。

4.4.1　Stability of the telegrapher equation

まず Israel-Stewart 理論は散逸の緩和を考慮することが本質であるため、次のような線形の方程式系に簡略化して考えることにする。

$$\partial_t Q + \nabla \cdot \mathbf{F} = 0, \tag{4.4.1}$$

$$\partial_t \mathbf{F} = -\frac{1}{\tau}(\mathbf{F} + \eta \nabla Q). \tag{4.4.2}$$

これらの方程式から $\mathbf{F}$ を消去すると次のように電信方程式になる。

$$\partial_t^2 Q + \frac{1}{\tau}\partial_t Q - \frac{\eta}{\tau}\triangle Q = 0. \tag{4.4.3}$$

電磁気の電信方程式と同様に考えて上の方程式 (4.4.3) の特性速度を求めると次のようになる。

$$v_c = \sqrt{\frac{\eta}{\tau}} \tag{4.4.4}$$

特性速度は方程式系の情報の伝播速度と考えられるので、因果律を守るためにはこの速度は光速を越えてはいけない。よって因果律を守るには、時間スケールのパラメータ τ には次のような条件が必要になる。

$$\tau \geq \tau_{min} \equiv \frac{\eta}{v_{\max}^2}, \tag{4.4.5}$$

この式で $v_{\max}$ は考えている物理系の最大の速度、例えば光速を表している。

次に安定性の条件を議論する。Israel-Stewart 理論は分布関数が平衡に近い場合には安定で因果律を守ることが示唆されている (5)。一方数値計算を行う場合は、安定な計算のために一般に CFL 条件と呼ばれる時間刻み幅の条件が現れる。例えば緩和時間が τ の緩和方程式を陽的差分法で計算する場合は次の条件が課される: $\Delta t < \tau$。以下では PES 法を用いた場合の式 (4.4.1)、(4.4.2) の安定性条件について von Neumann の安定性解析を行う。PES 法を用いる場合、基礎方程式は次のようになる。

$$\partial_t Q + \partial_x F = 0, \tag{4.4.6}$$

$$F = -\eta \partial_x Q + (F_0 + \eta \partial_x Q) e^{-\frac{\Delta t}{\tau}}. \tag{4.4.7}$$

次に上記の方程式の差分化を行う。時間 2 次精度を考慮した場合、次のようになる。

$$\frac{Q_j^{n+1} - Q_j^n}{\Delta t} + \frac{F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2}}{\Delta x} = 0, \tag{4.4.8}$$

$$F_j^{n+1} = -\eta \frac{Q_{j+1/2}^{n+1/2} - Q_{j-1/2}^{n+1/2}}{\Delta x} + \left[F_j^n + \eta \frac{Q_{j+1/2}^{n+1/2} - Q_{j-1/2}^{n+1/2}}{\Delta x} \right] e^{-\frac{\Delta t}{\tau}}. \tag{4.4.9}$$

安定性解析のために、上記の式に以下の形の Q_j^n と F_j^n を代入する。

$$Q_j^n = R^n e^{ij\theta}, \quad F_j^n = G^n e^{ij\theta}, \tag{4.4.10}$$

ここで R^n と G^n は定数 R、G の n 乗を表している。すると方程式 (4.4.8)、(4.4.9) は次のようになる。

$$R^{n+1} - R^n + 2\frac{\Delta t}{\Delta x} i \sin\frac{\theta}{2} G^{n+1/2} = 0, \tag{4.4.11}$$

$$G^{n+1} = -\frac{2\eta}{\Delta x} i \sin\frac{\theta}{2} R^{n+1/2} + G^n e^{-\frac{\Delta t}{\tau}} + \frac{2\eta}{\Delta x} i \sin\frac{\theta}{2} R^{n+1/2} e^{-\frac{\Delta t}{\tau}}. \tag{4.4.12}$$

これらの式から、次の式が得られる。

$$R^{n+1} - \left[1 + e^{-\frac{\Delta t}{\tau}} - 4\frac{\Delta t}{\Delta x^2}\eta(1 - e^{-\frac{\Delta t}{\tau}}) \sin^2\frac{\theta}{2} \right] R^n + e^{-\frac{\Delta t}{\tau}} R^{n-1} = 0. \tag{4.4.13}$$

この式を解くことで次のように Q が求まる。

$$R = \frac{1}{2}\left[1 + e^{-\frac{\Delta t}{\tau}} - 4\frac{\Delta t}{\Delta x^2}\eta(1 - e^{-\frac{\Delta t}{\tau}}) \sin^2\frac{\theta}{2} \right] \pm \sqrt{\frac{1}{4}\left[1 + e^{-\frac{\Delta t}{\tau}} - 4\frac{\Delta t}{\Delta x^2}\eta \sin^2\frac{\theta}{2}(1 - e^{-\frac{\Delta t}{\tau}}) \right]^2 - e^{-\frac{\Delta t}{\tau}}}. \tag{4.4.14}$$

安定のためには、$|R| \leq 1$ という条件を満たすことが必要になる。この条件を求めるために、式 (4.4.14) に $|R| = 1$ を代入すると、次のようになる。

$$\Delta t_{min} = \begin{cases} \frac{\Delta x^2}{2\eta \sin^2 \frac{\theta}{2}} & \text{if} \quad \Delta t \gg \tau \\ \frac{\Delta x}{\sin \frac{\theta}{2}} \sqrt{\frac{\tau}{\eta}} & \text{if} \quad \Delta t \ll \tau \end{cases} . \tag{4.4.15}$$

$R = 1$ の場合の厳密な解を図示すると Fig. 4.1 のようになる。式 (4.4.15) と Fig. 4.1 は PES 法の安定

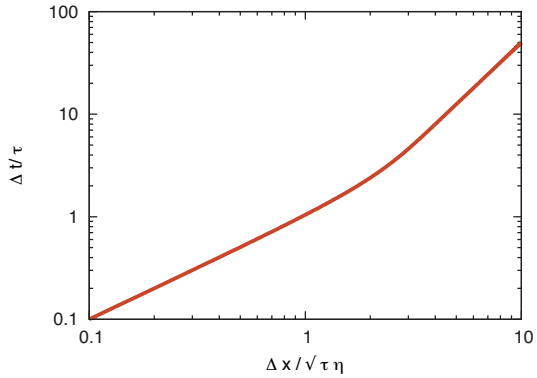


図 4.1: $|R| = 1$ の場合の式 (4.4.14) の数値計算結果。この図から $\Delta t/\tau$ は $\Delta t/\tau \ll 1$ の場合は Δx に比例し、$\Delta t/\tau \gg 1$ の場合は Δx^2 に比例していることが分かる。

性条件について次のことを示している。まず $\Delta t \geq \tau$ の場合、安定性の条件は通常の放物型の場合と同様の $\Delta t < \Delta x^2/\eta$ になる。一方 $\Delta t \leq \tau$ の場合、安定性の条件は双曲型の場合と同様の $\Delta t < \Delta x\sqrt{\tau/\eta}$ になる。

Israel-Stewart 理論は電信型の方程式であるため、上で求まった結果は Israel-Stewart 理論の実用解法についても適用出来ると思われる。そこで CFL 条件として次のような条件を課すことにする。

$$\Delta t = C_a \begin{cases} \min\left\{\Delta x/c_s, \alpha\rho h \Delta x^2/\max\{\kappa, \eta, \zeta\}\right\} & \text{if} \quad \Delta t > \tau, \\ \min\left\{\Delta x/c_s, \Delta x\sqrt{\tau/\eta}\right\} & \text{if} \quad \Delta t < \tau \end{cases} \tag{4.4.16}$$

ここで c_s は音速で、α は時間 2 次精度の場合は 1/2 で、時間 1 次精度の場合は 1/4 となる係数、そして $0 < C_a < 1$ はクーラン数である。なお Israel-Stewart 理論は非線形の散逸流体方程式であるため、上記の条件が真の安定性条件になっているかどうかは示されていない。しかし我々の数値計算は上記の条件を守ることで安定になっている。

4.5　テスト計算

この章では以上で説明した数値計算法を用いて行った計算結果を示す。以降の計算では CFL 数として 0.4 を用い、理想流体の状態方程式 $h \equiv 1 + \Gamma/(\Gamma - 1)p/\rho$ を仮定する。

4.5.1 1D shear flow

相対論的散逸理論である Israel-Stwart 理論は因果律と安定性のために変数が 9 個から 14 個に増えており、厳密解を得ることが非常に難しい。しかし shear flow の shear viscosity による緩和の問題は、相対論的な効果が出ない程度の速度を用いれば十分非相対論の厳密解に近づくと考えられる。そこでまず次のような初期条件を考える。

$$(\rho^L, p^L, v^{yL}) = (1.0, 1.0, -0.1) \qquad \text{for} \quad x < 0.0, \tag{4.5.1}$$

$$(\rho^R, p^R, v^{yR}) = (1.0, 1.0, 0.1) \qquad \text{for} \quad x \geq 0.0. \tag{4.5.2}$$

上記以外の速度場は全て 0 にする。

この場合、相対論的 Euler 方程式は次のようになる。

$$\rho h \partial_t [\gamma^2 v^y] + \partial_x \tau^{xy} = 0. \tag{4.5.3}$$

この問題では十分に小さな速度を用いているため、ロレンツ因子は $\gamma = 1$ と見なせる。この場合は粘性テンソルは次のような非相対論と同様の形に帰着する。

$$\tau^{xy} = -\eta \partial^x v^y. \tag{4.5.4}$$

式 (4.5.3)、(4.5.4) を用いると、次のような厳密解が得られる。

$$v^y = v_0^y \mathrm{erf}\left[\frac{|x - x_0|}{\chi t}\right], \tag{4.5.5}$$

$$\chi = \frac{\eta}{\rho h}. \tag{4.5.6}$$

Fig. 4.2 は $t = 4.0$ の場合の数値計算結果である。比熱比は $\Gamma = 4/3$ を用い、セルの大きさは $\Delta x = 0.01$

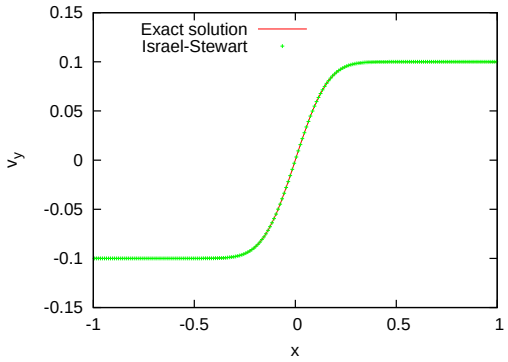


図 4.2: shear flow の粘性による緩和現象の $t = 4.0$ での数値計算結果。実線は厳密解 (4.5.5) を表し、点が数値計算結果を示している。我々の Israel-Stewart 理論の新しい数値解法は解析解を非常に精度よく再現出来ている。この問題ではセルの大きさとして $\Delta x = 0.01$ を用い、パラメータとして $\Gamma = 4/3, \eta = 0.01$ を用いた。

を用いた。この問題では shear viscosity 係数として $\eta = 0.01$ を用いている。Fig. 4.2 から数値計算結果は厳密解を精度よく再現していることが分かる。

4.5.2　shock tube test

この章では、shock tube 問題について考える。もし衝撃波を含む問題を『理想』流体の数値計算法を用いて解く場合、衝撃波の厚さは数値散逸で決まることになる。しかし実際には衝撃波の厚みは散逸係数によって決まることになる。以降では我々の数値解法を用いることで衝撃波の厚みが実際に散逸係数に依存して変化することを示す。厳密解の solver としては Martíと Müller によって作られたコードを用いて求めた (**?**)。

初期条件としては次のものを考える。

$$(\rho^L, p^L, v^{yL}) = (10.0, 10.0, 0.2) \qquad \text{for} \quad x < 0.5, \tag{4.5.7}$$

$$(\rho^R, p^R, v^{yR}) = (1.0, 1.0, -0.2) \qquad \text{for} \quad x \geq 0.5, \tag{4.5.8}$$

比熱比としては $\Gamma = 5/3$ を用い、上記以外の物理量は 0 にしている。数値計算は $t = 0.4$ まで行い、セルの大きさとしては $\Delta x = 0.0025$ を用いた。熱伝導係数 κ、shear 粘性 η、bulk 粘性 ζ は定数を仮定し、特に記載が無い場合は 10^{-15} としている。

まず Fig. 4.3 は shear 粘性係数を $\eta = 0.01, 0.05, 10^{-15}$ と変えた場合の v^y の数値計算結果と理想流体の場合の結果を比較した図である。この図から接触不連続面と衝撃波が共に粘性により散逸され、不連続面の厚みは $\eta = 0.05$ の場合は $\eta = 0.01$ のほぼ 5 倍程度になっていることが分かる。

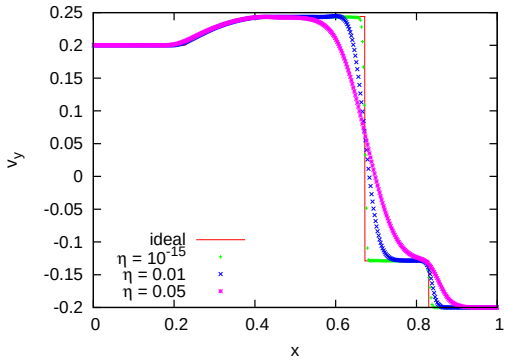


図 4.3: 相対論的 shock tubu 問題の数値計算結果。$t = 0.4$ の場合の v_y のプロファイルが点によって表されており、shear 粘性係数は $\eta = 10^{-15}, 0.01, 0.05$ の 3 つの場合において計算されている。また実線は理想流体の厳密解である。このテスト計算では不連続面の厚みは『物理的な』粘性によって定まっていることが分かる。このテスト計算ではセルの幅は $\Delta x = 0.0025$ を用いており、比熱比は $\Gamma = 5/3$ を、CFL 数は 0.4 を用いている。

次に Fig. 4.4 は温度 $T \equiv p/\rho$ についての数値計算結果である。この計算では熱伝導係数として $\kappa = 0.01, 0.05, 10^{-15}$ を用いている。この図から、熱伝導により温度が高い領域から低い領域へと熱が流れていることが見て取れ、接触不連続面と衝撃波面は散逸により滑らかになっている。また不連続面の厚みは $\kappa = 0.05$ の場合は $\kappa = 0.01$ のほぼ 5 倍程度になっていることが分かる。

Fig. 4.5 は速度 v^x についての数値計算結果である。この計算では bulk 粘性係数として $\zeta = 0.01, 0.04, 10^{-15}$ を用いている。bulk 粘性は流体の expansion $\theta = u^\mu_\mu$ を散逸させる粘性で、1 次元の場合は $\theta = \partial_x u^x$ により表される。Fig. 4.5 は v^x の勾配が粘性によって滑らかになっていることが分かる。

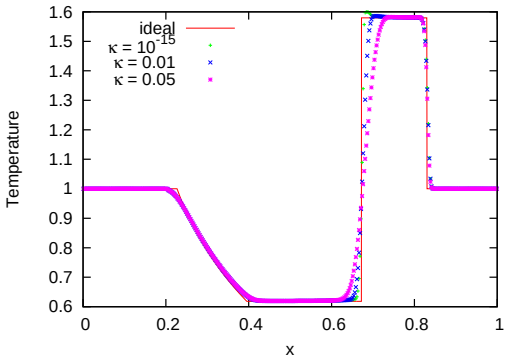


図 4.4: 相対論的 shock tubu 問題の数値計算結果。$t = 0.4$ の場合の温度 T のプロファイルが点によって表されており、熱伝導係数は $\kappa = 10^{-15}, 0.01, 0.05$ の 3 つの場合において計算されている。また実線は理想流体の厳密解である。全ての熱伝導係数の場合において衝撃波の厚みはほぼ同等だが、接触不連続面の厚みは『物理的な』熱伝導によって定まっていることが分かる。このテスト計算ではセルの幅は $\Delta x = 0.0025$ を用いており、比熱比は $\Gamma = 5/3$ を、CFL 数は 0.4 を用いている。

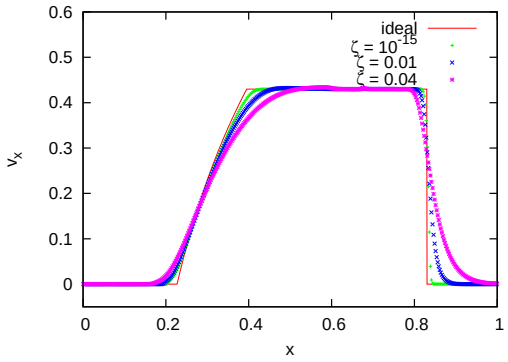


図 4.5: 相対論的 shock tubu 問題の数値計算結果。$t = 0.4$ の場合の v_x のプロファイルが点によって表されており、bulk 粘性係数は $\zeta = 10^{-15}, 0.01, 0.05$ の 3 つの場合において計算されている。また実線は理想流体の厳密解である。不連続面の厚みは『物理的な』粘性によって定まっていることが分かる。このテスト計算ではセルの幅は $\Delta x = 0.0025$ を用いており、比熱比は $\Gamma = 5/3$ を、CFL 数は 0.4 を用いている。

関連図書

[1] Takamoto, M., & Inutsuka, S.-I. 2011, Journal of Computational Physics, 230, 7002

[2] L. D. Landau, & E. M. Lifshitz, *Course of theoretical physics* (Oxford: Pergamon Press, 1959).

[3] W. A. Hiscock & L. Lindblom, Annals of Physics **151** (1983) 466.

[4] W. A. Hiscock & L. Lindblom, Phys. Rev. D **31** (1985) 725.

[5] W. Israel & J. M. Stewart, Annals of Physics **118** (1979) 341.

[6] S. R. de Groot, W. A. van Leeuwen, Ch. G. van Weert, *Relativistic kinetic theory - Principles and applications* (North-Holland, 1980).

[7] T. Inoue, & S. Inutsuka, ApJ 687 (2008) 303.

[8] J. M. Marti, & E. Müller, Journal of Fluid Mechanics 258 (1994) 317.

第5章　太陽表面対流のための輻射磁気流体方程式の数値解法

飯島陽久 (東京大学)

5.1　はじめに

太陽表面では放射冷却によるエネルギー損失を補うため対流が発生している。太陽では、表面対流と関わる磁場現象の観測とシミュレーションの比較研究が盛んで、輻射輸送と部分電離の効果を取り入れた輻射磁気流体計算が行われている。しかし、そのような研究は海外の一部のグループに限られていた。

我々は、その中でも並列性が高く高速だと思われる Max Planck のグループによる MURaM コード (Vögler 2003; Vögler et al. 2005; Rempel 2009) を参考に局所熱力学平衡、非散乱、灰色近似を仮定した輻射磁気流体計算コードを開発した。ここでは、その数値計算法を紹介する。

5.2　基礎方程式

基礎方程式は、重力と輻射加熱項 Q_{rad} を加えた磁気流体方程式

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0 \tag{5.2.1}$$

$$\frac{\partial (\rho \boldsymbol{V})}{\partial t} + \nabla \cdot \left[\rho \boldsymbol{V} \otimes \boldsymbol{V} + \left(p + \frac{B^2}{8\pi} \right) \underline{\mathrm{I}} - \frac{\boldsymbol{B} \otimes \boldsymbol{B}}{4\pi} \right] = \rho \boldsymbol{g} \tag{5.2.2}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \left[\left(e + p + \frac{B^2}{8\pi} \right) \boldsymbol{V} - \frac{1}{4\pi} \boldsymbol{B} (\boldsymbol{V} \cdot \boldsymbol{B}) \right] = \rho (\boldsymbol{g} \cdot \boldsymbol{V}) + Q_{rad} \tag{5.2.3}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \cdot (\boldsymbol{V} \otimes \boldsymbol{B} - \boldsymbol{B} \otimes \boldsymbol{V}) = 0 \tag{5.2.4}$$

である。太陽表面では輻射圧は十分に小さいので無視した。ここで、ρ は密度、$\boldsymbol{V}$ は速度場、$\boldsymbol{B}$ は磁場、$\underline{\mathrm{I}}$ は 3×3 の単位行列を表す。e は単位体積あたりの全エネルギーで、

$$e = e_{int} + \frac{1}{2} \rho V^2 + \frac{B^2}{8\pi} \tag{5.2.5}$$

ここで、e_{int} は単位体積あたりの内部エネルギーである。太陽表面付近では水素の電離度が急激に変化するため、電離平衡を仮定した非理想気体の状態方程式を用いる。

$$p = p(\rho, e_{int}) \tag{5.2.6}$$

状態方程式は OPAL コード (Rogers et al. 1996; Iglesias et al. 1996) で生成した数表を数値的に内挿して用いた。

輻射加熱項 Q_{rad} は、光速が十分大きい場合の輻射輸送方程式

$$\frac{dI}{d\tau} = S - I \tag{5.2.7}$$

$$d\tau = \rho\kappa ds \tag{5.2.8}$$

を直接解いて求める。ここで、簡単のため周波数方向の離散化は考えないものとした (灰色近似)。I は輻射強度、S は源泉関数、κ は Rosseland 平均した不透明度、τ は光学的厚み、s は幾何学的な厚みである。局所熱力学平衡と非散乱を仮定しているため、源泉関数 S や不透明度 κ は輻射強度によらず陽に決定される。

$$S = B(T) = \frac{\sigma_s}{\pi}T^4 \tag{5.2.9}$$

$$\kappa = \kappa(\rho, T) \tag{5.2.10}$$

$$T = T(\rho, e_{int}) \tag{5.2.11}$$

ここで、σ_s はステファン・ボルツマン係数である。よって、境界条件さえ与えられれば、輻射輸送方程式は単純に境界から積分していくだけで解くことが出来る。このようにして輻射強度 I が求められれば、輻射加熱項は以下の様に与えられる。

$$Q_{rad} = \nabla \cdot \boldsymbol{F} = 4\pi\rho\kappa(J - S) \tag{5.2.12}$$

$$\boldsymbol{F} = \int_{4\pi} d\Omega \boldsymbol{\mu} I \tag{5.2.13}$$

$$J = \int_{4\pi} d\Omega I \tag{5.2.14}$$

ここで、$\boldsymbol{\mu}$ は光線の角度、$\boldsymbol{F}$ は輻射フラックス、J は角度平均した輻射強度である。

5.3　流体部分

太陽表面付近は成層が強く、ボックス内で密度が 4 桁以上変化し、対流のオーバーシュートによって引き起こされた音波は衝撃波に変わるため、数値的にロバストであることが要求される。また、対流層の内部はレイノルズ数が非常に高く、限られたグリッド数で数値粘性を少なく抑えるため、高次精度のスキームである必要がある。ここでは、空間方向に 4 次の中央差分、時間方向に 4 次のルンゲ・クッタ法を用いた方法を紹介する。並列化は全て MPI を用いた空間領域分割で行った。

5.3.1　空間差分

方程式系は 1 次元の場合、以下の様に表せる。

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = s \tag{5.3.1}$$

この時、4 次精度のフラックスは

$$f_{j+1/2} = \frac{7}{12}\left[f(u_{j+1}) + f(u_j)\right] - \frac{1}{12}\left[f(u_{j+2}) + f(u_{j-1})\right] \tag{5.3.2}$$

で与えられる。

人工粘性は Rempel (2009) で提案されたものを用いた。区分的に線形であると仮定し、グリッド境界の両側での物理量を以下のように再構成する。

$$(u_x)_j = \mathrm{minmod}\left(\frac{u_j - u_{j-1}}{\Delta x}, \frac{u_{j+1} - u_j}{\Delta x}\right) \tag{5.3.3}$$

$$u^+_{j+1/2} = u_{j+1} - \frac{1}{2}(u_x)_{j+1} \tag{5.3.4}$$

$$u^-_{j+1/2} = u_j + \frac{1}{2}(u_x)_j \tag{5.3.5}$$

ここで、minmod 関数は絶対値の最も小さい引数を返す。人工粘性のフラックスは、以下の様に与えられる。

$$g_{j+1/2} = \frac{1}{2} c_{j+1/2} \phi_{j+1/2} (u^+_{j+1/2} - u^-_{j+1/2}) \tag{5.3.6}$$

$$\phi_{j+1/2} = \begin{cases} \left[(u^+_{j+1/2} - u^-_{j+1/2})/(u_{j+1} - u_j)\right]^2 \\ \qquad\qquad \text{for } (u^+_{j+1/2} - u^-_{j+1/2})(u_{j+1} - u_j) > 0 \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{5.3.7}$$

ここで、$c_{j+1/2}$ はグリッド境界での特性速度で、今はその最大値 $c = c_s + v_a + V$ (c_s は音速、v_a はアルフベン速度) を用いた。

結局、空間差分とフラックスは、

$$\frac{\partial}{\partial t} u_j = -\frac{H_{j+1/2} - H_{j-1/2}}{\Delta x} + s(u_j) \tag{5.3.8}$$

$$H_{j+1/2} \equiv f_{j+1/2} + g_{j+1/2} \tag{5.3.9}$$

のようになる。

5.3.2 時間積分

式 (5.3.8) は以下の様な形をしている。

$$\frac{\partial u}{\partial t} = R(u) \tag{5.3.10}$$

このとき、メモリ節約型の 4 次精度のルンゲ・クッタでの時間積分は、

$$u^{n+1/4} = u^n + \frac{\Delta t}{4} R(u^n) \tag{5.3.11}$$

$$u^{n+1/3} = u^n + \frac{\Delta t}{3} R(u^{n+1/4}) \tag{5.3.12}$$

$$u^{n+1/2} = u^n + \frac{\Delta t}{2} R(u^{n+1/3}) \tag{5.3.13}$$

$$u^{n+1} = u^n + \Delta t R(u^{n+1/2}) \tag{5.3.14}$$

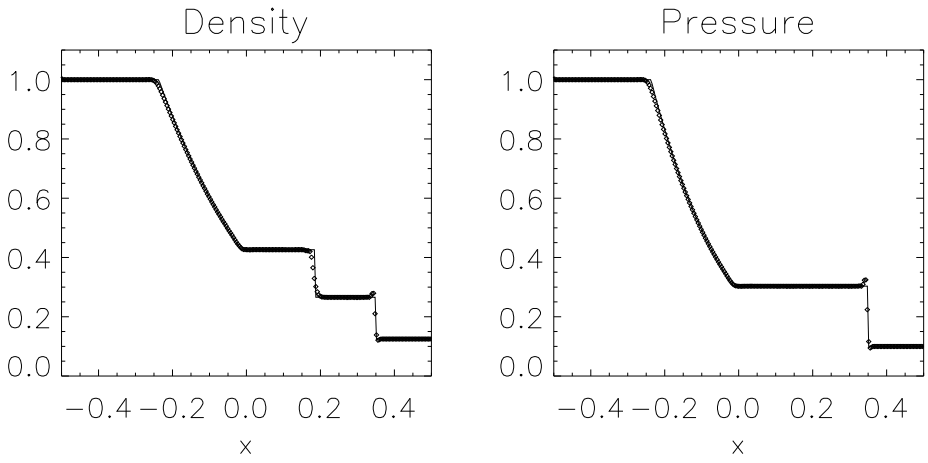


図 5.1: Sod の衝撃波管問題。グリッド数は 256。実線は解析解。

のようになる。

5.3.3　テスト問題

Sod の衝撃波管問題

Sod (1978) の 1 次元流体衝撃波管問題を行った。初期条件は衝撃波面の前後で、

$$\rho_R = 0.125$$
$$P_R = 0.1$$
$$\rho_L = 1.0$$
$$P_L = 1.0$$

とした。比熱比は $\gamma = 1.4$。境界条件は両側とも固定壁とした。図 5.1 では密度と圧力がプロットしてある。

Orszag-Tang テスト

Orszag & Tang (1979) によって提案された 2 次元磁気流体のテストを行った。領域は $0 \leq x \leq 1$, $0 \leq y \leq 1$、初期条件は、

$$\begin{aligned} \rho &= 25/(36\pi) \\ P &= 5/(12\pi) \\ V_x &= -\sin(2\pi y) \\ V_y &= \sin(2\pi x) \\ B_x &= -\sin(2\pi y) \\ B_y &= \sin(4\pi x) \end{aligned}$$

比熱比は $\gamma = 5/3$。境界条件は周期境界とした。結果は図 5.2 のようになった。これは Ryu et al. (1998) の fig. 3 と比較出来るように作ってある。

5.4 輻射輸送部分

5.4.1 Short Characteristic 法

ある光線方向 $\boldsymbol{\mu}$ について輻射輸送方程式を空間積分する。いくつかの方法があるが、ここではローカルで、通信の少ない Short Characteristic 法 (e.g. Kunasz & Auer 1988) を用いた。並列化は MPI で行った。

密度および不透明度が幾何学的厚みに対して局所的に線形だと仮定すると、光学的厚み $\Delta\tau$ は以下のように計算できる。

$$\rho(s) = \rho_0 + \rho_1' s \tag{5.4.1}$$

$$\kappa(s) = \kappa_0 + \kappa_1' s \tag{5.4.2}$$

$$\begin{aligned} \Delta\tau &= \int_0^{\Delta s} \rho(s)\kappa(s) ds \\ &= \rho_0\kappa_0\Delta s + \frac{1}{2}(\rho_0\kappa_1' + \rho_1'\kappa_0)(\Delta s)^2 + \frac{1}{3}\rho_1'\kappa_1'(\Delta s)^3 \end{aligned} \tag{5.4.3}$$

輻射強度は、同様に源泉関数を光学的厚みに対して線形だとして、輻射輸送方程式の形式解を用いると、

$$S(\tau) = S_0 + S_1'\tau \tag{5.4.4}$$

$$\begin{aligned} \Delta I &= \exp(-\Delta\tau) \int_0^{\Delta\tau} S(\tau)\exp(\tau) d\tau \\ &= S_0 W_0 + S_1' W_1 \end{aligned} \tag{5.4.5}$$

$$W_0 = 1 - \exp(-\Delta\tau) \tag{5.4.6}$$

$$W_1 = \Delta\tau - W_0 \tag{5.4.7}$$

$$I(\Delta\tau) = I(0)\exp(-\Delta\tau) + \Delta I \tag{5.4.8}$$

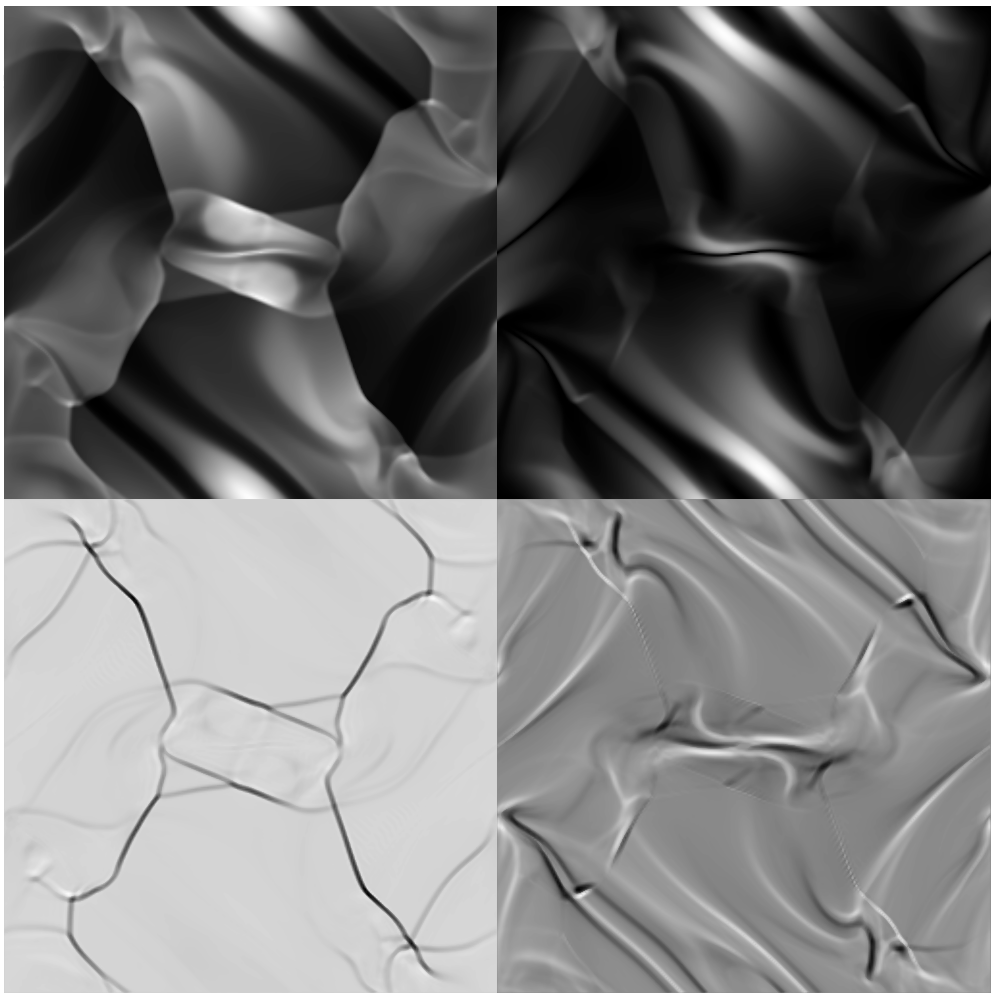


図 5.2: Orszag-Tang テストの結果（$t = 0.48$）。グリッド数は 256×256。左上：ガス圧、 右上：磁気圧、 左下：速度場の発散、 右下：渦度。

のように求められる。

積分の順序は、以下の様になる。

1. 輻射強度の初期値、境界値を設定する。
2. グリッド内部で計算。
   (a) 付近のグリッドから風上での輻射強度 $I(0)$ の値を補間する。
   (b) 式 (5.4.3)、(5.4.8) を用いてその場所での輻射強度の値 $I(\Delta\tau)$ を計算する。
3. 各グリッドでステップ 2 を繰り返し、コア内の全てのグリッドで輻射強度を計算する。
4. コア間で通信し、境界値を新しくする。
5. 輻射強度の変化が十分小さくなるまで、ステップ 2 から 4 を繰り返す。

シリアルプログラムでは必要ない反復計算が、並列化した場合には必要になる。

5.4.2 テスト問題

古典的な解析解のある 1 次元平行平板大気問題 (e.g., Rybicki & Lightman 1979) を解いた。z 軸に対し角度 $\mu_z = \pm 1/3$ で伝播する 2 本の光線を考える。境界条件を下部境界で $I(\mu_z = +1/3) = 1$、上部境界で $I(\mu_z = -1/3) = 1 - \exp(-\sqrt{3}\tau^*_{top})$ とすると、解析解は

$$J_{an} = 1 - \frac{1}{2}\exp\left(-\sqrt{3}\tau^*\right)$$

となる。ここで、τ^* は光学的深さを表す。

結果は図 5.3 の様になった。ここで、Q_F は輻射フラックスから求めた輻射加熱項 (式 (5.2.12) 中辺)、Q_J は角度平均した輻射強度から求めた輻射加熱項 (式 (5.2.12) 右辺)、Q_{an} は解析解から求めた輻射加熱項である。

5.5 計算例

太陽表面で最も小さい対流スケールである粒状斑 (水平スケール 2 Mm 程度) を計算してみた。磁場は入れていない。領域は $6 \times 6 \times 2$ Mm^3 、グリッド数は $288 \times 288 \times 100$ である。光線の角度方向には 24 本をとった。

図 5.4 では、鉛直上向きの輻射強度や太陽表面 (視線方向の光学的深さ $\tau^*_{Ross} = 1$) での鉛直速度などの物理量の水平分布を描いている。上昇流がある領域は対流層内部の高いエントロピーを持っているため明るい。水平流はターンオーバーして発散し、対流セルの境界に向かう。暗い場所では冷えた水平流が収束し、渦度を持ちながら再び対流層内部に沈み込む。

計算時間のうち、輻射輸送の計算にかかる時間は約 3 割程度で済んだ。これは、局所熱力学平衡と非散乱を仮定したため、問題として陽に積分することで解けてしまうことと、対流層のほとんどで輻射強度が源泉関数に充分近いため、初期値を適切に選ぶことで並列化に伴う反復計算も少なくて済むことによる。

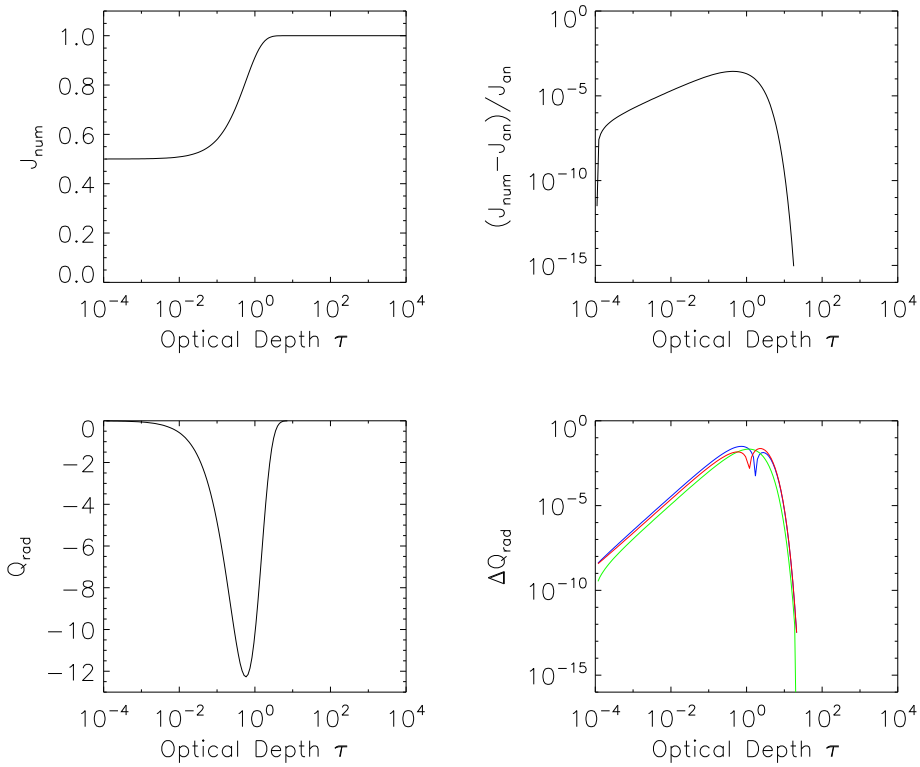


図 5.3: 輻射輸送のテスト問題 (1 次元平行平板大気)。グリッド数は、$40 \times 80 \times 160$(z 方向に 160)。左上：得られた輻射強度の角度平均 J、右上：J の解析解からの相対誤差、左下：得られた輻射加熱項 Q_{rad}、右下：輻射加熱項 Q_F、Q_J、Q_{an} の間のずれ (赤：$Q_F - Q_{an}$、緑：$Q_J - Q_{an}$、青：$Q_F - Q_J$)。

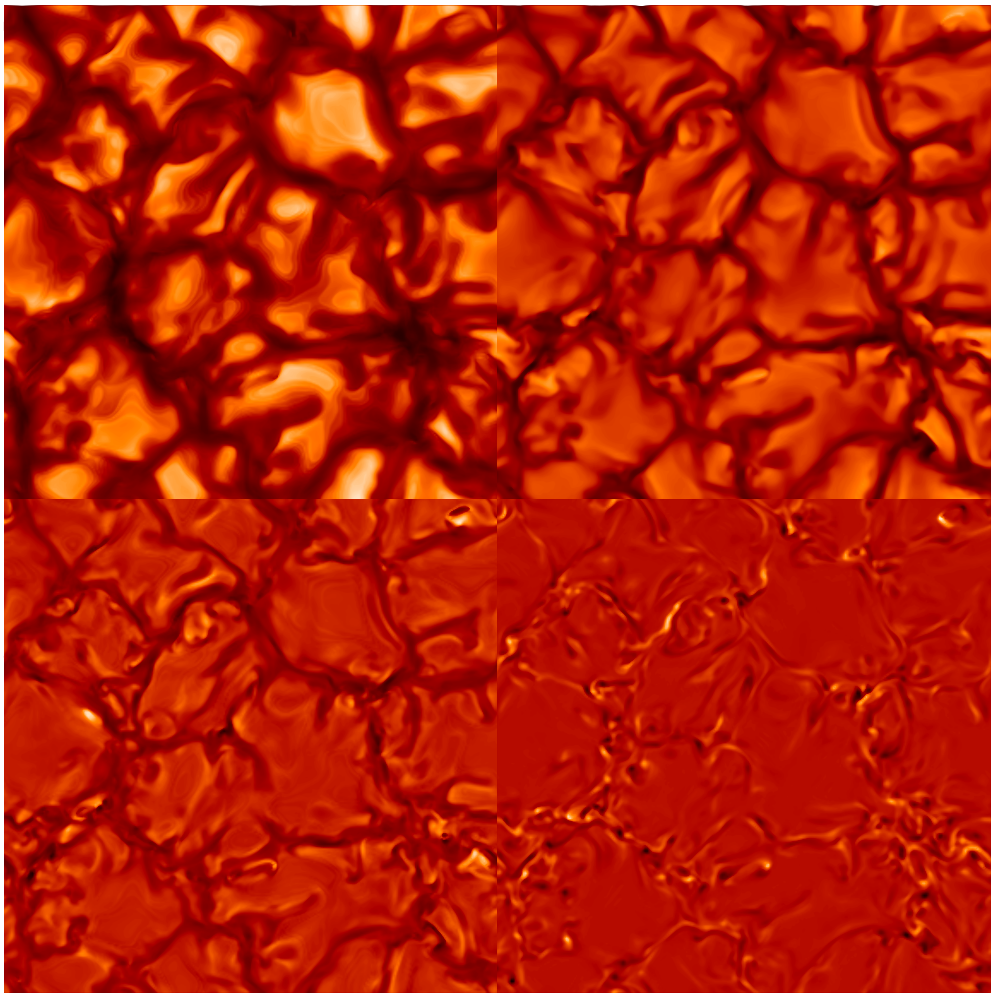


図 5.4: 太陽表面対流のシミュレーション結果。グリッド数は $288 \times 288 \times 100$。左上：上部境界での輻射強度 ($\mu_z = 1$)、 右上：鉛直速度、 左下：水平発散、 右下：鉛直渦度。輻射強度以外は表面 ($\tau^*_{Ross} = 1$) での物理量を描いている。 視野は全て 6×6 Mm^2 。

関連図書

[1] Iglesias, C. A. & Rogers, F. J. 1996, ApJ, 464, 943

[2] Kunasz, P. B., & Auer, L. 1988, J. Quant. Spectrosc. Radiat. Transfer, 39, 67

[3] Orszag, S. A. & Tang, C. M. 1979, Journal of Fluid Mechanics Digital Archive, 90, 129

[4] Rempel, M., Schüssler, M., & Knölker, M. 2009, ApJ, 691, 640

[5] Rogers, F. J., Swenson, F. J., & Iglesias, C. A. 1996, ApJ, 456, 902

[6] Rybicki, G. B. & Lightman, A. P. 1979, Radiative processes in astrophysics (Wiley-Interscience, New York)

[7] Ryu, D., Miniati, F., Jones, T. W., & Frank, A. 1998, ApJ, 509, 244

[8] Sod, G. A. 1978, Journal of Computational Physics, 27, 1

[9] Vögler, A. 2003 Ph.D. Thesis, Gottingen University

[10] Vögler, A., Shelyag, S., Schüssler, M., et al. 2005, A&A, 429, 335

第6章　ブラソフシミュレーション技法

簑島敬（海洋研究開発機構）

無衝突プラズマの運動論的シミュレーション技法であるブラソフシミュレーションについて紹介する。ブラソフシミュレーションは、これまで広く用いられてきた粒子法が持つ問題点を解決できる、大規模並列計算機に適した高精度の計算手法である。本記事では特に、磁化プラズマのブラソフシミュレーションに焦点を当て、ブラソフ方程式及びマクスウェル方程式の数値解法を解説し、計算例を紹介する。

6.1　イントロダクション

無衝突プラズマの運動論的シミュレーションは、宇宙プラズマ分野のみならず、様々な分野で精力的に研究されている。これまで広く用いられている手法は、粒子法である (1)。粒子法とは、本来は膨大な数のプラズマ粒子を、少数の「超粒子」で代表させ、それらの軌道はニュートンの運動方程式に従ってラグランジュ的に更新し、一方で電磁場はグリッド上に離散化し、オイラー的に更新する手法である。この手法は、粒子数が少数（=少ない計算機資源）でもシミュレーションが安定に実行でき、比較的満足な結果が得られること、コーディングが簡易であることなどの長所がある。一方で、プラズマ分布関数を有限個の超粒子で近似することから、数値ノイズが非常に大きく、これは粒子数の平方根でしか改善されないため、スケーラビリティが良くない。また、粒子と電磁場で変数の取扱いが異なるため、並列計算の効率化に問題がある。

これら粒子法固有の問題点を解決すべく提唱されている手法が、ブラソフシミュレーションである。この手法では、プラズマ分布関数を 6 次元位相空間で直接離散化し、電磁場と同様オイラー的に更新する。解くべき方程式は、無衝突プラズマの第一原理方程式であるブラソフ方程式

$$\frac{\partial f_s}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f_s}{\partial \boldsymbol{x}} + \frac{q_s}{m_s}\left(\boldsymbol{E} + \frac{\boldsymbol{v} \times \boldsymbol{B}}{c}\right) \cdot \frac{\partial f_s}{\partial \boldsymbol{v}} = 0, \tag{6.1.1}$$

及びマクスウェル方程式

$$\frac{\partial \boldsymbol{E}}{\partial t} = c\nabla \times \boldsymbol{B} - 4\pi \boldsymbol{j}, \tag{6.1.2}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -c\nabla \times \boldsymbol{E}, \tag{6.1.3}$$

$$\nabla \cdot \boldsymbol{E} = 4\pi\rho, \tag{6.1.4}$$

$$\nabla \cdot \boldsymbol{B} = 0. \tag{6.1.5}$$

である。ここで $f_s(\boldsymbol{x},\boldsymbol{v})$ は粒子種 s の分布関数、q,m は電荷及び質量、c は光速、$\boldsymbol{E}(\boldsymbol{x}),\boldsymbol{B}(\boldsymbol{x})$ は電磁場である。電荷密度 $\rho(\boldsymbol{x})$ 及び電流密度 $\boldsymbol{j}(\boldsymbol{x})$ は分布関数の速度空間モーメント

$$\rho(\boldsymbol{x}) = \sum_s q_s \int f_s(\boldsymbol{x},\boldsymbol{v})d\boldsymbol{v}, \tag{6.1.6}$$

$$\boldsymbol{j}(\boldsymbol{x}) = \sum_s q_s \int \boldsymbol{v} f_s(\boldsymbol{x},\boldsymbol{v})d\boldsymbol{v}, \tag{6.1.7}$$

を取ることで与えられ、これらを介してプラズマと電磁場が相互作用する。

ブラソフシミュレーションは、分布関数を直接解くため、粒子法では数値ノイズで埋もれてしまうような分布関数の詳細を明らかにできる可能性がある。また、プラズマも電磁場もオイラー変数として扱うため、並列計算の効率化が粒子法に比べて容易である。但し、最大で 6 次元の位相空間を扱うため、膨大な計算機資源を必要とする。以上の事から、ブラソフシミュレーションは、大規模並列計算機向けの技術といえる。

6.2　ブラソフ方程式の数値解法

まず、ブラソフ方程式の数値解法について解説する。式 (6.1.1) をそのまま解くのは難しいので、次のように実空間と速度空間の方程式に分離して、別々に更新することを考える。

$$\frac{\partial f_s}{\partial t} + \boldsymbol{v}\cdot\frac{\partial f_s}{\partial \boldsymbol{x}} = 0, \tag{6.2.1}$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s}\boldsymbol{E}\cdot\frac{\partial f_s}{\partial \boldsymbol{v}} = 0, \tag{6.2.2}$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s}\frac{\boldsymbol{v}\times\boldsymbol{B}}{c}\cdot\frac{\partial f_s}{\partial \boldsymbol{v}} = 0. \tag{6.2.3}$$

式 (6.2.1) は実空間での移流（自由流）を表している。移流速度 $\boldsymbol{v}$ がここでの独立変数 $(\boldsymbol{x},t)$ に依らないので、速度一定の最も単純な移流方程式である。式 (6.2.2) は速度空間での移流（電場加速）を表しており、これも速度一定の移流方程式である。式 (6.2.3) は速度空間での原点を中心とした剛体回転（ジャイロ運動）を表している。よって、いずれの式も、数多く開発されている双曲型偏微分方程式の数値計算スキームを用いて解くことが可能である。精度と計算コストのトレードオフを考慮して、適切なスキームを選択する。以上は、文献 (2) で提唱された Splitting スキームを、磁場がある場合に拡張したものである。

適切な時間刻み及び更新手順を選択する事で、シンプレクティックな時間積分[1]を行うことが出来る (7)。例えば、$t = n\Delta t$ における分布関数 f^n について

1. (6.2.1) を半ステップ更新。$f^n \to f^*$.
2. (6.2.2) を半ステップ更新。$f^* \to f^{**}$.
3. (6.2.3) を 1 ステップ更新。$f^{**} \to f^{***}$.

[1]ハミルトン系において、それに近い別のハミルトン系のハミルトニアン（影のハミルトニアン）が保存される手法。元の系のハミルトニアンとの誤差は Δt^m のオーダーで、時間的に振動するものの単調増大しない特徴がある。

4. (6.2.2) を半ステップ更新。$f^{***} \rightarrow f^{****}$.

5. (6.2.1) を半ステップ更新。$f^{****} \rightarrow f^{n+1}$.

とすれば時間 2 次精度で更新される。ただし、過程 2 から 4 の際に必要な電磁場は $t = (n + 1/2)\Delta t$ における値 $(\boldsymbol{E}^{n+1/2}, \boldsymbol{B}^{n+1/2})$ を用いる。これは、式 (6.2.1)-(6.2.3) をラグランジュ形式に書き直せば、粒子法における Buneman-Boris 法に相当することがわかる。

6.2.1 マルチモーメント移流法

ブラソフ方程式は拡散項の無い方程式であり、運動論的プラズマ中には様々なモードの波動が存在するため、その記述には高精度の数値計算スキームを必要とする。これまで様々な高精度スキームが開発・適用されてきた。それらの比較が文献 (3; 4) に紹介されている。

ブラソフシミュレーションは最大で 6 次元の位相空間を扱うため、膨大な計算機資源を必要とする。そのため、研究は実空間 1 次元、速度空間 1 次元の静電問題（$\boldsymbol{B}$=0）から始まった。近年の計算機能力の向上により、多次元の電磁ブラソフシミュレーションが可能になってきたが (12; 14)、物理的に意味のある結果を得ることは難しく、2012 年現在でも非磁化プラズマや旋回中心近似プラズマといった限られた問題を扱うことが多い。その主な原因は、式 (6.2.3) で表される剛体回転問題を正しく長時間安定に解くことが困難だからである。従来のスキームでは、剛体回転に伴って、数値分散や数値拡散が急速に発生する[2]。数値分散は速度分布関数の微細構造を発生させ、これは不要なプラズマ波動を発生させる。数値拡散はプラズマの数値加熱に相当し、圧力上昇、プラズマ波動の抑制をもたらす。運動論的シミュレーションでは、現象を追跡するため、イオンジャイロ周期の数～数十倍程度以上の時間を計算するので、電子は磁場の周りを数千回転程度ジャイロ運動することになる[3]。磁化プラズマのブラソフシミュレーションのためには、剛体回転問題を正しく長時間安定に解くスキームが必須である。

そこで著者らは、ブラソフシミュレーションのための新しいスキームとして、マルチモーメント移流（MMA）法を提唱している。この手法は、速度分布関数 $f(\boldsymbol{v})$ に加えて、その 0 次から 2 次までのモーメント

$$M^m(\boldsymbol{v}) = \frac{1}{m!}\int^{\boldsymbol{v}} \boldsymbol{v'}^m f d\boldsymbol{v'}, \quad (m = 0, 1, 2) \tag{6.2.4}$$

も変数として扱い、各々の支配方程式に従って独立に更新する。1 次元の場合は、

$$\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial v} = 0, \tag{6.2.5}$$

$$\frac{\partial M^m}{\partial t} + \frac{a}{m!}\int dv \frac{\partial}{\partial v}(v^m f) = \begin{cases} 0, & (m = 0) \\ \frac{a}{(m-1)!}\int v^{m-1} f dv, & (m = 1, 2) \end{cases} \tag{6.2.6}$$

となる。物理量は、グリッド上の分布関数 $f_i = f(v_i)$ 及びグリッド内の区分化モーメント

$$M^m_{i+1/2} = \frac{1}{m!}\int_{v_i}^{v_{i+1}} v^m f dv, \quad (m = 0, 1, 2) \tag{6.2.7}$$

[2]剛体回転問題を数回転程度解けることが、スキームのベンチマークテストになるほどである。

[3]実際は、計算機資源の制約から、現実より小さなイオン/電子質量比を用いることが多いが、それでも数十～数百回転程度は計算する必要がある。

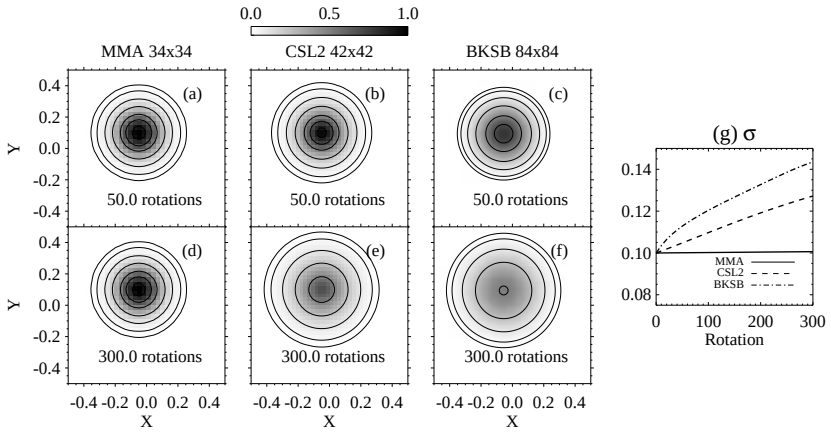


図 6.1: ガウス分布の 2 次元移流回転問題。 (a,b,c) MMA 法（34^2 グリッド）、CIP-CSL2 法（42^2 グリッド）、Backsubstitution 法（84^2 グリッド）で計算した 50 回転後の分布。 (d,e,f) 300 回転後の分布。(g) 分布の標準偏差の時間変化。実線、破線、点破線がそれぞれ MMA 法、CIP-CSL2 法、Backsubstitution 法で得られた結果。

を記憶し、グリッド内の分布関数はこれらを用いて局所多項式で補間する。式 (6.2.5) と (6.2.6) を更新する際は、分布の積分値（質量）のみならず、中心値と分散の保存を保証するように行う。多次元の場合も、同様の方法でスキームを設計する事が出来る。詳細は文献 (8; 9) を参考いただきたい。

図 6.1 は、2 次元移流回転問題（式 (6.2.2) と (6.2.3)）について、MMA 法、保存型 CIP（CIP-CSL2）法 (13)、Backsubstitution 法 (11) の計算比較を示している。各々のスキームで更新量の数が異なるので、メモリ使用量を揃えるために異なるグリッド数を用いている（MMA 法が一番粗い）。MMA 法では、数値拡散をほとんど発生させること無く、移流回転問題を計算できることが分かる。

本記事で紹介するブラソフシミュレーションでは、速度空間の更新に対して MMA 法を用いる。実空間の更新に対しては CIP-CSL2 法を用いる。

6.3　マクスウェル方程式の数値解法

次に、マクスウェル方程式 (6.1.2),(6.1.3) の数値解法について解説する。電磁場は粒子法でもオイラー的に更新されるため、同様の手法を用いることが出来る。例えば、古典的な Finite-Difference Time-Domain（FDTD）法や、フーリエ級数展開を用いたスペクトル法などが良く用いられる。双曲型偏微分方程式の高精度数値計算スキームの一つである CIP 法を用いることも出来る (10)。

ここでは、文献 (5; 6) で用いられている陰的解法について紹介する。マクスウェル方程式 (6.1.2),(6.1.3) を時間方向に差分近似すると、

$$\delta \boldsymbol{E} = \alpha c \Delta t \nabla \times \delta \boldsymbol{B} + c \Delta t \nabla \times \boldsymbol{B}^n - 4\pi \Delta t \boldsymbol{j}^{n+1/2}, \tag{6.3.1}$$

$$\delta \boldsymbol{B} = -\alpha c \Delta t \nabla \times \delta \boldsymbol{E} - c \Delta t \nabla \times \boldsymbol{E}^n, \tag{6.3.2}$$

と、$(\delta \boldsymbol{E}, \delta \boldsymbol{B}) = (\boldsymbol{E}^{n+1} - \boldsymbol{E}^n, \boldsymbol{B}^{n+1} - \boldsymbol{B}^n)$ についての方程式が得られる。ここで、$\alpha = 1$ の場合は 1 次精度の陰的解法、$\alpha = 0.5$ の場合は 2 次精度のクランク・ニコルソン法になる。1 次元 $(\partial/\partial y = \partial/\partial z = 0)$

の場合は、

$$\delta E_x = -4\pi\Delta t j_x^{n+1/2}, \tag{6.3.3}$$
$$\delta B_x = 0, \tag{6.3.4}$$
$$\delta E_y = -\alpha c\Delta t\frac{\partial\delta B_z}{\partial x} - c\Delta t\frac{\partial B_z^n}{\partial x} - 4\pi\Delta t j_y^{n+1/2}, \tag{6.3.5}$$
$$\delta B_y = \alpha c\Delta t\frac{\partial\delta E_z}{\partial x} + c\Delta t\frac{\partial E_z^n}{\partial x}, \tag{6.3.6}$$
$$\delta E_z = \alpha c\Delta t\frac{\partial\delta B_y}{\partial x} + c\Delta t\frac{\partial B_y^n}{\partial x} - 4\pi\Delta t j_z^{n+1/2}, \tag{6.3.7}$$
$$\delta B_z = -\alpha c\Delta t\frac{\partial\delta E_y}{\partial x} - c\Delta t\frac{\partial E_y^n}{\partial x}, \tag{6.3.8}$$

となる。式 (6.3.3) は静電成分であり、直ちに計算できる[4]。式 (6.3.5)-(6.3.8) を解くために、z 成分は整数グリッド、y 成分は半整数グリッド上に定義し、空間微分を 2 次中央差分で近似すると、

$$\delta E_{y,i+1/2} = -p\left(\delta B_{z,i+1} - \delta B_{z,i}\right) + K_{y,i+1/2}, \tag{6.3.9}$$
$$\delta B_{y,i+1/2} = p\left(\delta E_{z,i+1} - \delta E_{z,i}\right) + L_{y,i+1/2}, \tag{6.3.10}$$
$$\delta E_{z,i} = p\left(\delta B_{y,i+1/2} - \delta B_{y,i-1/2}\right) + K_{z,i}, \tag{6.3.11}$$
$$\delta B_{z,i} = -p\left(\delta E_{y,i+1/2} - \delta E_{y,i-1/2}\right) + L_{z,i}, \tag{6.3.12}$$

となる。ここで $p = \alpha c\Delta t/\Delta x$ であり、K, L は既知量で、

$$K_{y,i+1/2} = -\frac{c\Delta t}{\Delta x}\left(B_{z,i+1}^n - B_{z,i}^n\right) - 4\pi\Delta t j_{y,i+1/2}^{n+1/2}, \tag{6.3.13}$$
$$K_{z,i} = \frac{c\Delta t}{\Delta x}\left(B_{y,i+1/2}^n - B_{y,i-1/2}^n\right) - 4\pi\Delta t j_{z,i}^{n+1/2}, \tag{6.3.14}$$
$$L_{y,i+1/2} = \frac{c\Delta t}{\Delta x}\left(E_{z,i+1}^n - E_{z,i}^n\right), \tag{6.3.15}$$
$$L_{z,i} = -\frac{c\Delta t}{\Delta x}\left(E_{y,i+1/2}^n - E_{y,i-1/2}^n\right), \tag{6.3.16}$$

である。式 (6.3.9)-(6.3.12) を整理すると、以下の式が得られる。

$$-p^2\delta E_{z,i-1} + \left(1+2p^2\right)\delta E_{z,i} - p^2\delta E_{z,i+1} = p\left(L_{y,i+1/2} - L_{y,i-1/2}\right) + K_{z,i}, \tag{6.3.17}$$
$$-p^2\delta B_{z,i-1} + \left(1+2p^2\right)\delta B_{z,i} - p^2\delta B_{z,i+1} = -p\left(K_{y,i+1/2} - K_{y,i-1/2}\right) + L_{z,i}. \tag{6.3.18}$$

これらは 3 重対角な連立 1 次方程式のため、Thomas 法を用いて直ちに解くことができる[5]。$(\delta E_z, \delta B_z)$ を式 (6.3.9) と (6.3.10) に代入して $(\delta E_y, \delta B_y)$ を得る。

この手法は、時間空間精度は FDTD 法と同じ（$\alpha = 0.5$ の場合）だが、陰的に解くため、光速の CFL 条件に束縛されない。よって、プラズマ速度が光速に比べ非常に小さいパラメータの計算において、時間刻みを長く取っても安定に計算できる。

[4] このようにして決定した静電場が、ガウス則 (6.1.4) を満たしているよう注意する必要がある。電荷と電流が連続の式を満たすように決定されていれば、これは満たされる。そうでない場合は、電位に関するポアソン方程式を解いて、静電成分を修正する。

[5] 多次元の場合は共役勾配法などの反復法を用いる。

6.4　変数と方程式の規格化

数値シミュレーションでは、変数に実際の物理量を代入することは行わず、規格化した値を用いる。ブラソフシミュレーションでは電子スケールの現象を追跡するので、電子の特徴的な量で規格化すれば良い。ここでは、

$$t : \omega_{pe}^{-1}$$

$$\boldsymbol{v} : c$$

$$\boldsymbol{x} : \lambda_D$$

$$q : e\ (\text{電荷素量})$$

$$m : m_e\ (\text{電子質量})$$

$$\boldsymbol{E}, \boldsymbol{B} : B_0\ (\text{背景磁場強度})$$

$$f : n_0 c^{-3}$$

$$\rho : en_0$$

$$\boldsymbol{j} : ecn_0$$

を用いる。ここで、$\omega_{pe} = \sqrt{4\pi n_0 e^2/m_e}$ は電子プラズマ振動数、$\lambda_D = \sqrt{kT_e/4\pi n_0 e^2}$ はデバイ長、n_0 は背景密度、T_e は電子温度である。すると、方程式系 (6.1.1)-(6.1.7) は規格化された量を用いて次のように表せる。

$$\frac{\partial f_e}{\partial t} + \frac{\sqrt{2}\boldsymbol{v}}{v_{e;th}} \cdot \frac{\partial f_e}{\partial \boldsymbol{x}} - \frac{\omega_{ge}}{\omega_{pe}} (\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f_e}{\partial \boldsymbol{v}} = 0, \tag{6.4.1}$$

$$\frac{\partial f_i}{\partial t} + \frac{\sqrt{2}\boldsymbol{v}}{v_{e;th}} \cdot \frac{\partial f_i}{\partial \boldsymbol{x}} + Z_i \frac{m_e}{m_i} \frac{\omega_{ge}}{\omega_{pe}} (\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f_i}{\partial \boldsymbol{v}} = 0, \tag{6.4.2}$$

$$\frac{\partial \boldsymbol{E}}{\partial t} = \frac{\sqrt{2}}{v_{e;th}} \nabla \times \boldsymbol{B} - \frac{\omega_{pe}}{\omega_{ge}} \boldsymbol{j}, \tag{6.4.3}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\frac{\sqrt{2}}{v_{e;th}} \nabla \times \boldsymbol{E}, \tag{6.4.4}$$

$$\nabla \cdot \boldsymbol{E} = \frac{v_{e;th}}{\sqrt{2}} \frac{\omega_{pe}}{\omega_{ge}} \rho, \tag{6.4.5}$$

$$\rho = \int (Z_i f_i - f_e)\, d\boldsymbol{v}, \tag{6.4.6}$$

$$\boldsymbol{j} = \int \boldsymbol{v}\, (Z_i f_i - f_e)\, d\boldsymbol{v}. \tag{6.4.7}$$

ここで、$\omega_{ge} = eB_0/m_e c$ は電子ジャイロ振動数、$v_{e;th} = \sqrt{\beta_e}\omega_{ge}/\omega_{pe}$ は光速で規格化された電子熱速度、Z_i はイオンの電荷数である。熱速度はプラズマベータ値 $\beta_s = 8\pi n_0 k T_s/B_0^2$ をパラメータとして与える。イオンと電子の質量比 m_i/m_e 及び電子ジャイロ振動数とプラズマ振動数の比 ω_{ge}/ω_{pe} は運動論的シミュレーションにおける重要なパラメータである。

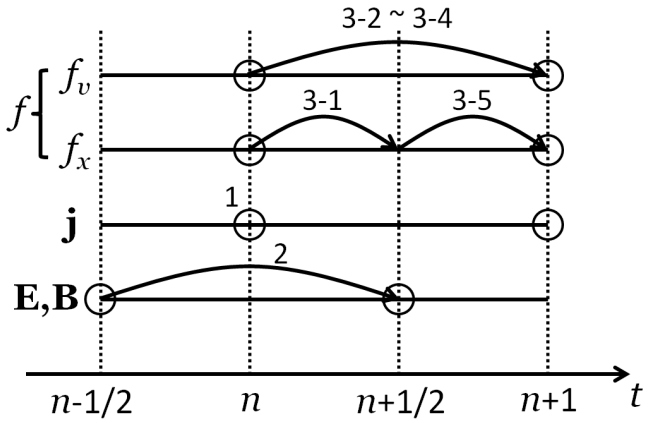


図 6.2: ブラソフ-マクスウェル系の時間発展のフローチャート。f_v, f_x はそれぞれ分布関数の速度空間及び実空間での更新を表す。

6.5 ブラソフ-マクスウェル系の時間更新

6.2-6.3 章で紹介した手法を組み合わせることで、ブラソフ-マクスウェル系の時間発展を追跡することが出来る。ここでは、系全体の時間精度が 2 次になるようなフローチャート（図 6.2）を紹介する。

0. プラズマ分布関数は $t = n\Delta t$ に、一方電磁場は $t = (n-1/2)\Delta t$ に定義する。$(f^n, \boldsymbol{E}^{n-1/2}, \boldsymbol{B}^{n-1/2})$.

1. 分布関数 f^n から電流密度 $\boldsymbol{j}^n$ を計算する。

2. $\boldsymbol{j}^n$ を用いて、$t = (n-1/2)\Delta t$ における電磁場を 1 ステップ更新する。$(\boldsymbol{E}^{n-1/2}, \boldsymbol{B}^{n-1/2}) \rightarrow (\boldsymbol{E}^{n+1/2}, \boldsymbol{B}^{n+1/2})$.

3. $(\boldsymbol{E}^{n+1/2}, \boldsymbol{B}^{n+1/2})$ を用いて、6.2 章で紹介した時間積分法に従って $t = n\Delta t$ における分布関数を 1 ステップ更新する。$f^n \rightarrow f^{n+1}$.

以上で、プラズマと電磁場が 1 ステップ分更新される。これは、粒子法で用いられる Leap-Frog 法に相当する。

陽的なスキームを使う限り、CFL 条件を満たさなければならない。ブラソフシミュレーションでは、ブラソフ方程式の実空間の更新、速度空間の更新、マクスウェル方程式の更新、それぞれに対しての CFL 条件がある。速度空間の最大値を $v_{\max}$ とすると、

$$\text{実空間移流：} \quad v_{\max}\Delta t \leq \Delta x, \tag{6.5.1}$$

$$\text{電場加速：} \quad \frac{q_s E}{m_s}\Delta t \leq \Delta v, \tag{6.5.2}$$

$$\text{ジャイロ運動：} \quad \frac{q_s v_{\max} B}{m_s c}\Delta t \leq \Delta v, \tag{6.5.3}$$

$$\text{マクスウェル方程式：} \quad c\Delta t \leq \Delta x, \tag{6.5.4}$$

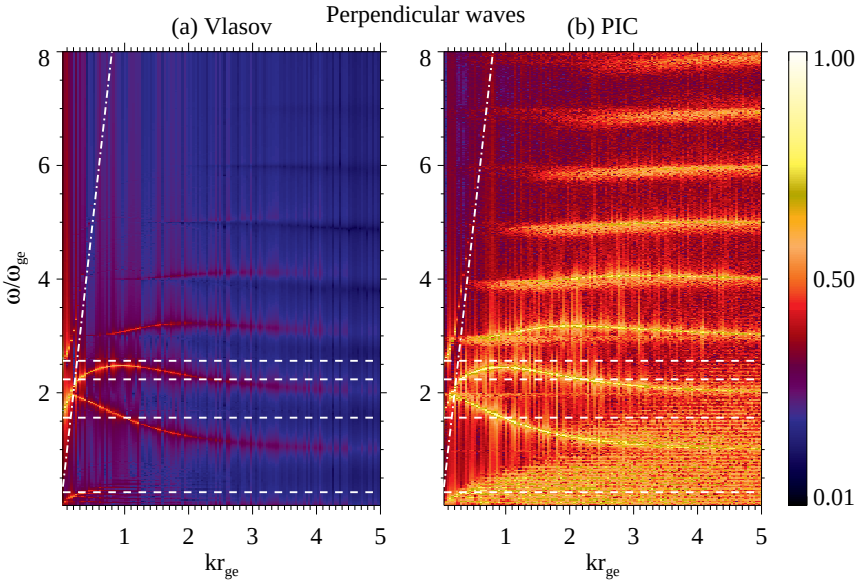


図 6.3: 磁場に垂直方向に伝播する線形波動の分散関係図。(a) ブラソフシミュレーションの結果。(b) 粒子法の結果。カラースケール（任意値）は静電成分 E_x のフーリエ振幅。横軸、縦軸はそれぞれ電子ジャイロ半径、ジャイロ周波数で規格化された波数及び周波数。点線は上から、右カットオフ周波数、高域混成周波数、左カットオフ周波数、低域混成周波数を表す。点破線は真空中の光速の分散関係を表す。

を満たす様に、実空間グリッド幅 Δx, 速度空間グリッド幅 Δv, 時間ステップ Δt を決定する。これらのうち厳しい条件は、式 (6.5.3) と (6.5.4) である[6]。加えて、図 6.2 ではシステムを陽的に更新しているので、プラズマ振動を分解する必要がある。よって

$$\omega_{pe}\Delta t \leq 1, \tag{6.5.5}$$

も満たさなければならない。

6.6　ブラソフシミュレーション

磁化プラズマのブラソフシミュレーションの例について紹介する[7]。実空間は x 方向のみの 1 次元とし、速度空間は 2 次元とする。すなわち $\boldsymbol{v} = (v_x, v_y, 0)\,, \boldsymbol{E} = (E_x, E_y, 0)\,, \boldsymbol{B} = (0, 0, B_z)$ である。プラズマは陽子（$Z_i = 1$）と電子で構成されている。

6.6.1　垂直伝播波動

まず線形問題として、静止しているプラズマ中の磁場に垂直方向に伝播する波動の計算を行った。シミュレーションパラメータは、$m_i/m_e = 16$、$\omega_{ge}/\omega_{pe} = 0.5$、$\beta_i = \beta_e = 0.04$、実空間長は 512 デバイ長、

[6] 6.3 章で紹介した陰的解法を用いれば、条件 (6.5.4) は不要である

[7] 最も基本的な問題である、実空間 1 次元、速度空間 1 次元の静電問題については、多くの論文が出版されているため、本記事では省略する。例えば文献 (4) などを参照いただきたい。

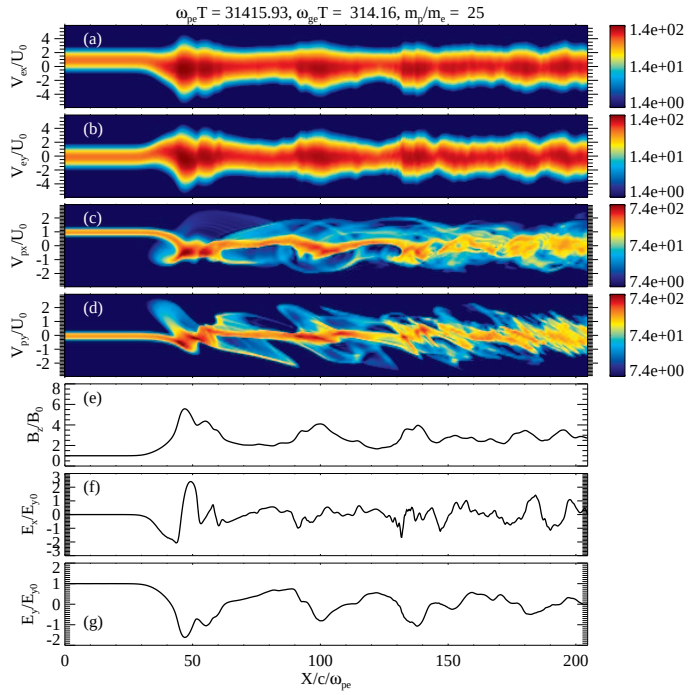


図 6.4: 1 次元垂直衝撃波のブラソフシミュレーション結果。(a,b) 電子位相空間分布 ($f_e(v_x, x), f_e(v_y, x)$)。(c,d) イオン位相空間分布 ($f_i(v_x, x), f_i(v_y, x)$)。(e,f,g) 電磁場 (B_z, E_x, E_y)。横軸は電子慣性長で規格化された空間長。速度は入射プラズマ速度、電場は上流での値 ($E_y = v_x B_z/c$) で規格化されている。

速度空間は電子 · イオン共に熱速度の ±4 倍まで取った。グリッド数は実空間に 512、速度空間に 50^2 である。時間刻みは $0.05/\sqrt{2}$ である。

図 6.3(a) は、$\omega_{ge}t = 361.7$ まで計算した静電成分 E_x のフーリエ振幅 (分散関係図) を示している。線形理論で示されている電子バーンスタイン波 (高波数で $\omega = n\omega_{ge}$ ($n = 1, 2, \dots$) に漸近するブランチ) を確認できる。また、低周波 · 低波数ではイオンバーンスタイン波も確認されている。この計算では、電子が 50 回転以上ジャイロ運動しているが、MMA 法を用いたため、エネルギー誤差は 10^{-6} である。

同じパラメータ条件の粒子法による結果を図 6.3(b) に示した。粒子法は、1 セルあたり 5,000 個の粒子[8]を用いて、ブラソフシミュレーションと同程度のメモリ使用で計算を行った。比較すると、ブラソフシミュレーションでは特に高波数のノイズがほぼ皆無であることがわかる。

6.6.2 垂直衝撃波

次に非線形問題として、垂直衝撃波の計算を行った。左境界から超音速の磁化プラズマを入射し、右境界で反射させる。すると入射プラズマと反射プラズマとの間で 2 流体不安定が発生し、衝撃波に成長する。衝撃波は左方向 (上流) に伝播する。シミュレーションパラメータは、$m_i/m_e = 25$、$\omega_{ge}/\omega_{pe} = 0.01$、

[8]通常は 1 セルあたり 100 個程度である。

$\beta_i = \beta_e = 1.0$、入射プラズマのアルフベンマッハ数 $M_A = 5$（衝撃波静止系では約 7.5）、実空間長は 20,480 デバイ長、電子速度空間は入射速度の ±6 倍、イオン速度空間は入射速度の ±3 倍まで取った。グリッド数は実空間に 1,024、速度空間に 72^2 である。時間刻みは 0.1π である。

図 6.4 は、$\omega_{ge}t = 314$ におけるシミュレーション結果を示している。上から、電子位相空間分布 $(f_e(v_x, x), f_e(v_y, x))$、イオン位相空間分布 $(f_i(v_x, x), f_i(v_y, x))$、電磁場 (B_z, E_x, E_y) である。衝撃波面 $(x = 50)$ を通過して電子圧力及び磁場が増加している様子がわかる（a,b,e）。衝撃波面においては、電子とイオンの慣性差に起因して、衝撃波静電ポテンシャルが形成され、一部のイオンが反射されている（f,c）。衝撃波における保存則（ランキン-ユゴニオ条件）が満たされていることも確認されている。

この計算は、実空間グリッド幅がデバイ長の 20 倍で行った。粒子法では、実空間グリッド幅がデバイ長より長くなると数値不安定を起こすが、ブラソフシミュレーションの場合、必ずしもそうではない。そのため、ブラソフシミュレーションは、より大規模スケールの高精度計算が出来る可能性がある。

6.7　まとめと今後

無衝突プラズマの運動論的シミュレーション技法であるブラソフシミュレーションについて紹介した。ブラソフシミュレーションは、粒子法に比べ複雑なテクニックと多くの計算機資源を必要とするが、その努力に見合う結果を得ることが出来る。大規模並列計算時代に向け、今後益々必要とされる技術であり、多くのユーザによる継続的な研究開発が求められる。そこで、具体的な課題を挙げておく。

- 数値振動と拡散の抑制。著者らはマルチモーメント移流法を開発することにより、数値拡散を徹底的に抑制する事に成功したが、数値振動は抑制しきれていない。任意のシミュレーション条件で両者を完璧に抑制するスキームを開発することは不可能なので、トレードオフを考慮しつつ両者を効果的に抑制するスキームを開発することが重要である。

- 速度空間更新における CFL 条件からの解放。剛体回転問題をオイラー的に解く場合、1 回転を数百分割程度の時間刻みを取る必要があり、計算負荷が大きい。CFL 条件に束縛されずに剛体回転問題を安定に更新するスキームの開発は、計算負荷軽減はもちろん、計算科学の点でもインパクトがある。

- 速度空間への解適合格子法の適用。例えば非常に冷たいプラズマの高マッハ数衝撃波問題を扱う場合、ドリフト速度が非常に速く、また非熱的粒子加速が発生する可能性もあるので、速度空間を広く取らなければならない。一方で冷たい分布関数を分解しなければならない。よって、速度空間に大量のメッシュを必要とする。解適合格子法を用いて、例えば分布中心には細かいメッシュを用い、裾野付近には粗いメッシュを用いれば、コストを大幅に節約できる可能性がある。

- 相対論への拡張。相対論的プラズマを解く場合は、独立変数として $\boldsymbol{v}$ ではなく $\boldsymbol{u} = \gamma\boldsymbol{v}$ を用いる（γ はローレンツ因子）。この際問題になるのは、やはり回転である。ジャイロ周波数が速度に依存するようになるので（$\omega_g = eB/\gamma mc$）、剛体回転ではなくなり、速度空間にシアが生じる。よって、数値振動の抑制がより重要になると考えられる。

謝辞

本記事は、松本洋介氏（千葉大学)、天野孝伸氏（東京大学）との共同研究結果に基づいて執筆しました。

関連図書

[1] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation.* Inst. of Phys. Publishing, Bristol/Philadelphia, 1991.

[2] C. Z. Cheng and G. Knorr. The integration of the Vlasov equation in configuration space. *Journal of Computational Physics*, 22:330–351, November 1976.

[3] N. Crouseilles, M. Mehrenberger, and E. Sonnendrücker. Conservative semi-Lagrangian schemes for Vlasov equations. *Journal of Computational Physics*, 229:1927–1953, March 2010.

[4] F. Filbet and E. Sonnendrücker. Comparison of Eulerian Vlasov solvers. *Computer Physics Communications*, 150:247–266, February 2003.

[5] M. Hoshino. *Theoretical and Computational Studies of Plasma Kinetic Phenomena: Tearing Mode Instability and Foreshock Cyclotron Interaction.* PhD thesis, Univ. Tokyo, 1986.

[6] M. Hoshino. The electrostatic effect for the collisionless tearing mode. *J. Geophys. Res.*, 92:7368–7380, July 1987.

[7] A. Mangeney, F. Califano, C. Cavazzoni, and P. Travnicek. A Numerical Scheme for the Integration of the Vlasov-Maxwell System of Equations. *Journal of Computational Physics*, 179:495–538, July 2002.

[8] T. Minoshima, Y. Matsumoto, and T. Amano. Multi-moment advection scheme for Vlasov simulations. *Journal of Computational Physics*, 230:6800–6823, July 2011.

[9] T. Minoshima, Y. Matsumoto, and T. Amano. Multi-moment advection scheme in three dimension for Vlasov simulations of magnetized plasma. *ArXiv e-prints*, February 2012.

[10] Y. Ogata, T. Yabe, and k. Odagaki. An Accurate Numerical Scheme for Maxwell Equation with CIP-Method of Characteristics. *Communications in Computational Physics*, 1:311–335, April 2006.

[11] H. Schmitz and R. Grauer. Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields. *Computer Physics Communications*, 175:86–92, July 2006.

[12] H. Schmitz and R. Grauer. Kinetic Vlasov simulations of collisionless magnetic reconnection. *Physics of Plasmas*, 13(9):092309–+, September 2006.

[13] K. Takizawa, T. Yabe, and T. Nakamura. Multi-dimensional semi-Lagrangian scheme that guarantees exact conservation. *Computer Physics Communications*, 148:137–159, October 2002.

[14] T. Umeda, J.-I. Miwa, Y. Matsumoto, T. K. M. Nakamura, K. Togano, K. Fukazawa, and I. Shinohara. Full electromagnetic Vlasov code simulation of the Kelvin-Helmholtz instability. *Physics of Plasmas*, 17(5):052311–+, May 2010.

pCANS Documentation

リリース *0.1*

2012 年 03 月 28 日

Contents

1 はじめに **1**
- 1.1 **pCANS** Web site . . . 1
- 1.2 **pCANS** for developers . . . 1
- 1.3 更新情報 . . . 1
- 1.4 問い合わせ . . . 1
- 1.5 開発メンバー . . . 2
- 1.6 謝辞 . . . 2

2 イントロダクション **3**
- 2.1 まずは始めましょう . . . 3
- 2.2 動作環境 . . . 4
- 2.3 環境変数 . . . 4
- 2.4 可視化について . . . 4
- 2.5 全体の構成 . . . 5

3 1 次元 PIC コード **7**
- 3.1 シリアル版 . . . 7
- 3.2 MPI 並列版 . . . 11

4 2 次元 PIC コード **15**
- 4.1 領域分割法による並列化 . . . 15
- 4.2 パラメタ設定 . . . 16

5 物理課題 **19**
- 5.1 1 次元課題 . . . 19
- 5.2 2 次元課題 . . . 22

6 References **31**

7 Appendix **33**
- 7.1 reST の書き方について . . . 33

A Appendix **35**
A.1 reST の書き方について 35

Bibliography **37**

CHAPTER 1

はじめに

pCANS は、 CANS (Coordinated Astronomical Numeical Software) の、電磁プラズマ粒子シミュレーションコード（Electromagnetic Particle-in-Cell simulation code）パッケージです。PIC シミュレーションは無衝突プラズマを記述する第一原理シミュレーション手法で、電磁場中のプラズマ粒子の運動を自己無撞着に記述できるため、宇宙・天文現象での粒子加速・輸送問題を理解するのに役立ちます。

1.1 pCANS Web site

http://www.astro.phys.s.chiba-u.ac.jp/pcans

1.2 pCANS for developers

http://bitbucket.org/ymatumot/pcans

1.3 更新情報

- Web: http://bitbucket.org/ymatumot/pcans/changesets
- RSS: http://bitbucket.org/ymatumot/pcans/rss

1.4 問い合わせ

下記より質問、バグレポート、機能要求などを受け付けています。

https://bitbucket.org/ymatumot/pcans/issues/new

これまでの履歴は下記よりご覧になれます。

https://bitbucket.org/ymatumot/pcans/issues/

1.5 開発メンバー

- 松本洋介（千葉大学理学研究科） - 統括、KH 不安定、無衝突衝撃波
- 天野孝伸（東京大学理学系研究科） - 情報技術参与、無衝突衝撃波、２流体不安定
- 加藤恒彦（広島大学理学研究科） - Weibel 不安定、無衝突衝撃波
- 銭谷誠司（国立天文台理論研究部） - 磁気リコネクション
- 高橋博之（国立天文台 CfCA） - 磁気リコネクション
- 三好由純（名古屋大学太陽地球環境研究所） - 電子温度異方性不安定
- 星野真弘（東京大学理学系研究科） - アドバイザー
- 松元亮治（千葉大学理学研究科） - アドバイザー

1.6 謝辞

pCANS は HPCI 戦略プログラム分野５「物質と宇宙の起源と構造」によってサポートされています。

CHAPTER 2

イントロダクション

2.1 まずは始めましょう

pCANS は Mercurial でバージョン管理されています。Mercurial は Windows, linux, Mac OS に対応したバージョン管理ソフトウェアです。Mercurial のインストールは、Fedora など Redhat 系の場合、

```
$ yum install mercurial
```

Ubuntu など Debian 系の場合、

```
$ sudo apt-get install mercurial
```

で簡単にできます。Mercurial のツールを利用してコードのインストール、アップデートを行います。

まずは、 **pCANS** のレポジトリを以下のようにダウンロードします。

```
$ hg clone https://bitbucket.org/ymatumot/pcans directory-name
```

pCANS は開発途上なため、不定期に更新されます。最新版を反映させるためには、上記で指定した *directory-name* 内で、

```
$ hg pull https://bitbucket.org/ymatumot/pcans
$ hg update
```

とすれば、最新版の差分情報が反映されます。 **この際、自分で修正を加えたファイルと更新ファイルが重なる場合はマージ（merge）する必要が出てきます。またその結果、衝突（conflict）する可能性もあります。** その場合は出力に従って、text editor 等で該当個所を編集してください。

Mercurial による分散バージョン管理の詳しい解説は、 本家 もしくは 日本語解説 をご覧ください。

2.2 動作環境

以下のソフトウェア・ライブラリがシステムにインストールされている必要があります。

- Mercurial。レポジトリのダウンロード、最新版のアップデートを行うのに必要です（上述）。
- MPI(Message Passing Interface)。2 次元以上のコードは MPI 並列化バージョンのみです。
- Fortran コンパイラ。ソースコードは Fortran 90 で書かれています。デフォルトは gfortran です。
- IDL(Interactive Data Language)。可視化ルーチンは IDL で用意されています。
- OS は Linux で動作確認済み。上記ソフト・ライブラリがインストールされていれば、Windows、Mac-OS でもたぶん可だと思います。

不具合等についての問い合わせは こちら まで。

2.3 環境変数

環境変数$PCANS_DIR を **pCANS** がインストールされたディレクトリの絶対パスとし、以下のように設定します（~/pcans にインストールした場合）。

bash の場合、

```
export PCANS_DIR = ~/pcans
```

tcsh の場合、

```
setenv PCANS_DIR ~/pcans
```

2.4 可視化について

IDL による可視化ルーチンが用意されています。$PCANS_DIR/idl 内には共通プロシージャ、各課題内には専用のプロシージャがあります。使うためには、環境変数$IDL_STARTUP に$PCANS_DIR/idl/init.pro を設定します。

bash の場合、

```
export IDL_STARTUP = $PCANS_DIR/idl/init.pro
```

tcsh の場合、

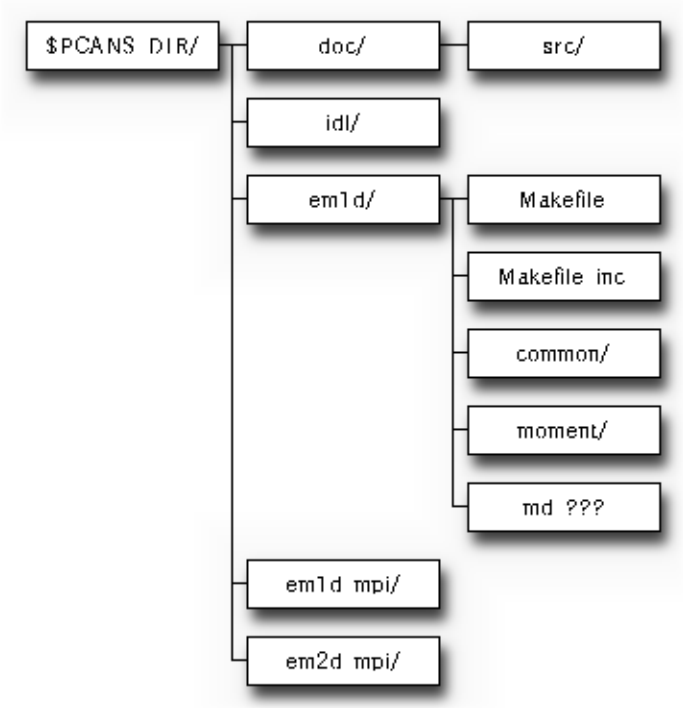
```
setenv IDL_STARTUP $PCANS_DIR/idl/init.pro
```

$PCANS_DIR/idl/init.pro 内には、path の設定、IDL 内の環境の設定等が含まれており、IDL 起動時に自動的に設定されます。各自の好みに合わせて修正してください。

IDL についてのさらなる詳細は こちら を参照ください。

2.5 全体の構成

pCANS では、1 次元及び 2 次元のコードが用意されています（2012 年 3 月現在）。1 次元コードは、シリアル版、MPI による並列版が用意され、2 次元コードは MPI 並列版のみとなっています。$PCANS_DIR 内は、以下のようにディレクトリとファイルで構成されています。



“doc/” 内には、本マニュアルのソースファイルが含まれています。本ディレクトリに含まれるファイルは開発者向けですので、一般ユーザーは編集する必要はありません。

“idl/” 内には、IDL による可視化ルーチンが含まれています。各課題で使用する可視化のための共通プロシージャが含まれています。

“em1d/”、”em1d_mpi/”、”em2d_mpi/” はそれぞれ、1 次元シリアル版、1 次元 MPI 並列版、2 次元 MPI 並列化版コードが含まれます。

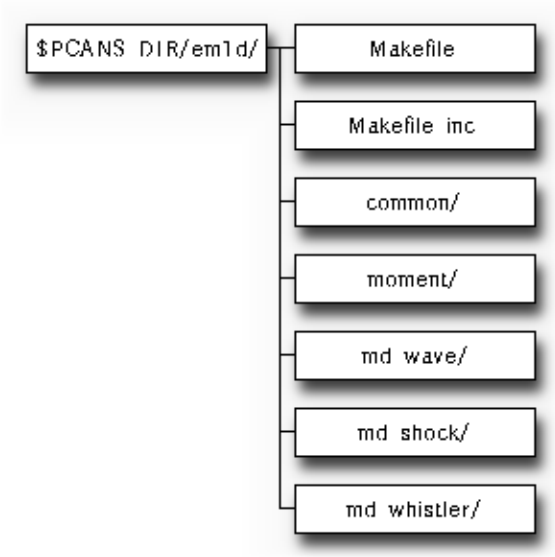
各コードのディレクトリ内には、コンパイル用の Makefile、コンパイル時の環境変数を設定した Makefile_inc が用意されています。各自の環境によってコンパイラ、コンパイラオプションを指定したい場合は、Makefile_inc 内に設定されている、”$FC” と ”$FFLAGS” を変更してください。”common/” には PIC コードの共通エンジンが収められています。”moment/” 内には、計算結果の粒子データからモーメント計算するためのコードが収められています。”md_???” は、各物理課題の初期設定等が含まれており、”???” に、物理現象の名前が付けられています。

CHAPTER 3

1次元PICコード

3.1 シリアル版

1次元シリアル版コードは以下のような構成になっています。



Makefileは本1次元コードのバイナリ生成、削除をコントロールしています。初期化するには$PCANS_DIR/em1d内で、

```
$ make clean
```

としてください。Makefile_incには、

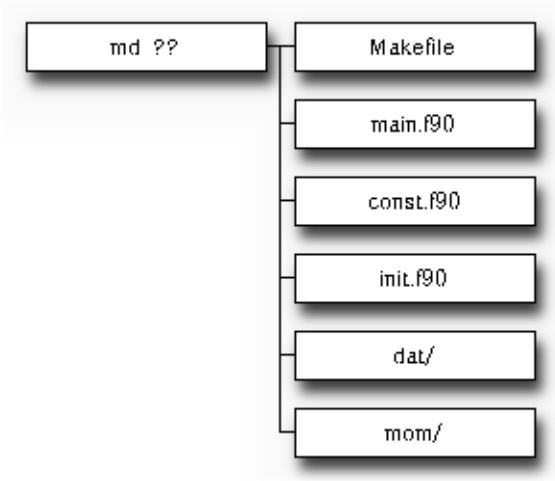
```
FC = gfortran
FFLAGS = -O2
```

が記述されており、makeする際の環境変数が設定されています。ここでは、"$FC"にはFortranコンパイラ、"$FFLAGS"にはコンパイラオプションが設定されています。コンパイラとコンパイラオプションを変更したい場合は、例えば、

```
FC = ifort
FFLAGS = -O3
```

のように修正してください。

シリアル版では、物理課題として「線形波動 (md_wave)」、「衝撃波 (md_shock)」、「電子温度異方性不安定 (md_whistler)」が用意されています（2012 年 3 月現在）。各課題には、



が含まれており、それぞれの課題に沿った初期設定のサンプルが置かれています。”dat/” は計算結果の出力先で、電磁場と粒子データが出力されます。”mom/” は”dat/” 内にある粒子データを元にモーメント計算をした時の出力先です。

例えば、衝撃波の計算を行うには、

```
$ cd $PCANS_DIR/em1d/md_shock
$ make
$ ./a.out
```

とします。結果は Fortran Unformatted 形式で”dat/” 内に出力されます。計算終了後にモーメントを計算するには、

```
$ make moment
```

とします。自動的に”em1d/moment/” 内のモーメント計算用のコードがコンパイルされ、計算が実行されます。結果は ASCII 形式で”mom/” 内に出力されます。

3.1.1 パラメタ設定

各課題に含まれる”const.f90” では、シミュレーション定数が設定されています。

const.f90:

```
module const

   implicit none
   integer, parameter :: nx  = 1200  ! number of grid points
   integer, parameter :: np  = 15000 ! number of particles in each cell
   integer, parameter :: nsp = 2     ! number of particle species
```

```
   integer, parameter :: bc  = -1     ! boundary condition (periodic:0, reflectiv

end module
```

nx はグリッド数、np は予想されるセル内の粒子数の最大値、nsp は粒子種数で通常はイオンー電子系なので、nsp=2 とします。bc は境界条件を指定するもので、**pCANS** では周期境界（bc=0）と反射端（bc=-1）が選べます。

“initt.f90” では、シミュレーション変数が設定されています。例えば、”md_wave” の場合、

init.f90:

```
!*****************************************************************
!   itmax   : number of iteration
!   it0     : base count
!   intvl1  : storage interval for particles & fields
!   intvl2  : printing interval for energy variation
!   intvl3  : printing interval for wave analysis
!   dir     : directory name for data output
!   file??  : output file name for unit number ??
!           :  9 - initial parameters
!           : 10 - for saving all data
!           : 11 - for starting from saved data
!           : 12 - for saving energy history
!           : 13~14 - for w-k diagram
!   gfac    : implicit factor
!             gfac < 0.5 : unstable
!             gfac = 0.5 : no implicit
!             gfac = 1.0 : full implicit
!*****************************************************************
itmax  = 10240
intvl1 = 512
intvl2 = 20
intvl3 = 20
dir    = './dat/'
file9  = 'init_param.dat'
file10 = 'file10.dat'
file12 = 'energy.dat'
file13 = 'wk_by.dat'
file14 = 'wk_bz.dat'
gfac   = 0.505
it0    = 0

if(it0 /= 0)then
   !start from the past calculation
   file11 = '002048_file10.dat'
   call fio__input(up,uf,np2,c,q,r,delt,delx,it0,np,nx,nsp,bc,dir,file11)
   return
endif
```

ここでは、ステップ数、出力先などが設定されています。各パラメタは上段のコメントに記述されています。”gfac” は、電磁場を陰解法で解くときの implicit factor で、通常 0.5 よ

り少し大きな値が設定されます。”it0” はステップ数のベースカウンタで、”it0=0” の時は最初から計算します。途中から計算を再開する時には、0 以外の値を設定します。その場合は、直後の if 文の中で”file11” で指定する中間ファイルを読み込み、計算を再開します。

その後、

init.f90:

```
!*************************************************************************
!   r(1)   : ion mass              r(2)   : electron mass
!   q(1)   : ion charge            q(2)   : electron charge
!   c      : speed of light        ldb    : debye length
!
!   rgi    : ion Larmor radius     rge    : electron Larmor radius
!   fgi    : ion gyro-frequency    fge    : electron gyro-frequency
!   vti    : ion thermal speed     vte    : electron thermal speed
!   b0     : magnetic field
!
!   alpha : wpe/wge
!   beta  : ion plasma beta
!   rtemp : Te/Ti
!*************************************************************************
pi    = 4.0*atan(1.0)
delx = 1.0
c     = 1.0
delt = 1.0
ldb   = delx

r(1) = 16.0
r(2) = 1.0
alpha = 2.0
beta  = 0.04
rtemp = 1.0
```

では、各物理変数（コメント内に詳述）が設定されています。これらを元にその他の物理変数を求めていきます。さらに、

init.f90:

```
np2(1:nx+bc,1) = 12500
np2(1:nx+bc,2) = np2(1:nx+bc,1)
```

では、初期のセル当たりの粒子数を設定しています。これは、”const.f90” 内で設定している”np” より小さくなるように注意してください。”np2” は計算途中で粒子が移動するに伴い変動し、位置によって値が異なるため、”np” をあらかじめ大きめにとり計算途中でも $\mathrm{np} > \max(\mathrm{np2})$ となるようにしてください。

3.1.2 モーメント計算

“moment/” 内には、粒子データからモーメントを計算するプログラムが格納されています。各課題で計算が終わったのち、

```
$ make moment
```

としてください。”dat/” 内にある粒子と電磁場データを読み込み、計算結果を”mom/” 内に出力します。Makefile 内の該当個所は、

```
############## moment calculation ###############
moment : $(LIB_DIR)/moment/mom.out
        $(LIB_DIR)/moment/mom.out ./dat/ `cd dat; \ls *_file10.dat`

$(LIB_DIR)/moment/mom.out :
        cd $(LIB_DIR)/moment ; make
#################################################
```

です。デフォルトでは”dat/” 内にある粒子データを全て計算することになりますが、”‘cd dat; \ls *_file10.dat‘” を修正することにより、任意の時間ステップにおける粒子・電磁場データに対して、モーメント計算を行います。

モーメント計算は、密度（0 次モーメント）、

$$n(x) = \int f(x, \mathbf{v}) d\mathbf{v}$$

速度（1 次モーメント）、

$$\mathbf{V}(x) = \frac{1}{n(x)} \int \mathbf{v} f(x, \mathbf{v}) d\mathbf{v}$$

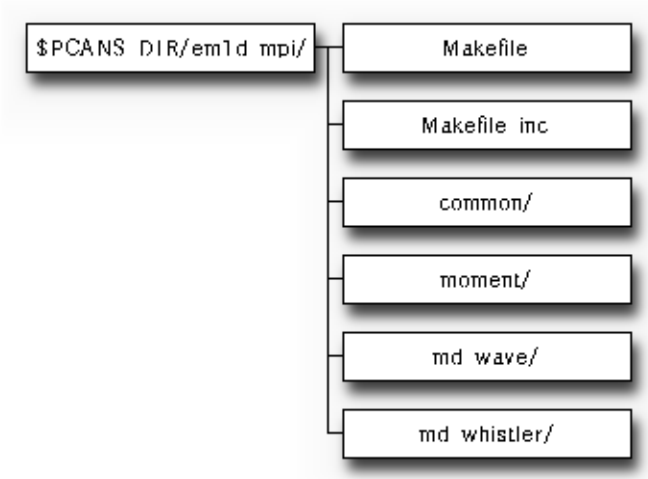
温度（2 次モーメント）、

$$T_{xx}(x) = \frac{1}{n(x)} \int v_x^2 f(x, \mathbf{v}) dv - V_x(x)^2$$
$$T_{yy}(x) = \frac{1}{n(x)} \int v_y^2 f(x, \mathbf{v}) dv - V_y(x)^2$$
$$T_{zz}(x) = \frac{1}{n(x)} \int v_z^2 f(x, \mathbf{v}) dv - V_z(x)^2$$

に従って計算します。

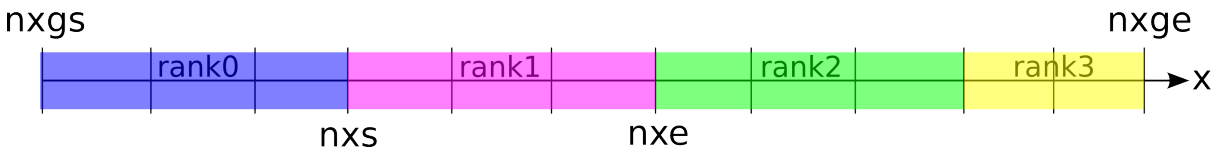
3.2 MPI 並列版

1 次元 MPI 並列版コードは以下のような構成になっています。



基本的な使い方は **シリアル版** と同じです。

3.2.1 領域分割法による並列化



MPI 並列版では、上図のように、プロセス数に従って 1 次元（x 方向）方向に領域を区分化し、各領域にある粒子の運動と場の発展を各プロセス（Rank）が分担することにより、並列化を行っています。MPI 並列化による違いを以下に示します。

ノート： 1 次元コードに対して領域分割をすると、系の発展に伴う粒子数の不均一化によって各プロセスの計算負荷が大きく異なるケースがあります（例：衝撃波）。その場合、一番大きな負荷のプロセスによって計算時間が決まるため、並列化効率が落ちることが知られています（ロードバランスの非均衡化）。 **pCANS** では、並列化プログラム例として 1 次元コードでも領域分割した並列化版コードを用意しています。

3.2.2 パラメタ設定

シミュレーション定数の設定（const.f90）は、以下のようになっています。

const.f90:

```
module const

  implicit none
  integer, parameter :: nx    = 2048       ! number of grid points
  integer, parameter :: nxgs  = 2          ! start point
  integer, parameter :: nxge  = nxgs+nx-1 ! end point
  integer, parameter :: np    = 25000      ! number of particles in each cell
  integer, parameter :: nsp   = 2          ! number of particle species
  integer, parameter :: nproc = 4          ! number of processors
  integer, parameter :: bc    = 0          ! boundary condition (periodic:0, refl
```

```
end module
```

MPI 並列版では、”nxgs” と”nxge” が設定されていて、グリッド番号の最初の値（nxgs）を任意に指定しています。”nproc” はプロセス数で、MPI で並列化するプロセス数を指定します（この例では 4 並列）。各担当領域を表す”nxs”、”nxe”（上図参照）は、init.f90 内で、”nxgs”、”nxge”、”nproc” から求めています。

init.f90:

```
!************** MPI settings  ******************!
    call mpi_set__init(nxgs,nxge,bc,nproc)
    if(nrank == nproc-1)then
       if(bc == -1) bcp = -1
       if(bc ==  0) bcp = 0
    else
       bcp = 0
    endif
    allocate(np2(nxs:nxe+bcp,nsp))
    allocate(uf(6,nxs1:nxe1))
    allocate(up(4,np,nxs:nxe+bcp,nsp))
    allocate(gp(4,np,nxs:nxe+bcp,nsp))
!************ End of MPI settings  ***************!
```

その他のユーザーが設定するパラメタは、 **シリアル版** と同じです。実行手続きは、

```
$ cd $PCANS_DIR/em1d_mpi/md_wave
$ make
$ mpiexec -n 4 ./a.out
```

と、最後の実行コマンドがシリアル版と違います。”-n” で並列数をしてしており、ここで与える数と上記の”nproc” の数が一致する必要があります（コード内で両数字が合っているかチェックしており、違う場合は終了します。）

3.2.3 モーメント計算

モーメント計算も、基本的には **シリアル版** と同じですが、

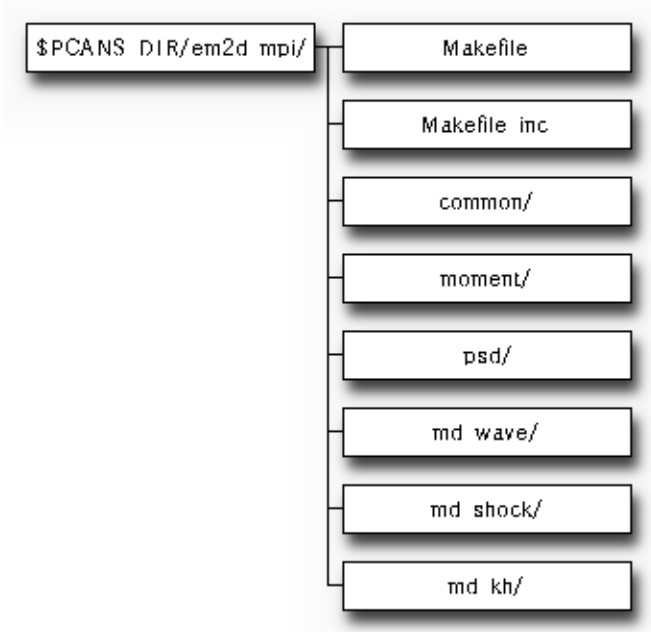
```
$ make moment
cd ../moment ; make
make[1]: ディレクトリ ~/pcans/em1d_mpi/moment に入ります
mpif90 -O2 -c boundary.f90
mpif90 -O2 -c particle.f90
mpif90 -O2 -c fio.f90
mpif90 -O2 -c mom_calc.f90
mpif90 -O2 -c main.f90
mpif90 -o mom.out -O2 boundary.o particle.o main.o mom_calc.o fio.o
make[1]: ディレクトリ ~/pcans/em1d_mpi/moment から出ます
../moment/mom.out ./dat/ `cd dat; \ls *_rank*.dat`
No. of processes?
```

のように、最後にプロセス数（並列数）が聞かれますので、ここで”mpiexec -n” で指定した並列数（上記例の場合は 4）を入力して下さい。結果は同様に、”mom/” 内に出力されます。

CHAPTER 4

2次元PICコード

2 次元コードは MPI 並列版のみで、以下のような構成になっています。

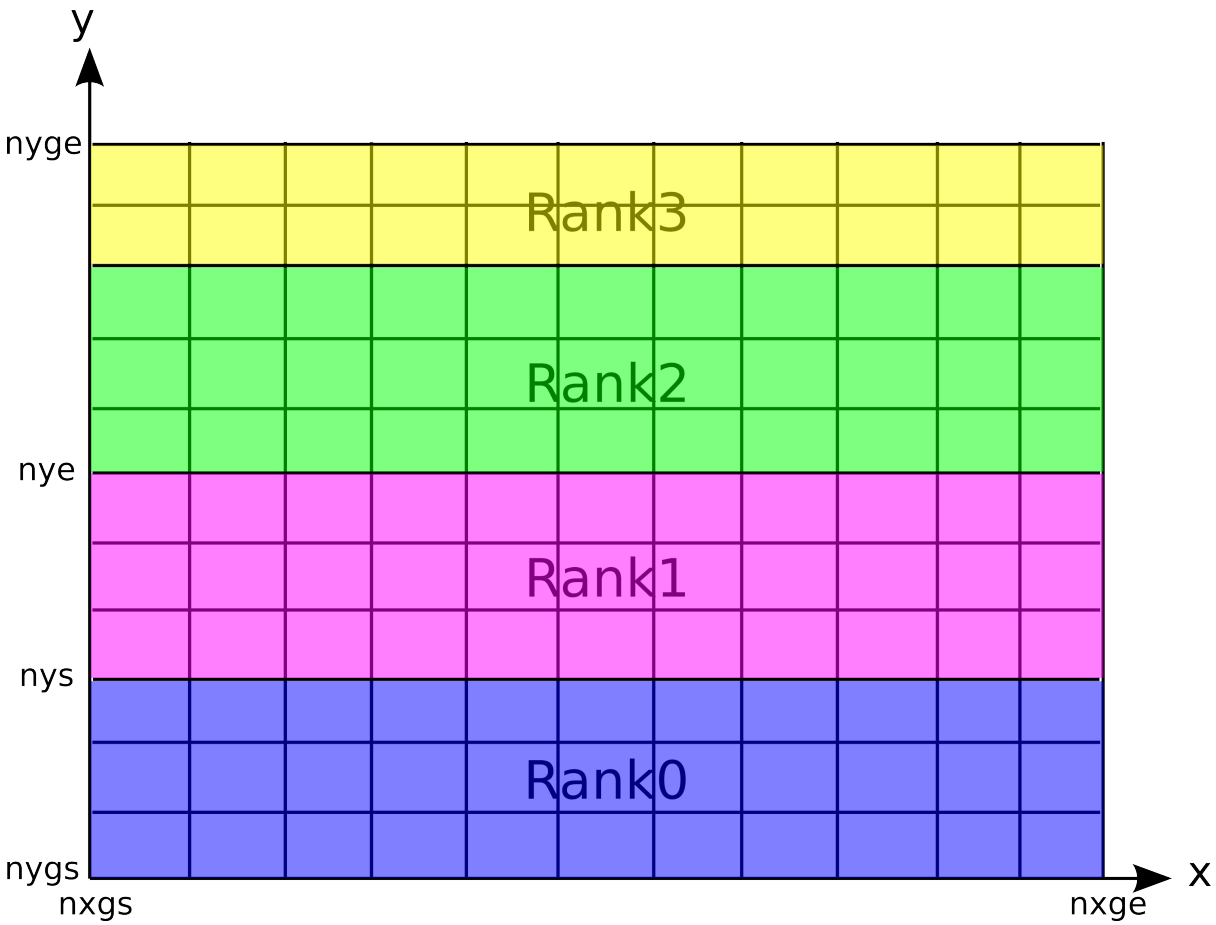


基本的な使い方は *1* **次元** *MPI* **並列版** と同じです。2 次元コードによる違いを以下に示します。

4.1 領域分割法による並列化

2 次元コードでは、 y 方向に 1 次元的に領域を区分化し、各領域にある粒子の運動と場の発展を各プロセス（Rank）が分担することにより、並列化を行っています（上図）。

ノート : 領域分割による粒子コードの並列化では、系の発展に伴う粒子数の不均一化によって各プロセスの計算負荷が大きく異なることが考えられます。その場合、一番大きな負荷のプロセスによって計算時間が決まるため、並列化効率が落ちることが知られています（ロードバランスの非均衡化）。各物理課題では、y 方向になるべく粒子数の偏りが少なくなるような初期設定をする必要があります（背景流れ場、磁場の向きなど）。



4.2 パラメタ設定

シミュレーション定数の設定（const.f90）は、以下のようになっています。

const.f90:

```
module const

  implicit none
  integer, parameter :: nx    = 128        ! number of grid points in x
  integer, parameter :: ny    = 128        ! number of grid points in y
  integer, parameter :: nxgs  = 2          ! start point in x
  integer, parameter :: nxge  = nxgs+nx-1 ! end point
  integer, parameter :: nygs  = 2          ! start point in y
  integer, parameter :: nyge  = nygs+ny-1 ! end point
  integer, parameter :: np    = 150*nx     ! number of particles in each cell
  integer, parameter :: nsp   = 2          ! number of particle species
  integer, parameter :: nproc = 4          ! number of processors
  integer, parameter :: bc    = -1         ! boundary condition in x (0:periodic,

end module
```

“nx”、”ny” はそれぞれ x、y 方向のグリッド数を設定します。”nxgs”、”nygs” はそれぞれ x、y 方向のグリッド開始番号を指定しています。本コードでは、2 次元領域分割は行わ

ず、y 方向に 1 次元領域分割をしています（上図）。また、y 方向は周期境界条件を仮定しています（x 方向は bc で指定可能）。このような制限があるものの、多くの 2 次元物理問題に対して適用可能です（衝撃波、磁気リコネクション、KH 不安定など）。

CHAPTER 5

物理課題

本章では、 **pCANS** で用意されている物理課題を紹介します。

5.1 1次元課題

5.1.1 線形波動

Author Yosuke MATSUMOTO (Chiba Univ.)

“md_wave” では、粒子の熱擾乱による線形波動の励起計算行っている。計算の実行手続きは、

```
$ cd $PCANS_DIR/em1d/md_wave
$ make
$ ./a.out
```

のようにして行う。計算結果を、空間－時間方向にフーリエ変換することにより、プラズマの固有波動の分散関係を数値的に得ることができる。下記では、その計算結果例を示す。（MPI 並列版でも同様であるので、以下ではシリアル版についてのみ説明する。）

パラメタ設定

本計算例でのパラメタ設定を以下に示す。

const.f90:

```
implicit none
integer, parameter :: nx  = 2048 ! number of grid points
integer, parameter :: np  = 1000 ! number of particles in each cell
integer, parameter :: nsp = 2    ! number of particle species
integer, parameter :: bc  = 0    ! boundary condition (periodic:0, reflective:
```

init.f90:

```
itmax  = 2048
intvl1 = 1024
intvl2 = 1024
intvl3 = 6
dir    = './dat/'
file9  = 'init_param.dat'
file10 = 'file10.dat'
file12 = 'energy.dat'
file13 = 'wk_by.dat'
file14 = 'wk_bz.dat'

pi   = 4.0*atan(1.0)
delx = 1.0
c    = 1.0
delt = 1.0
ldb  = delx

r(1) = 16.0
r(2) = 1.0

alpha = 2.0
beta  = 0.05
rtemp = 1.0

np2(1:nx+bc,1) = 500
np2(1:nx+bc,2) = np2(1:nx+bc,1)
```

本課題に固有なパラメタとして、"intvl3"、"file13"、"file14"が設定されている。intvl3は、サブルーチン wk_f（wk.f90）によって行われる電磁場データの出力頻度で、フーリエ解析で読み込むデータとして使われる。出力頻度は、求めたい波動の振動数によって調整する。シミュレーションパラメタは"dat/init_param.dat" に出力されているので、こちらを参照して決定する。"file13" 及び"file14" は、wk_f で出力するファイル名で、本例では横波成分（背景磁場は $\mathbf{B} = (B_0, 0, 0)$ ）を出力するので、上記のように指定している。解析するデータを修正するには（例えば、縦波成分、電場）、wk.f90 の出力部分

wk.f90:

```
!save data for w-k diagram
write(13,'(100000e13.4)')(uf(2,i),i=1,nx)
write(14,'(100000e13.4)')(uf(3,i),i=1,nx)
```

の箇所を変更する。

フーリエ解析

得られた結果を IDL によって読み込み、解析を行う。まずは、md_wave 内で

```
$ idl

IDL Version 6.4 (linux x86_64 m64). (c) 2007, ITT Visual Information Solutions
Installation number: 705264.
```

```
Licensed for use by: CHIBA-U

% Compiled module: LOADCT.
% Compiled module: FILEPATH.
% Compiled module: PATH_SEP.
% Compiled module: SET_X.

IDL>
```

のようにして idl を起動する。データの読み込みは、

```
IDL> input = file_read('dat/wk_by.dat')
column:         2048
line:          342
      0.  dat/wk_by.dat  Reading......
```

で行う。読み込んだデータ（input）を元に分散関係を描くプロシージャが同ディレクトリ内に用意されている（w_k_1d.pro）。ユーザーが指定するのは、同プロシージャの頭で設定されている物理パラメタで、

w_k_1d.pro:

```
;; DATA INFORMATION
dx = 1.0
dt = 6.0
c = 1.0D0
mr = 1.D0/16.D0
vai = 1.25D-1
vae = vai/sqrt(mr)
wgi = 2.47D-3
wge = wgi/mr
wpi = c/vai*wgi
wpe = c/vae*wge
beta = 0.05
```

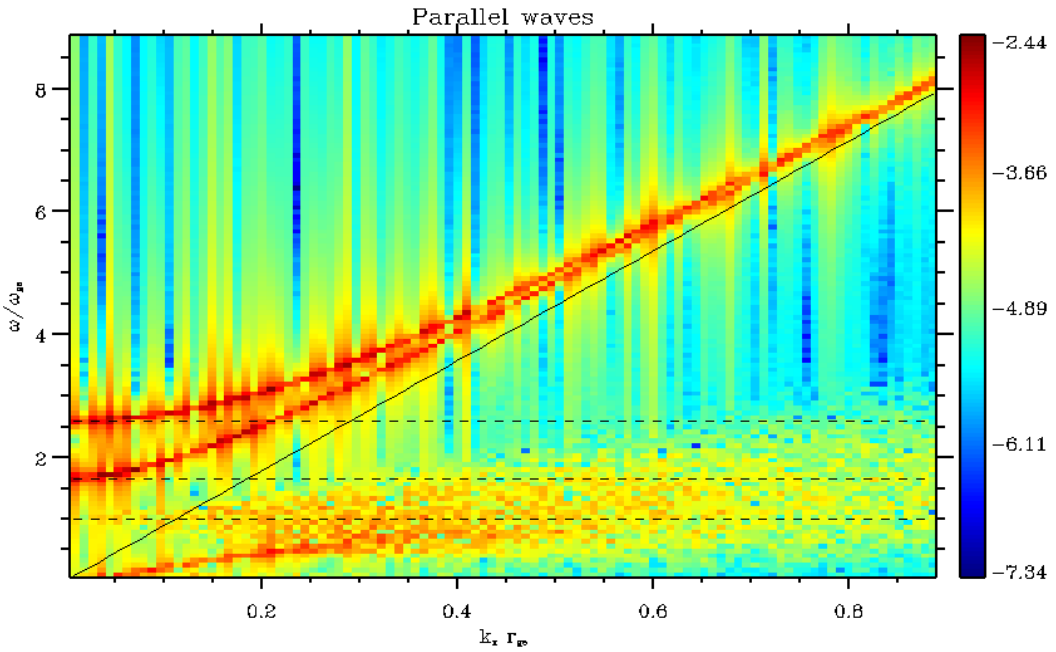
の部分を、”dat/init_param.dat” を参照して設定する。

```
IDL> .r w_k_1d
```

を行うと、

のように分散関係が得られる。破線は上から順に、R、L モードのカットオフ周波数、電子ジャイロ周波数を指す。実線は、$\omega = ck$ の真空中の電磁場モードを示している。本計算では、磁場に対して平行に伝搬する横波の分散関係を描いており、上から順に高周波 R モード, L モード、whistler 波といった、冷たいプラズマの分散関係が得られている。

初期設定の磁場配位（本計算例では、$\mathbf{B} = (B_0, 0, 0)$ ）を変えたり、解析する物理変数を変えることにより、その他の固有モードの分散関係を得ることができる。また、長時間計算することにより、上図ではつぶれてよく見えないイオン波動についても再現が可能である。プラズマ温度（β）を上げることにより、暖かいプラズマの分散関係を調べるのも、おもしろい課題である（例：Bernstein 波）。



5.1.2 電子温度異方性不安定

Author Yoshizumi MIYOSHI (STEL, Nagoya Univ.)

5.1.3 無衝突衝撃波

Author Yosuke MATSUMOTO (Chiba Univ.)

5.2 2 次元課題

5.2.1 無衝突衝撃波

Authors Takanobu AMANO (Univ. of Tokyo) and Yosuke MATSUMOTO (Chiba Univ.)

5.2.2 Weibel 不安定

Authors Tsunehiko KATO (Hiroshima Univ.)

5.2.3 磁気リコネクション

Authors Seiji ZENITANI (NAOJ) and Hiroyuki TAKAHASHI (CfCA,NAOJ)

5.2.4 ケルビン・ヘルムホルツ不安定

Author Yosuke MATSUMOTO (Chiba Univ.)

二つの運動している媒体が相対速度を持って接している場合、境界領域においては速度シア層が形成される。その速度シア境界は擾乱に対して不安定であることが 19 世紀後半より知られており、これを初期の研究者の名前から因んでケルビン－ヘルムホルツ（Kelvin-Helmholtz、以下 KH）不安定と言う。速度シア境界は、地球大気中から天体現象において普遍的に存在し、KH 不安定の発達は幅広い領域において応用されている。本巻で取り扱う実験室や宇宙空間でのプラズマ現象においても例外ではなく、速度シア層が存在するプラズマ物理領域に置いて KH 不安定が成長し、各領域において重要な役割を果たしている。

KH 不安定自体は流体的不安定であるため、プラズマ中での KH 不安定の非線形発展については主に電磁流体（Magnetohydrodynamic、以下 MHD）シミュレーションによる研究が行われている。例えば、活動銀河核などから放出される宇宙ジェットの MHD シミュレーションが多く報告されているが、KH 不安定がジェットの安定性に大きな影響を与えていることが明らかになっている（ *Ferrari et al. 1998*; *Baty et al. 2003* ）。このような MHD シミュレーションによるアプローチは、太陽風と接する地球磁気圏境界において成長する KH 不安定に対しても適用されてきた。しかしながら、本研究領域において重要となる、磁場を横切った無衝突プラズマの輸送過程は MHD シミュレーションでは記述できない現象である。そこで、個々のプラズマ粒子の運動を記述する粒子シミュレーションによる研究が近年始まってきており、無衝突プラズマの混合・輸送過程がどのような物理過程によって担われているかを明らかにすることを目指し、研究が行われている。本項では、KH 不安定の粒子シミュレーションを取り上げ、以下にその計算例を示す。

線形理論

数値計算による非線形発展を示す前に、線形理論から導かれる不安定条件と成長率を示す。これらは数値計算の初期設定、計算結果の妥当性を確認するのに有益である。

いま、プラズマが x 方向に流れ、速度が y 方向に差し渡し V_0 変化している速度シア層を考える。また、背景磁場は $\mathbf{B_0} = (B_{0x}, 0, B_{0z})$ の成分を持つ。このようなプラズマ中での KH 不安定の成長は、磁場の存在とプラズマの圧縮性に大きく影響を受ける（詳しくは文献 *Chandrasekhar 1961* ; *Miura & Pritchett 1982* を参照のこと）。

ここで、2 つの特別なケースを考えよう。まずは、背景磁場が背景速度と同じ方向に向いている場合、すなわち $\mathbf{B_0} = (B_{0x}, 0, 0)$ の場合、速度シアの大きさについて

$$V_A < \frac{V_0}{2} < C_S \tag{5.1}$$

の条件を満たす場合に不安定となる。ここで $V_A = |\mathbf{B_0}|/\sqrt{\mu_0 \rho_i}$ は Alfven 速度、 $C_S = \sqrt{\gamma P_i/\rho_i}$ は音速である（ ρ_i 、 P_i 、 γ はそれぞれイオン密度、圧力、比熱比である）。速度シアの大きさ $(V_0/2)$ が V_A より大きい必要があるのは、速度場の面内に磁場が存在すると磁気張力が安定化の働きをもたらすためであると直感的にも理解できる。また音速で制限されているのは、プラズマの圧縮性による安定化の効果である。このことから、平行磁場（ $\mathbf{V_0}//\mathbf{B_0}$ ）の場合は高 β プラズマ、例えば、宇宙ジェットのような弱い磁場領域に適用される。ここで、 $\beta = \frac{P_i+P_e}{P_{mag}}$ はイオンの圧力 $P_i \equiv \frac{1}{2} m_i V_{ti}^2$ と電子の圧力 $P_e \equiv \frac{1}{2} m_e V_{te}^2$ の和であるプラズマ圧力と磁気圧力 $P_{mag} \equiv |\mathbf{B}|^2/(2\mu_0)$ との比である。

一方、垂直磁場（ $\mathbf{V_0} \perp \mathbf{B_0}$ ）の場合は

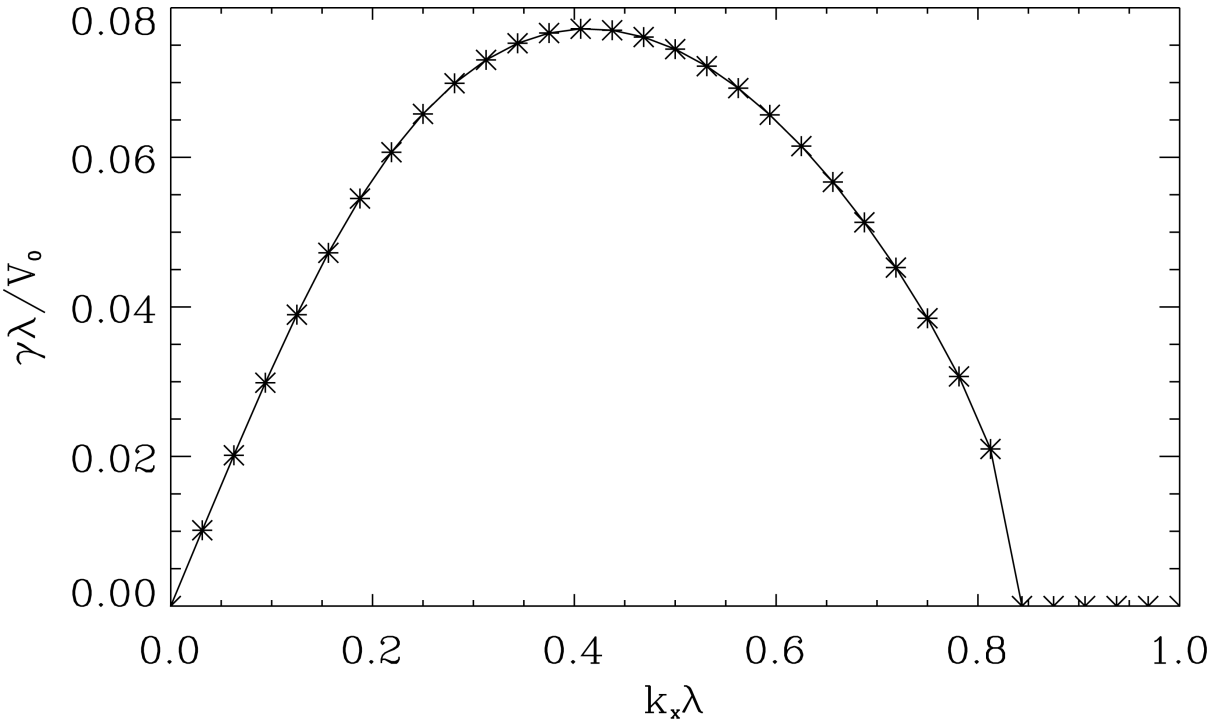
$$\frac{V_0}{2} < V_f = \sqrt{V_A^2 + C_S^2} \tag{5.2}$$

が不安定条件であり、亜磁気音速の流れであれば常に不安定である。このような条件は様々なプラズマ β 領域に適用可能であり、特に低緯度磁気圏境界においてはこの条件が満たされていると考えられている。現実には波数ベクトル、磁場、流れの向きが様々であり、それぞれの状況において不安定条件は違うが、一般に垂直磁場条件が最も不安定であり、平行磁場条件が最も安定化されやすい。

KH 不安定の成長率は上記条件に加え、速度シアの厚みが時間スケールを決める。以下では、数値シミュレーションでも用いられる一般的な速度シアの関数に対する成長率を求める。シミュレーションでは初期条件として

$$V_x = -\frac{V_0}{2} \tanh\left(\frac{y}{\lambda}\right) \tag{5.3}$$

の形で速度プロファイルがしばしば与えられる。また、ここでは垂直磁場配位 $\mathbf{B} = (0, 0, B_0)$ を考えよう。プラズマの圧力はプラズマ β より決まり、密度と共に一様に与える。このような条件の下で MHD 方程式を線形化し、得られた線形 MHD 方程式を固有値問題として数値的に解くことにより固有値（成長率）と固有関数が得られる（詳細は省く）。



上図は、 $\beta = 2.0$ 、 $V_0/V_f = 1.0$ の場合の成長率とそれに対応する不安定モードを表す。これより、最大成長率は $\gamma\lambda/V_0 = 0.077$ であり、対応する不安定モードは $k_x\lambda = 0.4$ であることがわかる。これらを元に計算領域のサイズを決める。また、計算結果から得られた成長率を求めることにより、計算の妥当性のチェックを行う。

以上は理想 MHD 近似の下での結果である。これら MHD 的特性に加え、速度シアの厚みがイオンスケール程度まで薄い場合には、イオンの慣性効果（ *Fujimoto & Terasawa 1991*

)、イオンの有限ラーマー半径効果（ *Huba 1996* ）などが、不安定モードや成長率に影響を与える。特に、二つの媒質中の磁場強度が違う場合は、磁場と渦度の内積 $\mathbf{B}\cdot\nabla\times\mathbf{V_0}$ の符号により、成長率が増減することが報告されている。また薄い速度シア層の場合は粒子シミュレーションにおいて初期平衡状態を設定する際に工夫が必要であることも指摘されている（ *Pritchett & Coroniti 1984* ； *Cai et al. 1993* ）。本小節ではこれらの影響が少ない、比較的厚いシア層について取り扱う。

以下では粒子シミュレーションによる計算例を示す。まず最初に、上記線形理論で示した条件の下での計算結果を示し、線形理論との一致を確認する。この計算では古典的な渦形成が非線形発展として特徴づけられ、得られる描像は MHD 的である。次に、粒子シミュレーションによるマルチスケール現象の例として、KH 不安定の乱流発展の計算例を示す。

ベンチマークテスト

まず最初の例では線形理論で用いた条件を初期設定とする。すなわち、流体的物理量として

$$
\begin{aligned}
\mathbf{V_0} &= (V_{0x}, 0, 0) \\
\mathbf{B_0} &= (0, 0, B_{0z}) \\
V_{0x}(y) &= -\frac{V_0}{2}\tanh\left(\frac{y}{\lambda}\right) \\
V_0 &= V_f \\
\beta &= 2.0 \\
\frac{P_i}{P_e} &= 1.0
\end{aligned}
\tag{5.4}
$$

となるように粒子を配置する。また、プラズマ固有のパラメタとして

$$
\begin{aligned}
m_i/m_e &= 16 \\
\omega_{pe}/\omega_{ce} &= c/V_{Ae} = 1.4 \\
\lambda/r_{ci} &= 4.0 \\
N_i &= N_e = 120/(\Delta x \Delta y)
\end{aligned}
\tag{5.5}
$$

を以下に続く計算では採用する。ここで上から順に、イオン・電子質量比、電子プラズマ－ジャイロ振動数比、イオンジャイロ半径に対するシアの厚み、およびセルあたりの各種の粒子数を表す。特に式 (5.5) では、速度シアの厚み λ をイオンジャイロ半径の 4 倍とることにより、初期平衡状態、線形成長が MHD 的描像で与えられる（以上の設定では質量比 m_i/m_e や振動数比 ω_{pe}/ω_{ce} の無次元量は系の発展に影響は及ぼさない）。

粒子は一様に配置し、速度はシフトしたマクスウェル分布

$$
f = \left(\frac{1}{2\pi v_t^2}\right)^{3/2}\exp\left(-\frac{(v_x - V_{0x}(y))^2 + v_y^2 + v_z^2}{2v_t^2}\right) \tag{5.6}
$$

となるように与える。速度シアがあり、速度場の面に対して垂直な磁場がある場合、対

流電場 $\mathbf{E_0} = (0, E_{0y}, 0)$ は空間的に非一様になるため、ガウスの法則

$$\begin{aligned} \nabla \cdot \mathbf{E_0} &= \frac{\partial E_{0y}}{\partial y} \\ &= \frac{\partial}{\partial y}\left(V_{0x} B_{0z}\right) \\ &= \frac{q(n_i - n_e)}{\epsilon_0} \end{aligned} \tag{5.7}$$

を満たす必要がある。したがって、システム全体としては電荷中性にはならない。ここでは電子個数密度を調整する。具体的には式 (5.3) より、

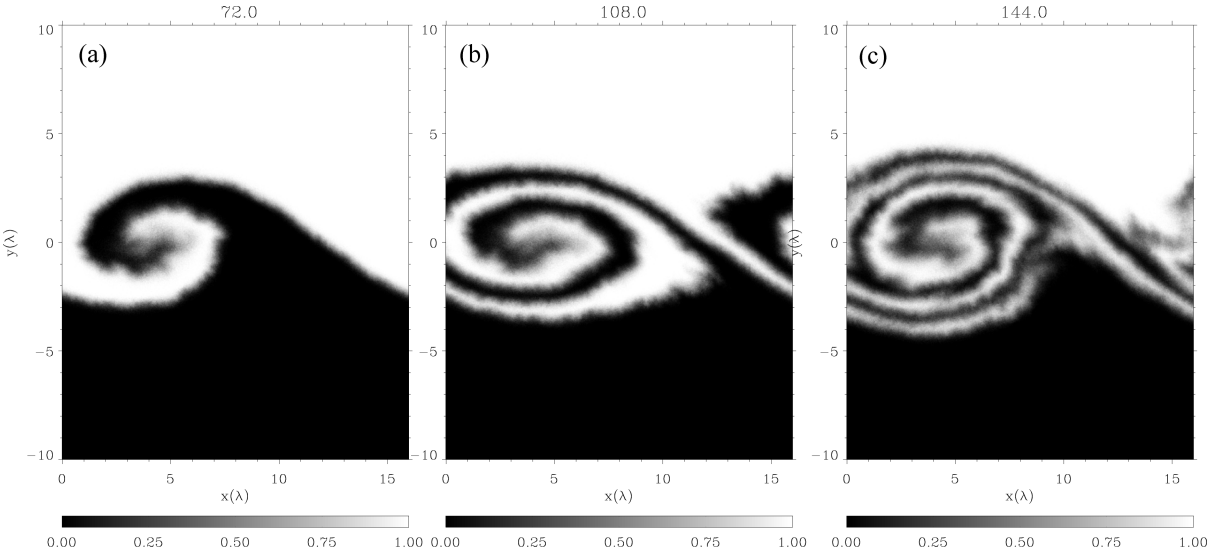
$$n_i(y) - n_e(y) = -\frac{\epsilon_0 B_0 V_0}{2qc\lambda} \cdot \frac{1}{\cosh^2(y/\lambda)} \tag{5.8}$$

となるように電子の数を加える。任意の関数の頻度分布（速度、空間分布）になるように粒子を配置するには累積分布関数（cumulative distribution function）を用いる（ *Birdsall & Langdon 1991* ）。粒子を配置する場合、結果として任意の分布 $f(y)$ となるようにするには、

$$R = F(y) = \frac{\int_{-y_b}^{y} f(y')dy'}{\int_{-y_b}^{+y_b} f(y')dy'} \tag{5.9}$$

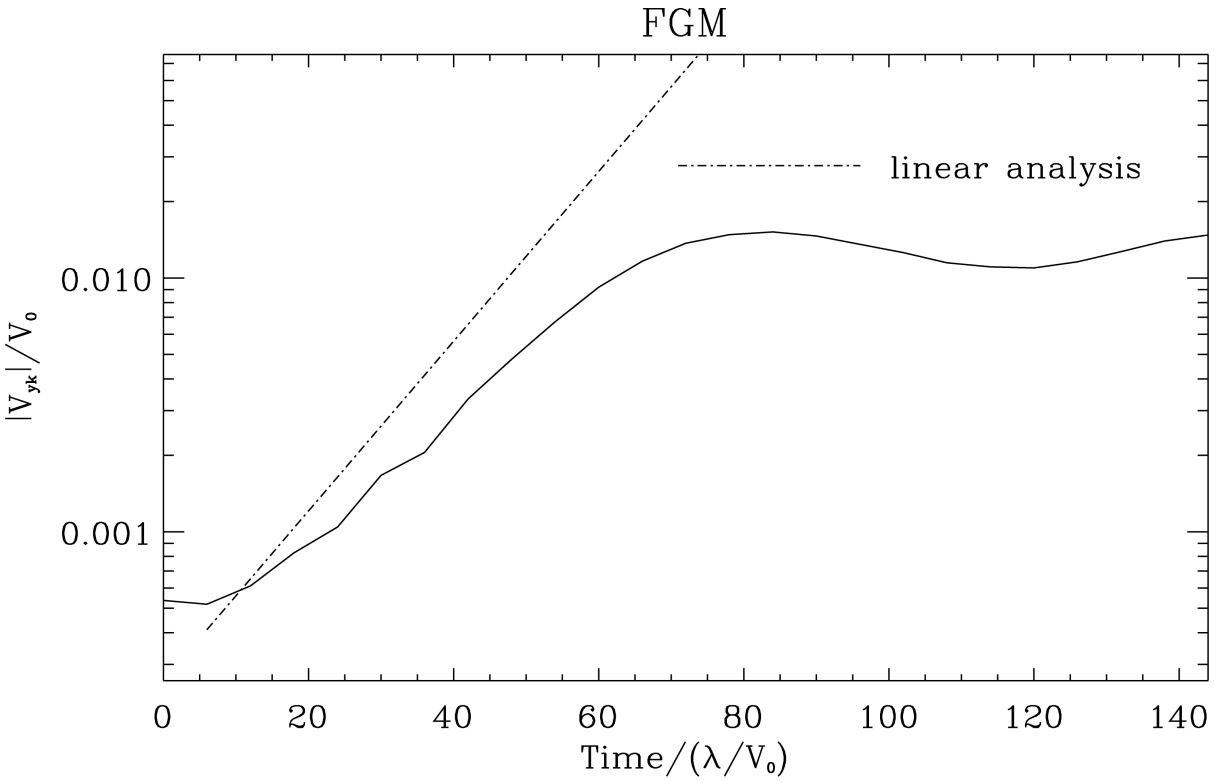
より、 y を求める。ここで $0 \le R \le 1$ の一様分布を持つ数（例えば一様乱数）、 $\pm y_b$ は y 方向の境界を表し、 $F(-y_b) = 0$ 、 $F(+y_b) = 1$ である。右辺の分子を解析的または数値的に求めることにより、頻度分布が $f(y)$ となるような R に対する位置 y を求めることができる。

線形解析で得られた最大成長率モード $k_x\lambda = 0.4$ より、シミュレーション領域の x 方向のサイズは $L_x = 15.7\lambda \sim 512\Delta x$ とし、 y 方向のサイズは $L_y = \pm 10\lambda = 641\Delta y$ とする。境界条件は、 x 方向は周期境界、 y 方向は反射端境界とした。グリッドサイズは通常どおりデバイ長程度（ $\Delta x = \Delta y = \Delta = \lambda_{De}$ ）に設定する。時間刻みは陰的 FDTD 法を用いると比較的大きくとることができ、ここではクーラン数として $c\Delta t = \Delta x, y$ とした。



図は KH 不安定の非線形発展を表す。初期の粒子（イオン）の位置により色分けしており、 $y(t=0) > 0$ の粒子は白色、 $y(t=0) \le 0$ の粒子の色は黒色を示している。境界付

近は両粒子が混合しているため混合率に比例した中間色を示している。MHD シミュレーションと同様、粒子シミュレーションにおいても速度シア層から渦が成長し、二つの媒質を巻き込むようにして混合していく様子が見られる。



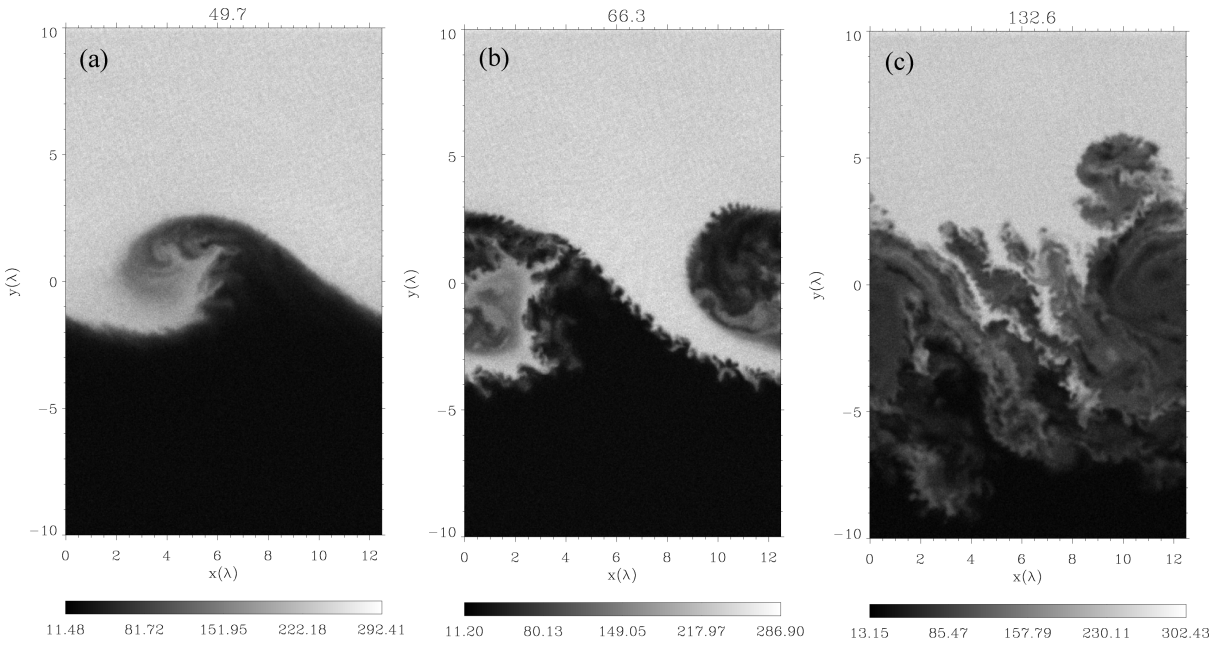
得られた計算結果の妥当性を調べるため、速度擾乱成分である V_y を x 方向にフーリエ変換し、y 方向に平均化した最大成長率モード (モード数１）のフーリエ振幅の時間発展を調べたのが上図である。また比較のため、線形解析で得られた成長率（ $\gamma\lambda/V_0 = 0.077$ ）の傾きを同時にプロットした。結果、粒子シミュレーションで得られた最大成長率モードの線形成長率は理想 MHD 近似での線形解析結果に近いことがわかる。また非線形段階での振幅の振動は MHD シミュレーションで見られる特徴である（ *Miura & Sato 1978* ）。これらの結果は、初期条件が MHD 近似に近い条件であるため、得られる結果は MHD 的描像であることがわかる。

マルチスケール現象としての乱流発展

以上の計算は非常に単純な設定であり、得られる描像は古典的な KH 不安定の成長であった。しかし、速度場の非一様性に加え、他の物理量の非一様性をも同時に扱う場合は様々な非線形現象が現れる。ここでは、地球磁気圏境界で特徴的な、密度の非一様性について着目し、密度成層構造がもたらす非線形現象を粒子シミュレーションで明らかにする。以下に示す例は上記設定とは一部異なり、

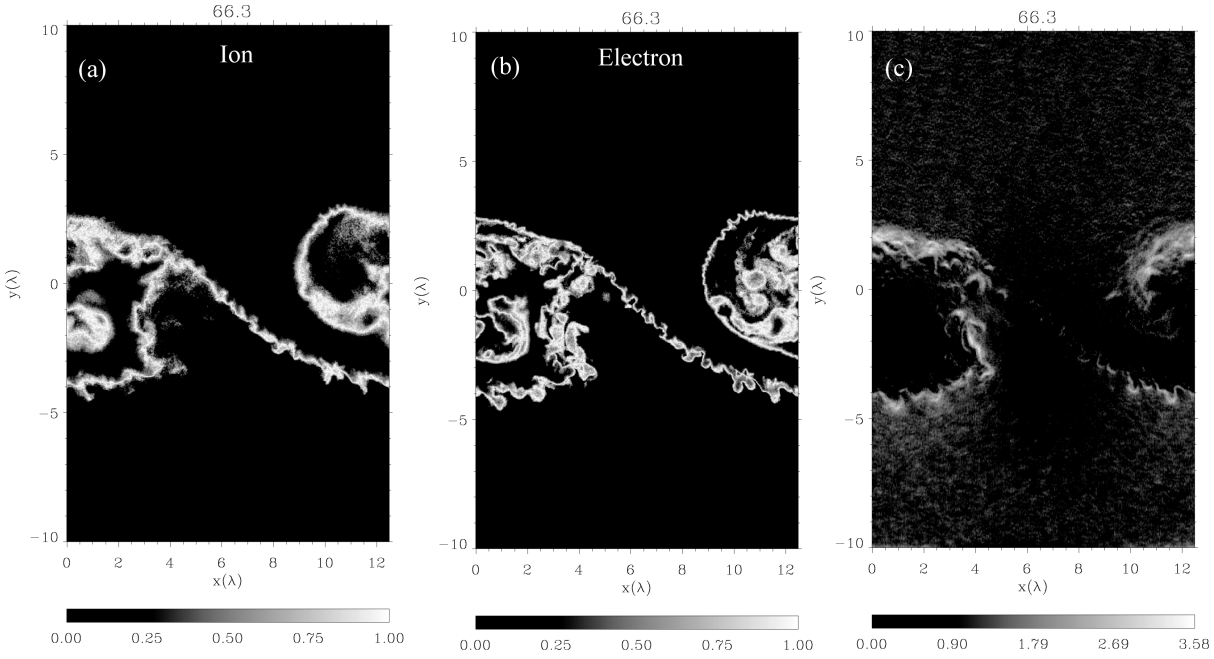
$$\begin{aligned} \beta &= \frac{P_i + P_e}{P_{mag}} = 0.3 \\ \omega_{pe}/\omega_{ce} &= c/V_{Ae} = 2.8 \end{aligned} \tag{5.10}$$

の値を採用している。



図は密度比が 1:0.1 の場合の計算結果である。図 (a) では、KH 不安定の特徴である渦発展が見られる。しかし非線形段階では一様な計算と違い、渦の巻き始めに境界付近で新たに小さなスケールの構造が生まれていることがわかる（b）。この小さな構造が 2 次的に成長することがきっかけとなり渦構造は砕波し、乱流へと移行する（c）。

このような、KH 不安定の非線形発展中に新たに励起する構造を 2 次的不安定性と呼ぶ。特に密度成層構造が励起する不安定性は、回転による遠心力が擬似的に重力の役割を果たすレイリーテイラー（Rayleigh-Taylor, 以下 R-T）不安定として知られ、2 次元 MHD シミュレーションによって提唱されている（ *Matsumoto & Hoshino 2004* ）。この 2 次的 R-T 不安定により乱流へと移行した結果、無衝突プラズマの混合過程が促進されている。



大局的な渦構造の中に 2 次的 R-T 不安定による小スケールの構造が生まれることにより、まずイオンが磁場の束縛から逃れ、混合が促進される。上図 (a)、(b) は各セルでの粒子の

混合比を表す。大局的には渦形成に伴って境界線が巻き込まれ、イオンは境界線を跨いだジャイロ半径程度の領域で混合している。しかし、2 次的不安定の励起に伴いイオンジャイロ半径を越えた領域までイオンが拡散していることが見て取れる。一方電子も同様に電子ジャイロスケール程度の狭い範囲で混合しているが、イオンの拡散に追随するように、細かく複雑な境界線を作りながら混合面積を増やしている。

2 次的 R-T 不安定に伴う粒子の拡散は強い静電場が担っている。これは、イオン一電子の 2 流体 R-T 不安定では重力ドリフトの違いから電荷が生じるためである。これは磁場勾配ドリフトによって励起される低周波混成ドリフト不安定（lower hybrid drift instability）と類似している（ *Winske 1988* ）。上図 (c) は初期の対流電場 $V_{0x}B_{0z}/c$ で規格化した電場強度を示したものである。2 次的 R-T 不安定が励起されている境界領域では初期の対流電場の約 4 倍もの強い電場が局所的に励起されており、それにともない粒子が拡散していることがわかる。このような 2 次的 R-T 不安定による静電場乱流の励起は MHD シミュレーションでは再現できない現象であり、粒子シミュレーションによって初めてプラズマ粒子の磁場垂直拡散のメカニズムが明らかになったものである。

まとめ

粒子シミュレーションによる KH 不安定の研究は世界的に見ても例が少なく（たとえば、*Wilber & Winglee 1995* ; *Matsumoto & Hoshino 2006* ）、上記例はその一つである。しかし計算機の能力の発達に伴い様々な計算が可能になってくると思われ、今後、無衝突プラズマの乱流発達とカスケード過程における粒子加速、散逸メカニズムの解明が重要なテーマとして挙げられよう。また宇宙ジェットなどの水平磁場での KH 不安定の発達は今後重要となる研究領域でもあり、MHD シミュレーション（ *Nykyri & Otto 2001* ）や 2 流体シミュレーション（ *Nykyri & Otto 2004* ; *Nakamura et al. 2008* ）で示されている磁気リコネクションとのカップリングが注目される。さらに近年の 3 次元 MHD シミュレーションでは 3 次元性による新たな KH 不安定の乱流移行が見出されてきており（ *Matsumoto & Seki 2007* ）、3 次元 KH 不安定の粒子シミュレーションについても今後重要なテーマとなっていく。

CHAPTER 6

References

CHAPTER 7

Appendix

7.1 reST の書き方について

このドキュメントは sphinx (日本語版ドキュメント) で生成されています。sphinx は reStructuredText (reST) という形式 (+いくつかの拡張) で書かれたテキストファイルを html や pdf へ変換するツールです。数式の表示 や ソースコードのハイライト表示 も可能です。

7.1.1 数式の表示

以下のように簡単に LaTeX 形式で数式を書くことができます。まずはディスプレイスタイルで Maxwell 方程式を表示します。

$$\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$
$$\frac{1}{c}\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B} - \frac{4\pi}{c}\mathbf{J}$$

文中に $\lim_{x \to 0} \frac{\sin x}{x}$ のようにインラインで埋め込むことも可能です。LaTeX でいうところの $数式$ です。

7.1.2 ソースコードのハイライト表示

以下のように自動的にソースコードのハイライト表示ができます:

```
program test
implicit none
integer*4 i

write(*,*) 'Hello World !'

do i = 1, 10
   write(*,*) i
end do
```

```
end program test
```

他の言語でも可能です。以下は C 言語の例です。

```
#include <stdio.h>

int main()
{
 printf("Hello World !\n");
 return 0;
}
```

また別のファイルを include して表示することもできます。

Appendix

A.1 reSTの書き方について

このドキュメントは sphinx (日本語版ドキュメント) で生成されています。sphinx は reStructuredText (reST) という形式 (+いくつかの拡張) で書かれたテキストファイルを html や pdf へ変換するツールです。数式の表示 や ソースコードのハイライト表示 も可能です。

A.1.1 数式の表示

以下のように簡単に LaTeX 形式で数式を書くことができます。まずはディスプレイスタイルで Maxwell 方程式を表示します。

$$\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$
$$\frac{1}{c}\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B} - \frac{4\pi}{c}\mathbf{J}$$

文中に $\lim_{x \to 0} \frac{\sin x}{x}$ のようにインラインで埋め込むことも可能です。LaTeX でいうところの $数式$ です。

A.1.2 ソースコードのハイライト表示

以下のように自動的にソースコードのハイライト表示ができます:

```
program test
implicit none
integer*4 i

write(*,*) 'Hello World !'

do i = 1, 10
   write(*,*) i
end do
```

```
end program test
```

他の言語でも可能です。以下は C 言語の例です。

```
#include <stdio.h>

int main()
{
 printf("Hello World !\n");
 return 0;
}
```

また別のファイルを include して表示することもできます。

Bibliography

[Baty03] Baty et al., ApJ, 2003.

[Birdsall-Langdon91] Birdsall and Langdon, “PLASMA PHYSICS VIA COMPUTER SIMULATION”, Adam Hilger, 1991.

[Cai93] Cai et al., J. Comput. Phys., 1993

[Chandrasekhar61] Chandrasekhar, “Hydrodynamic and Hydromagnetic Stability”, Dover, 1961.

[Ferrari98] Ferrari et al., Annu. Rev. Astron. Astrophys., 1998.

[Fujimoto-Terasawa91] Fujimoto and Terasawa, J. Geophys. Res., 1991.

[Huba96] Huba, Geophys. Res. Lett., 1996.

[Matsumoto-Hoshino04] Matsumoto and Hoshino, Geophys. Res. Lett., 2004.

[Matsumoto-Hoshino06] Matsumoto and Hoshino, J. Geophys. Res., 2006.

[Matsumoto-Seki07] Matsumoto and Seki, J. Geophys. Res., 2007.

[Miura-Sato78] Miura and Sato, J. Fluid Mech., 1978.

[Miura-Pritchett82] Miura and Pritchett, J. Geophys. Res., 1982.

[Nakamura08] Nakamura et al., J. Geophys. Res., 2008.

[Nykyri-Otto01] Nykyri and Otto, Geophys. Res. Lett., 2001.

[Nykyri-Otto04] Nykyri and Otto, Ann. Geophys, 2004.

[Pritchett-Coroniti84] Pritchett and Coroniti, J. Geophys. Res., 1984

[Winske88] Winske, J. Geophys. Res., 1988.

[Wilber-Winglee95] Wilber and Winglee, J. Geophys. Res., 1995