

数値天文学テクニカルマニュアル

まえがき

数値シミュレーションが天体やスペースのプラズマを理解する有力な手段であることは言を待たない。計算機のハードウェアおよびソフトウェアの発達により、大型で高精度の計算が極めて容易になった。修士論文でも2次元や3次元の高分解能な数値シミュレーションは決して珍しくない。またそのシミュレーション結果は、きれいな静止画や動画として発表されることが多い。

このように数値シミュレーションが普遍的な研究手段となったことは喜ばしい限りであるが、教育面から見ると喜んでばかりもいられない。数値計算の技法が洗練されるにつれ、その習得には時間が掛かるようになった。四半世紀前であれば、利用できる手法も限られており、独学でも比較的容易に習得することが可能であった。しかし現在では多くの技法が開発されていて、それらを一通り学ぶだけでも相当な努力が必要である。また個々の技法も複雑で、初学者にとって取っつきやすいものではない。

私たちはこのような現状認識から、これから数値シミュレーションを志す大学院生を対象として、数値シミュレーションサマースクールを開講してきた。運営の形態は途中で変化した。通算すると今年で第5回になる。サマースクールでは、数値シミュレーションを実行する具体的な方法や原理について基本的な事柄を講義するとともに、受講者に実際に数値シミュレーションを体験してもらっている。本書はサマースクールの流体・磁気流体コース卒業生を主たる対象としたアドバンスコースの集録である。

2004年のアドバンスコースでは各自が習ったあるいは開発した数値シミュレーションに関する技法を受講生にマニュアルの形でまとめていただいた。本書第1部では磁気流体力学方程式を数値的に解く方法、あるいはその準備のための解析計算がまとめられている。第2部では数値計算の結果を可視化するための技法、第3部では数値シミュレーション結果を解析するための技法が紹介されている。各章の著者には、後輩である修士課程1年の学生に親切に指導するという立場で書いていただいた。普通の教科書や論文では省略される、途中の式変形についてもできるだけ詳しく記述していただいた。プログラムの使い方も、1ステップずつ詳しく書いていただいた。本書が参加者だけでなく、これから数値天文学をはじめようと考えている方々の学習の役に立てばこれほど喜ばしいことはない。

本書の作成にあたっては著者の方々に多大な貢献をしていただいた。多くの著者の方には2004年9月6日より10日まで合宿形式で参加していただいた。改めてその努力に謝意を表すため、ここに氏名と所属を掲載させていただく。

分担	著者名	所属
1 章	花山 秀和	東京大学大学院理学系研究科
2 章	佐藤 裕司	千葉大学大学院自然科学研究科
3 章	上原 一浩	京都大学大学院理学研究科
4 章	木暮 宏光	京都大学大学院理学研究科
5 章	水野 陽介	京都大学大学院理学研究科
6 章	富高 真	千葉大学大学院自然科学研究科
7 章	天野 孝伸	東京大学大学院理学系研究科
8 章	越智 康浩	名古屋大学大学院理学研究科
9 章	錦織 弘充	千葉大学大学院自然科学研究科
10 章	大須賀 健	立教大学理学部
11, 12 章	杉本 香菜子	名古屋大学大学院理学研究科

また本書や付属のCD-ROMの作成にあたっては多くの方々の協力をいただいた。付属CD-ROMに採録されたCANSのソフトウェアやidlのプログラムの多くは東京大学の横山央明助教授や法政大学の松本倫明助教授が作成されたものである。さらに本書への採録も快諾していただいた。国立天文台天文学データ解析計算センターには付属CD-ROMの印刷を引き受けていただいた。千葉大学宇宙物理学研究室の川嶋恵理さんにはCD-ROMの表紙デザインと印刷を担当していただいた。これらの方々に改めて謝意を表したい。

本書の出版にあたっては費用の一部を科学研究費補助金基盤研究(B)(1)(15340062)より支出した。

2004年9月

編者

花輪 知幸

町田 正博

千葉大学 先進科学教育センター

目次

I	磁気流体力学方程式の数値解法	11
1	CANS を使った磁気流体力学シミュレーション	12
1.1	CANS とは	12
1.1.1	宇宙磁気流体シミュレーションソフトウェア	12
1.1.2	CANS で何ができるか？	12
1.1.3	CANS の特徴	13
1.1.4	CANS の構成	13
1.2	CANS を使ってみよう！	13
1.2.1	動作環境	13
1.2.2	インストール	14
1.2.3	内容確認	14
1.2.4	コンパイル	16
1.2.5	プログラムの実行	17
1.2.6	IDL の準備	18
1.2.7	IDL によるデータの読み込みと表示	18
1.3	計算の流れ	19
1.3.1	初期設定 -main.f.	19
1.3.2	計算スキームの選択 -main.f.	22
1.3.3	終了処理 -main.f.	24
1.3.4	モデルの設定 -model.f.	24
1.3.5	グリッド間隔の設定 -grdrdy.f.	27
1.3.6	ベクトルポテンシャルの設定 -bbtoaa.c.f.	27
1.3.7	境界条件の設定 -bnd.f.	27
1.3.8	下限値のチェック -chkdav.f.	28
1.3.9	CFL 条件の設定 -cfl_m.f.	28
1.4	修正 Lax-Wendroff スキーム	28
1.4.1	基礎方程式	28
1.4.2	磁気流体力学	29
1.4.3	基礎方程式の保存形式での表現	30
1.4.4	差分方程式	30
1.4.5	人工粘性の導入	36
1.4.6	計算エンジン -mlw_m.c.f.	37

1.5	謝辞	45
2	磁気流体力学方程式の近似 Riemann 解法	46
2.1	問題設定	46
2.1.1	背景	46
2.1.2	磁気流体力学方程式	46
2.1.3	中性子星形成過程の状態方程式	47
2.2	近似 Riemann 解の求め方	47
2.2.1	方針	47
2.2.2	座標変換	48
2.2.3	固有値	51
2.2.4	固有ベクトル	52
	Alfvén 波	52
	エントロピー波	53
	fast 波と slow 波	54
2.2.5	振幅	56
2.2.6	補正項 G についての考察	57
2.3	テスト計算	58
2.3.1	衝撃波管問題	58
2.4	公式のまとめ	61
	基礎方程式系	61
	Roe 平均	61
	固有値	62
	固有ベクトル (entropy)	62
	固有ベクトル (Alfvén)	63
	固有ベクトル (fast)	63
	固有ベクトル (slow)	64
	振幅	65
2.5	参考	65
2.5.1	1 次元流体力学方程式の場合	65
	Roe 平均	66
	固有値	66
	右固有ベクトル	67
	振幅	67
	固有モードへの分解	67
2.5.2	3 次元流体力学方程式の場合	67
	Roe 平均	68
	固有値	68
	右固有ベクトル	68
	振幅	69
	固有モードへの分解	69

3	CIP-MOCCT 法による磁気流体力学方程式の解法	71
3.1	はじめに	71
3.2	磁気流体力学方程式の移流形式での表現	71
3.3	CIP 法	72
3.3.1	CIP 法の基礎	73
3.3.2	3次元への拡張と流体力学への応用	74
3.3.3	移流項	75
3.3.4	非移流項	77
3.4	CT 法	79
3.5	MOC 法	80
3.5.1	誘導方程式	80
3.5.2	磁気張力項	82
3.6	人工粘性	82
3.6.1	von-Neumann 型人工粘性	82
3.6.2	Ogata & Yabe (1999) の人工粘性	83
3.7	まとめ	83
3.7.1	計算の流れ	83
3.7.2	テスト計算	84
4	保存保証型 CIP コードの開発に向けて	91
4.1	イントロダクション	91
4.2	CIP-CSL2 法	91
4.3	時間発展	92
4.4	計算例	92
4.4.1	1次元衝撃波管問題	94
4.4.2	ケルビンヘルムホルツ不安定性	98
5	一般相対論的磁気流体力学の数値解法	101
5.1	相対論的磁気流体力学シミュレーションの必要性和現状	101
5.2	基礎方程式	102
5.2.1	GRMHD 方程式の4次元形式	102
5.2.2	GRMHD 方程式の3+1形式	103
5.2.3	GRMHD 方程式のベクトル形式	107
5.2.4	ブラックホールの計量	108
5.3	GRMHD 方程式の数値解法	109
5.3.1	Simplified TVD Method	109
5.3.2	保存量からの基本量の導出	110
5.4	シミュレーションの計算例	111
5.4.1	宇宙ジェット	111
5.4.2	ブラックホールからのエネルギーの引き抜き	113
5.5	現状の問題点と今後	115

6	相対論的磁気流体における保存量からの基本量の導出	119
6.1	はじめに	119
6.2	相対論的磁気流体力学方程式 (保存系)	119
6.3	反復解法の必要性	121
6.4	2-D Newton's method (KO : Koide et al. 1999)	121
6.5	1-D Newton's method (DZ : Del Zanna et al. 2002)	123
6.5.1	Del Zanna の 1-D Newton's method	123
6.5.2	Cardano's method	125
6.5.3	Cardano 法に基づいて W を求める	126
6.5.4	W' に関する記述	127
6.5.5	変換後の基本量の値	128
6.6	HLL scheme	128
6.6.1	なぜ HLL scheme なのか?	128
6.6.2	HLL Flux の導出	129
6.7	テスト計算	130
6.7.1	Alfvén Wave の伝播	130
6.7.2	MHD Shock-tube Riemann problems	133
6.7.3	二種類の Blast Wave	135
6.7.4	反復法 (KO と DZ) の比較	137
6.8	考察	141
7	Vlasov 方程式の数値解法	143
7.1	イントロダクション	143
7.2	2 次元 Vlasov-Poisson System	144
7.2.1	The Splitting Scheme	144
7.2.2	CIP 法	145
7.2.3	CIP 法の Vlasov 方程式への適用	146
7.2.4	解法のまとめ	146
7.2.5	線形ランダウ減衰	147
7.3	3 次元 Vlasov-Maxwell System	148
7.3.1	Symplectic Integrator	148
7.3.2	3 次元 Vlasov 方程式の解法	150
7.3.3	Maxwell 方程式の解法	151
	電磁波の伝播	151
	電流項	151
7.3.4	解法のまとめ	151
7.3.5	計算例: Buneman 不安定	154
II	数値シミュレーションの可視化	156
8	IDL を使った可視化	157

8.1	準備	157
8.1.1	使用するファイル	157
8.1.2	設定	159
	メインプログラム、プロシージャの保存	159
	パスの設定	160
	デバイスの設定	161
8.2	データの読み込み	162
8.3	tvcn による図の作成	164
8.3.1	表示したい物理量の選択	164
8.3.2	カラースケール範囲の指定	167
8.3.3	コントラストや矢印の指定	167
8.3.4	タイトルの指定	172
8.3.5	文字の指定	173
8.3.6	一部の領域だけ拡大表示させる方法	174
8.3.7	複数の図を表示する方法	176
8.3.8	流線を表示させる方法	179
8.4	ファイルの出力方法	181
8.4.1	PNG ファイルの出力	181
8.4.2	EPS ファイルの出力	183
8.4.3	連続スナップショットの作成方法	185
8.4.4	ムービーの作成	188
8.5	カラーの選択	189
8.5.1	白黒の EPS ファイルを作成する場合	189
8.5.2	カラーの EPS ファイルを作成する場合	190
8.6	トラブルシューティング	191
8.6.1	“No such file or directory”	191
8.6.2	“Corrupted f77 unformatted file detected.”	191
8.7	ソースファイル	192
8.7.1	サンプルデータ用のソースファイル	192
9	AVS を使った可視化	196
9.1	可視化ソフトウェア AVS とは	196
9.1.1	AVS/Express の特徴	196
9.1.2	AVS の画面	197
	ネットワークエディタ	198
	コントロールパネル／ビューワーウィンドウ	199
9.1.3	モジュール	199
9.1.4	モジュールネットワークファイルの読み込み・保存	200
9.1.5	AVS/Express の終了	200
9.1.6	付属 CD-ROM の集録内容	200
9.2	1 time step の 3 次元データの可視化	201
9.2.1	field データファイルの作成	201

9.2.2	表示領域の設定	203
	field データファイルの読み込み -Read Field	203
	境界の表示 -bounds / 座標軸の表示 -Axis3D	204
9.2.3	スカラーデータの可視化	204
	等値面の表示 -isosurface	204
	等値ボリュームの表示 -isovolume	206
	データの最大値、最小値の設定 -set minmax	206
	数式演算処理を行う -data math	209
	格子に沿った断面の密度コントラストの表示 -orthoslice	209
9.2.4	ベクトルデータの可視化	212
	流線の表示 -streamlines	212
	矢印の表示 -Arrow1	214
9.2.5	補助的なデータの可視化	217
	カラーバーの表示 -LegendHoriz/LegendVert	217
	タイトルの表示 -TextTitle	220
9.2.6	モジュールごとの表示の設定	220
9.2.7	画像の出力	225
	ビューワーウィンドウの出力 1 - 画像として保存 -OutputField/Write Image	225
	ビューワーウィンドウの出力 2 -PostScript ファイルとして保存	226
9.3	netCDF のデータの可視化	229
9.4	応用編	231
9.4.1	連続の field データファイルの読み込み	231
9.4.2	アニメーションの作成	232
	時間変化の動画	232
	スライス面が変化する動画	236
	動画ファイルを使ったムービーの作成	240
9.5	付録	243
10	観測的可視化の目的	247
10.1	モンテカルロ輻射輸送計算法	247
10.1.1	計算手順	248
10.1.2	光子発生	248
10.1.3	コンプトン散乱	249
10.1.4	スペクトル&イメージ	249
10.2	3次元磁気降着円盤の輻射スペクトル	250
10.2.1	計算手順	250
10.2.2	電子温度	250
10.2.3	結果	250
10.3	まとめと今後の展開	251

III 数値シミュレーションのデータ解析	255
11 Fourier 解析	256
11.1 数値計算ライブラリ SSLII による高速フーリエ変換	257
11.2 スペクトルの計算法	263
11.3 おまけ：乱流の発生法、スペクトルの制御、非圧縮性ゆらぎの生成	267
11.3.1 乱流の発生法 (逆実 FFT の注意点)	267
11.3.2 任意のパワースペクトルを持つゆらぎの与え方	267
11.3.3 非圧縮性ゆらぎの与え方	269
11.4 富士通 SSL II の離散型実フーリエ変換サブルーチン	269
12 Minkowski functionals を用いた構造解析	273
12.1 定義と計算法	273
12.1.1 1次元の Minkowski functionals $(\mathcal{G}, \mathcal{C})$	274
12.1.2 2次元の Minkowski functionals $(\mathcal{G}, \mathcal{C}, \mathcal{S})$	275
12.1.3 3次元の Minkowski functionals $(\mathcal{G}, \mathcal{C}, \mathcal{S}, \mathcal{V})$	276
12.2 閾値 ρ_{th} による Minkowski functionals の変化	277
12.3 異方性の検出	279
12.4 閾値 ρ_{th} を変化させた計算	280

第 I 部

磁気流体力学方程式の数値解法

第 1 章

CANS を使った磁気流体力学シミュレーション

東京大学 花山 秀和

1.1 CANS とは

1.1.1 宇宙磁気流体シミュレーションソフトウェア

¹ 計算機と計算科学技術の発達に伴って、物理法則にしたがって時間発展する系の進化を計算機を用いて追跡するという数値実験が従来の理論と実験・観測に加わる第3の科学研究手法として確立してきた。天体现象は地上での実験が困難であることが多い。従って、計算機の中に仮想的な宇宙をつくって実験を行い、その結果を観測と比較することを通してモデルの妥当性を検証し、現象の物理機構を明らかにしていくという計算科学的手法が特に有効である。

千葉大学を始めとするグループでは、科学技術振興事業団計算科学技術活用型のプロジェクトとして「宇宙シミュレーション・ネットラボラトリーシステムの開発」を実施し、高速ネットワークを活用した宇宙シミュレーションのバーチャルラボラトリーの構築を目標として研究開発を進めてきた。このラボラトリーの構成要素として開発されたのが、宇宙磁気流体現象のシミュレーション研究を支援するソフトウェアライブラリー 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、さまざまな宇宙磁気流体現象の数値シミュレーションを行い、シミュレーション結果を可視化することができる。宇宙は様々な活動性と多様性に満ちているが、フレアやジェットをはじめとして、磁場と強く相互作用するプラズマのマクロな運動がエネルギー解放や質量輸送の起源になっている現象が数多くある。CANS が扱うことができるのは、このようにマクロな流体・磁気流体方程式で記述できる現象である。

CANS では自己重力、磁力線方向に依存する非等方熱伝導、磁気拡散、輻射冷却などの物理過程を取り入れることができ、1次元から3次元に至る種々の天体磁気流体現象の数値実験に適用できる。

¹ 1.1.1 から 1.1.4 までは、松元亮治氏 (千葉大学) による「全体の説明」<http://www.kwasan.kyoto-u.ac.jp/~tanuma/JST/matsumoto.pdf> および田沼俊一氏 (京都大学) による「とりあえずの簡単な使い方」http://www.kwasan.kyoto-u.ac.jp/~tanuma/cans_howtouse.html を参考にさせていただいた。

1.1.3 CANS の特徴

CANS の最大の特徴は、シミュレーションコードに加えて、宇宙シミュレーションの典型的問題（基本課題）のシミュレーションモデル（初期条件、境界条件等）、推奨パラメータ、各モデルの解説、シミュレーション結果の動画、可視化・解析ツール等をセットにして公開している点である。基本課題のテーマは各種流体・磁気流体不安定性をはじめとして、衝撃波伝播、ジェット、フレア、星形成など多岐にわたっている。シミュレーションコードだけを公開しても、それを使いこなすことは一般には困難である。CANS では、各種初期モデルのソースコードとその解説、入力パラメータのサンプル等が公開されているため、これらを手がかりにして、利用者は容易にシミュレーションモデルを再構成し、新たなシミュレーションを実施することができる。CANS ではシミュレーション結果データの入出力フォーマットとして netCDF 形式も利用可能にしている。これにより、netCDF 形式をサポートしている各種可視化ソフトウェアを用いてシミュレーション結果を可視化することが可能である。

1.1.4 CANS の構成

CANS は流体・磁気流体シミュレーションコードの共通部分と、各基本課題のモジュール及び解説ページから構成されている。磁気流体方程式に従う時間発展を解くシミュレーションエンジンとして、修正 Lax-Wendroff 法、Roe 法、CIP-MOCCT 法の3種類のエンジンがあり、エンジン部分だけを取替えてシミュレーションを実施することも可能である。CANS には、1次元パッケージ、2次元、3次元パッケージに加えて並列計算機用に MPI を用いて並列化したモジュールもある。

1.2 CANS を使ってみよう！

まず、CANS のインストールから計算実行および計算結果の可視化まで簡単に解説する。CANS の使用方法に関しては開発者である松元亮治氏（千葉大学）、横山央明氏（東京大学）らによる説明が載せられている以下の web ページを是非参照してもらいたい。

CANS の Web ページ

<http://www.astro.phys.s.chiba-u.ac.jp/netlab/astro/>

CANS の配信ページ

<http://www-space.eps.s.u-tokyo.ac.jp/~yokoyama/etc/cans/>

CANS の説明書（ドキュメント）

<http://www.kwasan.kyoto-u.ac.jp/~tanuma/JST.html>

1.2.1 動作環境

CANS による数値計算コードは Fortran で記述されている。したがって動作環境として Fortran の計算コードがコンパイルできる環境が必要がある。UNIX 環境における開発者によるテストにより、少なくとも以下の環境での動作が確認されている。以下の環境以外でも UNIX 環境であれば、多くの場合で問題なく動作するようコーディングがなされている。

FreeBSD4.2

Linux 2.2

IRIX64 6.5

SunOS5.7
HP-UX 11.0
SUPER-UX 11.1 (SX5)
UXP/V (VPP)

1.2.2 インストール

CANS ホームページから圧縮された CANS データファイルを各自のホームディレクトリにダウンロードして、解凍する。CANS には「開発版」と「リリース版」があり、現在バージョン 3.0 が配布されている。開発版は主に開発中の計算コードによりテストがなされている段階のものであり、最新の修正が加えられている一方でバグなどが含まれている場合もある。リリース版では既に様々なテストがなされて安定して動作することが確認されているものである。従ってリリース版は計算コードの信頼性が高い一方で更新が開発版に比べてやや遅くなっている。ここではリリース版を使用する。

リリース版

`http://www.space.eps.s.u-tokyo.ac.jp/~yokoyama/etc/cans/src/cans-release.tgz`

リリース版をホームディレクトリにダウンロードし tar で展開する。

```
>cd  
>wget http://www.space.eps.s.u-tokyo.ac.jp/~yokoyama/etc/cans/src/cans-release.tgz  
>tar -zxvf cans-release.tgz
```

1.2.3 内容確認

CANS の解凍後、CANS のディレクトリに入り内容を確認する。

```
>cd cans  
>ls  
Develop.txt Models.pdf README      cans1d/ cans3d/ htdocs/  
Makefile    NonLTE      Readme.pdf cans2d/ cansnc/ idl
```

pdf ファイルや txt ファイルおよび README ファイルは松元亮治氏（千葉大学）、横山央明氏（東京大学）、福田尚也氏（岡山理科大学）によって作成された CANS にある様々な計算モジュール群の解説文や使用に当たっての手引きであり一読されることを是非お勧めする。また、他のディレクトリにおいてもそれぞれのプログラムの機能などの説明が詳しく記載された README などのファイルがあり、CANS を用いて計算を行うための手助けとなる。

ディレクトリ cans1d、cans2d、cans3d には 1 次元、2 次元、3 次元における基本課題や共通して使用するプログラムが入っている。ディレクトリ cansnc には netcdf フォーマットで入出力するためのプログラムが入っている。ディレクトリ IDL には IDL で可視化を行う際使用するプログラムが入っている。

次にディレクトリ cans2d に入り内容を確認する。

```

>cd cans2d
>ls
bc/          cndbicg/      cndsor/      cndsormpi    common/
commonmpi/   hdmlw/        htcl/        md_advect/   md_awdecay/
md_cloud/    md_cme/       md_cndsp/    md_cndtb/    md_corjet/
md_efr/      md_itmhdshktb/ md_itshktb/  md_jetprop/  md_kh/
md_mhd3dkh/  md_mhd3dshktb/ md_mhdcloud/ md_mhdcondtb/ md_mhdgwave/
md_mhdkh/    md_mhdshktb/  md_mhdsn/    md_mhdwave/  md_mri/
md_parker/   md_reccnd/    md_recon/    md_recon3/   md_rt/
md_sedov/    md_shkref/    md_shktb/    md_sndwave/  md_thinst/
mdp_awdecay/ mdp_cme/      mdp_cndsp/   mdp_cndtb/   mdp_corjet/
mdp_efr/     mdp_itmhdshktb/ mdp_itshktb/ mdp_jetprop/ mdp_kh/
mdp_mhd3kh/  mdp_mhd3shktb/ mdp_mhdcndtb/ mdp_mhdkh/   mdp_mhdshktb/
mdp_mhdsn/   mdp_mhdwave/   mdp_mri/     mdp_recon/   mdp_recon3/
mdp_rt/      mdp_sedov/     mdp_shkref/  mdp_shktb/   mdp_thinst/

```

それぞれのディレクトリの内容に関して以下に記載する。

```

hdmlw/  流体力学方程式・MHD 方程式を改良 Lax-Wendroff + 人工粘性法で解くた
        めのモジュール

bc/      境界条件を定義するためのプロシジャ
common/  計算で使うさまざまな共通ルーチンを集めたモジュール

cndsor/  熱伝導を陰解法（時間精度 1 次：行列反転は Red Black SOR 法）で解くた
        めのモジュール
cndbicg/ 熱伝導を陰解法（時間精度 1 次：行列反転は BICG 法）で解くための
        モジュール
htcl/    放射冷却・静的加熱を陽解法で解くためのモジュール

commonmpi/ 計算で使うさまざまな共通ルーチンを集めたモジュール（MPI）
cndsormpi/ 熱伝導を陰解法（時間精度 1 次：行列反転は Red Black SOR 法）で解く
        ためのモジュール（MPI）

md_***/  2 次元基本課題モジュール [スカラー計算機用]（後述）
mdp_***/ 2 次元基本課題モジュール [パラレル計算機用]（後述）

```

ほとんどの 2 次元基本課題に対してスカラー計算機用（1CPU での計算）とパラレル計算機用（複数の CPU による並列計算）の 2 種類のパッケージが用意されている。それぞれはディレクトリ名で区別されており、例えば md_mhdsn と mdp_mhdsn とはスカラー計算機とパラレル計算機で基本的に同じ計算を行う。

2 次元基本課題のそれぞれの内容について以下にまとめたものを記載する。

```

md_advect/ 単純移流 [移流]
md_awdecay/ 大振幅 Alfvén 波減衰不安定 [等温 3 成分 MHD]
md_cloud/ 等温自己重力収縮 [等温流体、自己重力]
md_cme/ 太陽コロナ質量放出: Low 解 [MHD、球座標]
md_cndsp/ 球対称単純熱伝導 [熱伝導、円柱・球座標]
md_cndtb/ 単純熱伝導 [熱伝導]
md_corjet/ 太陽コロナジェット [MHD、重力、抵抗]
md_efr/ 太陽浮上磁場: Parker 不安定 [MHD、重力]
md_itmhdshktb/ 等温 MHD 衝撃波管 [等温 MHD]
md_itshktb/ 等温衝撃波管 [等温流体]
md_jetprop/ ジェット伝播 [流体、円柱座標]
md_kh/ Kelvin-Helmholtz 不安定性 [流体]
md_mhd3kh/ 3 成分 MHD Kelvin-Helmholtz 不安定性 [3 成分 MHD]
md_mhd3shktb/ 3 成分 MHD 衝撃波管 [3 成分 MHD]
md_mhdcld/ 等温 MHD 自己重力収縮 [等温 MHD、重力]
md_mhdcndtb/ 単純 MHD 熱伝導 [MHD、熱伝導]
md_mhdgwave/ 成層大気中の MHD 波動 [MHD]
md_mhdkh/ MHD Kelvin-Helmholtz 不安定性 [MHD]
md_shktb/ 衝撃波管 [流体]
md_mhdsn/ MHD 超新星残骸 [MHD、円柱・球座標]
md_mhdwave/ 線形 MHD 波動伝播 [MHD]
md_mri/ 磁気回転 (Balbus-Hawley) 不安定 [MHD、潮汐 Coriolis 力]
md_parker/ 銀河 Parker 不安定 [MHD、重力]
md_mhdshktb/ MHD 衝撃波管 [MHD]
md_reccnd/ 熱伝導磁気リコネクション [MHD、抵抗、熱伝導]
md_recon/ 磁気リコネクション [MHD、抵抗]
md_recon3/ 3 成分磁気リコネクション [3 成分 MHD、抵抗]
md_rt/ Rayleigh-Taylor 不安定性 [流体、重力]
md_sedov/ 超新星残骸: Sedov 解 [流体、円柱・球座標]
md_shkref/ 反射衝撃波 [流体]
md_shktb/ 衝撃波管 [流体]
md_thinst/ 熱不安定 [放射冷却]

```

1.2.4 コンパイル

~/cans において、make コマンドにより libcans1d.a などのライブラリファイルを作成する。(使用マシンが変わったり様子がおかしい場合は再コンパイルが必要になる)

```

>cd ~/cans
>make 'FC=f77'
>ls
libcans1d.a, libcans2d.a, libcans3d.a, libcansnc.a

```

ここで'FC=f77' というのは、システムの Fortran コンパイラのコマンド名を入力する。システムによって f90、g77、f77 などのコンパイラを使用する。

2 次元基本課題で並列計算を行う場合には、まず~/cans/cansnc のディレクトリにおいて make clean した後、make 'FC=mpif77' を実行してコンパイルを行い、その後~/cans/cans2d のディレクトリにおいて make clean した後、make 'FC=mpif77' を実行し、最後に同ディレクトリで make mpi というコマンドを実行する。

```
>cd ~/cans/cansnc
>make clean
>make 'FC=mpif77'
>cd ../cans2d
>make clean
>make 'FC=mpif77'
>make mpi
```

1.2.5 プログラムの実行

例えば、2次元基本課題の一つである MHD 超新星残骸について計算を行うとする。
まず、cans2d/md_mhdsn に入り、表 1.1にあるファイルが作成されているか確認する。

```
>cd cans2d/md_mhdsn
>ls
Makefile Makefile-nc Makefile-pgnc anime.pro bnd.f
main.f    model.f      pldt.pro      rddt.pro      rdnc.pro
```

ファイル名	内容・機能
Makefile	'make' コマンドの内容
Makefile-nc	netcdf で出力したいときに用いる
Makefile-pgnc	pgplot で可視化させたいときに用いる
main.f	メインプログラム
model.f	初期条件のサブルーチン
bnd.f	境界条件のサブルーチン
rddt.pro	IDL で出力データを読み込むためのファイル
rdnc.pro	netcdf で出力されたデータを読み込むためのファイル
pldt.pro	IDL で 2 次元コントアをプロットするためのファイル
anime.pro	IDL でアニメーション表示させるためのファイル

表 1.1: MHD 超新星残骸モジュールにある初期ファイルとその説明

make を実行すると、実行ファイル'a.out' と共に、'ay.dac' などの計算結果が格納されたファイル（拡張子:dac）、および計算パラメーターが出力された params.txt が作成される²。

²Segmentation fault のエラーが発生した場合は、/md_mhdsn/main.f の 5 行目にある ix や jx というパラメーターを変更するとうまくいく場合がある。例 parameter(ix=50,jx=50)

```
>make 'FC=f77' ←並列計算の時は'FC=mpif77' とする
>ls
Makefile Makefile-nc Makefile-pgnc a.out      anime.pro
ay.dac   bnd.f        bnd.o      bx.dac    bz.dac
main.f   main.o        model.f    model.o   params.txt
pldt.pro pr.dac         rddt.pro   rdnc.pro  ro.dac
t.dac    vx.dac        vz.dac     x.dac     z.dac
```

1.2.6 IDL の準備

使用マシンには計算結果を可視化するためのソフト IDL(The Interactive Data Language) がインストールされているものとする。IDL による可視化については第 7 章に詳しく説明されている。

~/cshrc 中の一文に、以下のように IDL のパスを追加する (詳しくは第 7 章参照)。

```
setenv IDL_PATH +/usr/local/rsi/idl/lib:~/cans/idl/
```

これにより~/cans/idl 内の IDL プログラム (dacgetparam.pro など) が使用可能になる。ただし、/usr/local/rsi/idl/lib の部分は使用する環境に依存するので注意が必要である (詳しくは第 7 章参照)。

1.2.7 IDL によるデータの読み込みと表示

まず IDL を起動させ、計算結果が格納されている dac ファイルを読み込む。

```
>idl ← IDL の起動
IDL>.r rddt ←データの読み込み
```

[読み込みの例]

密度コントラストなどのデータを 2 次元プロットして表示させる。

```
IDL>.r pldt ←データの表示
Plot columns & rows ? : 1,1 ←window に表示させるプロットの配列を指定
Variable for color-maps ? (ro,pr,te) : ro ←表示させる物理量を選択
Start step ? : 10 ←時間 (書き出し番号) を指定
```

以上のように入力すると図 1.1 のように window に計算結果が表示される。

カラーでコントラストを表示させるには次のようにカラーテーブルを起動させる。

```
IDL>device,decomposed=0
IDL>loadct,5
```

カラーテーブルを変更するときは xloadct と入力しカラーテーブルを選択する。同様の手順でアニメーションも表示される。

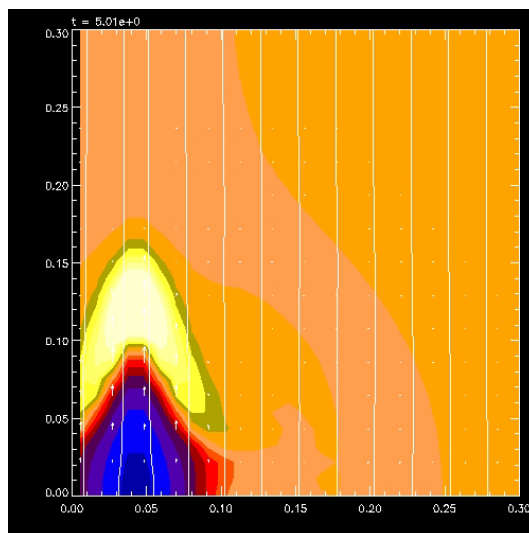


図 1.1: 密度コントアの2次元プロット。白い縦線は磁力線を表す。

```
IDL>.r anime    ←アニメーションで表示
Select a variable ? (ro,pr,te,vx,vy,bx,by,az):ro
```

他の各モジュールでの計算結果に関しては、デモページとして以下の Web ページで公開されているので参照してもらいたい。

CANS 2D デモページ

<http://www.astro.phys.s.chiba-u.ac.jp/netlab/cans/movie2/frame.html>

1.3 計算の流れ

計算の全体的な流れは

`~/cans/cans2d/md_mhdsn/main.f`

に記述されているプログラムに沿うものである。

`main.f`はモデルの設定や計算エンジンなどの多くのサブルーチンで構成されているが、大まかには初期設定、主計算、終了処理の3つの項目に構成を分けることができる。以下、実際のプログラムを参照しながら各項目について説明する。

1.3.1 初期設定 - `main.f`

初期設定の項目では、各パラメーターの定義、出力の設定、グリッドやモデルの設定、再計算プログラム、境界条件、磁力線の設定、エラーチェックなど、流束計算を行うために必要な初期条件の設定を行う。

```

c=====|
c   array definitions
c=====|
c   implicit double precision (a-h,o-z)
c   parameter (ix=103,jx=102)
!ix は X 軸のグリッド数
!jx は Z 軸のグリッド数

c   dimension x(ix),xm(ix),dx(ix),dxm(ix)
c   dimension z(jx),zm(jx),dz(jx),dzm(jx)

c   dimension ro(ix,jx),pr(ix,jx)
c   dimension vx(ix,jx),vz(ix,jx)
c   dimension bx(ix,jx),bz(ix,jx)
c   dimension ay(ix,jx)
! 各パラメーターの配列を定義

c=====|
c   prologue
c=====|
c   mcont=0
c   ndi=1000
!mcont、ndi は再計算用のパラメーター
!mcont=1 で再計算モード
! 再計算とはこの場合 1 度目の計算の後に続きの計算をすることを指す
! 再計算は終了時間 tend や終了ステップ数 nstop を変更して行う

c-----|
c   parameters
c   margin: 1 in MLW, 2 in Roe, 4 in CIP

c   margin=4
!margin は計算領域の境界の外側に余分に確保するグリッド数

c-----|
c   file open
!(netcdf 省略)
!netcdf フォーマットはここでは扱わない
c   mf_params=9
c   call dacdefparam(mf_params,'params.txt')
c   mf_t =10
c   call dacdef0s(mf_t,'t.dac',6)
c   mf_ro=20
c   call dacdef2s(mf_ro,'ro.dac',6,ix,jx)
c   mf_pr=21
c   call dacdef2s(mf_pr,'pr.dac',6,ix,jx)
c   mf_vx=22
c   call dacdef2s(mf_vx,'vx.dac',6,ix,jx)
c   mf_vz=24
c   call dacdef2s(mf_vz,'vz.dac',6,ix,jx)
c   mf_bx=25
c   call dacdef2s(mf_bx,'bx.dac',6,ix,jx)
c   mf_bz=27
c   call dacdef2s(mf_bz,'bz.dac',6,ix,jx)
c   mf_ay=28
c   call dacdef2s(mf_ay,'ay.dac',6,ix,jx)
c   call dacputparamc(mf_params,'comment','cans2d md_mhdsn')
c   call dacputparami(mf_params,'ix',ix)
c   call dacputparami(mf_params,'jx',jx)
c   call dacputparami(mf_params,'margin',margin)
! 出力は /cansnc の中にある dacdefparam.f などのサブルーチンにより行われる
! ここでは出力ファイル (dac ファイル) の作成とパラメーターの出力がされている

```

```

c-----|
c  initialize counters
      nd=1
!ndは書き出し回数
      time = 0.0
!timeは時間
      timep = 0.0
      ns = 0
!nsは計算ステップ数
      merr = 0
!merrはエラーカウンタの数
c-----|
c  time control parameters
c      nstop : number of total time steps for the run
      tend=5.0
!tendは終了時刻
      dtout=0.5
!dtoutは書き出し時間の間隔
      nstop=1000000
!nstopは終了ステップ数
!time > tend または ns > nstop になったとき計算は終了する
c      dtout=1.d-10
c      nstop=3
c-----|
c  setup numerical model (grid, initial conditions, etc.)
      call model(idf,ro,pr,vx,vz,bx,bz,gm,margin,x,ix,z,jx
&      ,mf_params)
! サブルーチンmodel.f
! グリッド間隔の作成、および、周囲のガス、磁場、初期の爆発などを配置
! 各サブルーチンについては後述する

c-----|
! (再計算用の読み込みプログラムは省略)
c-----|
c      ready

      call grdrdy(dx,xm,dxm,x,ix)
      call grdrdy(dz,zm,dzm,z,jx)
! サブルーチンgrdrdy.f
! セルとセルの間隔やグリッドとグリッドの間隔を定義

      call bbtoaa_c(ay,bz,bx,dzm,dxm,x,ix,jx)
! サブルーチンbbtoaa_c.f
! 磁力線を描くためにベクトルポテンシャルを作成

      call bnd(margin,ro,pr,vx,vz,bx,bz,ay,ix,jx)
! サブルーチンbnd.f
! 境界条件を適用

      floor=1.d-9
!floorは密度や圧力などの下限値

      call chkdav(n_floor,ro,vx,vz,floor,ix,jx)
      call chkdav(n_floor,pr,vx,vz,floor,ix,jx)
! サブルーチンchkdav.f
! 密度や圧力がfloorを下回っていた場合はfloorの値に修正する

```

```

c-----|
c      data output
!(netcdf 省略)
  mf_x=11
  call dacdef1d(mf_x,'x.dac',6,ix)
  write(mf_x) x
  mf_z=12
  call dacdef1d(mf_z,'z.dac',6,jx)
  write(mf_z) z
  call dacputparamd(mf_params,'gm',gm)
  write(mf_t) time
  write(mf_ro) ro
  write(mf_pr) pr
  write(mf_vx) vx
  write(mf_vz) vz
  write(mf_bx) bx
  write(mf_bz) bz
  write(mf_ay) ay
  write(6,913) ns,time,nd
  nd=nd+1
! 初期状態の出力をする

```

1.3.2 計算スキームの選択-main.f

この部分では CFL(Courant-Friedrich-Lewy) 条件により dt を決定し、修正 Lax-Wendroff スキームにより移流計算を行う。境界条件を適用後、エラーチェックを行い、出力設定で決めた条件により計算結果の出力が行われる。

```

c=====|
c      time integration
c=====|
1000 continue
      ns = ns+1
      mwflag=0
!mwflag はサブルーチン cfl_m.f で異常終了するときは 0 の値のままで、その時の状態を出力させるために使うパラメーター

c-----|
c      obtain time spacing

      safety=0.4d0
!safety は dt を安全に決定するためのパラメーター（クーラン数）

      dtmin=1.d-10
!dtmin は dt の最小値
!dt が dtmin を下回ると数値不安定が起きるとみなされて計算は終了する
      call cfl_m(dt,safety,dtmin,merr,gm,ro,pr,vx,vz,bx,bz
&              ,dx,ix,dz,jx)
! サブルーチン cfl_m.f
! CFL 条件から dt を決定する

```

```

        if (merr.ne.0) goto 9999
! ここでエラーが発生したら終了する

        timep = time
        time = time+dt
! それまでの時間に dt を足して時間を進める

c-----|
c      solve hydrodynamic equations
c
c                                hdmlw - start >>>
        qav=3.d0
! qav は人工粘性の大きさを調整するパラメーター
        call mlw_m_c(ro,pr,vx,vz,bx,bz,ay
&                ,dt,qav,gm,x,xm,dx,dxm,ix,dz,dzm,jx)
! サブルーチン mlw_m_c.f
! 修正 Lax-Wendroff スキームで差分方程式を計算し状況を更新するメインエンジン
c                                hdmlw - end <<<
! (Roe 法は未完成なので省略)

        call bnd(margin,ro,pr,vx,vz,bx,bz,ay,ix,jx)
! サブルーチン bnd.f
! 境界条件を適用

        floor=1.d-9
        call chkdav(n_floor,ro,vx,vz,floor,ix,jx)
        call chkdav(n_floor,pr,vx,vz,floor,ix,jx)
! サブルーチン chkdav.f
! 更新後の密度や圧力が下限値を下回っていないかチェック

c-----|
c      data output
! ここでは計算結果の出力を行う
        mw=0
! mw は出力させるためのパラメーター、普段は0の値をとる
        nt1=int(timep/dtout)
        nt2=int(time/dtout)
        if (nt1.lt.nt2) mw=1
! ある書き出し時刻から時間間隔 dtout だけ経過したとき、mw=1 となって計算結果が出力される

        if (mw.ne.0) then
! (netcdf 省略)
        write(mf_t) time
        write(mf_ro) ro
        write(mf_pr) pr
        write(mf_vx) vx
        write(mf_vz) vz
        write(mf_bx) bx
        write(mf_bz) bz
        write(mf_ay) ay
        write(6,913) ns,time,nd
        nd=nd+1
        mwflag=1
! 異常なく計算が終了する時にはmwflag=1 となり終了時に出力はされない
        endif

c-----|
c      loop test
        if (ns .lt. nstop .and. time .lt. tend) goto 1000
! ns が nstop を超えたときまたは time が tend を超えたときは計算終了
! そうでない場合はループして再び主計算が行われる

```

1.3.3 終了処理 -main.f

最後に、初期に設定した終了条件により計算は終了する。また、エラーチェックなどにより異常終了する場合は終了直前の状態を出力するように設定されている。

```

c=====|
c   epilogue
c=====|
9999 continue
! 異常終了の時はここにくる
c-----|
c data output
c   if (mwflag.eq.0) then
! サブルーチン cfl_m.f で異常終了される時にはmwflag=0になっているため計算結果が出力される
c       write(6,913) ns,time,nd
! (netcdf 省略)
c       write(mf_t) time
c       write(mf_ro) ro
c       write(mf_pr) pr
c       write(mf_vx) vx
c       write(mf_vz) vz
c       write(mf_bx) bx
c       write(mf_bz) bz
c       write(mf_ay) ay
c   endif
c-----|
c file close
! (netcdf 省略)
c-----|
c ending message
c   write(6,915) ns,time
c   if (merr.eq.0) then
c       write(6,*) '   ### normal stop ###'
c   else
c       write(6,*) '   ### abnormal stop ###'
c   endif
! エラーがない場合はnormal stop、エラーがあった場合はabnormal stop の表示が出力され計算が終了する
c   stop
913 format (1x,' write      ','step=',i8,' time=',e10.3,' nd =',i3)
915 format (1x,' stop      ','step=',i8,' time=',e10.3)
end

```

1.3.4 モデルの設定 -model.f

main.fと同じディレクトリにある

X 軸および Z 軸の値 $x(i)$ 、 $z(j)$ をそれぞれ定義する

密度 $\rho(1.0)$ 、圧力 $\text{prism}(10^{-8})$ 、プラズマベータ $(10^{-5})^3$ 、磁場 $b_0(0.1853)$ を決める

爆発のエネルギーを、圧力（ピーク値 1）のガウス分布として与える

³プラズマベータ = ガス圧 / 磁気圧

```

c=====|
      subroutine model(idf,ro,pr,vx,vz,bx,bz,gm,margin,x,ix,z,jx
      & ,mf_params)
c=====|
      implicit double precision (a-h,o-z)
c-----|
      dimension x(ix),dxm(ix)
      dimension z(jx),dzm(jx)
      dimension ro(ix,jx),pr(ix,jx),vx(ix,jx)
      dimension vz(ix,jx),bx(ix,jx),bz(ix,jx)
!x(i)、z(j)は、X軸、Z軸の座標の値
!例えばx(5)=0.d0、z(5)=0.d0の時は、(i,j)=(5,5)がxy座標での原点(0,0)に対応する
!dxm(i)、dzm(j)はxz座標でのグリッドの間隔の値

c-----|
c  parameters
c-----|
      gm=5./3.
!gmは比熱比ガンマ

c-----|
c  grid
c-----|
      dx0=1./real(ix-margin*2+1)
!X軸のグリッド間隔の基準値を定義、iは1からはじまることに注意

      do i=1,ix
        dxm(i)=dx0
      enddo
!ここでは間隔はどの場所でも同じdx0とする

      izero=margin+1
      izero=1
!対称境界上(軸上)に原点を置くためにはizeroは本来margin+1とりたいが
!ここではZ軸上で起こる問題を未然に回避するためにizero=1としている
      x(izero)=0.5*dx0
!これはiがizeroとizero-1の間(つまりセル上)にX軸の原点があることを意味する
      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
!x(izero)を基準としてdxmを加えていくことにより座標を定義する

      dz0=1./real(jx-margin*2+1)
!Z軸のグリッド間隔の基準値を定義、jは1からはじまることに注意

      do j=1,jx
        dzm(j)=dz0
      enddo
!ここでは間隔はどの場所でも同じdz0とする

```

```

    jzero=margin+1
    z(jzero)=0.
! X 軸上では計算上の問題が生じないのでこのように設定している
! グリッド上に Z 軸の原点がある (セル上でも特にかまわない)
    do j=jzero+1,jx
        z(j) = z(j-1)+dzm(j-1)
    enddo

    do j=jzero-1,1,-1
        z(j) = z(j+1)-dzm(j)
    enddo
! z(jzero) を基準として dzm を加えていくことにより座標を定義する

c-----|
c      store initial condition into common area
c-----|

    prism=1.e-8
! prism は星間ガスの圧力
    wexp=0.02
! wexp は爆発をガウス分布で配置するとき、圧力の値が圧力ピーク値の e 分の 1 になるまでの幅
c      prism=1/gm
    betai=1.0d5
! betai はプラズマベータ (= ガス圧 / 磁気圧) の逆数
    pi = acos(-1.0d0)
    b0=sqrt(prism*8*pi*betai)
! 圧力とプラズマベータから磁場の値を設定

    do j=1,jx
    do i=1,ix
        ro(i,j) = 1.
! 初期の星間ガスの密度の値を 1. にする
        vx(i,j) = 0.0
        vz(i,j) = 0.0
! 初期の星間ガスの速度は 0
        ss=sqrt(x(i)**2+z(j)**2)
! ss は xz 座標での原点からの距離
        pr(i,j) = prism*(1.-prism)*exp(-(ss/wexp)**2)
! 爆発を圧力のガウス分布として与える
! 爆発の中心は原点
c      pr(i,j) = 1/gm+0.1/gm*exp(-(ss/wexp)**2)
        bx(i,j) = 0.0
        bz(i,j) = b0
! 磁場は Z 方向に一樣に分布すると仮定
    enddo
    enddo

c-----|
c      write parameters to file
c-----|
! (netcdf 省略)
    call dacputparamd(mf_params,'gm',gm)
    call dacputparamd(mf_params,'wexp',wexp)
    call dacputparamd(mf_params,'prism',prism)
    call dacputparamd(mf_params,'betai',betai)
! 必要な定数を出力
    return
end

```

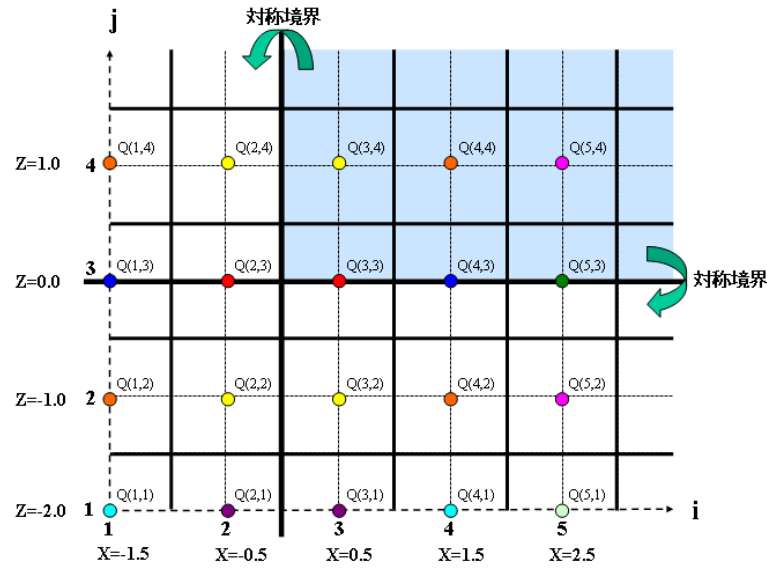


図 1.2: X 軸、Z 軸での境界条件、それぞれのグリッドの物理量が境界で対称的になっている様子を色分けであらわした

1.3.5 グリッド間隔の設定 -grdrdy.f

cans2d/common にある。セルの間隔 dx やグリッドの間隔 dxm を定義する。またグリッド間の中点 xm も定義する。

詳細は cans2d/common 内の README 参照。

1.3.6 ベクトルポテンシャルの設定 -bbtoa.c.f

/cans2d/common にある。磁力線を描くためにベクトルポテンシャル a_y を作成する。

詳細は cans2d/common 内の README 参照。

1.3.7 境界条件の設定 -bnd.f

bnd.f 自体は main.f と同じディレクトリにあるが、境界条件を計算領域に適用するサブルーチンは cans2d/bc ディレクトリ内にあって、bnd.f で指定されたサブルーチンが指定された境界で機能するようになっている。MHD 超新星残骸のモジュールは 2 次元円筒座標系での計算が行われるため、軸上は対称境界、外側は自由境界となっている。詳細は cans2d/bc 内の README 参照。以下に margin=2 のときの軸付近の境界を図 1.2 に示す。

1.3.8 下限値のチェック -chkdavl.f

/cans2d/common にある。あるパラメーターが floor を下回っていた場合、floor の値に修正する。そのとき n_events をカウントして何回 floor を使ったかわかるようにする。詳細は cans2d/common 内の README 参照。

1.3.9 CFL 条件の設定 -cfl_m.f

/cans2d/common にある。CFL 条件に基づいて dt を決定する。各セルに対しアルフベン速度、星間ガスの音速、ガスの運動速度を求め、全セルでの最大値を求める、さらにセルどうしの間隔 dx もしくは dy から dt の最小値を求める。 dt を決めるときには *safety* パラメーターをかけて数値不安定の成長を抑制する。詳細は cans2d/common 内の README 参照。

1.4 修正 Lax-Wendroff スキーム

サブルーチン mlw_m.c.f の内容の理解の助けとして磁気流体方程式の差分解法についてここで解説する。物理法則から導かれる基礎方程式を差分化して解くための具体例として、CANS で用いられている数値計算スキームの一つである修正 Lax-Wendroff スキームについて述べる。

1.4.1 基礎方程式

磁気流体力学の基礎方程式として、2次元円筒座標系における質量保存、運動量保存、エネルギー保存、磁場の誘導方程式は以下のように表される。

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1.1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p - \nabla \left(\frac{\mathbf{B}^2}{8\pi} \right) + \frac{1}{4\pi} (\mathbf{B} \cdot \nabla) \mathbf{B} \quad (1.2)$$

$$\frac{\partial}{\partial t} \left(e + \frac{\mathbf{B}^2}{8\pi} \right) + \nabla \cdot [(e + p) \mathbf{v} + \frac{1}{4\pi} \{ \mathbf{B} \times (\mathbf{v} \times \mathbf{B}) \}] = 0 \quad (1.3)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) \quad (1.4)$$

ここで $\rho, \mathbf{v}, p, t, \mathbf{B}$ はそれぞれ星間ガスの密度、速度、圧力、時間、磁場を表す。ここで、 $e = p/(\gamma - 1) + \rho v^2/2$ (単位体積あたりの熱+運動エネルギー)、 $\gamma = 5/3$ (比熱比) とした。

ここで、円筒座標系の ϕ 成分は、等方的な進化を仮定するため扱わず、動径方向 (R 方向) および軸方向 (Z 方向) の 2 次元の座標系を考える。

2次元円筒座標の場合、 $\mathbf{A}$ 、 ψ をそれぞれ任意のベクトル、スカラーとして

$$\nabla \cdot \mathbf{A} = \frac{1}{r} \frac{\partial}{\partial r} (r A_r) + \frac{\partial A_z}{\partial z} \quad (1.5)$$

$$\nabla \psi = \left(\frac{\partial \psi}{\partial r}, \frac{\partial \psi}{\partial z} \right) \quad (1.6)$$

であることに注意しなければならない。

$\rho \mathbf{v} \mathbf{v}$ はテンソル積で、

$$\begin{pmatrix} \rho v_r v_r & \rho v_r v_z \\ \rho v_z v_r & \rho v_z v_z \end{pmatrix} \quad (1.7)$$

であり、

$$(\nabla \cdot \rho \mathbf{v} \mathbf{v})_r = \frac{1}{r} \frac{\partial}{\partial r} (r \rho v_r v_r) + \frac{\partial \rho v_r v_z}{\partial z} \quad (1.8)$$

$$(\nabla \cdot \rho \mathbf{v} \mathbf{v})_z = \frac{1}{r} \frac{\partial}{\partial r} (r \rho v_z v_r) + \frac{\partial \rho v_z v_z}{\partial z} \quad (1.9)$$

を表す。

移流方程式を用いて基礎方程式を数値的に解く方法について述べる前に磁気流体力学について簡単に述べる。

1.4.2 磁気流体力学

電磁場に関するマクスウェルの方程式は、

$$\nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} \quad (1.10)$$

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{j} \quad (1.11)$$

のように表される。 c は光速、 $\mathbf{j}$ は電流密度である。ここで、星間ガスの条件として、流体現象の時間スケールが電磁波の振動周期よりはるかに長いことから変位電流は無視した。またガスが低密度であることから、媒質の透磁率も真空の場合と同じように 1 とする。式 (1.10)、(1.11) とオームの法則

$$\mathbf{j} = \sigma \left(\mathbf{E} + \frac{\mathbf{v} \times \mathbf{B}}{c} \right) \quad (1.12)$$

と組み合わせることにより磁場の誘導方程式

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) + \frac{c^2}{4\pi\sigma} \nabla^2 \mathbf{B} \quad (1.13)$$

が得られる。ここで σ は媒質の電気伝導度である。

流体が完全導体ならば電気伝導度 $\sigma \rightarrow \infty$ であり、磁場の誘導方程式は式 (1.4) となる。電気伝導度 $\sigma \rightarrow \infty$ を理想磁気流体力学極限と呼ぶ。

磁場によるローレンツ力は、

$$\begin{aligned} \frac{\mathbf{j} \times \mathbf{B}}{c} &= \frac{(\nabla \times \mathbf{B}) \times \mathbf{B}}{4\pi} \\ &= -\nabla \left(\frac{B^2}{8\pi} \right) + \frac{1}{4\pi} (\mathbf{B} \cdot \nabla) \mathbf{B} \end{aligned} \quad (1.14)$$

であり、運動方程式は、

$$\rho \left\{ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right\} = -\nabla p - \nabla \left(\frac{B^2}{8\pi} \right) + \frac{1}{4\pi} (\mathbf{B} \cdot \nabla) \mathbf{B} \quad (1.15)$$

となる。運動方程式 (1.15) と質量保存の式 (1.1) から運動量保存の式 (1.2) が得られる。エネルギー保存の式 (1.3) は、全エネルギーに磁場のエネルギーを加え、エネルギー流束にポインティングベクトル $(c/4\pi) \mathbf{E} \times \mathbf{B}$ を加えることによって得られる。

1.4.3 基礎方程式の保存形式での表現

円筒座標系における基礎方程式 (1.1) ～ (1.4) を解くために、まず基礎方程式 (1.1) ～ (1.4) を移流方程式の保存形式で表すことを考える。

Q を物理量、 F_r 、 F_z を各成分の方向に流れる流束、ソース項を S' として、式 (1.5)、(1.6) を用いて円筒座標系における移流方程式の保存形式

$$\frac{\partial Q}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r}(r F_r) + \frac{\partial}{\partial z} F_z = S' \quad (1.16)$$

で各成分ごとに表すと以下ようになる。

式 (1.1) は、

$$\frac{\partial \rho}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r}(r \rho v_r) + \frac{\partial}{\partial z}(\rho v_z) = 0 \quad (1.17)$$

式 (1.2) の R 成分は、

$$\frac{\partial \rho v_r}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r}[r\{\rho v_r^2 + p + \frac{1}{8\pi}(B_z^2 - B_r^2)\}] + \frac{\partial}{\partial z}(\rho v_r v_z - \frac{1}{4\pi} B_r B_z) = \frac{1}{r}(p + \frac{B_r^2 + B_z^2}{8\pi}) \quad (1.18)$$

同じく Z 成分は、

$$\frac{\partial \rho v_z}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r}\{r(\rho v_z v_r - \frac{B_r B_z}{4\pi})\} + \frac{\partial}{\partial z}(\rho v_z^2 + p + \frac{B_r^2 - B_z^2}{8\pi}) = 0 \quad (1.19)$$

式 (1.3) は、

$$\frac{\partial}{\partial t}(e + \frac{B^2}{8\pi}) + \frac{1}{r} \frac{\partial}{\partial r}\{r(e_p v_r + \frac{B_z e_y}{4\pi})\} + \frac{\partial}{\partial z}(e_p v_z - \frac{B_r e_y}{4\pi}) = 0 \quad (1.20)$$

式 (1.4) の R 成分は、

$$\frac{\partial B_r}{\partial t} + \frac{\partial}{\partial z}(-e_y) = 0 \quad (1.21)$$

同じく Z 成分は、

$$\frac{\partial B_z}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r}(r e_y) = 0 \quad (1.22)$$

である。ここで、 $e_p = \gamma p / (\gamma - 1) + \rho v^2 / 2$ 、 $e_y = -v_z B_r + v_r B_z$ とした。

1.4.4 差分方程式

MHD 超新星残骸モジュールでは移流方程式を解くスキームとして 2 段階修正 Lax-Wendroff スキームを用いている。修正 Lax-Wendroff スキームを適用させるために、式 (1.16) 以下のような直交座標系における移流方程式の保存形式に書き換える。

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial r} F_r + \frac{\partial}{\partial z} F_z = S' - \frac{F_r}{r} \quad (1.23)$$

ここで $S = S' - F_r/r$ とおき、便宜上 r を x に置き換えることにすると、式 (1.23) は

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} F_x + \frac{\partial}{\partial z} F_z = S \quad (1.24)$$

のようになる。

物理量 Q	流束 F_x	流束 F_z	ソース項 S
ρ	ρv_x	ρv_z	$-\frac{\rho v_x}{x}$
E	$e_p v_x + \frac{B_z e_y}{4\pi}$	$e_p v_z - \frac{B_x e_y}{4\pi}$	$-\frac{e_p v_x + \frac{B_z e_y}{4\pi}}{x}$
ρv_x	$\rho v_x^2 + p + \frac{B_z^2 - B_x^2}{8\pi}$	$\rho v_x v_z - \frac{B_x B_z}{4\pi}$	$\frac{\rho v_x^2}{x} + \frac{B_x^2}{4\pi x}$
ρv_z	$\rho v_x v_z - \frac{B_x B_z}{4\pi}$	$\rho v_z^2 + p + \frac{B_x^2 - B_z^2}{8\pi}$	$-\frac{\rho v_x v_z - \frac{B_x B_z}{4\pi}}{x}$
B_x	0	$-e_y$	0
B_z	e_y	0	$-\frac{e_y}{x}$

表 1.2: 物理量 Q 、流束 F_x 、 F_z 、ソース項 S

この一連の式変形を、式 (1.17) ～ (1.22) に対して同様に行い、各成分ごとに整理すると、表 1.2 のようになる。

ただし、

$$e = \rho\varepsilon + \frac{1}{2}\rho v^2, \quad \rho\varepsilon = \frac{p}{\gamma - 1} \quad (1.25)$$

$$\begin{aligned} E &= \rho\varepsilon + \frac{1}{2}\rho v^2 + \frac{B^2}{8\pi} \\ &= \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2 + \frac{B^2}{8\pi} \end{aligned} \quad (1.26)$$

$$\begin{aligned} e_p &= e + p = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2 + p \\ &= \frac{\gamma p}{\gamma - 1} + \frac{1}{2}\rho v^2 \end{aligned} \quad (1.27)$$

$$e_y = -v_z B_x + v_x B_z \quad (1.28)$$

である。

そこで、この 2 次元スカラー移流方程式 (1.24) を差分法（ここでは 2 段階修正 Lax-Wendroff スキーム）により具体的に解くことを考える。

2 次元スカラー移流方程式を差分法により解くとは、2 次元スカラー移流方程式の微分 $\partial Q/\partial r$ 、 $\partial F_x/\partial x$ 、 $\partial F_z/\partial z$ を差分 $\Delta Q/\Delta x$ 、 $\Delta F_x/\Delta x$ 、 $\Delta F_z/\Delta z$ に置き換えることによって、時間 Δt 後の Q の値を求めることである。ここで Δ は格子間の差をとる演算子である。

したがって式 (1.24) を、

$$\frac{\Delta Q}{\Delta t} + \frac{\Delta F_x}{\Delta x} + \frac{\Delta F_z}{\Delta z} = S \quad (1.29)$$

として扱う。n を時間、i, j それぞれ X 軸のグリッド (格子) 番号、Z 軸のグリッド番号として物理量 Q を $Q_{i,j}^n$ のように表して、以下 $Q_{i,j}^n$ から $Q_{i,j}^{n+1}$ を求める方法を考える。

Lax-Wendroff スキームはテイラー展開にもとづく差分法であり、1 次元スカラー移流方程式の場合以下のようにして導かれる。

$$Q_i^{n+1} = Q_i^n + \Delta t \frac{\partial F}{\partial t} + \frac{1}{2} \Delta t^2 \frac{\partial^2 F}{\partial t^2} + O(\Delta t^3) \quad (1.30)$$

右辺第 2 項、第 3 項に

$$\frac{\partial Q}{\partial t} = -\frac{\partial F}{\partial t} \quad (1.31)$$

$$\frac{\partial^2 Q}{\partial t^2} = \frac{\partial^2 F}{\partial t^2} \quad (1.32)$$

を代入すると、

$$Q_i^{n+1} = Q_i^n - \Delta t \frac{\partial F}{\partial x} + \frac{1}{2} \Delta t^2 \frac{\partial^2 F}{\partial x^2} + O(\Delta t^3) \quad (1.33)$$

空間微分 $\partial Q / \partial x$ 、 $\partial^2 Q / \partial x^2$ をそれぞれ中心差分で近似すると、

$$Q_i^{n+1} = Q_i^n - \frac{1}{2} \Delta t \frac{F_i^n - F_{i-1}^n}{\Delta x} + \frac{1}{2} \left(\frac{\Delta t}{\Delta x} \right)^2 (F_{i+1}^n - 2F_i^n + F_{i-1}^n) \quad (1.34)$$

これが Lax-Wendroff のスキームである。以上の導出からわかるように、Lax-Wendroff スキームは空間、時間についていずれも 2 次精度の解法になっている。1 次元スカラー移流方程式の場合、Lax-Wendroff スキームは 以下のように 2 段階に分けたスキームと同等である。この方法を 2 段階 Lax-Wendroff スキームと呼ぶ。

$$Q_{i+1/2}^{n+1/2} = \frac{Q_{i+1}^n + Q_i^n}{2} - \frac{1}{2} \frac{\Delta t}{\Delta x} (F_{i+1}^n - F_i^n) \quad (1.35)$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2}) \quad (1.36)$$

この 2 段階 Lax-Wendroff スキームをさらに改良したものが CANS で用いられている 2 段階修正 Lax-Wendroff スキームである。簡単のため、1 次元スカラー移流方程式 を解く場合の 2 段階修正 Lax-Wendroff スキームについて述べる。

このスキームではまず第 0 段階として、グリッド上の物理量 Q_i^n を用いてセル (グリッド間の境界のこと) 上の物理量を求め、グリッド上の流束 F_i^n, F_{i-1}^n (表 1.2 参照) を用いて Δt だけ時間を進めた物理量 $Q_{i+1/2}^{n+1}$ を求める (図 1.3 参照)。

第 0 段階 セル上の物理量作成:

$$\tilde{Q}_{i+1/2}^{n+1} = Q_{i+1/2}^n - \Delta t \frac{F_{i+1}^n - F_i^n}{\Delta x} + \Delta t \cdot S_{i+1/2}^n \quad (1.37)$$

すなわち、

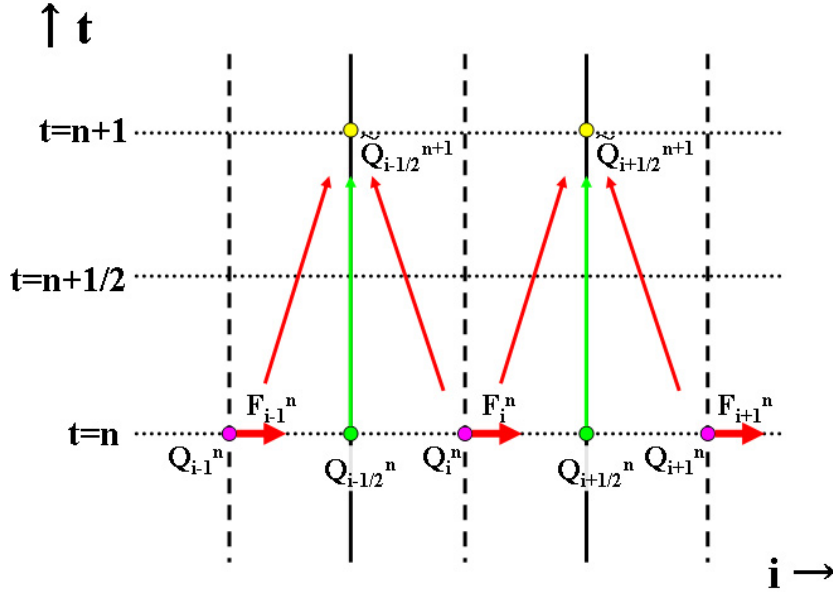


図 1.3: 1 次元第 0 段階のイメージ

$$\tilde{Q}_{i+1/2}^{n+1} = \frac{Q_{i+1}^n + Q_i^n}{2} - \Delta t \frac{F_{i+1}^n - F_i^n}{\Delta x} + \Delta t \cdot \frac{S_{i+1}^n + S_i^n}{2} \quad (1.38)$$

ここでセル上の物理量は $\tilde{}$ をつけてあらわした。

次に、第 1 段階として、左右グリッド上の流束 F_{i-1}^n, F_{i+1}^n (表 1.2 参照) の中心差分から、時間を $\Delta t/2$ だけ進めた物理量 $Q_i^{n+1/2}$ を求める (図 1.4 参照)。

第 1 段階 流束中心差分:

$$Q_i^{n+1/2} = Q_i^n - \frac{\Delta t}{2} \frac{F_{i+1}^n - F_{i-1}^n}{2\Delta x} + \frac{\Delta t}{2} \cdot S_i^n \quad (1.39)$$

最後に、第 2 段階として、セル上の流束 $\tilde{F}_{i+1/2}^{n+1}, \tilde{F}_{i-1/2}^{n+1}$ (これらはそれぞれ第 0 段階で求めた $\tilde{Q}$ を用いて表 1.2 をもとに作成する) の差分を物理量 $Q_i^{n+1/2}$ に加え、 Δt 時間を進めた物理量 Q_i^{n+1} を求める (図 1.5 参照)。

第 2 段階 セル上流束中心差分:

$$Q_i^{n+1} = Q_i^{n+1/2} - \frac{\Delta t}{2} \frac{\tilde{F}_{i+1/2}^{n+1} - \tilde{F}_{i-1/2}^{n+1}}{\Delta x} + \frac{\Delta t}{2} \cdot \tilde{S}_i^{n+1} \quad (1.40)$$

すなわち、

$$Q_i^{n+1} = Q_i^{n+1/2} - \frac{\Delta t}{2} \frac{\tilde{F}_{i+1/2}^{n+1} - \tilde{F}_{i-1/2}^{n+1}}{\Delta x} + \frac{\Delta t}{2} \cdot \frac{\tilde{S}_{i+1/2}^{n+1} + \tilde{S}_{i-1/2}^{n+1}}{2} \quad (1.41)$$

同様に 2 次元に拡張した式は以下のようにあらわされる (図 1.6 図 1.7 図 1.8 参照)。

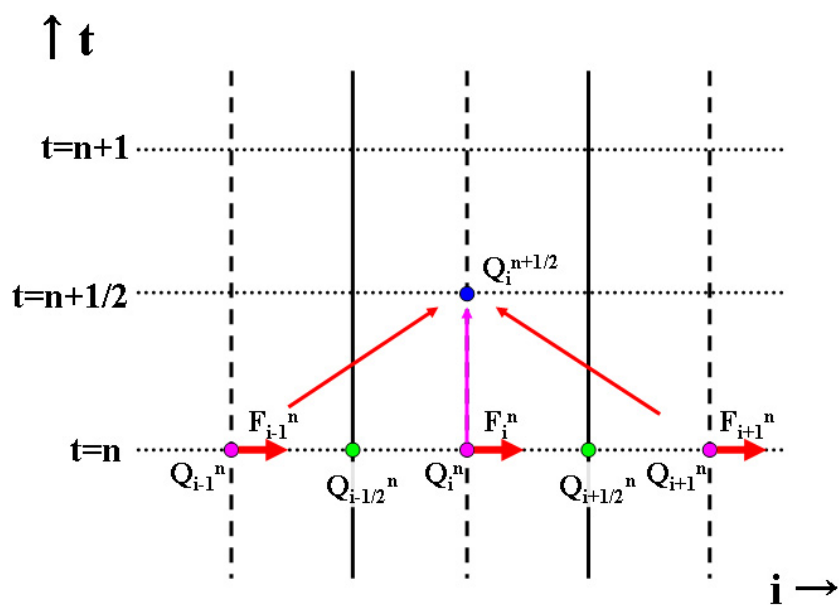


図 1.4: 1次元第1段階のイメージ

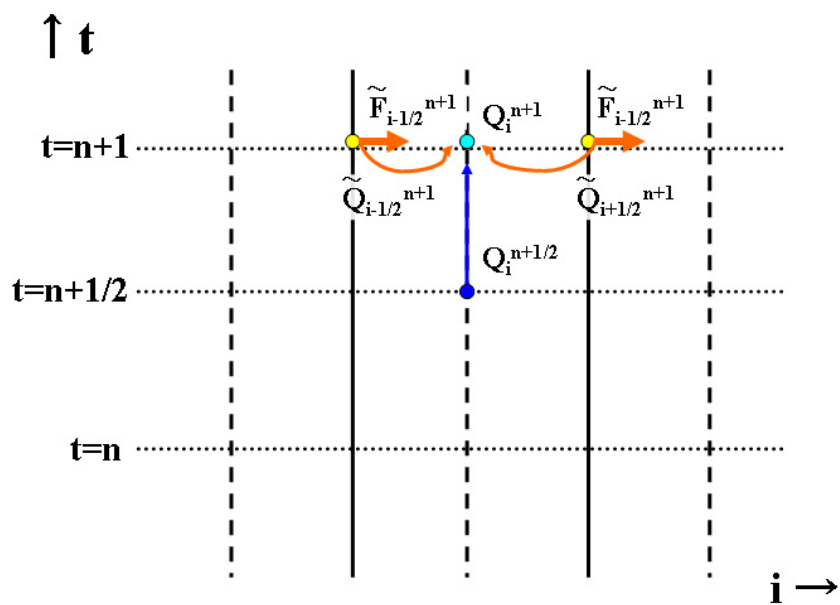


図 1.5: 1次元第2段階のイメージ

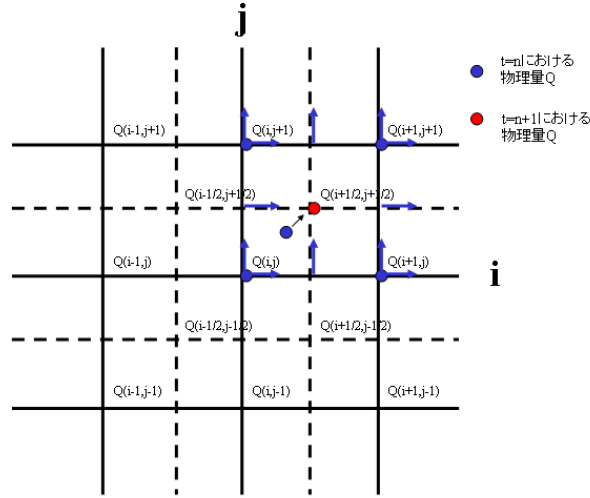


図 1.6: 2次元第0段階のイメージ

第0段階 セル上の物理量作成:

$$\begin{aligned} \tilde{Q}_{i+1/2,j+1/2}^{n+1} &= Q_{i+1/2,j+1/2}^n - \Delta t \frac{F_{i+1,j+1/2}^n - F_{i,j+1/2}^n}{\Delta x} - \Delta t \frac{F_{i+1/2,j+1}^n - F_{i+1/2,j}^n}{\Delta z} \\ &\quad + \Delta t \cdot S_{i+1/2,j+1/2}^n \end{aligned} \quad (1.42)$$

すなわち、

$$\begin{aligned} \tilde{Q}_{i+1/2,j+1/2}^{n+1} &= \frac{1}{2} \left(\frac{Q_{i+1,j+1}^n + Q_{i+1,j}^n}{2} + \frac{Q_{i,j+1}^n + Q_{i,j}^n}{2} \right) \\ &\quad - \Delta t \frac{1}{\Delta x} \left(\frac{F_{i+1,j+1}^n + F_{i+1,j}^n}{2} - \frac{F_{i,j+1}^n + F_{i,j}^n}{2} \right) \\ &\quad - \Delta t \frac{1}{\Delta z} \left(\frac{F_{i+1,j+1}^n + F_{i,j+1}^n}{2} - \frac{F_{i+1,j}^n + F_{i,j}^n}{2} \right) \\ &\quad + \Delta t \cdot \frac{1}{2} \left(\frac{S_{i,j+1}^n + S_{i,j}^n}{2} + \frac{S_{i+1,j+1}^n + S_{i+1,j}^n}{2} \right) \end{aligned} \quad (1.43)$$

第1段階 流束中心差分:

$$\begin{aligned} Q_{i,j}^{n+1/2} &= Q_{i,j}^n - \frac{\Delta t}{2} \frac{F_{i+1,j}^n - F_{i-1,j}^n}{2\Delta x} - \frac{\Delta t}{2} \frac{F_{i,j+1}^n - F_{i,j-1}^n}{2\Delta z} \\ &\quad + \frac{\Delta t}{2} \cdot S_{i,j}^n \end{aligned} \quad (1.44)$$

第2段階 セル上流束中心差分:

$$Q_{i,j}^{n+1} = Q_{i,j}^{n+1/2} - \frac{\Delta t}{2} \frac{\tilde{F}_{i+1/2,j}^{n+1} - \tilde{F}_{i-1/2,j}^{n+1}}{\Delta x} - \frac{\Delta t}{2} \frac{\tilde{F}_{i,j+1/2}^{n+1} - \tilde{F}_{i,j-1/2}^{n+1}}{\Delta z}$$

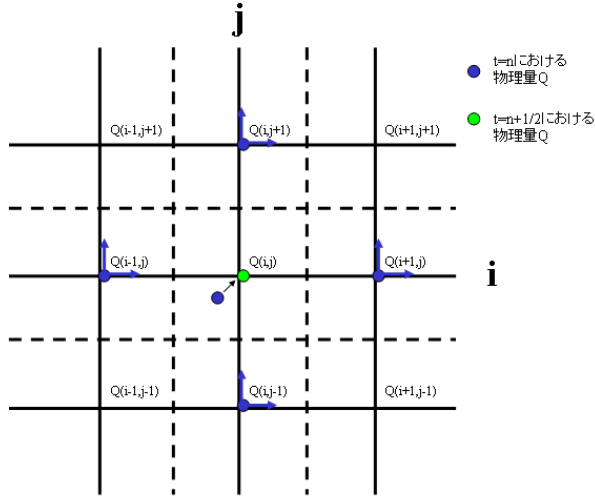


図 1.7: 2次元第1段階のイメージ

$$+\frac{\Delta t}{2} \cdot \tilde{S}_{i,j}^{n+1} \quad (1.45)$$

すなわち、

$$\begin{aligned} Q_{i,j}^{n+1} = & Q_{i,j}^{n+1/2} \\ & -\frac{\Delta t}{2} \frac{1}{\Delta x} \left(\frac{\tilde{F}_{i+1/2,j+1/2}^{n+1} + \tilde{F}_{i+1/2,j-1/2}^{n+1}}{\Delta 2} - \frac{\tilde{F}_{i-1/2,j+1/2}^{n+1} + \tilde{F}_{i-1/2,j-1/2}^{n+1}}{\Delta 2} \right) \\ & -\frac{\Delta t}{2} \frac{1}{\Delta z} \left(\frac{\tilde{F}_{i+1/2,j+1/2}^{n+1} + \tilde{F}_{i-1/2,j+1/2}^{n+1}}{\Delta 2} - \frac{\tilde{F}_{i+1/2,j-1/2}^{n+1} + \tilde{F}_{i-1/2,j-1/2}^{n+1}}{\Delta 2} \right) \\ & +\frac{\Delta t}{2} \cdot \frac{1}{2} \left(\frac{\tilde{S}_{i+1/2,j+1/2}^{n+1} + \tilde{S}_{i-1/2,j+1/2}^{n+1}}{2} + \frac{\tilde{S}_{i+1/2,j-1/2}^{n+1} + \tilde{S}_{i-1/2,j-1/2}^{n+1}}{2} \right) \end{aligned} \quad (1.46)$$

1.4.5 人工粘性の導入

修正 Lax-Wendroff 法は空間、時間についていずれも 2 次精度の方法であるが、不連続面近傍で数値振動を生じるという欠点を持つ。詳しくは「流体力学の数値計算法 (1994) 東京大学出版会、藤井孝蔵著」を参照してもらいたい。数値振動を回避するために、人工粘性として拡散項が各方程式に導入されている。一般に拡散項は拡散係数を κ として、各式の物理量 Q を用いて $\kappa \nabla^2 Q$ の形で表される。

CANS で用いられている 2 次元計算での人工粘性の拡散係数 (κ_x, κ_z) は、 Q_v をパラメータとして、速度の不連続面で大きな値をとるように

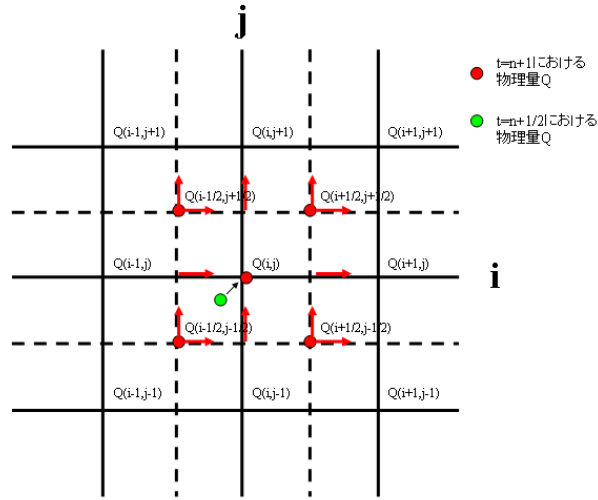


図 1.8: 2次元第2段階のイメージ

$$(\kappa_x, \kappa_z) = Q_v \left(\left| \frac{\partial v_x}{\partial x} \right|, \left| \frac{\partial v_z}{\partial z} \right| \right) \quad (1.47)$$

のようにとられている。

また CANS では基礎方程式に拡散項を導入した形は、各式の物理量 Q を用いて以下のように表される。

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x} F_x + \frac{\partial}{\partial z} F_z = S + \left(\frac{\partial}{\partial x} (\kappa_x \nabla Q) + \frac{\partial}{\partial z} (\kappa_z \nabla Q) \right) \quad (1.48)$$

1.4.6 計算エンジン -mlw_m.c.f

/cans2d/hdmlw にある。2段階修正 Lax-Wendroff スキームのそれぞれの段階を計算するサブルーチン mlwhalf.f、mlwsrch.f、mlwfull.f、mlwsrcf.f および人工粘性を計算するサブルーチン mlwartv.f から構成される。第0段階および第1段階は mlwhalf.f と mlwsrch.f において、第2段階は mlwfull.f と mlwsrcf.f において計算される。前節までは2段階修正 Lax-Wendroff スキームの内容に関して述べてきたが、本節では実際に実行されるプログラム mlw_m.c.f について具体的に説明する。

```

c=====|
      subroutine mlw_m_c(ro,pr,vx,vz,bx,bz,ay
      &
      ,dt,qav,gm,x,xm,dx,dxm,ix,dz,dzm,jx)
!ro は密度、 pr は圧力、 vx は X 方向の速度、 vz は Z 方向の速度
!bx は X 方向の磁場、 bz は Z 方向の磁場、 ay はベクトルポテンシャル
c=====|
      implicit double precision (a-h,o-z)
      dimension dx(ix),dxm(ix)
      dimension dxi(ix),dxim(ix)
      dimension ux0(ix),ux1(ix)
      dimension x(ix),xm(ix)
      dimension dz(jx),dzm(jx)
      dimension dzi(jx),dzim(jx)
      dimension uz0(jx),uz1(jx)
      dimension ro(ix,jx),pr(ix,jx),vx(ix,jx),vz(ix,jx)
      &
      ,bx(ix,jx),bz(ix,jx)
      dimension ey(ix,jx)
      dimension ay(ix,jx)
      dimension ee(ix,jx),rx(ix,jx),rz(ix,jx)
      dimension roh(ix,jx),eeh(ix,jx),rxh(ix,jx),rzh(ix,jx)
      &
      ,bxh(ix,jx),bzh(ix,jx)
      dimension eyh(ix,jx)
      dimension ayh(ix,jx)
      dimension prh(ix,jx),vxh(ix,jx),vzh(ix,jx)
      dimension dro(ix,jx),dee(ix,jx),drx(ix,jx),drz(ix,jx)
      &
      ,dbx(ix,jx),dbz(ix,jx)
      &
      ,day(ix,jx)
      dimension fx(ix,jx),qx(ix,jx)
      dimension fz(ix,jx),qz(ix,jx)
      dimension ss(ix,jx)

c-----|
      pi = acos(-1.0d0)
      pi8i=1./pi/8.
      pi4i=1./pi/4.

c-----|
c      ready
c-----|

      do i=1,ix
         dxi(i) = 1.0/dx(i)
         dxim(i) = 1.0/dxm(i)
      enddo
!dx、dxmの逆数を定義
!dx、dxmはgrdrdy.fで作成された

      do i=2,ix-1
         ux1(i) = 0.5*dxm(i-1)/dx(i)
         ux0(i) = 0.5*dxm(i)/dx(i)
      enddo
!ux1(i)、ux0(i)はmlwfull.f、mlwsrct.fで使われる

      do j=1,jx
         dzi(j) = 1.0/dz(j)
         dzim(j) = 1.0/dzm(j)
      enddo
      do j=2,jx-1
         uz1(j) = 0.5*dzm(j-1)/dz(j)
         uz0(j) = 0.5*dzm(j)/dz(j)
      enddo
!X方向で行ったことと同様

```

```

c-----|
c   initialize dro etc.
c-----|

do j=1,jx
do i=1,ix
    dro(i,j)= 0.0
    dee(i,j)= 0.0
    drx(i,j)= 0.0
    drz(i,j)= 0.0
    dbx(i,j)= 0.0
    dbz(i,j)= 0.0
    day(i,j)= 0.0
enddo
enddo
!dro などは時刻 n から n+1 までに ro などが変わったかという変化量
!dro などのこれから計算される変化量はいつもここで 0 にしておく。

c-----|
c   calculate energy from pressure
c-----|

do j=1,jx
do i=1,ix
    vv=vx(i,j)**2+vz(i,j)**2
    bb=bx(i,j)**2+bz(i,j)**2
    ee(i,j) = pr(i,j)/(gm-1)+0.5*ro(i,j)*vv+pi8i*bb
    rx(i,j) = ro(i,j)*vx(i,j)
    rz(i,j) = ro(i,j)*vz(i,j)
enddo
enddo
!ro、pr、vx、vz、bx、bz から計算に必要な物理量を作成している
!ee は全エネルギー、エネルギー保存における物理量 Q に相当 表 1.2 参照
!rx、rz は運動量、運動量保存における物理量 Q に相当 表 1.2 参照
!rx、rz は運動量保存における物理量 Q に相当する一方で、質量保存における流束 Fx、Fz にも相当している (表
1.2 参照) が、下で実際に計算される質量保存では rx、rz は用いずに改めて ro*vx と ro*vz を作成している

do j=1,jx
do i=1,ix
    ey(i,j) = -vz(i,j)*bx(i,j)+vx(i,j)*bz(i,j)
enddo
enddo
!ey はエネルギー保存で流束の作成に使われる量であり、磁場の誘導方程式で流束として用いられる量でもある 表
1.2 参照

c-----|
c   step intermediate results for flux calculation
c-----|
!ここから第0段階および第1段階の計算が行われる
c--- density ---
do j=1,jx
do i=1,ix
    fx(i,j)= ro(i,j)*vx(i,j)
    fz(i,j)= ro(i,j)*vz(i,j)
    ss(i,j)= -fx(i,j)/x(i)
enddo
enddo
!fx は流束 Fx、fz は流束 Fz、ss はソース項を表す
!質量保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

call mlwhalf(ro ,roh ,dro,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
call mlwsrch(roh ,dro ,dt,ss,ix,jx)

```

! 物理量 ro に対して第0段階および第1段階すなわち式(1.43)、(1.44)に相当する計算が行われている
 ! 第0段階で dt だけ時間更新されたセル上の物理量を roh としていることに注意
 ! 第1段階で dt/2 だけ時間更新するための「物理量の時間的差分」が dro になっていることに注意

```
c--- energy ---
do j=1,jx
do i=1,ix
  vv=vx(i,j)**2+vz(i,j)**2
! vv は ep の作成のときに用いられる
  ep = pr(i,j)*gm/(gm-1.)+0.5*ro(i,j)*vv
! ep はエネルギー保存の流束 Fx、Fz の作成のときに用いられる 表 1.2 参照
  fx(i,j)= ep*vx(i,j) +(bz(i,j)*ey(i,j))*pi4i
  fz(i,j)= ep*vz(i,j) +(-bx(i,j)*ey(i,j))*pi4i
  ss(i,j)= -fx(i,j)/x(i)
! ss はソース項 表 1.2 と対応している
enddo
enddo
! エネルギー保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照
```

```
call mlwhalf(ee ,eeh ,dee ,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
call mlwsrch(eeh ,dee ,dt,ss,ix,jx)
! 第0段階で dt だけ時間更新されたセル上の物理量を eeh としていることに注意
! 第1段階で dt/2 だけ時間更新するための「物理量の時間的差分」が dee になっていることに注意
```

```
c--- x-momentum ---
do j=1,jx
do i=1,ix
  fx(i,j)= ro(i,j)*vx(i,j)**2+pr(i,j)
&
  +pi8i*(bz(i,j)**2-bx(i,j)**2)
  fz(i,j)= ro(i,j)*vx(i,j)*vz(i,j)-pi4i*bx(i,j)*bz(i,j)
  ss(i,j)= -(ro(i,j)*(vx(i,j)**2)
&
  +pi4i*(-bx(i,j)**2))/x(i)
enddo
enddo
! X 方向の運動量保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照
```

```
call mlwhalf(rx,rhx,drx,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
call mlwsrch(rhx ,drx ,dt,ss,ix,jx)
! 第0段階で dt だけ時間更新されたセル上の物理量を rxh としていることに注意
! 第1段階で dt/2 だけ時間更新するための「物理量の時間的差分」が drx になっていることに注意
```

```
c--- z-momentum ---
do j=1,jx
do i=1,ix
  fx(i,j)= ro(i,j)*vz(i,j)*vx(i,j)-pi4i*bx(i,j)*bx(i,j)
  fz(i,j)= ro(i,j)*vz(i,j)**2+pr(i,j)
&
  +pi8i*(bx(i,j)**2-bz(i,j)**2)
  ss(i,j)= -fx(i,j)/x(i)
enddo
enddo
! Z 方向の運動量保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照
```

```
call mlwhalf(rz,rzh,drz,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
call mlwsrch(rzh ,drz ,dt,ss,ix,jx)
! 第0段階で dt だけ時間更新されたセル上の物理量を rzh としていることに注意
! 第1段階で dt/2 だけ時間更新するための「物理量の時間的差分」が drz になっ
ていることに注意
```

```

c--- x-magnetic ---
do j=1,jx
do i=1,ix
    fx(i,j)= 0.
    fz(i,j)=-ey(i,j)
enddo
enddo
! X 方向の磁場の誘導方程式における流束 Fx、Fz の作成 表 1.2 参照
! ソース項は 0 なので作成していない

    call mlwhalf(bx,bxh,dbx,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
! 第 0 段階で dt だけ時間更新されたセル上の物理量を bxh としていることに注意
! 第 1 段階で dt/2 だけ時間更新するための「物理量の時間的差分」が dbx になっていることに注意
! ソース項は 0 なので mlwsrch.f は用いない

c--- z-magnetic ---
do j=1,jx
do i=1,ix
    fx(i,j)= ey(i,j)
    fz(i,j)= 0.
    ss(i,j)= -fx(i,j)/x(i)
enddo
enddo
! Z 方向の磁場の誘導方程式における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

    call mlwhalf(bz,bzh,dbz,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
    call mlwsrch(bzh ,dbz ,dt,ss,ix,jx)
! 第 0 段階で dt だけ時間更新されたセル上の物理量を bzh としていることに注意
! 第 1 段階で dt/2 だけ時間更新するための「物理量の時間的差分」が dbz になっていることに注意

c--- y-magnetic potential ---
do j=1,jx
do i=1,ix
    ss(i,j)= -ey(i,j)
enddo
enddo
    call mlwsrch(ayh ,day ,dt,ss,ix,jx)
! ベクトルポテンシャルの計算
! ss(i,j)= -ey(i,j) はベクトルポテンシャルの式および磁場の誘導方程式から容易に導かれる

c-----|
c    convert from total energy to pressure
c-----|

do j=1,jx-1
do i=1,ix-1
    vxh(i,j)  = rxh(i,j)/roh(i,j)
    vzh(i,j)  = rzh(i,j)/roh(i,j)
    vv=vxh(i,j)**2+vzh(i,j)**2
    bb=bxh(i,j)**2+bzh(i,j)**2
    prh(i,j)  = (gm-1)*(eeh(i,j)-0.5*roh(i,j)*vv-pi8i*bb)
enddo
enddo    call mlwhalf(rz,rzh,drz,dt,fx,dxi,dxim,ix,fz,dzi,dzim,jx)
    call mlwsrch(rzh ,drz ,dt,ss,ix,jx)
! 第 0 段階で dt だけ時間更新されたセル上の物理量を rzh としていることに注意
! 第 1 段階で dt/2 だけ時間更新するための「物理量の時間的差分」が drz になっていることに注意

```

! 第 0 段階で dt だけ時間更新されたセル上の物理量 roh, eeh, rxh, rzh, bxh, bzh を用いて、作成されていない vxh や vzh および prh などの物理量を作成している

! この作業により dt だけ時間更新されたセル上の流束が計算可能になる

```
do j=1,jx
do i=1,ix
  eyh(i,j)=-vzh(i,j)*bxh(i,j)+vxh(i,j)*bzh(i,j)
enddo
enddo
```

! eyh は第 1 段階における ey に相当

! エネルギー保存で流束の作成に使われる量であり、磁場の誘導方程式で流束として用いられる量でもある 表 1.2 参照

```
c-----|
c   step intermediate results for full step
c-----|
```

! ここから第 2 段階の計算が行われる

```
c--- density ---
do j=1,jx-1
do i=1,ix-1
  fx(i,j)= roh(i,j)*vxh(i,j)
  fz(i,j)= roh(i,j)*vzh(i,j)
  ss(i,j)= -fx(i,j)/xm(i)
enddo
enddo
```

! fx は流束 Fx、fz は流束 Fz、ss はソース項を表す

! 質量保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

```
call mlwfull(dro ,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)
call mlwsrccf(dro,dt,ss,ux0,ux1,ix,uz0,uz1,jx)
```

! 物理量 roh に対して第 2 段階すなわち式 (1.46) に相当する計算が行われている

! 第 1 段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が dro になっていることに注意

! すなわちこの dro を加えると ro が時間 n から dt だけ更新され n+1 になったことになる

```
c--- energy ---
do j=1,jx-1
do i=1,ix-1
  vv=vxh(i,j)**2+vzh(i,j)**2
```

! vv は ep の作成のときに用いられる

```
  ep = prh(i,j)*gm/(gm-1.)+0.5*roh(i,j)*vv
```

! ep はエネルギー保存の流束 Fx、Fz の作成のときに用いられる 表 1.2 参照

```
  fx(i,j)= ep*vxh(i,j)
&      +(bzh(i,j)*eyh(i,j))*pi4i
  fz(i,j)= ep*vzh(i,j)
&      +(-bxh(i,j)*eyh(i,j))*pi4i
  ss(i,j)= -fx(i,j)/xm(i)
```

! ss はソース項 表 1.2 と対応している

```
enddo
enddo
```

! fx は流束 Fx、fz は流束 Fz、ss はソース項を表す

! エネルギー保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

```
call mlwfull(dee ,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)
call mlwsrccf(dee,dt,ss,ux0,ux1,ix,uz0,uz1,jx)
```

! 物理量 eeh に対して第 2 段階すなわち式 (1.46) に相当する計算が行われている

! 第 1 段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が dee になっていることに注意

! すなわちこの dee を加えると ee が時間 n から dt だけ更新され n+1 になったことになる

```

c--- x-momentum ---
do j=1,jx-1
do i=1,ix-1
    fx(i,j)= roh(i,j)*vxh(i,j)**2+prh(i,j)
    &
    +pi8i*(bzh(i,j)**2-bxh(i,j)**2)
    fz(i,j)= roh(i,j)*vxh(i,j)*vzh(i,j)-pi4i*bxh(i,j)*bzh(i,j)
    ss(i,j)= -(roh(i,j)*(vxh(i,j)**2)
    &
    +pi4i*(-bxh(i,j)**2))/xm(i)
enddo
enddo

```

! X 方向の運動量保存における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

```

call mlwfull(drx,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)
call mlwsrcf(drx,dt,ss,ux0,ux1,ix,uz0,uz1,jx)

```

! 物理量 rxh に対して第 2 段階すなわち式 (1.46) に相当する計算が行われている

! 第 1 段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が drx になっていることに注意

! すなわちこの drx を加えると rx が時間 n から dt だけ更新され n+1 になったことになる

```

c--- z-momentum ---
do j=1,jx-1
do i=1,ix-1
    fx(i,j)= roh(i,j)*vzh(i,j)*vxh(i,j)-pi4i*bzh(i,j)*bxh(i,j)
    fz(i,j)= roh(i,j)*vzh(i,j)**2+prh(i,j)
    &
    +pi8i*(bxh(i,j)**2-bzh(i,j)**2)
    ss(i,j)= -fx(i,j)/xm(i)
enddo
enddo

```

! Z 方向の磁場の誘導方程式における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

```

call mlwfull(drz,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)
call mlwsrcf(drz,dt,ss,ux0,ux1,ix,uz0,uz1,jx)

```

! 物理量 rzxh に対して第 2 段階すなわち式 (1.46) に相当する計算が行われている

! 第 1 段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が drz になっていることに注意

! すなわちこの drz を加えると rz が時間 n から dt だけ更新され n+1 になったことになる

```

c--- x-magnetic ---
do j=1,jx-1
do i=1,ix-1
    fx(i,j)= 0.
    fz(i,j)= -eyh(i,j)
enddo
enddo

```

! X 方向の磁場の誘導方程式における流束 Fx、Fz の作成 表 1.2 参照

! ソース項は 0 なので作成していない

```

call mlwfull(dbx,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)

```

! 物理量 bxh に対して第 2 段階すなわち式 (1.46) に相当する計算が行われている

! 第 1 段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が dbx になっていることに注意

! すなわちこの dbx を加えると bx が時間 n から dt だけ更新され n+1 になったことになる

```

c--- z-magnetic ---
do j=1,jx-1
do i=1,ix-1
    fx(i,j)= eyh(i,j)
    fz(i,j)= 0.
    ss(i,j)= -fx(i,j)/x(i)
enddo
enddo

```

! Z 方向の磁場の誘導方程式における流束 Fx、Fz およびソース項 S の作成 表 1.2 参照

```

      call mlwfull(dbz,dt,fx,dxi,ux0,ux1,ix,fz,dzi,uz0,uz1,jx)
      call mlwsrctf(dbz,dt,ss,ux0,ux1,ix,uz0,uz1,jx)
! 物理量 bzh に対して第2段階すなわち式(1.46)に相当する計算が行われている
! 第1段階で dt/2 だけ更新された時間からさらに dt/2 だけ更新するための「物理量の時間的差分」が dbz になっていることに注意
! すなわちこの dbz を加えると bz が時間 n から dt だけ更新され n+1 になったことになる

c--- y-magnetic potential ---
      do j=1,jx
      do i=1,ix
          ss(i,j)= -eyh(i,j)
      enddo
      enddo
      call mlwsrctf(day,dt,ss,ux0,ux1,ix,uz0,uz1,jx)
! ベクトルポテンシャルの計算
! ss(i,j)= -eyh(i,j) はベクトルポテンシャルの式および磁場の誘導方程式から容易に導かれる

c-----|
c      diffusion coefficients for artificial viscosity
c-----|
c      qav=3.0
      zero=0.0
      do j=1,jx-1
      do i=1,ix-1
          qx(i,j)=qav*dxm(i)*max(zero,abs(vx(i+1,j)-vx(i,j)))-1.0e-4)
      enddo
      enddo
      do j=1,jx-1
      do i=1,ix
          qz(i,j)=qav*dzm(j)*max(zero,abs(vz(i,j+1)-vz(i,j)))-1.0e-4)
      enddo
      enddo
! ここでは人工粘性を計算するための拡散係数について決めている
! 「1.4.5 人工粘性の導入」および式(1.47) 参照

c-----|
c      apply artificial viscosity
c-----|
      call mlwartv(ro,dro,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
      call mlwartv(ee,dee,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
      call mlwartv(rx,drx,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
      call mlwartv(rz,drz,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
      call mlwartv(bx,dbx,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
      call mlwartv(bz,dbz,dt,qx,dxi,dxim,ix,qz,dzi,dzim,jx)
! サブルーチン mlwartv.f によって人工粘性の値が計算され、dro などの各物理量の時間的な差分に加えられている
! 「1.4.5 人工粘性の導入」および式(1.48) 参照

c-----|
c      update internal points
c-----|
      do j=2,jx-1
      do i=2,ix-1
          ro(i,j) = ro(i,j) +dro(i,j)
          ee(i,j) = ee(i,j) +dee(i,j)
          rx(i,j) = rx(i,j) +drx(i,j)
          rz(i,j) = rz(i,j) +drz(i,j)
          bx(i,j) = bx(i,j) +dbx(i,j)
          bz(i,j) = bz(i,j) +dbz(i,j)
          ay(i,j) = ay(i,j) +day(i,j)
      enddo
      enddo
! 第0段階～第2段階で計算された各物理量の時間的な差分を、もとの物理量(時間 t=n)に加え、時間を dt 更新する

```

```

c-----|
c   convert from total energy to pressure
c-----|

do j=2,jx-1
do i=2,ix-1
  vx(i,j) = rx(i,j)/ro(i,j)
  vz(i,j) = rz(i,j)/ro(i,j)
  vv=vx(i,j)**2+vz(i,j)**2
  bb=bx(i,j)**2+bz(i,j)**2
  pr(i,j) = (gm-1)*(ee(i,j) - 0.5*ro(i,j)*vv - pi8i*bb)
enddo
enddo
! 最後に時間 dt 更新されたグリッド上の物理量 ro、ee、rx、rz、bx、bz を用いて、作成されていない vx や vz
! および pr などの物理量を作成している
! この作業により dt だけ時間更新されたグリッド上の必要な物理量はすべてそろえることになる。

return
end

```

1.5 謝辞

今回の執筆にあたって、この機会を与えて下さった千葉大学 花輪知幸教授に感謝する。指導教官である国立天文台 富阪幸治教授には、天体とスペースプラズマのシミュレーションサマースクールに参加することを勧めてくださり、また執筆にあたっての準備等、面倒をみて頂いたことに心からお礼を言いたい。また、CANSを開発した千葉大学 松元亮治教授、東京大学 横山央明助教授には執筆全般に関して貴重な助言を頂いた。法政大学 松本倫明助教授、国立天文台 西合一矢博士、千葉大学 町田正博博士には計算手法に関して貴重な助言をいただいた。その他、校正等様々な点で協力をして頂いた共著者や千葉大学宇宙物理学研究室の人々には心より感謝の気持ちを述べたい。

参考文献

- [1] 宇宙流体力学 (1997) 培風館、坂下志郎、池内了著
- [2] 流体力学の数値計算法 (1994) 東京大学出版会、藤井孝蔵著
- [3] 数値天体物理学サマースクールのテキスト (2000)、富阪幸治、花輪知幸著
- [4] 流体力学 (1972) 培風館、巽友正

第 2 章

磁気流体力学方程式の近似 Riemann 解法

千葉大学 佐藤 裕司

2.1 問題設定

2.1.1 背景

筆者は大質量星のコアが重力崩壊を起こして中性子星が形成する過程の3次元磁気流体力学シミュレーションを行ってきた。中性子星形成のシミュレーションを行うためには、縮退したガスの圧力、超新星爆発に至る衝撃波の発生、それと同時に生成する熱エネルギーを取り扱う必要がある。またパルサー形成を考慮すると磁場と回転を伴うので、磁気流体力学方程式を解く必要がある。特に衝撃波面での数値振動をなくすためには、近似 Riemann 解法が有効である。理想気体の磁気流体力学方程式に対する近似 Riemann 解法の数値流束 (eg. Fukuda & Hanawa 1999) は求められているが、一般のガスに対するものは求められていない。そこで私は、理想気体の場合の磁気流体力学方程式の近似 Riemann 解法の数値流束と、一般のガスの場合の流体力学方程式の近似 Riemann 解法の数値流束 (Nobuta & Hanawa 1999) を組み合わせて、縮退したガスの圧力を考慮した状態方程式に対する磁気流体力学方程式の近似 Riemann 解法の数値流束を新たに求めた。

この章では、その新たに求めた近似 Riemann 解の導出方法を紹介する。

2.1.2 磁気流体力学方程式

磁気流体力学方程式の基礎方程式系は保存系で以下のように表される。

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (2.1)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ B_y \\ B_z \\ \rho E \end{pmatrix} \quad (2.2)$$

$$\mathbf{F} = \begin{pmatrix} \rho v_x^2 + P + \frac{\rho v_x}{B_y^2 + B_z^2 - B_x^2} \\ \rho v_x v_y - \frac{B_x 8\pi}{B_y} \\ \rho v_x v_z - \frac{4\pi}{B_x B_z} \\ v_x B_y - v_y B_x \\ v_x B_z - v_z B_x \\ \rho H v_x - \frac{B_x (B_x v_x + B_y v_y + B_z v_z)}{4\pi} \end{pmatrix} \quad (2.3)$$

$$E = \frac{\mathbf{v}^2}{2} + \varepsilon_{\text{int}} + \frac{\mathbf{B}^2}{8\pi\rho} \quad (2.4)$$

$$H = \frac{\mathbf{v}^2}{2} + \varepsilon_{\text{int}} + \frac{\mathbf{B}^2}{4\pi\rho} + \frac{P}{\rho} \quad (2.5)$$

ただし、 ε_{int} は内部エネルギーである。理想気体の場合は、

$$\varepsilon_{\text{int}} = \frac{P}{(\gamma - 1)\rho} \quad (2.6)$$

である。中性子星形成では、

$$\varepsilon_{\text{int}} = \varepsilon_t + \varepsilon_c \quad (2.7)$$

のように Thermal part と Cold part の和になる。

2.1.3 中性子星形成過程の状態方程式

密度 ρ と熱エネルギー ε_t に比例する P_t と、密度の大きさによって指数が変化する、レプトンの縮退圧に相当する P_c の和で表される状態方程式を採用する。

$$P_{\text{int}}(\rho, \varepsilon_t) = P_c(\rho) + P_t(\rho, \varepsilon_t) \quad (2.8)$$

$$P_t(\rho, \varepsilon_t) = (\gamma_t - 1)\rho\varepsilon_t \quad (2.9)$$

$$P_c(\rho) = \kappa\rho^\gamma \quad (2.10)$$

ただし、 κ と γ は区間ごとに異なる。内部エネルギーは

$$\varepsilon_{\text{int}} = \int_0^\rho \frac{P_c}{\rho^2} d\rho + \varepsilon_t \quad (2.11)$$

で表される。

2.2 近似 Riemann 解の求め方

2.2.1 方針

本節では筆者が近似 Riemann 解を導いた過程の詳細を解説する。Roe 法自体の詳しい解説はここではしない。Roe 法の詳細は原論文 (Roe, 1981) や教科書 (藤井, 1994) を参照して欲しい。

Roe 法による数値流束を求めるための一般的な手順は、

1. 微分方程式を

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = 0 \quad (2.12)$$

の形に書き直し、「速度行列」 $\mathbf{A}$ を求める。

ここで $\mathbf{A}$ はヤコビアンであり、各成分は Roe 平均と呼ぶ平均量で評価する。

2. 速度行列 $\mathbf{A}$ の固有値 λ を求める。3. 速度行列 $\mathbf{A}$ の固有ベクトル $\mathbf{r}$ を求める。4. 固有値と固有ベクトルから振幅 δw を求め、以下の式で表される数値流束を求める。

$$\mathbf{F}_{j+1/2} = \frac{\mathbf{F}_j + \mathbf{F}_{j+1}}{2} - \frac{1}{2} \sum_{i=1}^m |\lambda_i| \delta w_i \mathbf{r}_i \quad (2.13)$$

である。一般の状態方程式の場合も以上の手続きによって数値流束を求めることができるが、状態方程式が複雑になると速度行列 $\mathbf{A}$ を求める事が大変になる。数値流束を求めるためには、固有値・固有ベクトル・振幅がわかれば良いので、行列 $\mathbf{A}$ の具体的な成分はわからなくても良い。従って今回は $\mathbf{A}$ を求める事はせずに、 $\mathbf{F}$ から直接7つの固有ベクトルを分離して求め、数値流束を構成する方針を取った。

2.2.2 座標変換

固有ベクトルを分離するのに見通しを良くするための作業変数として、保存量で表されるベクトル空間 $\mathbf{U}$ から、物理量で表されるベクトル空間 $\mathbf{U}'$ に移る。この際に軸の方向としてデカルト座標ではなく、磁場の方向に垂直な方向（下付き添字 $\perp$ で表す）と平行な方向（下付き添字 $\parallel$ で表す）を軸に選ぶ。後述するが、それによって横波と縦波、すなわち Alfvén 波とその他の波を分離する事が容易になる。

2つの座標系は行列 $\mathbf{B}$ で変換される。

$$\Delta \mathbf{U} = \mathbf{B} \Delta \mathbf{U}' \quad (2.14)$$

ただし、

$$\Delta \mathbf{U} = \begin{pmatrix} \Delta \rho \\ \Delta(\rho v_x) \\ \Delta(\rho v_y) \\ \Delta(\rho v_z) \\ \Delta B_y \\ \Delta B_z \\ \Delta(\rho E) \end{pmatrix} \quad (2.15)$$

$$\Delta \mathbf{U}' = \begin{pmatrix} \Delta \rho \\ \Delta v_x \\ \Delta v_{\parallel} \\ \Delta v_{\perp} \\ \Delta B_{\parallel} \\ \Delta B_{\perp} \\ \Delta \varepsilon_t \end{pmatrix} \quad (2.16)$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{v}_x & \bar{\rho} & 0 & 0 & 0 & 0 & 0 \\ \bar{v}_y & 0 & \bar{\rho}\beta_y & -\bar{\rho}\beta_z & 0 & 0 & 0 \\ \bar{v}_z & 0 & \bar{\rho}\beta_z & \bar{\rho}\beta_y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_y & -\beta_z & 0 \\ 0 & 0 & 0 & 0 & \beta_z & \beta_y & 0 \\ B_{7,1} & \bar{\rho}\bar{v}_x & \bar{\rho}(\beta_y\bar{v}_y + \beta_z\bar{v}_z) & \bar{\rho}(-\beta_z\bar{v}_y + \beta_y\bar{v}_z) & \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} & 0 & \bar{\rho} \end{pmatrix} \quad (2.17)$$

である。\$\Delta\$ がついた変数はその変数の差分を表し、文字の上に ‘-’ がついている変数はその Roe 平均を表す。ここで、

$$B_{7,1} = \bar{\varepsilon}_t + \bar{\varepsilon}_c + \frac{\gamma_t - 2}{\gamma_t - 1} \delta b^2 + \frac{\bar{v}^2}{2} + \frac{\Delta \varepsilon_c}{\Delta \rho} \bar{\rho} \quad (2.18)$$

$$\beta_y = \frac{\bar{B}_y}{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}} \quad (2.19)$$

$$\beta_z = \frac{\bar{B}_z}{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}} \quad (2.20)$$

$$\delta b^2 = \frac{\gamma_t - 1}{\gamma_t - 2} \frac{(B_{y,j+1} - B_{y,j})^2 + (B_{z,j+1} - B_{z,j})^2}{8\pi (\sqrt{\rho_{j+1}} + \sqrt{\rho_j})^2} \quad (2.21)$$

である。上記の式を計算する際に以下の式が必要である。

$$\varepsilon_{\text{int}} = \varepsilon_c + \varepsilon_t \quad (2.22)$$

$$P = P_c + (\gamma_t - 1)\rho\varepsilon_t \quad (2.23)$$

$$\Delta P = \Delta P_c + (\gamma_t - 1)\bar{\rho}\Delta\varepsilon_t + (\gamma_t - 1)\bar{\varepsilon}_t\Delta\rho \quad (2.24)$$

$$= \left[\frac{\Delta P_c}{\Delta\rho} + (\gamma_t - 1)\bar{\varepsilon}_t \right] \Delta\rho + (\gamma_t - 1)\bar{\rho}\Delta\varepsilon_t \quad (2.25)$$

$$\Delta\varepsilon_{\text{int}} = \Delta\varepsilon_c + \Delta\varepsilon_t \quad (2.26)$$

$$= \frac{\Delta\varepsilon_c}{\Delta\rho}\Delta\rho + \Delta\varepsilon_t \quad (2.27)$$

また、Roe 平均は以下のように表される。

$$\bar{\rho} = \sqrt{\rho_{j+1}\rho_j} \quad (2.28)$$

$$\bar{v}_x = \frac{\sqrt{\rho_{j+1}}v_{x,j+1} + \sqrt{\rho_j}v_{x,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.29)$$

$$\bar{v}_y = \frac{\sqrt{\rho_{j+1}}v_{y,j+1} + \sqrt{\rho_j}v_{y,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.30)$$

$$\bar{v}_z = \frac{\sqrt{\rho_{j+1}}v_{z,j+1} + \sqrt{\rho_j}v_{z,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.31)$$

$$\bar{B}_y = \frac{\sqrt{\rho_{j+1}}B_{y,j} + \sqrt{\rho_j}B_{y,j+1}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.32)$$

$$\bar{B}_z = \frac{\sqrt{\rho_{j+1}}B_{z,j} + \sqrt{\rho_j}B_{z,j+1}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.33)$$

$$\bar{\varepsilon}_c = \frac{\sqrt{\rho_{j+1}} \varepsilon_{c,j+1} + \sqrt{\rho_j} \varepsilon_{c,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.34)$$

$$\bar{\varepsilon}_t = \frac{\sqrt{\rho_{j+1}} \varepsilon_{t,j+1} + \sqrt{\rho_j} \varepsilon_{t,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.35)$$

$$\bar{P}_c = \frac{\sqrt{\rho_{j+1}} P_{c,j+1} + \sqrt{\rho_j} P_{c,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.36)$$

$\mathbf{F}$ についても同様に、 $\mathbf{U}'$ で表すと、

$$\Delta \mathbf{F} = \mathbf{C}' \Delta \mathbf{U}' \quad (2.37)$$

$$\mathbf{C}' = \begin{pmatrix} \bar{v}_x & \bar{\rho} & 0 & 0 & 0 & 0 & 0 \\ C'_{2,1} & 2\bar{\rho}\bar{v}_x & 0 & 0 & \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} & 0 & (\gamma_t - 1)\bar{\rho} \\ \bar{v}_x\bar{v}_y & \bar{\rho}\bar{v}_y & \bar{\rho}\bar{v}_x\beta_y & -\bar{\rho}\bar{v}_x\beta_z & -\frac{B_x}{4\pi}\beta_y & \frac{B_x}{4\pi}\beta_z & 0 \\ \bar{v}_x\bar{v}_z & \bar{\rho}\bar{v}_z & \bar{\rho}\bar{v}_x\beta_z & \bar{\rho}\bar{v}_x\beta_y & -\frac{B_x}{4\pi}\beta_z & -\frac{B_x}{4\pi}\beta_y & 0 \\ 0 & \bar{B}_y & -B_x\beta_y & B_x\beta_z & \bar{v}_x\beta_y & -\bar{v}_x\beta_z & 0 \\ 0 & \bar{B}_z & -B_x\beta_z & -B_x\beta_y & \bar{v}_x\beta_z & \bar{v}_x\beta_y & 0 \\ C'_{7,1} & C'_{7,2} & C'_{7,3} & C'_{7,4} & C'_{7,5} & C'_{7,6} & \gamma_t\bar{\rho}\bar{v}_x \end{pmatrix} \quad (2.38)$$

となる。ただし、

$$C'_{2,1} = \bar{v}_x^2 + \frac{\gamma_t - 2}{\gamma_t - 1} \delta b^2 + (\gamma_t - 1)\bar{\varepsilon}_t + \frac{\Delta P_c}{\Delta \rho} \quad (2.39)$$

$$C'_{7,1} = (\bar{\varepsilon}_t + \bar{\varepsilon}_c)\bar{v}_x + 2\bar{v}_x \frac{\gamma_t - 2}{\gamma_t - 1} \delta b^2 + \frac{\bar{v}_x \bar{\mathbf{v}}^2}{2} + \bar{\rho}\bar{v}_x \frac{\Delta \varepsilon_c}{\Delta \rho} \quad (2.40)$$

$$+ \bar{v}_x \left[\frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1)\bar{\varepsilon}_t \right] \quad (2.41)$$

$$C'_{7,2} = \bar{\rho}(\bar{\varepsilon}_c + \bar{\varepsilon}_t) + \bar{P}_c + \frac{\bar{\rho}\bar{\mathbf{v}}^2}{2} + \bar{\rho}\bar{v}^2 + \frac{\bar{B}_y^2 + \bar{B}_z^2}{4\pi} + (\gamma_t - 1)\bar{\rho}\bar{\varepsilon}_t \quad (2.42)$$

$$C'_{7,3} = -\frac{B_x \sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} + \bar{\rho}\bar{v}_x(\beta_y\bar{v}_y + \beta_z\bar{v}_z) \quad (2.43)$$

$$C'_{7,4} = \bar{\rho}\bar{v}_x(-\beta_z\bar{v}_y + \beta_y\bar{v}_z) \quad (2.44)$$

$$C'_{7,5} = \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{2\pi} \bar{v}_x - \frac{B_x}{4\pi}(-\beta_z\bar{v}_y + \beta_y\bar{v}_z) \quad (2.45)$$

$$C'_{7,6} = -\frac{B_x}{4\pi}(-\beta_z\bar{v}_y + \beta_y\bar{v}_z) \quad (2.46)$$

である。

ここで簡単のためにもう一回座標変換する。すべての固有ベクトルの固有値は $\bar{v}_x \pm C$ の形になると予想できるので、速度 $\bar{v}_x$ で移動する座標系に移る。これは、 $\Delta \mathbf{F}$ から $\bar{v}_x \Delta \mathbf{U}'$ を引く事に相当する。そこで新たに行列 $\mathbf{C}$ を次のように定義する。

$$\Delta \mathbf{F} - \bar{v}_x \Delta \mathbf{U}' = \mathbf{C} \Delta \mathbf{U}' \quad (2.47)$$

$$\mathbf{C} = \begin{pmatrix} 0 & \bar{\rho} & 0 & 0 & 0 & 0 & 0 \\ C_{2,1} & \bar{\rho}\bar{v}_x & 0 & 0 & \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} & 0 & (\gamma_t - 1)\bar{\rho} \\ 0 & \bar{\rho}\bar{v}_y & 0 & 0 & -\frac{\bar{B}_x}{4\pi}\beta_y & \frac{B_x}{4\pi}\beta_z & 0 \\ 0 & \bar{\rho}\bar{v}_z & 0 & 0 & -\frac{\bar{B}_x}{4\pi}\beta_z & -\frac{\bar{B}_x}{4\pi}\beta_y & 0 \\ 0 & \bar{B}_y & -B_x\beta_y & B_x\beta_z & 0 & 0 & 0 \\ 0 & \bar{B}_z & -B_x\beta_z & -B_x\beta_y & 0 & 0 & 0 \\ C_{7,1} & C_{7,2} & C_{7,3} & 0 & C_{7,5} & C_{7,6} & (\gamma_t - 1)\bar{\rho}\bar{v}_x \end{pmatrix} \quad (2.48)$$

ただし、

$$C_{2,1} = \frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1)\bar{\varepsilon}_t + \frac{\gamma_t - 2}{\gamma_t - 1}\delta b^2 \quad (2.49)$$

$$C_{7,1} = \bar{v}_x \left[\frac{\gamma_t - 2}{\gamma_t - 1}\delta b^2 + \frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1)\bar{\varepsilon}_t \right] \quad (2.50)$$

$$C_{7,2} = \bar{\rho}\bar{\varepsilon}_c + \bar{P}_c + \frac{\bar{\rho}\bar{v}^2}{2} + \frac{\bar{B}_y^2 + \bar{B}_z^2}{4\pi} + \gamma_t\bar{\rho}\bar{\varepsilon}_t \quad (2.51)$$

$$C_{7,3} = -\frac{B_x\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} \quad (2.52)$$

$$C_{7,5} = \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi}\bar{v}_x - \frac{B_x}{4\pi}(\beta_y\bar{v}_y + \beta_z\bar{v}_z) \quad (2.53)$$

$$C_{7,6} = -\frac{B_x}{4\pi}(-\beta_z\bar{v}_y + \beta_y\bar{v}_z) \quad (2.54)$$

である。

2.2.3 固有値

ここまで計算できると、固有値を計算する事ができる。行列 $\mathbf{B}$ と $\mathbf{C}$ は以下の関係式を満たしている。

$$\Delta \mathbf{U} = \mathbf{B} \Delta \mathbf{U}' \quad (2.55)$$

$$\Delta \mathbf{F} - \bar{v}_x \Delta \mathbf{U}' = \mathbf{C} \Delta \mathbf{U}' \quad (2.56)$$

$$= \mathbf{C} \mathbf{B}^{-1} \Delta \mathbf{U} \quad (2.57)$$

従って、次の行列式を計算すれば固有値が求まる。

$$|\mathbf{C} \mathbf{B}^{-1} - \lambda| = 0 \quad (2.58)$$

両辺に右から $|\mathbf{B}|$ をかけると、次のように書ける。

$$|\mathbf{C} - \lambda \mathbf{B}| = 0 \quad (2.59)$$

7 次行列式の計算法は一意ではないので詳細は省くが、行列の成分を代入して行列式を計算すると、

$$\lambda(\lambda^2 - b_x^2)(\lambda^4 - a_*^2\lambda^2 + b_x^2a^2) = 0 \quad (2.60)$$

となる。従って、固有値は元々の座標系 $\mathbf{U}$ において、次のように求まる。

$$\lambda_0 = \bar{v}_x \quad (2.61)$$

$$\lambda_{A+} = \bar{v}_x + b_x \quad (2.62)$$

$$\lambda_{A-} = \bar{v}_x - b_x \quad (2.63)$$

$$\lambda_{f+} = \bar{v}_x + c_f \quad (2.64)$$

$$\lambda_{f-} = \bar{v}_x - c_f \quad (2.65)$$

$$\lambda_{s+} = \bar{v}_x + c_s \quad (2.66)$$

$$\lambda_{s-} = \bar{v}_x - c_s \quad (2.67)$$

ここで a は音速であり、

$$a^2 = (\gamma_t - 1) \left(\bar{H} - \frac{\bar{v}_x^2 + \bar{v}_y^2 + \bar{v}_z^2}{2} - G + \frac{B_x^2 + \bar{B}_y^2 + \bar{B}_z^2}{4\pi\rho} - \delta b^2 \right) \quad (2.68)$$

$$a_*^2 = (\gamma_t - 1) \left(\bar{H} - \frac{\bar{v}_x^2 + \bar{v}_y^2 + \bar{v}_z^2}{2} - G - \delta b^2 \right) \quad (2.69)$$

$$- (\gamma_t - 2) \left(\frac{B_x^2 + \bar{B}_y^2 + \bar{B}_z^2}{4\pi\rho} \right) \quad (2.70)$$

$$b_x = \frac{|B_x|}{\sqrt{4\pi\rho}} \quad (2.71)$$

$$c_{f,s}^2 = \frac{a_*^2 \pm \sqrt{a_*^4 - 4a^2 b_x^2}}{2} \quad (2.72)$$

$$G = \bar{\varepsilon}_c + \bar{\rho} \frac{\Delta \varepsilon_c}{\Delta \rho} - \frac{1}{\gamma_t - 1} \frac{\Delta P_c}{\Delta \rho} \quad (2.73)$$

である。また解の公式から導かれる次の関係式は計算を進める上で有用なので指摘しておく。

$$\frac{c_f c_s}{b_x a} = 1 \quad (2.74)$$

このように理想気体の場合の磁気流体力学方程式の Roe 法の数値流束に対して、音速に G という補正項を加えた形で固有値が求まった。

2.2.4 固有ベクトル

Alfvén 波

最初に Alfvén 波を分離する事を考える。Alfvén 波は横波であり、磁場に垂直な方向の速度成分と磁場成分を運ぶ。従って、

$$\begin{pmatrix} \Delta \rho \\ \Delta v_x \\ \Delta v_{\parallel} \\ \Delta v_{\perp} \\ \Delta B_{\parallel} \\ \Delta B_{\perp} \\ \Delta \varepsilon_t \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ K_{\perp} \\ 0 \\ L_{\perp} \\ 0 \end{pmatrix} \quad (2.75)$$

のように2つの未知の成分を持つベクトルを考え、このベクトルが以下の固有値方程式を満たすように成分を決定できるかどうかを試みる。

$$\Delta \mathbf{U} = \begin{pmatrix} 0 \\ 0 \\ -\bar{\rho}\beta_z K_\perp \\ \bar{\rho}\beta_y K_\perp \\ -\beta_z L_\perp \\ \beta_y L_\perp \\ \bar{\rho}(\beta_y v_z + \beta_z v_y) K_\perp \end{pmatrix} \quad (2.76)$$

$$\begin{aligned} \Delta \mathbf{F} - \bar{v}_x \Delta \mathbf{U}' &= \begin{pmatrix} 0 \\ 0 \\ \frac{B_x}{4\pi} \beta_z L_\perp \\ -\frac{B_x}{4\pi} \beta_y L_\perp \\ B_x \beta_z K_\perp \\ -B_x \beta_y K_\perp \\ -\frac{B_x}{4\pi} (-\beta_z v_y + \beta_y v_z) L_\perp \end{pmatrix} \\ &= \pm \frac{B_x}{\sqrt{4\pi}\bar{\rho}} \Delta \mathbf{U} \end{aligned} \quad (2.77)$$

この連立方程式は解が存在し、以下のベクトルが Alfvén 速度を固有値とする固有ベクトルになっている事がわかる。

$$\mathbf{r}_{A\pm} = \begin{pmatrix} 0 \\ 0 \\ \mp \beta_z \operatorname{sgn}(B_x) \\ \pm \beta_y \operatorname{sgn}(B_x) \\ \beta_z \sqrt{4\pi/\bar{\rho}} \\ -\beta_y \sqrt{4\pi/\bar{\rho}} \\ \mp (\beta_z \bar{v}_y - \beta_y \bar{v}_z) \operatorname{sgn}(B_x) \end{pmatrix} \quad (2.79)$$

エントロピー波

エントロピー波についても同様にして固有ベクトルを求める。エントロピー波は密度とエネルギー成分のみを運ぶ波であるから、

$$\begin{pmatrix} \Delta \rho \\ \Delta v_x \\ \Delta v_\parallel \\ \Delta v_\perp \\ \Delta B_\parallel \\ \Delta B_\perp \\ \Delta \varepsilon_t \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ K_0 \end{pmatrix} \quad (2.80)$$

のような形のベクトルが、以下の固有値方程式を満たすように成分を決定すればよい。

$$\Delta \mathbf{U} = \begin{pmatrix} 1 \\ \bar{v}_x \\ \bar{v}_y \\ \bar{v}_z \\ 0 \\ 0 \\ \bar{\varepsilon}_c + \bar{\varepsilon}_t + \frac{\bar{v}^2}{2} + \frac{\gamma_t - 2}{\gamma_t - 1} \delta b^2 + \frac{\Delta \varepsilon_c}{\Delta \rho} \bar{\rho} + \bar{\rho} K_0 \end{pmatrix} \quad (2.81)$$

$$\Delta \mathbf{F} - \bar{v}_x \Delta \mathbf{U}' = \begin{pmatrix} 0 \\ \frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1) \bar{\varepsilon}_t + \delta b^2 + (\gamma_t - 1) \bar{\rho} K_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \bar{v}_x \left[\frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1) \bar{\varepsilon}_t + \delta b^2 + (\gamma_t - 1) \bar{\rho} a \right] \end{pmatrix} \quad (2.82)$$

$$= 0 \quad (2.83)$$

すると以下のようにエントロピー波の固有ベクトルを決定する事が出来る。

$$\mathbf{r}_0 = \begin{pmatrix} 1 \\ \bar{v}_x \\ \bar{v}_y \\ \bar{v}_z \\ 0 \\ 0 \\ \frac{(\bar{v})^2}{2} + \delta b^2 + G \end{pmatrix} \quad (2.84)$$

fast 波と slow 波

fast 波と slow 波についても同様にして計算する。fast 波の場合は、以下のような、垂直方向の成分だけが0である形のベクトルを探す。

$$\begin{pmatrix} \Delta \rho \\ \Delta v_x \\ \Delta v_{\parallel} \\ \Delta v_{\perp} \\ \Delta B_{\parallel} \\ \Delta B_{\perp} \\ \Delta \varepsilon_t \end{pmatrix} = \begin{pmatrix} \alpha_f \\ K_{\parallel} \\ L_{\parallel} \\ 0 \\ M_{\parallel} \\ 0 \\ N_{\parallel} \end{pmatrix} \quad (2.85)$$

ここで、

$$\alpha_f = \frac{\sqrt{c_f^2 - b_x^2}}{\sqrt{c_f^2 - c_s^2}} \quad (2.86)$$

$$\alpha_s = \frac{\sqrt{c_f^2 - a^2}}{\sqrt{c_f^2 - c_s^2}} = \frac{c_f}{b_x} \frac{\sqrt{b_x^2 - c_s^2}}{\sqrt{c_f^2 - c_s^2}} \quad (2.87)$$

$$\alpha_f^2 + \frac{b_x^2}{c_f^2} \alpha_s^2 = 1 \quad (2.88)$$

である。

これまでと同様にして、

$$\Delta \mathbf{U} = \begin{pmatrix} \alpha_f \\ \bar{v}_x \alpha_f + \bar{\rho} K_{\parallel} \\ \bar{v}_y \alpha_f + \bar{\rho} \beta_y L_{\parallel} \\ \bar{v}_z \alpha_f + \bar{\rho} \beta_z L_{\parallel} \\ \beta_y M_{\parallel} \\ \beta_z M_{\parallel} \\ X \end{pmatrix} \quad (2.89)$$

$$X = \left(\bar{\varepsilon}_c + \bar{\varepsilon}_t + \frac{\mathbf{v}^2}{2} + \delta b^2 + \frac{\Delta \varepsilon_c}{\Delta \rho} \bar{\rho} \right) \alpha_f \quad (2.90)$$

$$+ \bar{\rho} \bar{v}_x K_{\parallel} + \bar{\rho} (\beta_y v_y + \beta_z v_z) L_{\parallel} + \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} M_{\parallel} + \bar{\rho} N_{\parallel} \quad (2.91)$$

$$\Delta \mathbf{F} - \bar{v}_x \Delta \mathbf{U}' = \begin{pmatrix} \bar{\rho} K_{\parallel} \\ Y \\ \bar{\rho} \bar{v}_y K_{\parallel} - \frac{B_x \beta_y}{4\pi} M_{\parallel} \\ \bar{\rho} \bar{v}_z K_{\parallel} - \frac{B_x \beta_z}{4\pi} M_{\parallel} \\ \bar{B}_y K_{\parallel} - B_x \beta_y L_{\parallel} \\ \bar{B}_z K_{\parallel} - B_x \beta_z L_{\parallel} \\ X \end{pmatrix} \quad (2.92)$$

$$= \pm c_f \Delta \mathbf{U} \quad (2.93)$$

$$(2.94)$$

ただし、

$$Y = \left[\frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1) \bar{\varepsilon}_t + \delta b^2 \right] \alpha_f + \bar{\rho} \bar{v}_x K_{\parallel} \quad (2.95)$$

$$+ \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} M_{\parallel} + (\gamma_t - 1) \bar{\rho} N_{\parallel} \quad (2.96)$$

$$X = \alpha_f \bar{v}_x \left\{ \delta b^2 + \frac{\Delta P_c}{\Delta \rho} + (\gamma_t - 1) \bar{\varepsilon}_t \right\} \quad (2.97)$$

$$+ K_{\parallel} \left\{ \bar{\rho} (\bar{\varepsilon}_c + \bar{\varepsilon}_t) + \bar{P}_c + (\gamma_t - 1) \bar{\rho} \bar{\varepsilon}_t + \frac{\bar{\rho} \mathbf{v}^2}{2} + \frac{\bar{B}_y^2 + \bar{B}_z^2}{4\pi} \right\} \quad (2.98)$$

$$- \frac{B_x \sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} L_{||} + \left\{ \frac{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}}{4\pi} \bar{v}_x - \frac{B_x}{4\pi} (\beta_y v_y + \beta_z v_z) \right\} M_{||} \quad (2.99)$$

$$+ \bar{\rho}(\gamma_t - 1) \bar{v}_x N_{||} \quad (2.100)$$

連立方程式を解くと、fast 波の固有ベクトルとして、

$$\mathbf{r}_{f\pm} = \begin{pmatrix} \alpha_f \\ \alpha_f (\bar{v}_x \pm c_f) \\ \alpha_f \bar{v}_y \mp \alpha_s \beta_y b_x \operatorname{sgn}(B_x) \\ \alpha_f \bar{v}_z \mp \alpha_s \beta_z b_x \operatorname{sgn}(B_x) \\ \alpha_s \beta_y c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_f \left\{ \frac{(\bar{v})^2}{2} + \delta b^2 + G \pm c_f \bar{v}_x + \frac{c_f^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_f^2 - a^2) \right\} \pm \alpha_s b_x \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.101)$$

が得られる。

slow 波についても同様の手続きにより、固有ベクトルが以下のように求まる。

$$\mathbf{r}_{s\pm} = \begin{pmatrix} \alpha_s \\ \alpha_s (\bar{v}_x \pm c_s) \\ \alpha_s \bar{v}_y \pm \alpha_f \beta_y a \operatorname{sgn}(B_x) \\ \alpha_s \bar{v}_z \pm \alpha_f \beta_z a \operatorname{sgn}(B_x) \\ - \frac{\alpha_f \beta_y a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ - \frac{\alpha_f \beta_z a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \left\{ \frac{(\bar{v})^2}{2} + \delta b^2 + G \pm c_s \bar{v}_x + \frac{c_s^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_s^2 - a^2) \right\} \pm \alpha_f a \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.102)$$

2.2.5 振幅

固有値と固有ベクトルが完全に求まったので、振幅を求める事ができる。だが新たに計算しなくとも、G の補正項が打ち消されて理想気体の場合と変わらない式で書ける。

$$\delta w_{A+} = \frac{1}{2} \left[-\bar{\rho}(\beta_z \Delta v_y - \beta_y \Delta v_z) \operatorname{sgn}(B_x) + \sqrt{\frac{\bar{\rho}}{4\pi}} (\beta_z \Delta B_y - \beta_y \Delta B_z) \right] \quad (2.103)$$

$$\delta w_{A-} = \frac{1}{2} \left[\bar{\rho}(\beta_z \Delta v_y - \beta_y \Delta v_z) \operatorname{sgn}(B_x) + \sqrt{\frac{\bar{\rho}}{4\pi}} (\beta_z \Delta B_y - \beta_y \Delta B_z) \right] \quad (2.104)$$

$$\delta w_{f+} + \delta w_{f-} = \frac{\alpha_f}{c_f^2} \left(\Delta P + \frac{\bar{B}_y \Delta B_y + \bar{B}_z \Delta B_z}{4\pi} \right) \quad (2.105)$$

$$+ \left\{ \frac{\alpha_s}{a^2 c_f} [(\gamma_t - 1)c_s^2 - (\gamma_t - 2)a^2] \sqrt{4\pi\bar{\rho}} \right. \quad (2.106)$$

$$+ (\gamma_t - 2) \sqrt{\bar{B}_y^2 + \bar{B}_z^2} \frac{\alpha_f}{c_f^2} \left. \right\} \frac{\beta_y \Delta B_y + \beta_z \Delta B_z}{4\pi} \quad (2.107)$$

$$\delta w_{f+} - \delta w_{f-} = \frac{\alpha_f}{c_f} \bar{\rho} \Delta v_x - \frac{\alpha_s c_s}{c_f a} \text{sgn}(B_x) \bar{\rho} (\beta_y \Delta v_y + \beta_z \Delta v_z) \quad (2.108)$$

$$\delta w_{s+} + \delta w_{s-} = \frac{\alpha_s}{a^2} \left(\Delta P + \frac{\tilde{B}_y \Delta B_y + \tilde{B}_z \Delta B_z}{4\pi} \right) \quad (2.109)$$

$$+ \left\{ \alpha_f \left[\frac{\gamma_t - 2}{c_f} - (\gamma_t - 1) \frac{c_f}{a^2} \right] \sqrt{4\pi\bar{\rho}} \right. \quad (2.110)$$

$$+ (\gamma_t - 2) \sqrt{\bar{B}_y^2 + \bar{B}_z^2} \frac{\alpha_s}{a^2} \left. \right\} \frac{\beta_y \Delta B_y + \beta_z \Delta B_z}{4\pi} \quad (2.111)$$

$$\delta w_{s+} - \delta w_{s-} = \frac{\alpha_s b_x}{c_f a} \bar{\rho} \Delta v_x + \frac{\alpha_f}{a} \text{sgn}(B_x) \bar{\rho} (\beta_y \Delta v_y + \beta_z \Delta v_z) \quad (2.112)$$

$$\delta w_0 = \Delta \rho - \alpha_f (\delta w_1 + \delta w_7) - \alpha_s (\delta w_3 + \delta w_5) \quad (2.113)$$

以上で固有値・固有ベクトル・振幅が完全に求まったので、数値流束を構成する事ができる。

2.2.6 補正項 G についての考察

ここで補正項 G について説明する。 G はもともと次の式で表される量である。

$$\Delta(\rho E)_{\text{New}} - \Delta(\rho E)_{\text{Org}} = \Delta(\rho \varepsilon_c + \rho \varepsilon_t) - \Delta \left(\frac{P}{\gamma_t - 1} \right) \quad (2.114)$$

$$= \Delta \left(\rho \varepsilon_c - \frac{P_c}{\gamma_t - 1} \right) \quad (2.115)$$

$$= \Delta \rho \left(\bar{\varepsilon}_c + \bar{\rho} \frac{\Delta \varepsilon_c}{\Delta \rho} - \frac{1}{\gamma_t - 1} \frac{\Delta P_c}{\Delta \rho} \right) \quad (2.116)$$

$$= (\Delta \rho) G \quad (2.117)$$

ただし、New の添字は今回採用した状態方程式を用いる場合のエネルギー成分の式に代入する事を意味し、Org は理想気体の状態方程式を用いる場合のエネルギー成分の式に代入する事を意味する（圧力 P には新しい状態方程式を代入する事に注意）。すなわち、状態方程式を変えた事によるエネルギー成分の増加量に対応している。

これまで述べてきたように、状態方程式を変更した事による効果は全て G の項で表される。従って、今回採用している状態方程式以外の場合は、行列を計算し直す事なく、この G を新たに計算するだけで新たな数値流束を求める事ができる。

従って今回新たに求めた数値流束は、状態方程式が、

$$P_{\text{tot}}(\rho, \varepsilon_t) = (\gamma_t - 1) \rho \varepsilon_t + f(\rho) \quad (2.118)$$

と表される場合にまで一般化されている。

物理量としての G は、密度変化 $\Delta \rho$ が微小の場合以下の式で表される。

$$G = \varepsilon_c + \frac{P_c}{\rho} - \frac{1}{\gamma_t - 1} \frac{dP_c}{d\rho} \quad (2.119)$$

2.3 テスト計算

2.3.1 衝撃波管問題

コードのテストとして (Ryu & Jones, 1995) のテスト問題を基に、縮退圧を加えた場合のテストを実行した。以下にその結果を掲載する。3つのモデルを計算し、それらを比較している。モデル 1 は Ryu & Jones のモデル、モデル 2 はモデル 1 に縮退圧を $\kappa = 1$ 、 $\gamma = 4/3$ として加えたモデル、モデル 3 はモデル 1 に縮退圧を $\rho \leq 1$ のとき $\kappa = 1$ 、 $\gamma = 1.3$ 、 $\rho \geq 1$ のとき $\kappa = 1$ 、 $\gamma = 2.0$ として加えたモデルである。また各モデルで $\gamma_t = 1.3$ 、クーラン数を 0.5 とした。

Ryu & Jones の初期条件は、

$$\begin{pmatrix} \rho \\ v_x \\ v_y \\ v_z \\ B_x \\ B_y \\ B_z \\ P_t \end{pmatrix}_L = \begin{pmatrix} 1.08 \\ 1.2 \\ 0.01 \\ 0.5 \\ 2 \\ 3.6 \\ 2 \\ 0.95 \end{pmatrix} \quad (2.120)$$

$$\begin{pmatrix} \rho \\ v_x \\ v_y \\ v_z \\ B_x \\ B_y \\ B_z \\ P_t \end{pmatrix}_R = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 2 \\ 4 \\ 2 \\ 1 \end{pmatrix} \quad (2.121)$$

である。添字 L は不連続面の左側の値、添字 R は不連続面の右側の値である事を示す。

以下の図は $t = 0.17$ における各物理量の分布図である。

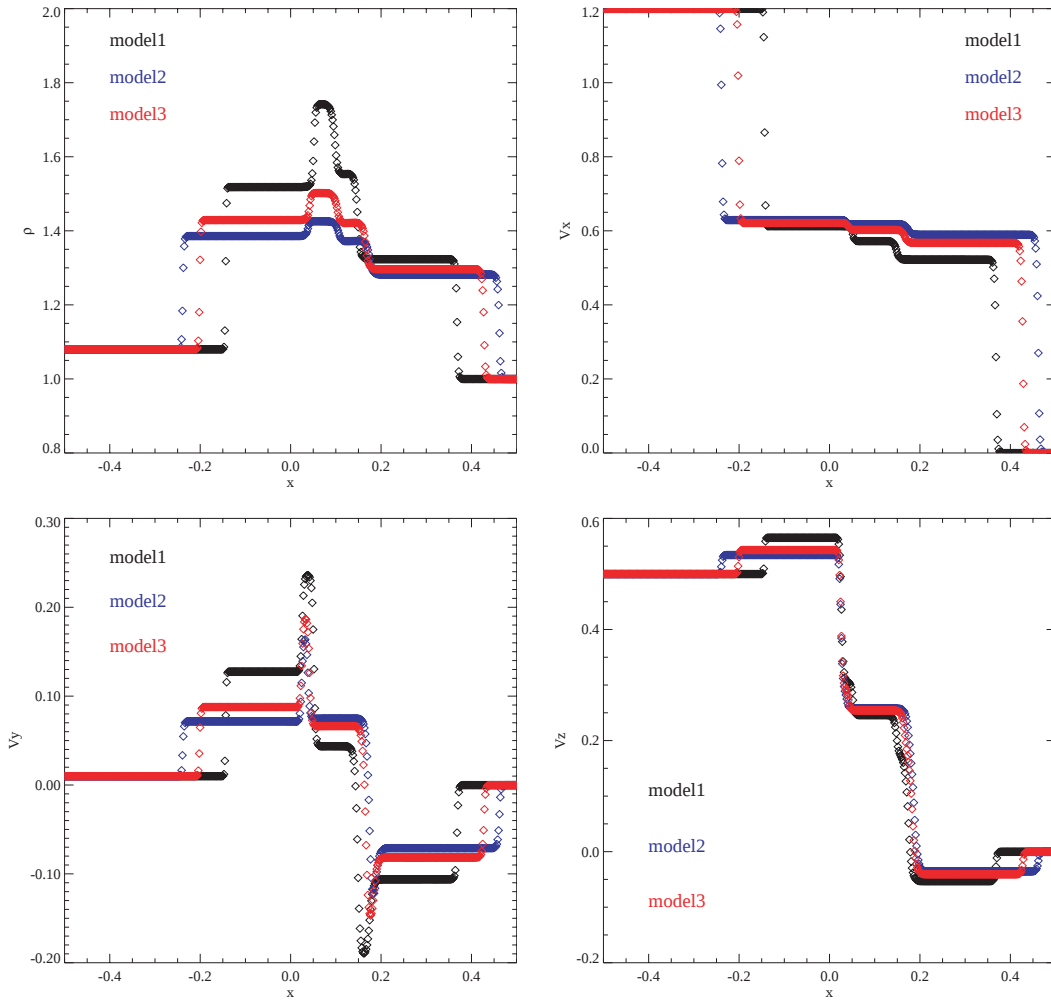


図 2.1: テスト問題の計算結果。上段左が密度、上段右が v_x 、下段左が v_y 、下段右が v_z の時刻 0.17 での分布図である。黒線が model1、青線が model2、赤線が model3 の結果を表している。

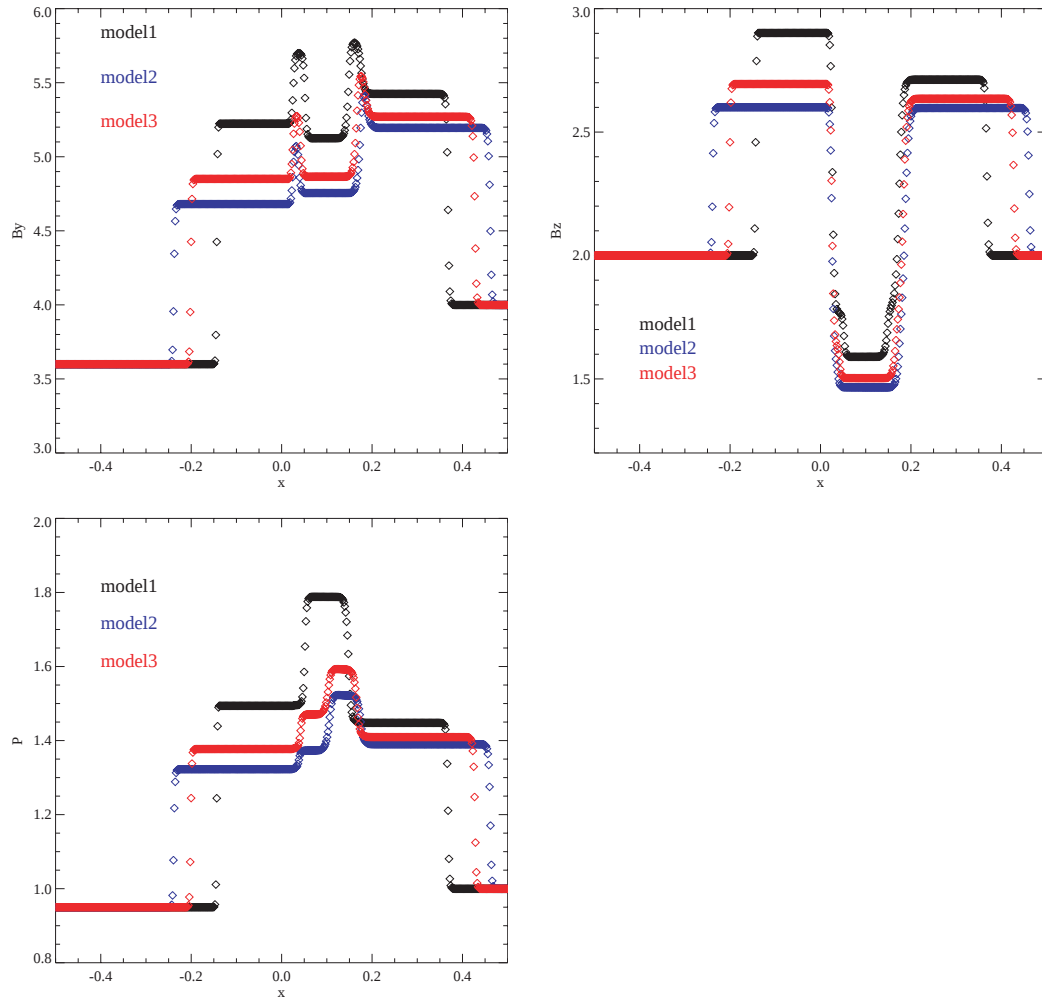


図 2.2: テスト問題の計算結果。上段左が B_y 、上段右が B_z 、下段左が P の時刻 0.17 での分布図である。黒線が model1、青線が model2、赤線が model3 の結果を表している。

2.4 公式のまとめ

基礎方程式系

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (2.122)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ B_y \\ B_z \\ \rho E \end{pmatrix} \quad (2.123)$$

$$\mathbf{F} = \begin{pmatrix} \rho v_x^2 + P + \frac{\rho v_x}{B_y^2 + B_z^2 - B_x^2} \\ \rho v_x v_y - \frac{B_x B_y}{4\pi} \\ \rho v_x v_z - \frac{B_x B_z}{4\pi} \\ v_x B_y - v_y B_x \\ v_x B_z - v_z B_x \\ \rho H v_x - \frac{B_x (B_x v_x + B_y v_y + B_z v_z)}{4\pi} \end{pmatrix} \quad (2.124)$$

Roe 平均

$$\bar{\rho} = \sqrt{\rho_{j+1} \rho_j} \quad (2.125)$$

$$\bar{v}_x = \frac{\sqrt{\rho_{j+1}} v_{x,j+1} + \sqrt{\rho_j} v_{x,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.126)$$

$$\bar{v}_y = \frac{\sqrt{\rho_{j+1}} v_{y,j+1} + \sqrt{\rho_j} v_{y,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.127)$$

$$\bar{v}_z = \frac{\sqrt{\rho_{j+1}} v_{z,j+1} + \sqrt{\rho_j} v_{z,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.128)$$

$$\bar{B}_y = \frac{\sqrt{\rho_{j+1}} B_{y,j} + \sqrt{\rho_j} B_{y,j+1}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.129)$$

$$\bar{B}_z = \frac{\sqrt{\rho_{j+1}} B_{z,j} + \sqrt{\rho_j} B_{z,j+1}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.130)$$

$$\bar{\varepsilon}_c = \frac{\sqrt{\rho_{j+1}} \varepsilon_{c,j+1} + \sqrt{\rho_j} \varepsilon_{c,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.131)$$

$$\bar{\varepsilon}_t = \frac{\sqrt{\rho_{j+1}} \varepsilon_{t,j+1} + \sqrt{\rho_j} \varepsilon_{t,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.132)$$

$$\bar{P}_c = \frac{\sqrt{\rho_{j+1}} P_{c,j+1} + \sqrt{\rho_j} P_{c,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.133)$$

固有値

$$\lambda_0 = \bar{v}_x \quad (2.134)$$

$$\lambda_{A+} = \bar{v}_x + b_x \quad (2.135)$$

$$\lambda_{A-} = \bar{v}_x - b_x \quad (2.136)$$

$$\lambda_{f+} = \bar{v}_x + c_f \quad (2.137)$$

$$\lambda_{f-} = \bar{v}_x - c_f \quad (2.138)$$

$$\lambda_{s+} = \bar{v}_x + c_s \quad (2.139)$$

$$\lambda_{s-} = \bar{v}_x - c_s \quad (2.140)$$

$$\delta b^2 = \frac{\gamma_t - 1}{\gamma_t - 2} \frac{(B_{y,j+1} - B_{y,j})^2 + (B_{z,j+1} - B_{z,j})^2}{8\pi (\sqrt{\rho_{j+1}} + \sqrt{\rho_j})^2} \quad (2.141)$$

$$a^2 = (\gamma_t - 1) \left(\bar{H} - \frac{\bar{v}_x^2 + \bar{v}_y^2 + \bar{v}_z^2}{2} - G + \frac{B_x^2 + \bar{B}_y^2 + \bar{B}_z^2}{4\pi\rho} - \delta b^2 \right) \quad (2.142)$$

$$a_*^2 = (\gamma_t - 1) \left(\bar{H} - \frac{\bar{v}_x^2 + \bar{v}_y^2 + \bar{v}_z^2}{2} - G - \delta b^2 \right) \quad (2.143)$$

$$- (\gamma_t - 2) \left(\frac{B_x^2 + \bar{B}_y^2 + \bar{B}_z^2}{4\pi\rho} \right) \quad (2.144)$$

$$b_x = \frac{|B_x|}{\sqrt{4\pi\bar{\rho}}} \quad (2.145)$$

$$c_{f,s}^2 = \frac{a_*^2 \pm \sqrt{a_*^4 - 4a^2 b_x^2}}{2} \quad (2.146)$$

固有ベクトル (entropy)

$$\boldsymbol{r}_0 = \begin{pmatrix} 1 \\ \bar{v}_x \\ \bar{v}_y \\ \bar{v}_z \\ 0 \\ 0 \\ \frac{(\bar{\boldsymbol{v}})^2}{2} + \delta b^2 + G \end{pmatrix} \quad (2.147)$$

固有ベクトル (Alfvén)

$$\mathbf{r}_{A+} = \begin{pmatrix} 0 \\ 0 \\ -\beta_z \operatorname{sgn}(B_x) \\ \beta_y \operatorname{sgn}(B_x) \\ \beta_z \sqrt{4\pi/\bar{\rho}} \\ -\beta_y \sqrt{4\pi/\bar{\rho}} \\ -(\beta_z \bar{v}_y - \beta_y \bar{v}_z) \operatorname{sgn}(B_x) \end{pmatrix} \quad (2.148)$$

$$\mathbf{r}_{A-} = \begin{pmatrix} 0 \\ 0 \\ \beta_z \operatorname{sgn}(B_x) \\ -\beta_y \operatorname{sgn}(B_x) \\ \beta_z \sqrt{4\pi/\bar{\rho}} \\ -\beta_y \sqrt{4\pi/\bar{\rho}} \\ (\beta_z \bar{v}_y - \beta_y \bar{v}_z) \operatorname{sgn}(B_x) \end{pmatrix} \quad (2.149)$$

固有ベクトル (fast)

$$\mathbf{r}_{f+} = \begin{pmatrix} \alpha_f \\ \alpha_f (\bar{v}_x + c_f) \\ \alpha_f \bar{v}_y - \alpha_s \beta_y b_x \operatorname{sgn}(B_x) \\ \alpha_f \bar{v}_z - \alpha_s \beta_z b_x \operatorname{sgn}(B_x) \\ \alpha_s \beta_y c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_f \left\{ \frac{(\bar{\mathbf{v}})^2}{2} + \delta b^2 + G + c_f \bar{v}_x + \frac{c_f^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_f^2 - a^2) \right\} - \alpha_s b_x \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.150)$$

$$\mathbf{r}_{f-} = \begin{pmatrix} \alpha_f \\ \alpha_f (\bar{v}_x - c_f) \\ \alpha_f \bar{v}_y + \alpha_s \beta_y b_x \operatorname{sgn}(B_x) \\ \alpha_f \bar{v}_z + \alpha_s \beta_z b_x \operatorname{sgn}(B_x) \\ \alpha_s \beta_y c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_f \left\{ \frac{(\bar{\mathbf{v}})^2}{2} + \delta b^2 + G - c_f \bar{v}_x + \frac{c_f^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_f^2 - a^2) \right\} + \alpha_s b_x \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.151)$$

固有ベクトル (slow)

$$\mathbf{r}_{s+} = \begin{pmatrix} \alpha_s \\ \alpha_s (\bar{v}_x + c_s) \\ \alpha_s \bar{v}_y + \alpha_f \beta_y a \operatorname{sgn}(B_x) \\ \alpha_s \bar{v}_z + \alpha_f \beta_z a \operatorname{sgn}(B_x) \\ - \frac{\alpha_f \beta_y a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ - \frac{\alpha_f \beta_z a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \left\{ \frac{(\bar{\mathbf{v}})^2}{2} + \delta b^2 + G + c_s \bar{v}_x + \frac{c_s^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_s^2 - a^2) \right\} + \alpha_f a \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.152)$$

$$\mathbf{r}_{s+} = \begin{pmatrix} \alpha_s \\ \alpha_s (\bar{v}_x - c_s) \\ \alpha_s \bar{v}_y - \alpha_f \beta_y a \operatorname{sgn}(B_x) \\ \alpha_s \bar{v}_z - \alpha_f \beta_z a \operatorname{sgn}(B_x) \\ - \frac{\alpha_f \beta_y a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ - \frac{\alpha_f \beta_z a^2 \sqrt{4\pi}}{c_f \sqrt{\bar{\rho}}} \\ \alpha_s \beta_z c_f \sqrt{4\pi/\bar{\rho}} \\ \alpha_s \left\{ \frac{(\bar{\mathbf{v}})^2}{2} + \delta b^2 + G - c_s \bar{v}_x + \frac{c_s^2}{\gamma_t - 1} \right. \\ \left. + \frac{\gamma_t - 2}{\gamma_t - 1} (c_s^2 - a^2) \right\} - \alpha_f a \operatorname{sgn}(B_x) (\beta_y \bar{v}_y + \beta_z \bar{v}_z) \end{pmatrix} \quad (2.153)$$

$$\alpha_f = \frac{\sqrt{c_f^2 - b_x^2}}{\sqrt{c_f^2 - c_s^2}} \quad (2.154)$$

$$\alpha_s = \frac{\sqrt{c_f^2 - a^2}}{\sqrt{c_f^2 - c_s^2}} = \frac{c_f}{b_x} \frac{\sqrt{b_x^2 - c_s^2}}{\sqrt{c_f^2 - c_s^2}} \quad (2.155)$$

$$c_{f,s}^2 = \frac{a_*^2 \pm \sqrt{a_*^4 - 4a^2 b_x^2}}{2} \quad (2.156)$$

$$\beta_y = \frac{\bar{B}_y}{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}} \quad (2.157)$$

$$\beta_z = \frac{\bar{B}_z}{\sqrt{\bar{B}_y^2 + \bar{B}_z^2}} \quad (2.158)$$

$$\beta_y^2 + \beta_z^2 = 1 \quad (2.159)$$

$$\alpha_f^2 + \frac{b_x^2}{c_f^2} \alpha_s^2 = 1 \quad (2.160)$$

振幅

$$\delta w_2 = \frac{1}{2} \left[-\bar{\rho}(\beta_z \Delta v_y - \beta_y \Delta v_z) \text{sgn}(B_x) + \sqrt{\frac{\bar{\rho}}{4\pi}} (\beta_z \Delta B_y - \beta_y \Delta B_z) \right] \quad (2.161)$$

$$\delta w_6 = \frac{1}{2} \left[\bar{\rho}(\beta_z \Delta v_y - \beta_y \Delta v_z) \text{sgn}(B_x) + \sqrt{\frac{\bar{\rho}}{4\pi}} (\beta_z \Delta B_y - \beta_y \Delta B_z) \right] \quad (2.162)$$

$$\delta w_1 + \delta w_7 = \frac{\alpha_f}{c_f^2} \left(\Delta P + \frac{\tilde{B}_y \Delta B_y + \tilde{B}_z \Delta B_z}{4\pi} \right) \quad (2.163)$$

$$+ \left\{ \frac{\alpha_s}{a^2 c_f} [(\gamma_t - 1)c_s^2 - (\gamma_t - 2)a^2] \sqrt{4\pi \bar{\rho}} \right. \quad (2.164)$$

$$+ (\gamma_t - 2) \sqrt{\bar{B}_y^2 + \bar{B}_z^2} \frac{\alpha_f}{c_f^2} \left. \right\} \frac{\beta_y \Delta B_y + \beta_z \Delta B_z}{4\pi} \quad (2.165)$$

$$\delta w_1 - \delta w_7 = \frac{\alpha_f}{c_f} \bar{\rho} \Delta v_x - \frac{\alpha_s c_s}{c_f a} \text{sgn}(B_x) \bar{\rho} (\beta_y \Delta v_y + \beta_z \Delta v_z) \quad (2.166)$$

$$\delta w_3 + \delta w_5 = \frac{\alpha_s}{a^2} \left(\Delta P + \frac{\tilde{B}_y \Delta B_y + \tilde{B}_z \Delta B_z}{4\pi} \right) \quad (2.167)$$

$$+ \left\{ \alpha_f \left[\frac{\gamma_t - 2}{c_f} - (\gamma_t - 1) \frac{c_f}{a^2} \right] \sqrt{4\pi \bar{\rho}} \right. \quad (2.168)$$

$$+ (\gamma_t - 2) \sqrt{\bar{B}_y^2 + \bar{B}_z^2} \frac{\alpha_s}{a^2} \left. \right\} \frac{\beta_y \Delta B_y + \beta_z \Delta B_z}{4\pi} \quad (2.169)$$

$$\delta w_3 - \delta w_5 = \frac{\alpha_s b_x}{c_f a} \bar{\rho} \Delta v_x + \frac{\alpha_f}{a} \text{sgn}(B_x) \bar{\rho} (\beta_y \Delta v_y + \beta_z \Delta v_z) \quad (2.170)$$

$$\delta w_4 = \Delta \rho - \alpha_f (\delta w_1 + \delta w_7) - \alpha_s (\delta w_3 + \delta w_5) \quad (2.171)$$

2.5 参考

参考のために、1次元と3次元の流体の場合に今回の状態方程式を採用した場合の公式を掲載する。

2.5.1 1次元流体力学方程式の場合

$$\frac{\partial U}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (2.172)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v \\ \rho E \end{pmatrix} \quad (2.173)$$

$$\mathbf{F} = \begin{pmatrix} \rho v \\ \rho v^2 + P \\ \rho H v \end{pmatrix} \quad (2.174)$$

$$\Delta \mathbf{U} = \begin{pmatrix} \rho_{j+1} - \rho_j \\ \rho_{j+1} v_{j+1} - \rho_j v_j \\ \rho_{j+1} E_{j+1} - \rho_j E_j \end{pmatrix} \quad (2.175)$$

$$= \begin{pmatrix} \rho_{j+1} - \rho_j \\ \rho_{j+1} v_{j+1} - \rho_j v_j \\ \rho_{j+1} H_{j+1} - \rho_j H_j + P_{j+1} - P_j \end{pmatrix} \quad (2.176)$$

$$\Delta \mathbf{F} = \begin{pmatrix} \rho_{j+1} v_{j+1} - \rho_j v_j \\ \rho_{j+1} v_{j+1}^2 + P_{j+1} - \rho_j v_j^2 - P_j \\ \rho_{j+1} H_{j+1} v_{j+1} - \rho_j H_j v_j \end{pmatrix} \quad (2.177)$$

Roe 平均

$$\bar{\rho} = \sqrt{\rho_{j+1} \rho_j} \quad (2.178)$$

$$\bar{v} = \frac{\sqrt{\rho_{j+1}} v_{j+1} + \sqrt{\rho_j} v_j}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.179)$$

$$\bar{H} = \frac{\sqrt{\rho_{j+1}} H_{j+1} + \sqrt{\rho_j} H_j}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.180)$$

$$\bar{c}^2 = (\gamma_t - 1) \left(\bar{H} - \frac{(\bar{v})^2}{2} - G \right) \quad (2.181)$$

$$G = (\Delta \rho)^{-1} \Delta \left(\rho \varepsilon_c - \frac{P_c}{\gamma_t - 1} \right) \quad (2.182)$$

$$= \bar{\varepsilon}_c + \bar{\rho} \frac{\Delta \varepsilon_c}{\Delta \rho} - \frac{1}{\gamma_t - 1} \frac{\Delta P_c}{\Delta \rho} \quad (2.183)$$

固有値

$$\lambda_+ = \bar{v} + \bar{c} \quad (2.184)$$

$$\lambda_- = \bar{v} - \bar{c} \quad (2.185)$$

$$\lambda_0 = \bar{v} \quad (2.186)$$

右固有ベクトル

$$\mathbf{r}_+ = \begin{pmatrix} 1 \\ \bar{v} + \bar{c} \\ \bar{H} + \bar{c}\bar{v} \end{pmatrix} \quad (2.187)$$

$$\mathbf{r}_- = \begin{pmatrix} 1 \\ \bar{v} - \bar{c} \\ \bar{H} - \bar{c}\bar{v} \end{pmatrix} \quad (2.188)$$

$$\mathbf{r}_0 = \begin{pmatrix} 1 \\ \frac{(\bar{v})^2}{2} + G \end{pmatrix} \quad (2.189)$$

振幅

$$\delta w_+ = \frac{1}{2} \left(\frac{\Delta P}{\bar{c}^2} + \frac{\bar{\rho}}{\bar{c}} \Delta v \right) \quad (2.190)$$

$$\delta w_- = \frac{1}{2} \left(\frac{\Delta P}{\bar{c}^2} - \frac{\bar{\rho}}{\bar{c}} \Delta v \right) \quad (2.191)$$

$$\delta w_0 = \Delta \rho - \frac{\Delta P}{\bar{c}^2} \quad (2.192)$$

固有モードへの分解

$$\Delta \mathbf{U} = \delta w_+ \mathbf{r}_+ + \delta w_- \mathbf{r}_- + \delta w_0 \mathbf{r}_0 \quad (2.193)$$

$$\Delta \mathbf{F} = \delta w_+ \lambda_+ \mathbf{r}_+ + \delta w_- \lambda_- \mathbf{r}_- + \delta w_0 \lambda_0 \mathbf{r}_0 \quad (2.194)$$

2.5.2 3次元流体力学方程式の場合

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho E \end{pmatrix} \quad (2.195)$$

$$\mathbf{F} = \begin{pmatrix} \rho v \\ \rho v_x^2 + P \\ \rho v_x v_y \\ \rho v_x v_z \\ \rho H v_x \end{pmatrix} \quad (2.196)$$

Roe 平均

$$\bar{\rho} = \sqrt{\rho_{j+1}\rho_j} \quad (2.197)$$

$$\bar{v}_x = \frac{\sqrt{\rho_{j+1}} v_{x,j+1} + \sqrt{\rho_j} v_{x,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.198)$$

$$\bar{v}_y = \frac{\sqrt{\rho_{j+1}} v_{y,j+1} + \sqrt{\rho_j} v_{y,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.199)$$

$$\bar{v}_z = \frac{\sqrt{\rho_{j+1}} v_{z,j+1} + \sqrt{\rho_j} v_{z,j}}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.200)$$

$$\bar{H} = \frac{\sqrt{\rho_{j+1}} H_{j+1} + \sqrt{\rho_j} H_j}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \quad (2.201)$$

$$\bar{c}^2 = (\gamma_t - 1) \left(\bar{H} - \frac{(\bar{\mathbf{v}})^2}{2} - G \right) \quad (2.202)$$

$$G = (\Delta\rho)^{-1} \Delta \left(\rho \varepsilon_c - \frac{P_c}{\gamma_t - 1} \right) \quad (2.203)$$

$$= \bar{\varepsilon}_c + \bar{\rho} \frac{\Delta \varepsilon_c}{\Delta \rho} - \frac{1}{\gamma_t - 1} \frac{\Delta P_c}{\Delta \rho} \quad (2.204)$$

固有値

$$\lambda_+ = \bar{v}_x + \bar{c} \quad (2.205)$$

$$\lambda_- = \bar{v}_x - \bar{c} \quad (2.206)$$

$$\lambda_0 = \bar{v}_x \quad (2.207)$$

$$\lambda_y = \bar{v}_x \quad (2.208)$$

$$\lambda_z = \bar{v}_x \quad (2.209)$$

右固有ベクトル

$$\mathbf{r}_+ = \begin{pmatrix} 1 \\ \bar{v}_x + \bar{c} \\ \bar{v}_y \\ \bar{v}_z \\ \bar{H} + \bar{c}\bar{v}_x \end{pmatrix} \quad (2.210)$$

$$\mathbf{r}_- = \begin{pmatrix} 1 \\ \bar{v}_x - \bar{c} \\ \bar{v}_y \\ \bar{v}_z \\ \bar{H} - \bar{c}\bar{v}_x \end{pmatrix} \quad (2.211)$$

$$\mathbf{r}_0 = \begin{pmatrix} 1 \\ \bar{v}_x \\ \bar{v}_y \\ \bar{v}_z \\ \frac{(\bar{v})^2}{2} + G \end{pmatrix} \quad (2.212)$$

$$\mathbf{r}_y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \bar{v}_y \end{pmatrix} \quad (2.213)$$

$$\mathbf{r}_z = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \bar{v}_z \end{pmatrix} \quad (2.214)$$

振幅

$$\delta w_+ = \frac{1}{2} \left(\frac{\Delta P}{\bar{c}^2} + \frac{\bar{\rho}}{\bar{c}} \Delta v_x \right) \quad (2.215)$$

$$\delta w_- = \frac{1}{2} \left(\frac{\Delta P}{\bar{c}^2} - \frac{\bar{\rho}}{\bar{c}} \Delta v_x \right) \quad (2.216)$$

$$\delta w_0 = \Delta \rho - \frac{\Delta P}{\bar{c}^2} \quad (2.217)$$

$$\delta w_y = \bar{\rho} \Delta v_y \quad (2.218)$$

$$\delta w_z = \bar{\rho} \Delta v_z \quad (2.219)$$

$$(2.220)$$

固有モードへの分解

$$\Delta U = \delta w_+ \mathbf{r}_+ + \delta w_- \mathbf{r}_- + \delta w_0 \mathbf{r}_0 + \delta w_y \mathbf{r}_y + \delta w_z \mathbf{r}_z \quad (2.221)$$

$$\Delta \mathbf{F} = \delta w_+ \lambda_+ \mathbf{r}_+ + \delta w_- \lambda_- \mathbf{r}_- + \delta w_0 \lambda_0 \mathbf{r}_0 \quad (2.222)$$

$$+ \delta w_y \lambda_y \mathbf{r}_y + \delta w_z \lambda_z \mathbf{r}_z \quad (2.223)$$

参考文献

- [1] Fukuda, N., & Hanawa, T., 2000, ApJ, 533, 911
- [2] Nobuta, K., & Hanawa, T., 1999, ApJ, 510, 614
- [3] Roe, P. L. 1981, J. Comput. Phys., 43, 357
- [4] Ryu, D., & Jones, T. W., 1995, ApJ, 442, 228
- [5] 藤井孝藏, 1994, 流体力学の数値計算法, 東大出版会

第 3 章

CIP-MOCCT 法による磁気流体力学方程式の解法

京都大学 上原 一浩

3.1 はじめに

この章では、CIP-MOCCT 法を用いて磁気流体力学方程式を解く方法を紹介する。CIP-MOCCT 法は、Kudoh et al. (1997) が開発し、CIP 法、MOC 法、CT 法という 3 つの数値計算法を組み合わせたものである。

CIP 法 (Cubic Interpolated Propagation / Constrained Interpolated Profile)

移流方程式を解く方法として Yabe & Aoki (1991) によって提案された高次精度差分法の一つである。物理量だけでなく、その微分値も独立変数として用いることにより、補間精度を向上させた。これまで流体力学など様々な分野で応用されている。

CT 法 (Constrained Transport)

Evans & Hawley (1988) によって開発された。磁場と電場を計算格子上の異なる位置で定義することにより、 $\nabla \cdot \mathbf{B} = 0$ を保証する。

MOC 法 (Method Of Characteristic)

特性曲線をもちいることで、磁気流体波を安定に解くことができるように考案された。(Stone & Norman 1992, Hawley & Stone 1995)

計算格子 計算格子の上で、各物理量は次の位置に定義されている。密度・温度などのスカラー量は、セルの中心 (zone center)、速度・磁場はセル境界面の中心 (face center)、電場はセルの辺の中心 (edge center) に置く。図 3.1 はセルの 8 分の 1 を切り取って描いたものである。このように、グリッドをずらして定義するような格子は、スタガード格子とよばれる。

これから、これらの手法の磁気流体方程式への応用をみていく。

3.2 磁気流体力学方程式の移流形式での表現

磁気流体力学の基礎方程式は次のようにあたえられる。

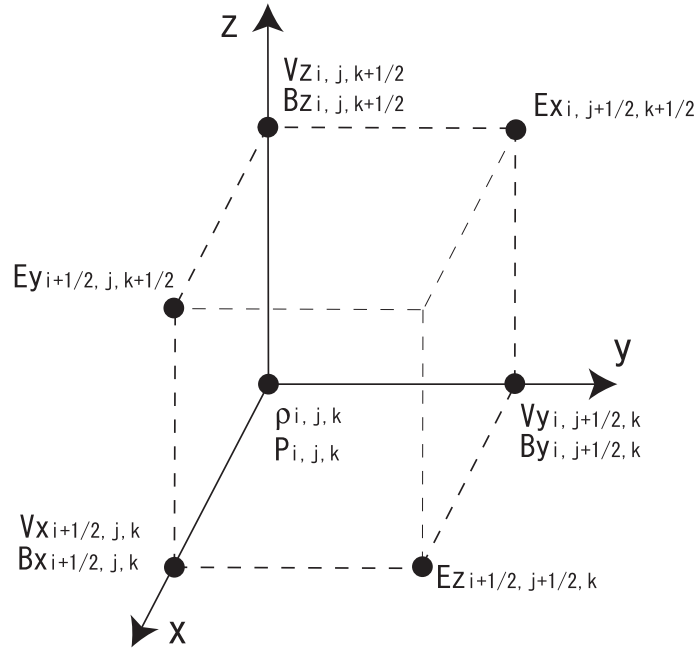


図 3.1: CIP-MOCCT 法における計算格子。密度・圧力はゾーンセンター、速度・磁場は、フェイスセンター、電場はエッジセンターとする (スタガード格子)。

$$\frac{\partial \rho}{\partial t} + (\mathbf{v} \cdot \nabla) \rho = -\rho \nabla \cdot \mathbf{v} \quad (3.1)$$

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla \left(p + \frac{B^2}{8\pi} \right) + \frac{1}{4\pi\rho} (\mathbf{B} \cdot \nabla) \mathbf{B} + \mathbf{g} \quad (3.2)$$

$$\frac{\partial p}{\partial t} + (\mathbf{v} \cdot \nabla) p = -\gamma p \nabla \cdot \mathbf{v} \quad (3.3)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) \quad (3.4)$$

流体計算の部分に CIP 法を適用するため、連続の式 (3.1)、運動方程式 (3.2)、エネルギー式 (3.3)、を移流項と非移流項にわけると。磁気張力項と誘導方程式の計算には MOC 法を用いる。 γ は比熱比、 $\mathbf{g}$ は、重力などの外力項である。

3.3 CIP 法

CIP 法の基礎的な部分については、Yabe & Aoki (1991) や天体とスペースプラズマのシミュレーションサマースクールのテキスト、より詳しいことは CIP 法 (矢部、内海、尾形 森北出版、2003) を参照。ここでは、1 次元で基本的な考え方を理解してから、3 次元へ拡張しその具体的な表式をあたえる。

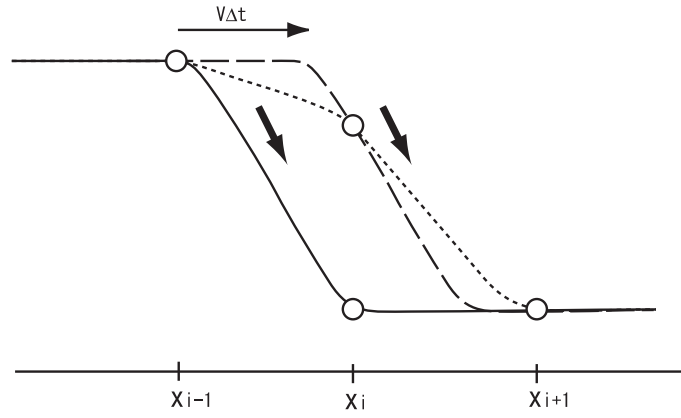


図 3.2: CIP 法の考え方。

3.3.1 CIP 法の基礎

CIP 法の特長は、物理量だけでなく微分量まで考慮にいれたスキームになっているところである。図 3.2 は、CIP 法の考え方を示したものである。いま、初期に実線のようなある物理量のプロファイルがあったとする。このプロファイルが移流方程式、

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} = 0 \quad (3.5)$$

にしたがって、移流速度 $v_x = \text{const} > 0$ で、 Δt の時間、移動するとき、厳密解は破線のようにになる。しかし、データは離散的なので、格子点上でしか情報は得られない。この離散的な情報から、セル内のプロファイルを単純に補間してしまうと、点線のようになり、厳密解からのずれが生じる。そこで、式 (3.5) の微分 ($g = \partial f / \partial x$) を考えてみる。

$$\frac{\partial g}{\partial t} + v_x \frac{\partial g}{\partial x} = 0 \quad (3.6)$$

式 (3.5) と (3.6) は全く同じように移流方程式に従うことがわかる。この微分量も物理量と同様に移流させれば、格子点上では物理量とさらに微分量もわかることになる。もし格子点上でプロファイルの傾き (微分) の情報が加わると、格子点 x_i での傾きが修正され、より厳密解に近づくことがわかる。このように、CIP 法では微分の情報を使うことで計算の精度を向上させている。

具体的には、格子点 x_{i-1} 、 x_i の間を 3 次関数、

$$F(X) = aX^3 + bX^2 + cX + d \quad (3.7)$$

で補間する。ここで $X = x - x_i$ である。格子点上で、物理量 f_{i-1} 、 f_i 、微分量 g_{i-1} 、 g_i が連続であるという条件を課す。

$$F(x_i) = f_i = d \quad (3.8)$$

$$F(x_{i-1}) = f_{i-1} = -a\Delta x^3 + b\Delta x^2 - c\Delta x + d \quad (3.9)$$

$$\frac{dF(x_i)}{dx} = g_i = c \quad (3.10)$$

$$\frac{dF(x_{i-1})}{dx} = g_{i-1} = 3a\Delta x^2 - 2b\Delta x + c \quad (3.11)$$

これらの条件から、係数を求めることができる。

$$a = \frac{g_i + g_{i-1}}{\Delta x^2} - \frac{2(f_i - f_{i-1})}{\Delta x^3} \quad (3.12)$$

$$b = \frac{3(f_{i-1} - f_i)}{\Delta x^2} + \frac{2g_i + g_{i-1}}{\Delta x} \quad (3.13)$$

格子点 x_i での時刻 Δt 後の値 $f_i(\Delta t)$ 、 $g_i(\Delta t)$ は、このプロファイルを x_i から $v_x \Delta t$ だけ遡ればよいから、

$$f_i(\Delta t) = a\xi^3 + b\xi^2 + c\xi + d \quad (3.14)$$

$$g_i(\Delta t) = 3a\xi^2 + 2b\xi + c \quad (3.15)$$

である。ここで、 $\xi = -v_x \Delta t$ である。 $v_x < 0$ の場合は、 $i - 1$ を $i + 1$ に $-\Delta x$ を Δx に置き換える。これは、風上（移流の上流側）を判断して移流させていることになる。

3.3.2 3次元への拡張と流体力学への応用

これまででは、1次元、移流速度一定として考えてきた。つぎは、3次元かつ移流速度も一定でない場合について考えていく。式 (3.1)-(3.3) は一般的に以下のようにかける。

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} + v_z \frac{\partial f}{\partial z} = S \quad (3.16)$$

ここで、 f は ρ 、 $\mathbf{v}$ 、 p をあらわす。 S は非移流項である。さらに、式 (3.16) を x 、 y 、 z で偏微分する。

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{\partial f}{\partial x} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial x} \right) \\ = \frac{\partial S}{\partial x} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial x} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial x} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial x} \end{aligned} \quad (3.17)$$

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{\partial f}{\partial y} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial y} \right) \\ = \frac{\partial S}{\partial y} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial y} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial y} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial y} \end{aligned} \quad (3.18)$$

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{\partial f}{\partial z} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial z} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial z} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial z} \right) \\ = \frac{\partial S}{\partial z} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial z} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial z} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial z} \end{aligned} \quad (3.19)$$

ここで、式 (3.16)、-(3.19) のうち、移流項と非移流項の部分の2段階にわけて計算することにする。すなわち、移流フェイズ、

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} + v_z \frac{\partial f}{\partial z} = 0, \quad (3.20)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial f}{\partial x} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial x} \right) = 0 \quad (3.21)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial f}{\partial y} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial y} \right) = 0 \quad (3.22)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial f}{\partial z} \right) + v_x \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial z} \right) + v_y \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial z} \right) + v_z \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial z} \right) = 0 \quad (3.23)$$

非移流フェイズ、

$$\frac{\partial f}{\partial t} = S \quad (3.24)$$

$$\frac{\partial}{\partial t} \left(\frac{\partial f}{\partial x} \right) = \frac{\partial S}{\partial x} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial x} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial x} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial x} \quad (3.25)$$

$$\frac{\partial f}{\partial t} \left(\frac{\partial f}{\partial x} \right) = \frac{\partial S}{\partial y} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial y} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial y} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial y} \quad (3.26)$$

$$\frac{\partial f}{\partial t} \left(\frac{\partial f}{\partial x} \right) = \frac{\partial S}{\partial z} - \frac{\partial f}{\partial x} \frac{\partial v_x}{\partial z} - \frac{\partial f}{\partial y} \frac{\partial v_y}{\partial z} - \frac{\partial f}{\partial z} \frac{\partial v_z}{\partial z} \quad (3.27)$$

の部分に分け、移流させたあとでその値を使って非移流項を解くという操作をする。(非移流項 → 移流項の順番も可能である)

3.3.3 移流項

以下では、 ρ 、 p の移流についてみていく。(v_x の場合は i を $i + 1/2$ 、 v_y の場合は j を $j + 1/2$ 、 v_z の場合は k を $k + 1/2$ に置き換える)

時刻 $t = n$ (移流前) での値を上付きの添字 n で、時刻 $t = n + a$ (移流後) での値を添字 $n + a$ であらわす。移流の間、速度が一定と考えると、時刻 $t = n + a$ での値は次のように与えられる。

$$f^{n+a}(x_i, y_j, z_k) = f^n(x_i - v_x \Delta t, y_i - v_y \Delta t, z_i - v_z \Delta t) \quad (3.28)$$

$v_x < 0$ 、 $v_y < 0$ 、 $v_z < 0$ のとき $(x_i - v_x \Delta t, y_i - v_y \Delta t, z_i - v_z \Delta t)$ は (x_i, y_j, z_k) と $(x_{i+1}, y_{j+1}, z_{k+1})$ を含むセル内にあるはずである。ここで、時刻 $t = n$ におけるこのセル内の物理量を CIP 法で補間し、 $f^n(x_i - v_x \Delta t, y_i - v_y \Delta t, z_i - v_z \Delta t)$ を決定する。

セル内の物理量は、

$$\begin{aligned} F(X, Y, Z) = & a_1 X^3 + a_2 Y^3 + a_3 Z^3 \\ & + a_4 X^2 Y + a_5 X^2 Z + a_6 Y^2 X + a_7 Y^2 Z + a_8 Z^2 X + a_9 X^2 Y + a_{10} X Y Z \\ & + a_{11} X^2 + a_{12} Y^2 + a_{13} Z^2 + a_{14} X Y + a_{15} X Z + a_{16} Y Z \\ & + a_{17} X + a_{18} Y + a_{19} Z + a_{20} \end{aligned} \quad (3.29)$$

で補間する。ここで、 $X = x - x_i$ 、 $Y = y - y_j$ 、 $Z = z - z_k$ である。式 (3.29) 中の 20 個の未知係数は、次のように決める。

各格子点上には、物理量とその微分量の計 4 つの値が与えられている。このうち、 (i, j, k) とこれに近い位置にある格子点 $(i + 1, j, k)$ 、 $(i, j + 1, k)$ 、 $(i, j, k + 1)$ では、物理量と微分量のすべての連続性を条件とする。一方、 (i, j, k) から遠い位置にある格子点 $(i + 1, j + 1, k)$ 、 $(i + 1, j, k + 1)$ 、 $(i, j + 1, k + 1)$ 、 $(i + 1, j + 1, k + 1)$ では、物理量の連続性だけを条件とする。このようにすると、20 個の条件がそろい係数を決定することができる。以下はその係数である。($f_{i,j,k}^n = f^n(x_i, y_j, z_k)$ のようにあらわすことにする)

$$a_1 = \frac{\partial}{\partial x} (f_{i+1,j,k} + f_{i,j,k}) / \Delta x^2 - 2d_i / \Delta x^3 \quad (3.30)$$

$$a_2 = \frac{\partial}{\partial y} (f_{i,j+1,k} + f_{i,j,k}) / \Delta y^2 - 2d_j / \Delta y^3 \quad (3.31)$$

$$a_3 = \frac{\partial}{\partial z} (f_{i,j,k+1} + f_{i,j,k}) / \Delta z^2 - 2d_k / \Delta z^3 \quad (3.32)$$

$$a_4 = \frac{\partial d_i}{\partial y} / \Delta x^2 - a_{14} / \Delta x \quad (3.33)$$

$$a_5 = \frac{\partial d_i}{\partial z} / \Delta x^2 - a_{15} / \Delta x \quad (3.34)$$

$$a_6 = \frac{\partial d_j}{\partial x} / \Delta y^2 - a_{14} / \Delta y \quad (3.35)$$

$$a_7 = \frac{\partial d_j}{\partial z} / \Delta y^2 - a_{16} / \Delta y \quad (3.36)$$

$$a_8 = \frac{\partial d_k}{\partial x} / \Delta z^2 - a_{15} / \Delta z \quad (3.37)$$

$$a_9 = \frac{\partial d_k}{\partial y} / \Delta z^2 - a_{16} / \Delta z \quad (3.38)$$

$$\begin{aligned} a_{10} = & \frac{1}{\Delta x \Delta y \Delta z} [f_{i+1,j+1,k+1} \\ & - a_1 \Delta x^3 - a_2 \Delta y^3 - a_3 \Delta z^3 - a_4 \Delta x^2 \Delta y - a_5 \Delta x^2 \Delta z \\ & - a_6 \Delta y^2 \Delta x - a_7 \Delta y^2 \Delta z - a_8 \Delta z^2 \Delta x - a_9 \Delta z^2 \Delta y \\ & - a_{11} \Delta x^2 - a_{12} \Delta y^2 - a_{13} \Delta z^2 - a_{14} \Delta x \Delta y - a_{15} \Delta x \Delta z - a_{16} \Delta y \Delta z \\ & - a_{17} \Delta x - a_{18} \Delta y - a_{19} \Delta z - a_{20}] \end{aligned} \quad (3.39)$$

$$a_{11} = 3d_i / \Delta x^2 - \frac{\partial}{\partial x} (f_{i+1,j,k} + 2f_{i,j,k}) / \Delta x \quad (3.40)$$

$$a_{12} = 3d_j / \Delta y^2 - \frac{\partial}{\partial y} (f_{i,j+1,k} + 2f_{i,j,k}) / \Delta y \quad (3.41)$$

$$a_{13} = 3d_k / \Delta z^2 - \frac{\partial}{\partial z} (f_{i,j,k+1} + 2f_{i,j,k}) / \Delta z \quad (3.42)$$

$$\begin{aligned} a_{14} = & - (f_{i+1,j+1,k} - f_{i+1,j,k} - f_{i,j+1,k} + f_{i,j,k}) / \Delta x \Delta y \\ & + \frac{\partial d_j}{\partial x} / \Delta y + \frac{\partial d_i}{\partial y} / \Delta x \end{aligned} \quad (3.43)$$

$$\begin{aligned} a_{15} = & - (f_{i+1,j,k+1} - f_{i+1,j,k} - f_{i,j,k+1} + f_{i,j,k}) / \Delta x \Delta z \\ & + \frac{\partial d_k}{\partial x} / \Delta z + \frac{\partial d_i}{\partial z} / \Delta x \end{aligned} \quad (3.44)$$

$$\begin{aligned} a_{16} = & - (f_{i,j+1,k+1} - f_{i,j+1,k} - f_{i,j,k+1} + f_{i,j,k}) / \Delta y \Delta z \\ & + \frac{\partial d_j}{\partial z} / \Delta y + \frac{\partial d_k}{\partial y} / \Delta z \end{aligned} \quad (3.45)$$

$$a_{17} = \frac{\partial}{\partial x} f_{i,j,k} \quad (3.46)$$

$$a_{18} = \frac{\partial}{\partial y} f_{i,j,k} \quad (3.47)$$

$$a_{19} = \frac{\partial}{\partial z} f_{i,j,k} \quad (3.48)$$

$$a_{20} = f_{i,j,k} \quad (3.49)$$

$$d_i = f_{i+1,j,k} - f_{i,j,k}, \quad d_j = f_{i,j+1,k} - f_{i,j,k}, \quad d_k = f_{i,j,k+1} - f_{i,j,k} \quad (3.50)$$

$v_x > 0$ のときは、 $i+1 \Rightarrow i-1$ 、 $\Delta x \Rightarrow -\Delta x$ 、 $v_y > 0$ のときは、 $j+1 \Rightarrow j-1$ 、 $\Delta y \Rightarrow -\Delta y$ 、 $v_z > 0$ のときは、 $k+1 \Rightarrow k-1$ 、 $\Delta z \Rightarrow -\Delta z$ としなければならない。

このように、セル内の物理量を補間することができたら、時刻 $t = n + a$ での値は次のように求められる。

$$\begin{aligned} f^{n+a} = & a_1 \xi^3 + a_2 \eta^3 + a_3 \zeta^3 \\ & + a_4 \xi^2 \eta + a_5 \xi^2 \zeta + a_6 \eta^2 \xi + a_7 \eta^2 \zeta + a_8 \zeta^2 \xi + a_9 \zeta^2 \eta + a_{10} \xi \eta \zeta \\ & + a_{11} \xi^2 + a_{12} \eta^2 + a_{13} \zeta^2 + a_{14} \xi \eta + a_{15} \xi \zeta + a_{16} \eta \zeta \\ & + a_{17} \xi + a_{18} \eta + a_{19} \zeta + a_{20} \end{aligned} \quad (3.51)$$

$$\begin{aligned} \frac{\partial f^{n+a}}{\partial x} = & 3a_1 \xi^2 + 2a_4 \xi \eta + 2a_5 \xi \zeta + a_6 \eta^2 + a_8 \zeta^2 \\ & + a_{10} \eta \zeta + 2a_{11} \xi + a_{14} \eta + a_{15} \zeta + a_{17} \end{aligned} \quad (3.52)$$

$$\begin{aligned} \frac{\partial f^{n+a}}{\partial y} = & 3a_2 \eta^2 + a_4 \xi^2 + 2a_6 \eta \xi + 2a_7 \eta \zeta + a_9 \zeta^2 \\ & + a_{10} \xi \zeta + 2a_{12} \eta + a_{14} \xi + a_{16} \zeta + a_{18} \end{aligned} \quad (3.53)$$

$$\begin{aligned} \frac{\partial f^{n+a}}{\partial z} = & 3a_3 \zeta^2 + a_5 \xi^2 + a_7 \eta^2 + 2a_8 \xi \zeta + 2a_9 \zeta \eta \\ & + a_{10} \xi \eta + 2a_{13} \zeta + a_{15} \xi + a_{16} \eta + a_{19} \end{aligned} \quad (3.54)$$

ここで、 $\xi = -v_x \Delta t$ 、 $\eta = -v_y \Delta t$ 、 $\zeta = -v_z \Delta t$ である。

3.3.4 非移流項

非移流項に関しては中心差分を用いて計算する。時刻 $t = n + a$ から $t = n + 1$ へは、次のようにして進められる。

$$f_{i,j,k}^{n+1} = f_{i,j,k}^{n+a} + S_{i,j,k} \Delta t \quad (3.55)$$

$$\begin{aligned} \left(\frac{\partial f}{\partial x} \right)_{i,j,k}^{n+1} = & \left(\frac{\partial f}{\partial x} \right)_{i,j,k}^{n+a} + \frac{\left(f_{i+1,j,k}^{n+1} - f_{i-1,j,k}^{n+1} - f_{i+1,j,k}^{n+a} + f_{i-1,j,k}^{n+a} \right)}{2\Delta x} \\ & - \left(\frac{\partial f}{\partial x} \right)_{i,j,k}^{n+a} \frac{\left(v_{x \ i+1/2,j,k}^{n+a} - v_{x \ i-1/2,j,k}^{n+a} \right) \Delta t}{\Delta x} \\ & - \left(\frac{\partial f}{\partial y} \right)_{i,j,k}^{n+a} \frac{\left(v_{y \ i+1/2,j,k}^{n+a} - v_{y \ i-1/2,j,k}^{n+a} \right) \Delta t}{\Delta x} \end{aligned} \quad (3.56)$$

$$\begin{aligned}
& - \left(\frac{\partial f}{\partial z} \right)_{i,j,k}^{n+a} \frac{\left(v_{x \ i+1/2,j,k}^{n+a} - v_{x \ i-1/2,j,k}^{n+a} \right) \Delta t}{\Delta x} \\
\left(\frac{\partial f}{\partial y} \right)_{i,j,k}^{n+1} &= \left(\frac{\partial f}{\partial y} \right)_{i,j,k}^{n+a} + \frac{\left(f_{i,j+1,k}^{n+1} - f_{i,j-1,k}^{n+1} - f_{i,j+1,k}^{n+a} + f_{i,j-1,k}^{n+a} \right)}{2\Delta y} \\
& - \left(\frac{\partial f}{\partial x} \right)_{i,j,k}^{n+a} \frac{\left(v_{x \ i,j+1/2,k}^{n+a} - v_{x \ i,j-1/2,k}^{n+a} \right) \Delta t}{\Delta y} \\
& - \left(\frac{\partial f}{\partial y} \right)_{i,j,k}^{n+a} \frac{\left(v_{y \ i,j+1/2,k}^{n+a} - v_{y \ i,j-1/2,k}^{n+a} \right) \Delta t}{\Delta y} \\
& - \left(\frac{\partial f}{\partial z} \right)_{i,j,k}^{n+a} \frac{\left(v_{z \ i,j+1/2,k}^{n+a} - v_{z \ i,j-1/2,k}^{n+a} \right) \Delta t}{\Delta y}
\end{aligned} \tag{3.57}$$

$$\begin{aligned}
\left(\frac{\partial f}{\partial z} \right)_{i,j,k}^{n+1} &= \left(\frac{\partial f}{\partial z} \right)_{i,j,k}^{n+a} + \frac{\left(f_{i,j,k+1}^{n+1} - f_{i,j,k-1}^{n+1} - f_{i,j,k+1}^{n+a} + f_{i,j,k-1}^{n+a} \right)}{2\Delta z} \\
& - \left(\frac{\partial f}{\partial x} \right)_{i,j,k}^{n+a} \frac{\left(v_{x \ i,j,k+1/2}^{n+a} - v_{x \ i,j,k-1/2}^{n+a} \right) \Delta t}{\Delta z} \\
& - \left(\frac{\partial f}{\partial y} \right)_{i,j,k}^{n+a} \frac{\left(v_{y \ i,j,k+1/2}^{n+a} - v_{y \ i,j,k-1/2}^{n+a} \right) \Delta t}{\Delta z} \\
& - \left(\frac{\partial f}{\partial z} \right)_{i,j,k}^{n+a} \frac{\left(v_{z \ i,j,k+1/2}^{n+a} - v_{z \ i,j,k-1/2}^{n+a} \right) \Delta t}{\Delta z}
\end{aligned} \tag{3.58}$$

ここで、微分量の非移流項にでてくる $S_{i,j,k}$ (の微分) は、式 (3.55) から

$$S_{i,j,k} = \left(f_{i,j,k}^{n+1} - f_{i,j,k}^{n+a} \right) / \Delta t \tag{3.59}$$

で置き換えられている。また、 $v_{x \ i,j+1/2,k}^{n+a}$ のように、本来その格子上では定義されていない値は、つぎのように平均を使った。

$$v_{x \ i,j+1/2,k}^{n+a} = \frac{v_{x \ i-1/2,j,k}^{n+a} + v_{x \ i+1/2,j,k}^{n+a} + v_{x \ i-1/2,j+1,k}^{n+a} + v_{x \ i+1/2,j+1,k}^{n+a}}{4} \tag{3.60}$$

3 次元 Cartesian のときの ρ 、 v_x 、 p の差分式を具体的に書くと、

$$\rho_{i,j,k}^{n+1} = \rho_{i,j,k}^{n+a} + S_{\rho \ i,j,k} \Delta t \tag{3.61}$$

$$\begin{aligned}
S_{\rho \ i,j,k} &= -\rho_{i,j,k}^{n+a} \left(\frac{v_{x \ i+1/2,j,k}^{n+a} - v_{x \ i-1/2,j,k}^{n+a}}{\Delta x} \right. \\
&\quad \left. + \frac{v_{y \ i,j+1/2,k}^{n+a} - v_{y \ i,j-1/2,k}^{n+a}}{\Delta y} + \frac{v_{z \ i,j,k+1/2}^{n+a} - v_{z \ i,j,k-1/2}^{n+a}}{\Delta z} \right)
\end{aligned} \tag{3.62}$$

$$p_{i,j,k}^{n+1} = p_{i,j,k}^{n+a} + S_{p\ i,j,k} \Delta t \quad (3.63)$$

$$S_{p\ i,j,k} = -\gamma p_{i,j,k}^{n+a} \left(\frac{v_{x\ i+1/2,j,k}^{n+a} - v_{x\ i-1/2,j,k}^{n+a}}{\Delta x} + \frac{v_{y\ i,j+1/2,k}^{n+a} - v_{y\ i,j-1/2,k}^{n+a}}{\Delta y} + \frac{v_{z\ i,j,k+1/2}^{n+a} - v_{z\ i,j,k-1/2}^{n+a}}{\Delta z} \right) \quad (3.64)$$

$$v_{x\ i+1/2,j,k}^{n+1} = v_{x\ i+1/2,j,k}^{n+a} + S_{v_x\ i+1/2,j,k} \Delta t \quad (3.65)$$

$$S_{v_x\ i+1/2,j,k} = g_{x\ i+1/2} - \frac{1}{\rho_{i+1/2,j,k}^{n+a}} \frac{p_{i+1,j,k}^{n+a} - p_{i,j,k}^{n+a}}{\Delta x} - \frac{B_{y\ i+1/2,j,k}}{4\pi\rho_{i+1/2,j,k}^{n+a}} \frac{B_{y\ i+1,j,k} - B_{y\ i,j,k}}{\Delta x} - \frac{B_{z\ i+1/2,j,k}}{4\pi\rho_{i+1/2,j,k}^{n+a}} \frac{B_{z\ i+1,j,k} - B_{z\ i,j,k}}{\Delta x} + \frac{B_{y\ i+1/2,j,k}}{4\pi\rho_{i+1/2,j,k}^{n+a}} \frac{B_{x\ i+1/2,j+1/2,k}^s - B_{x\ i+1/2,j-1/2,k}^s}{\Delta y} + \frac{B_{z\ i+1/2,j,k}}{4\pi\rho_{i+1/2,j,k}^{n+a}} \frac{B_{x\ i+1/2,j,k+1/2}^s - B_{x\ i+1/2,j,k-1/2}^s}{\Delta z} \quad (3.66)$$

となる。ここで、 $\rho_{i+1/2,j,k}^{n+a} = (\rho_{i+1,j,k}^{n+a} + \rho_{i,j,k}^{n+a})/2$ 、 $B_{y\ i,j,k} = (B_{y\ i,j+1/2,k} + B_{y\ i,j-1/2,k})/2$ 、 $B_{y\ i+1/2,j,k} = (B_{y\ i+1,j,k} + B_{y\ i,j,k})/2$ 、 $B_{z\ i,j,k} = (B_{z\ i,j,k+1/2} + B_{z\ i,j,k-1/2})/2$ 、 $B_{z\ i+1/2,j,k} = (B_{z\ i+1,j,k} + B_{z\ i,j,k})/2$ である。磁気張力項の上付きの添字 s はMOC法で求められる値を使う(後述)。上付きの添字のない磁場の成分は、時刻 $t = n$ のものを使う。

3.4 CT法

CT法は、数値計算で解く方程式には現われてこない $\nabla \cdot \mathbf{B} = 0$ を保証する。磁場はセルの面の中心で、電場はセルの辺の中心で定義し(図3.1)、誘導方程式を中心差分で考えると、

$$\begin{aligned} \frac{1}{\Delta t} (B_{x\ i+1/2,j,k}^{n+1} - B_{x\ i+1/2,j,k}^n) &= -\frac{1}{\Delta y} (E_{z\ i+1/2,j+1/2,k} - E_{z\ i+1/2,j-1/2,k}) \\ &\quad + \frac{1}{\Delta z} (E_{y\ i+1/2,j,k+1/2} - E_{y\ i+1/2,j,k-1/2}) \end{aligned} \quad (3.67)$$

$$\begin{aligned} \frac{1}{\Delta t} (B_{y\ i,j+1/2,k}^{n+1} - B_{y\ i,j+1/2,k}^n) &= -\frac{1}{\Delta z} (E_{x\ i,j+1/2,k+1/2} - E_{x\ i,j+1/2,k-1/2}) \\ &\quad + \frac{1}{\Delta x} (E_{z\ i+1/2,j+1/2,k} - E_{z\ i-1/2,j+1/2,k}) \end{aligned} \quad (3.68)$$

$$\begin{aligned} \frac{1}{\Delta t} (B_{z\ i,j,k+1/2}^{n+1} - B_{z\ i,j,k+1/2}^n) &= -\frac{1}{\Delta x} (E_{y\ i+1/2,j,k+1/2} - E_{y\ i-1/2,j,k+1/2}) \\ &\quad + \frac{1}{\Delta y} (E_{x\ i,j+1/2,k+1/2} - E_{x\ i,j-1/2,k+1/2}) \end{aligned} \quad (3.69)$$

となる。ここで、式 (3.67)-(3.69) を足しあわせてみると、

$$\begin{aligned} & \frac{B_{x\ i+1/2,j,k}^{n+1} - B_{x\ i-1/2,j,k}^{n+1}}{\Delta x} + \frac{B_{y\ i,j+1/2,k}^{n+1} - B_{y\ i,j-1/2,k}^{n+1}}{\Delta y} + \frac{B_{z\ i,j,k+1/2}^{n+1} - B_{z\ i,j,k-1/2}^{n+1}}{\Delta z} \\ &= \frac{B_{x\ i+1/2,j,k}^n - B_{x\ i-1/2,j,k}^n}{\Delta x} + \frac{B_{y\ i,j+1/2,k}^n - B_{y\ i,j-1/2,k}^n}{\Delta y} + \frac{B_{z\ i,j,k+1/2}^n - B_{z\ i,j,k-1/2}^n}{\Delta z} \end{aligned} \quad (3.70)$$

が導かれる。これにより、初期に $\nabla \cdot \mathbf{B} = 0$ が満たされていれば、その後も数値誤差の範囲内で満たされることがわかる。したがって、格子を図 3.1 のように定義してさえいれば、 $\nabla \cdot \mathbf{B} = 0$ は自動的に満たされることになる。

3.5 MOC 法

3.5.1 誘導方程式

CT 法では、式 (3.67)-(3.69) の電場をどのように求めるかはなにも制限されていない。Stone & Norman (1992) は、特性曲線を用いて電場を求める方法を考案した。

ここからは、簡単のため 1.5 次元 ($\partial/\partial y = \partial/\partial z = 0$)、 $B_z = 0$ の場合を考える。1.5 次元での物理量が定義されている点は図 3.3 のようになる。このとき E_z は次のように書ける。

$$E_z = - (v_x B_y^* - v_y^* B_x) \quad (3.71)$$

ここで、上付きの添字 * は MOC 法で求められる値である。運動方程式、誘導方程式について、アルフベン波の特性曲線を使って B_y^* 、 v_y^* を見積もる方法を考える。誘導方程式の y 成分は、

$$\frac{\partial B_y}{\partial t} = B_x \frac{\partial v_y}{\partial x} - \frac{\partial}{\partial x} (v_x B_y) \quad (3.72)$$

のように書ける。一方、運動方程式の y 成分については、非圧縮を考慮すると ($\rho, v_x, B_x = \text{const}$)、

$$\frac{\partial v_y}{\partial t} = \frac{B_x}{4\pi\rho} \frac{\partial B_y}{\partial x} - \frac{\partial}{\partial x} (v_x v_y) \quad (3.73)$$

のようになる。式 (3.72) に $(4\pi\rho)^{-1/2}$ を掛けたものと、式 (3.73) の和と差をとると、

$$\left[\frac{\partial}{\partial t} + \left(v_x + \frac{B_x}{\sqrt{4\pi\rho}} \right) \frac{\partial}{\partial x} \right] \left(v_y - \frac{B_y}{\sqrt{4\pi\rho}} \right) = 0 \quad (3.74)$$

$$\left[\frac{\partial}{\partial t} + \left(v_x - \frac{B_x}{\sqrt{4\pi\rho}} \right) \frac{\partial}{\partial x} \right] \left(v_y + \frac{B_y}{\sqrt{4\pi\rho}} \right) = 0 \quad (3.75)$$

式 (3.74)、(3.75) は、アルフベン波の特性方程式である。この特性方程式のリーマン不変量を使って次の関係が導かれる。

$$(v_y^* - v_y^+) - \frac{1}{\sqrt{4\pi\rho^+}} (B_y^* - B_y^+) = 0 \quad (3.76)$$

$$(v_y^* - v_y^-) + \frac{1}{\sqrt{4\pi\rho^-}} (B_y^* - B_y^-) = 0 \quad (3.77)$$

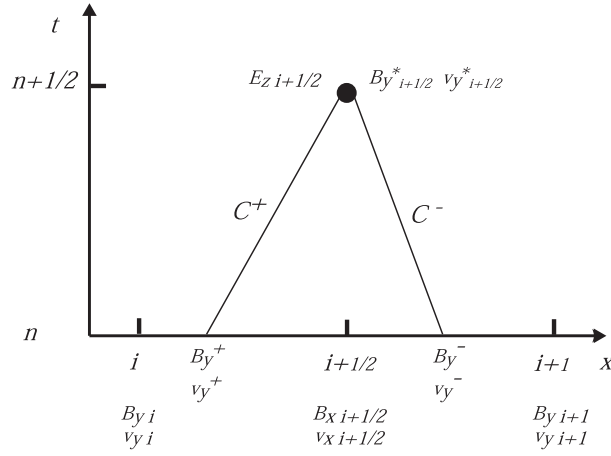


図 3.3: 1.5 次元での MOC 法。

ここで、添字 ± は、対応する特性曲線上の時刻 $t = n$ での値である式 (3.76)、(3.77) を v_y^* 、 B_y^* について解くと、

$$v_y^* = \frac{v_y^+ \sqrt{4\pi\rho^+} + v_y^- \sqrt{4\pi\rho^-} - B_y^+ + B_y^-}{\sqrt{4\pi\rho^+} + \sqrt{4\pi\rho^-}} \quad (3.78)$$

$$B_y^* = \frac{-v_y^+ + v_y^- + B_y^+ / \sqrt{4\pi\rho^+} + B_y^- / \sqrt{4\pi\rho^-}}{1/\sqrt{4\pi\rho^+} + 1/\sqrt{4\pi\rho^-}} \quad (3.79)$$

が得られる。ここでは、簡単のため $\rho^+ = \rho_i$ 、 $\rho^- = \rho_{i+1}$ とする。 $v_y^\pm$ 、 $B_y^\pm$ は、van Leer 法や CIP 法による補間などが使える。CIP 法に関しては、すでに述べた通りである。また、テスト計算において、単純に CIP 法を用いたコードでは数値振動がでたことなどから、まだまだ改良が必要であり、注意が必要である (表 3.4、図 3.10)。ここでは、安定に解くことができる van Leer 法を解説する。

van Leer 法 van Leer 法では、 $i - 1/2$ と $i + 1/2$ を 1 次補間する。

$$v_y^+ = \begin{cases} v_{y i}^n + \frac{1}{2} \left(\Delta x - C_{i+1/2}^+ \Delta t \right) \frac{dv_y}{dx}_i & \text{if } C_{i+1/2}^+ \geq 0 \\ v_{y i+1}^n - \frac{1}{2} \left(\Delta x + C_{i+1/2}^+ \Delta t \right) \frac{dv_y}{dx}_{i+1} & \text{if } C_{i+1/2}^+ < 0 \end{cases} \quad (3.80)$$

ここで、 $C_{i+1/2}^+ = (v_x + B_x / \sqrt{4\pi\rho})_{i+1/2}$ はアルフベン波の特性速度、

$$\frac{dv_y}{dx}_i = \begin{cases} \frac{2\Delta v_{y i-1/2} \Delta v_{y i+1/2}}{\Delta v_{y i-1/2} + \Delta v_{y i+1/2}} & \text{if } \Delta v_{y i-1/2} \Delta v_{y i+1/2} \geq 0 \\ 0 & \text{if } \Delta v_{y i-1/2} \Delta v_{y i+1/2} < 0 \end{cases} \quad (3.81)$$

$\Delta v_{y i+1/2} = (v_{y i+1} - v_{y i}) / \Delta x$ である。また、 $C_{i+1/2}^+$ を求めるときの移流後に v_x^{n+a} を用いた。これは、磁気流体衝撃波管を解くときにより良い解を得るためである。 v_y^+ と同様にして、 v_y^- 、 B_y^+ 、 B_y^- も求めることができる。

3次元への拡張 基本的には、1.5次元のときと変わらない。3次元での電場は、

$$E_x = -(v_y^* B_z^* - v_z^* B_y^*) \quad (3.82)$$

$$E_y = -(v_z^* B_x^* - v_x^* B_z^*) \quad (3.83)$$

$$E_z = -(v_x^* B_y^* - v_y^* B_x^*) \quad (3.84)$$

のようにすべての項に添字 * がついているが、 v_y^* と同様の手順で求めることができる。

3.5.2 磁気張力項

MOC 法は、運動方程式中の磁気張力項の計算にも使う。たとえば、 v_y の運動方程式中の磁気張力項を取りだして考えてみる。

$$\frac{v_y^{n+1} - v_y^{n+a}}{\Delta t} = -\frac{B_x}{4\pi\rho} \frac{B_y^s{}_{i+1/2} - B_y^s{}_{i-1/2}}{\Delta x} \quad (3.85)$$

ここでは、 B_y^s が MOC 法で求められる値である。磁気張力項を計算するときは、 v_y^* を求めたときの特性速度 $C^\pm = v_x \pm B_x/\sqrt{4\pi\rho}$ の代わりに $C^\pm = \pm B_x/\sqrt{4\pi\rho}$ を用いる。それ以外は、 v_y^* を求めたときと同様である。これは、移流に関する部分はすでに CIP 法で考慮されているためである。

3.6 人工粘性

圧縮性流体の計算では、超音速になると衝撃波が発生する。衝撃波は数学的には不連続面であるが、実際は流体に粘性があるため、衝撃波は分子の平均自由行程程度の厚さを持っている。数値解析では、衝撃波の厚さを格子幅 Δx 程度で捉えることができればよいので、衝撃波のみに作用する人工的な粘性を圧力項に加えることで衝撃波を捉えることが可能になる。ここでは、von-Neumann 型と、Ogata & Yabe (1999) の人工粘性を紹介する。

3.6.1 von-Neumann 型人工粘性

von-Neumann 型人工粘性は、ランキン - ユゴニオ関係を基礎にしている。

$$\Delta p = \frac{\gamma + 1}{4} \rho_0 (\Delta v)^2 + \rho_0 |\Delta v| \left[\left(\frac{(\gamma + 1)^2}{4} (\Delta v)^2 + C_s^2 \right) \right]^{1/2} \quad (3.86)$$

$$\Delta v = v_1 - v_0 \quad (3.87)$$

ここで、 Δp と Δv は衝撃波面前後での圧力、速度の差、 C_s は音速である。添え字 0、1 はそれぞれ、衝撃波前面、後面での物理量をあらわしている。ランキン - ユゴニオ関係を満たすために、 Δp を数値粘性として圧力に加える。 $C_s \gg v$ 、 $C_s \ll v$ のときを考えると、式 (3.86) から粘性項 q は、以下のようになる。

$$q = \begin{cases} \alpha \left[-\rho C_s \Delta v + \frac{\gamma+1}{2} \rho (\Delta v)^2 \right] & \text{if } \Delta v \leq 0 \\ 0 & \text{if } \Delta v > 0 \end{cases} \quad (3.88)$$

ここで、 α は無次元パラメータで、 $\alpha = 1$ が定義から考えられる値であるが、一般に 0.6 から 1.0 が用いられる。

これを3次元に拡張するときは、各方向において粘性項 q_x 、 q_y 、 q_z をそれぞれ Δv_x 、 Δv_y 、 Δv_z から計算し、運動方程式中の圧力項、エネルギー式に加える。

$$\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + v_z \frac{\partial v_x}{\partial z} = -\frac{1}{\rho} \frac{\partial}{\partial x} (p + q_x) \quad (3.89)$$

$$\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + v_z \frac{\partial v_y}{\partial z} = -\frac{1}{\rho} \frac{\partial}{\partial y} (p + q_y) \quad (3.90)$$

$$\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} = -\frac{1}{\rho} \frac{\partial}{\partial z} (p + q_z) \quad (3.91)$$

$$\frac{\partial p}{\partial t} + (\mathbf{v} \cdot \nabla) p = -\gamma p \nabla \cdot \mathbf{v} - (\gamma - 1) \left[q_x \frac{\partial v_x}{\partial x} + q_y \frac{\partial v_y}{\partial y} + q_z \frac{\partial v_z}{\partial z} \right] \quad (3.92)$$

この方法では、人工粘性は方向に依存してしまう。例えば点源爆発の場合、 $x(y, z)$ 軸方向とそうでない方向への衝撃波の伝播速度が変わってくる。(図 3.8、図 3.9)

3.6.2 Ogata & Yabe (1999) の人工粘性

Ogata & Yabe 1999 では、von-Neumann 型人工粘性における Δv_x 、 Δv_y 、 Δv_z を $(\nabla \cdot \mathbf{v}) \lambda$ で置き換えることで、方向に対する依存性を軽減した。

$$q_x = q_y = q_z = \alpha \left[-\rho C_s (\nabla \cdot \mathbf{v}) \lambda + \frac{\gamma + 1}{2} (\nabla \cdot \mathbf{v})^2 \lambda^2 \right] \text{ if } \nabla \cdot \mathbf{v} < 0 \quad (3.93)$$

$$q_x = q_y = q_z = 0 \quad \text{if } \nabla \cdot \mathbf{v} > 0 \quad (3.94)$$

ここで、 λ は衝撃波の厚さであり、 $\lambda = \Delta x = \Delta y = \Delta z$ とすればよい。 α は、von-Neumann 型人工粘性の項でのべたものと同様なもので、数値粘性の効果を差し引いて、 $\alpha = 0.75$ くらいが CIP には適当なようである (Ogata & Yabe 1991)。

3.7 まとめ

3.7.1 計算の流れ

CIP-MOCCT 法での磁気流体方程式を解く手順を以下にまとめておく。(図 3.4)

1. 密度、速度、圧力を移流させる $\rho^n, \mathbf{v}^n, p^n \Rightarrow \rho^{n+a}, \mathbf{v}^{n+a}, p^{n+a}$
2. 人工粘性項を計算する $\rho^{n+a}, \mathbf{v}^{n+a}, p^{n+a} \Rightarrow q$
3. 密度、圧力の非移流項を計算 $\rho^{n+a}, p^{n+a}, q \Rightarrow \rho^{n+1}, p^{n+1}$
4. 非移流項の磁気張力項を計算 $\rho^{n+a}, \mathbf{B}^n \Rightarrow \mathbf{B}^s$
5. 速度の非移流項を計算 $\rho^{n+a}, \mathbf{v}^{n+a}, p^{n+a}, \mathbf{B}^n, \mathbf{B}^s, q \Rightarrow \mathbf{v}^{n+1}$
6. 電場を計算 $\rho^{n+a}, \mathbf{v}^{n+a}, \mathbf{B}^{n+a} \Rightarrow \mathbf{v}^*, \mathbf{B}^*$
7. 磁場の時間発展 $\mathbf{v}^*, \mathbf{B}^* \Rightarrow \mathbf{B}^{n+1}$

以上くりかえし。

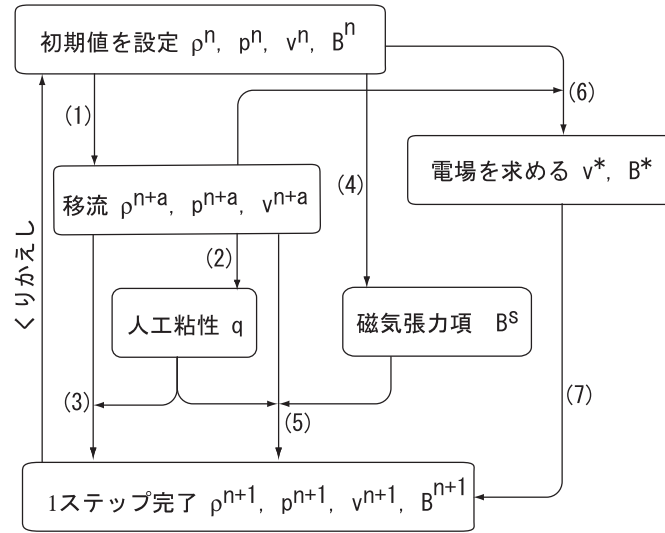


図 3.4: CIP-MOCCT 法のフローチャート。

3.7.2 テスト計算

流体衝撃波管問題 Sod (1978) の衝撃波管問題を解いた。グリッド数は 200、 $\Delta x = 0.005$ 、比熱比 $\gamma = 1.4$ でおこなった。初期に $x = 0.5$ に不連続がある。その他の初期条件は、表 3.1 にまとめている。図 3.5 は時刻 $t = 0.2$ での密度、圧力、速度のプロファイルである。破線は厳密解。

アルフベン波の伝播 グリッド数は 200、 $\Delta x = 0.005$ 。 $\rho = 1, p = 1, v_x = 0, v_y = 0, B_x/\sqrt{4\pi} = 1, B_y = 0$ の一様な場に、 $0.4 < x < 0.6$ の領域に $v_y = 0.001$ の摂動を与える。図 3.6 は時刻 $t = 0.2$ のときの $v_y, B_y/\sqrt{4\pi}$ のプロファイルである。点線は厳密解。

磁気流体衝撃波管問題 Brio & Wu (1984) の磁気流体衝撃波管問題を解いた。グリッド数は 400、 $\Delta x = 0.0025$ 、比熱比は $\gamma = 2$ である。その他の初期値は、表 3.2 にまとめている。図 3.7 は $t = 0.1$ でのプロファイル、表 3.3 はこのときの各場所での物理量の値である。

点源爆発 人工粘性の影響をみるため、2次元の点源爆発問題のシミュレーションを行なった。初期条件は、中心からの距離 $R < 0.08$ では、 $\rho = 1.0, p = 3.0$ 、 $R > 0.08$ では $\rho = 0.1, p = 0.1$ とし、磁場は考えていない。図 3.8、図 3.9 をみると、斜め方向の衝撃波面の位置が改善されていることがわかる。

MOC 法における補間法の比較 Brio & Wu (1984) の磁気流体衝撃波管問題を、MOC 法における補間法を変えて実行した。電場と磁気張力項の 2 カ所に補間する部分があるので、van Leer 法と CIP 法の 2 種類とあわせて 4 通りである。補間法とモデルは表 3.4 にまとめている。図 3.10 は密度のプロファイルを示している。

質量保存 上記2つの衝撃波管問題における、質量保存。全質量を時間の関数として次のように定義する。

$$M(t) = \int_0^1 \rho(t, x) dx \quad (3.95)$$

図3.11は、各時間での全質量を時刻 $t = 0$ での全質量でわったものをあらわしている。

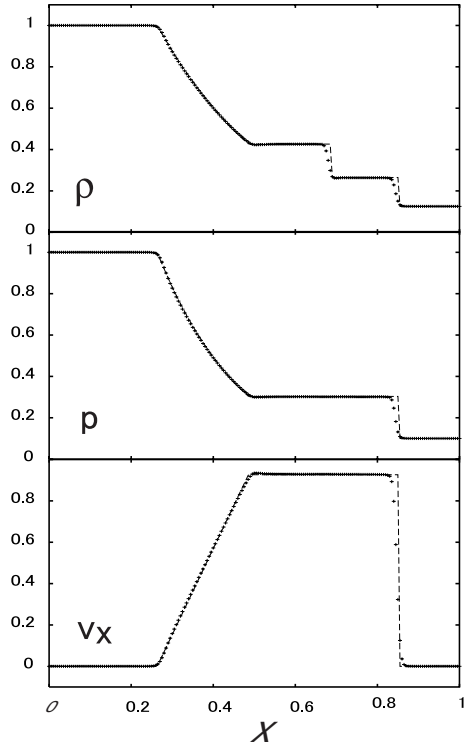


図 3.5: Sod (1978) の衝撃波管問題。

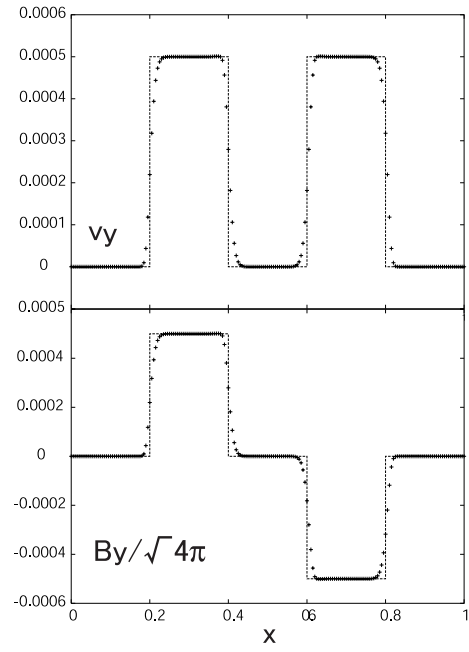


図 3.6: アルフベン波の伝搬。

	ρ	p	v_x
左	1	1	0
右	0.125	0.1	0

表 3.1: Sod (1978) の衝撃波管問題の初期値。

	ρ	p	v_x	v_y	$Bx/\sqrt{4\pi}$	$By/\sqrt{4\pi}$
左	1	1	0	0	0.75	1
右	0.125	0.1	0	0	0.75	-1

表 3.2: Brio & Wu (1984) の磁気流体衝撃波管問題の初期値。

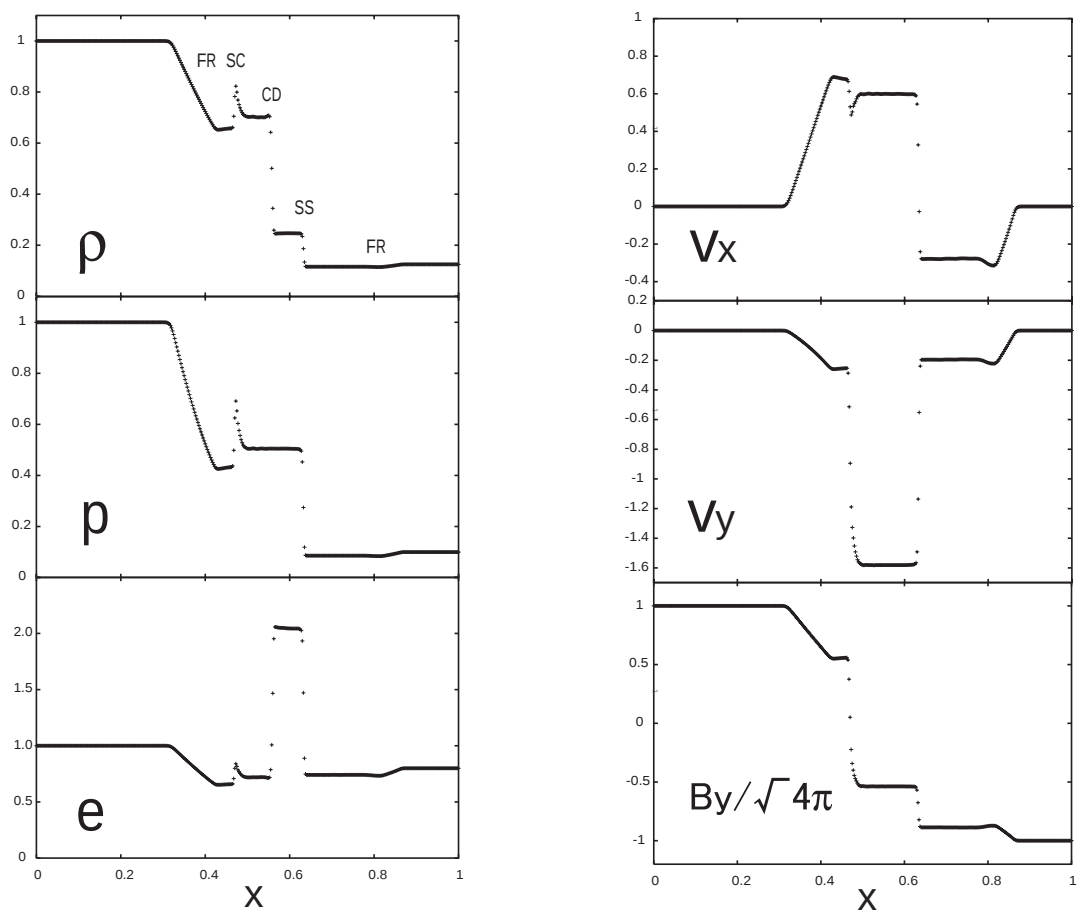


図 3.7: Brio & Wu (1984) の磁気流体衝撃波管問題。

Variable	Left	FR	SC	CD _l	CD _r	FR	Right
ρ	1.0	0.662	0.823	0.703	0.246	0.116	0.125
p	1.0	0.438	0.691	0.505	0.505	0.087	0.1
e	1.0	0.661	0.804	0.719	2.06	0.741	0.8
v_x	0	0.665	0.487	0.600	0.599	0.-279	0
v_y	0	-0.287	-0.119	-0.158	-0.158	-0.1972	0
$B_y/\sqrt{4\pi}$	1.0	0.539	-0.2231	-0.538	-0.537	-0.886	-1.0

表 3.3: Brio & Wu (1984) の磁気流体衝撃波での、各物理量の値。

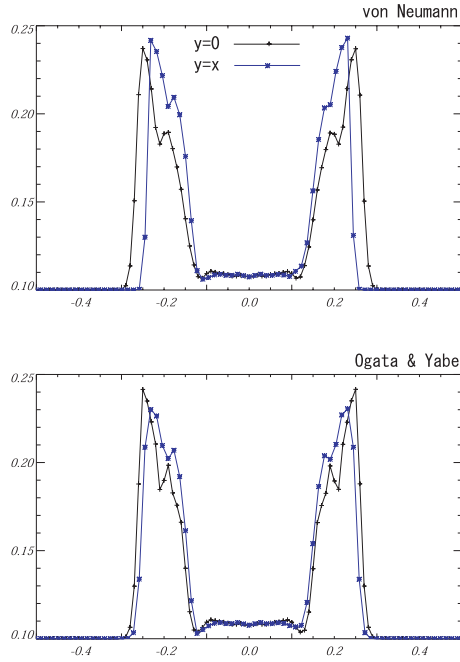


図 3.8: 点源爆発での人工粘性の比較。密度のプロファイル。 $y = 0$ 上の密度 (黒) と $y = x$ 上の密度 (青)。上が von Neumann 型粘性、下が Ogata & Yabe の粘性。

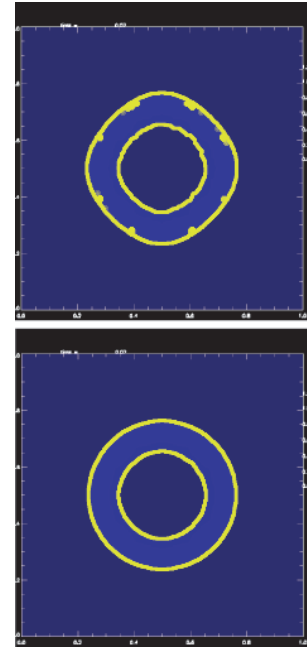


図 3.9: 点源爆発での人工粘性の比較。圧力の等値線。上が von Neumann 型粘性、下が Ogata & Yabe の粘性。

モデル	電場	磁気張力項
Kudoh (1997)	van Leer	van Leer
ct	CIP	van Leer
st	van Leer	CIP
all	CIP	CIP

表 3.4: MOC 法における、補間法の組み合わせ。図 3.10 参照

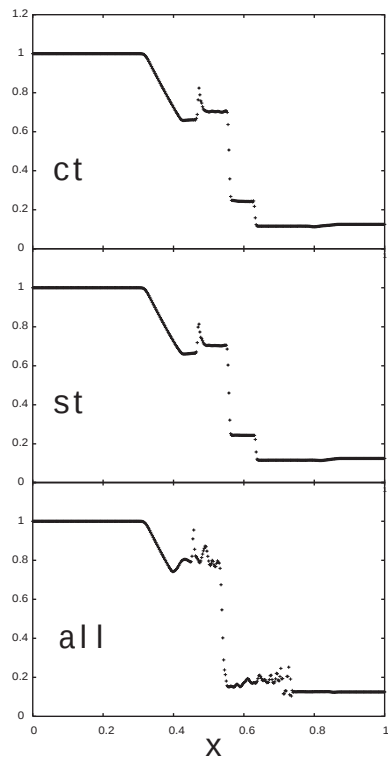


図 3.10: MOC 法での補間法の比較。補間法を変えて磁気流体衝撃波管問題を解いた。すべて密度プロファイルを示している。モデルについては表 3.4 参照。

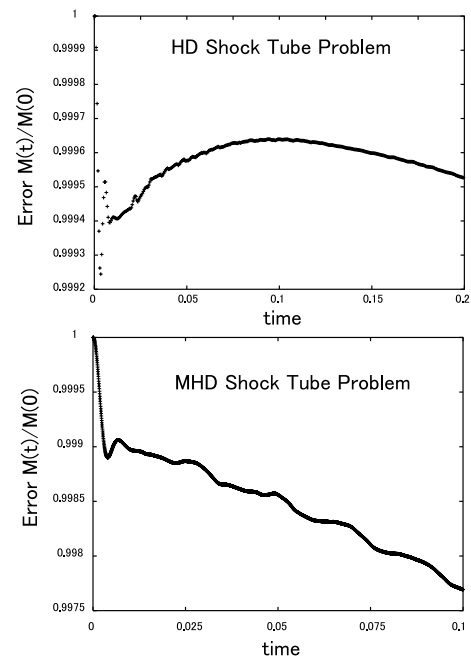


図 3.11: 数値エラーの時間変化。Sod (1978) の衝撃波管問題 (上) と Brio & Wu (1984) の磁気流体衝撃波管問題 (下)。 $M(0)$ は初期の全質量、 $M(t)$ は各時間での全質量。

参考文献

- [1] Brio, M. & Wu, C. C. 1984, *Journal of Computational Physics*, 140, 17
- [2] Evans, C. R. & Hawley, J. F. 1988, *ApJ*, 332, 659
- [3] Hawley, J. F. & Stone, J. M. 1995, *Computer Physics Communications*, 89, 127
- [4] Kudoh, T. & Shibata, K. 1997, *ApJ*, 476, 632
- [5] Ogata, Y. & Yabe, T. 1999, *Computer Physics Communications*, 119, 179
- [6] Sod, G. A. 1978, *Journal of Computational Physics*, 27, 1
- [7] Stone, J. M. & Norman, M. L. 1992, *ApJS*, 80, 791
- [8] Yabe, T. & Aoki, A. 1991, *Computer Physics Communications*, 66, 219
- [9] 矢部 孝, 内海 隆行, 尾形 陽一, 2003, CIP 法, 森北出版

第 4 章

保存保証型 CIP コードの開発に向けて

京都大学 木暮 宏光

4.1 イントロダクション

CIP 法による解法は、方程式系を移流方程式の形をした部分とそれ以外の部分とに分けて、移流方程式の部分は物理量の空間微分の値も用いて、3 次関数で格子点間のプロファイルを近似する。つまり保存形の式であっても非保存形に書き換えて解くために、保存性が問題となってくる。そのために物理量の保存を保証した CIP 法が提案されている (矢部、内海、尾形 2003 の教科書に解説が掲載されている)。その方法は CIP-CSL2 法と CIP-CSL4 法と呼ばれているが、今回は CIP-CSL2 法について述べる。

4.2 CIP-CSL2 法

保存性を保証するための考え方は、保存形の式を解く解法と同様で、セル境界からの流入出を評価する事である。まず以下のような式を考える。

$$\frac{\partial D}{\partial t} + u \frac{\partial D}{\partial x} = 0 \quad (4.1)$$

この式を x で微分すると

$$\frac{\partial D'}{\partial t} + \frac{\partial}{\partial x} (D'u) = 0 \quad (4.2)$$

という保存形の式となる。つまり保存形の式で書ける物理量の積分量は、移流方程式で書けるということである。まず質量 D を

$$D_i(x) = \int_{x_i}^x f(x') dx' \quad (4.3)$$

と定義し、この量を 3 次関数

$$D_i(x) = AX^3 + BX^2 + f_i^n X \quad (4.4)$$

で評価する。ここで $X = x - x_i$ である。 $X = 0 (x = x_i)$ の時は定義から $D = 0$ であるので、3 次関数の X の 0 次の係数は 0 である。 Δx をグリッド間隔とした時、 $X = \Delta x$ であれば、 D はグリッド間のセル中の質量を意味することになる。密度 ρ は D の微分で与えられる。

$$\rho_i(x) = \frac{\partial D(x)}{\partial x} = 3AX^2 + 2BX + f_i^n \quad (4.5)$$

と書ける。補間関数の係数 A,B を関数の連続性から求める。考え方は通常の CIP 法と同様である。

$$D_i(x_{iup}) = -\text{sgn}(u_i)M_{icell}^n \quad (4.6)$$

$$\partial_x D_i(x_i) = f_i^n \quad (4.7)$$

$$\partial_x D_i(x_{iup}) = f_{iup}^n \quad (4.8)$$

$$\text{sgn}(u_i) = \begin{cases} 1 & \text{if } u_i \geq 0 \\ -1 & \text{if } u_i < 0 \end{cases} \quad (4.9)$$

ここで M_{icell}^n は上流側のセル内の全質量である。 iup は上流側のグリッドの番号を意味している。これらの関係式から各係数は

$$A = \frac{f_i^n + f_{iup}^n}{\Delta x_i^2} + \frac{2\text{sgn}(u_i)M_{icell}^n}{\Delta x_i^3} \quad (4.10)$$

$$B = -\frac{3\text{sgn}(u_i)M_{icell}^n}{\Delta x_i^2} - \frac{2f_i^n + f_{iup}^n}{\Delta x_i} \quad (4.11)$$

ここで $\Delta x_i = x_{iup} - x_i$ であり、 u_i が正の時は Δx_i が負になる事に注意。

4.3 時間発展

上記の補間関数によって、シミュレーション上では取り扱われていないグリッド間での物理量分布が定義された。これによって通常の CIP 法と同様に、移流方程式の部分の関しては解を求めることができる。密度に関しては通常の CIP 法と同様に保存形の式、

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \quad (4.12)$$

を非保存形の式、

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} = -\rho \frac{\partial u}{\partial x} \quad (4.13)$$

という形に変形して解く。質量に関しては、セル境界からの流入出量を考える。時間 Δt の間にセル境界 x_i から流入（出）した質量は

$$\Delta M_i^n = \int_{x_i+\xi}^{x_i} \rho(x') dx' = -D_i(x_i + \xi) \quad (4.14)$$

と書ける。ここで $\xi = -u_i \Delta t$ である。

4.4 計算例

以下には流体力学方程式を解いた結果を示す。まだ確認が不十分でバグ等が残っている可能性がある参考程度と考えて欲しい。

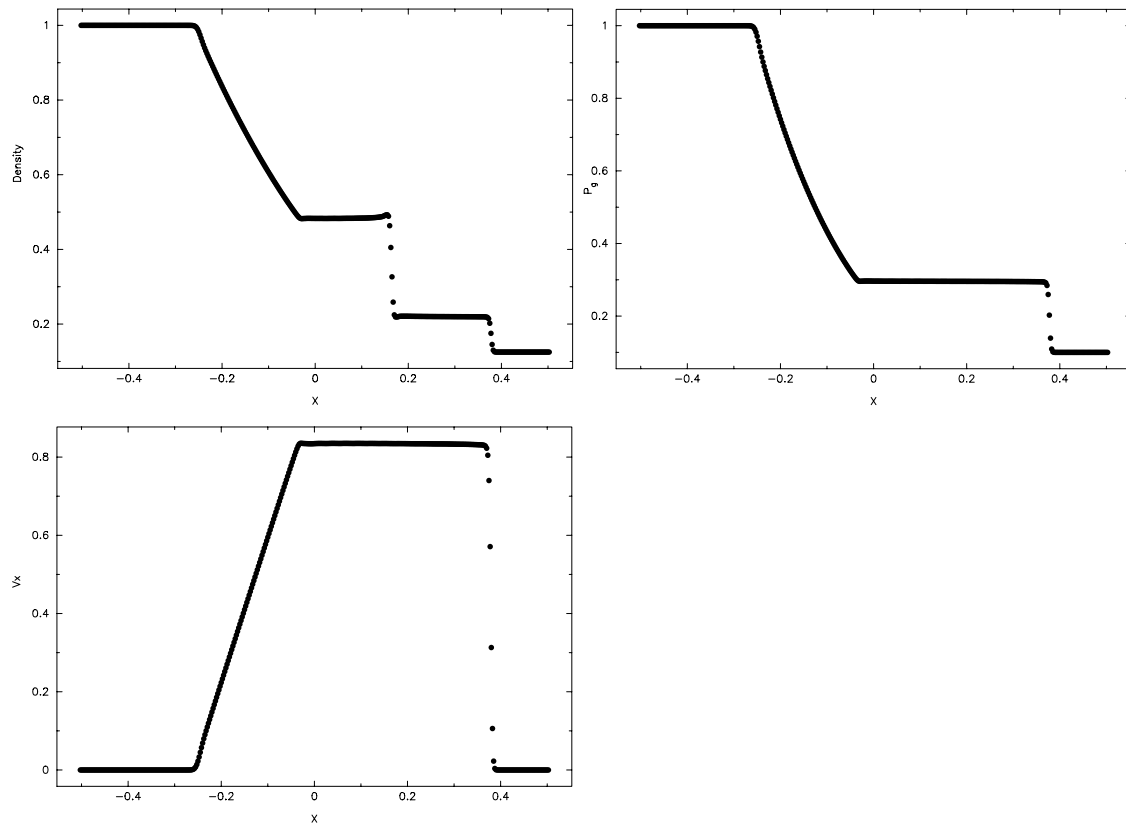


図 4.1: 通常の CIP 法で解いた結果。先に移流方程式部分（以後、移流相と呼ぶ）を、後にそれ以外の部分（以後、非移流相と呼ぶ）を解くコードでの計算結果。左上が密度、右上が圧力、下は速度。

4.4.1 1次元衝撃波管問題

グリッド数は 400、グリッド間隔 $\Delta x = 0.0025$ 、クーラン数は 0.4、人工粘性係数は 0.6、比熱比 $\gamma = 1.66666$ とした。t=0.2、計算ステップ数は約 480 の結果である。

保存性を見るために密度 ρ の計算領域中での総和、

$$\rho_{\Sigma} = \sum_i \rho_i dx_i \quad (4.15)$$

を計算した。そして初期値からのズレの初期値に対する割合の時間変化を t=0.26（波が計算領域の境界に達する時間）まで見た（図 4.2）。

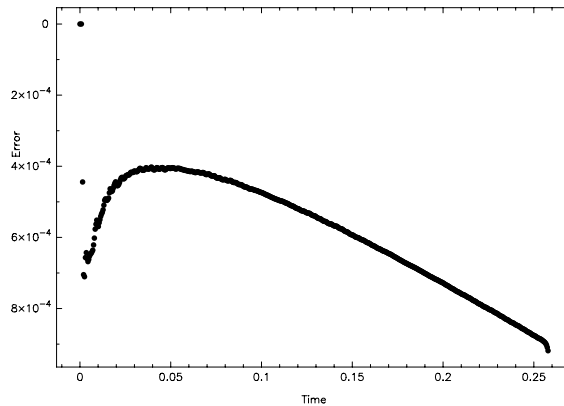


図 4.2: ρ_{Σ} の初期値に対するエラーの時間変化。

次に通常の CIP 法でも先に非移流相、後に移流相を解いた結果を示す。

同様に ρ_{Σ} のエラー時間変化を見ると以下ようになった。

先に移流相を解くコードでは最終的にエラーが -9.2×10^{-4} であったのに対して、先に非移流相を解くコードではエラーが -4.7×10^{-3} であった。

次に保存保証型（CIP-CSL2）のコードで解いた結果を示す。先に移流相を解くコードの結果を図 4.5 に示した。注目すべきは、接触不連続面前後で密度の overshoot、undershoot が見られる点である。しかし質量の保存性を見ると 2.7×10^{-6} となっていて、絶対値で比較すると通常の CIP 法に比べて 1 桁以上良くなっている（図 4.6 参照）。さらにそれは時間に対してほぼ一定の値となっている。

CIP-CSL2 法で先に非移流相を解くコードでの結果を図 4.7 に示した。先ほどと同様に接触不連続面前後で overshoot、undershoot が見られている。保存性の結果を図 4.8 に示した。エラーは 3.4×10^{-4} でほぼ一定となっている。やはり通常の CIP 法に比べて絶対値で 1 桁以上良くなっている。このように流体力学方程式を解く場合、保存性を考えれば移流相から解く、また CIP-CSL2 を用いる、というのが良いということができる。ただし今回は 1 つのテスト計算しか行っていないので、厳密な議論のためにはさまざまな初期条件からのテスト計算を行う必要がある。

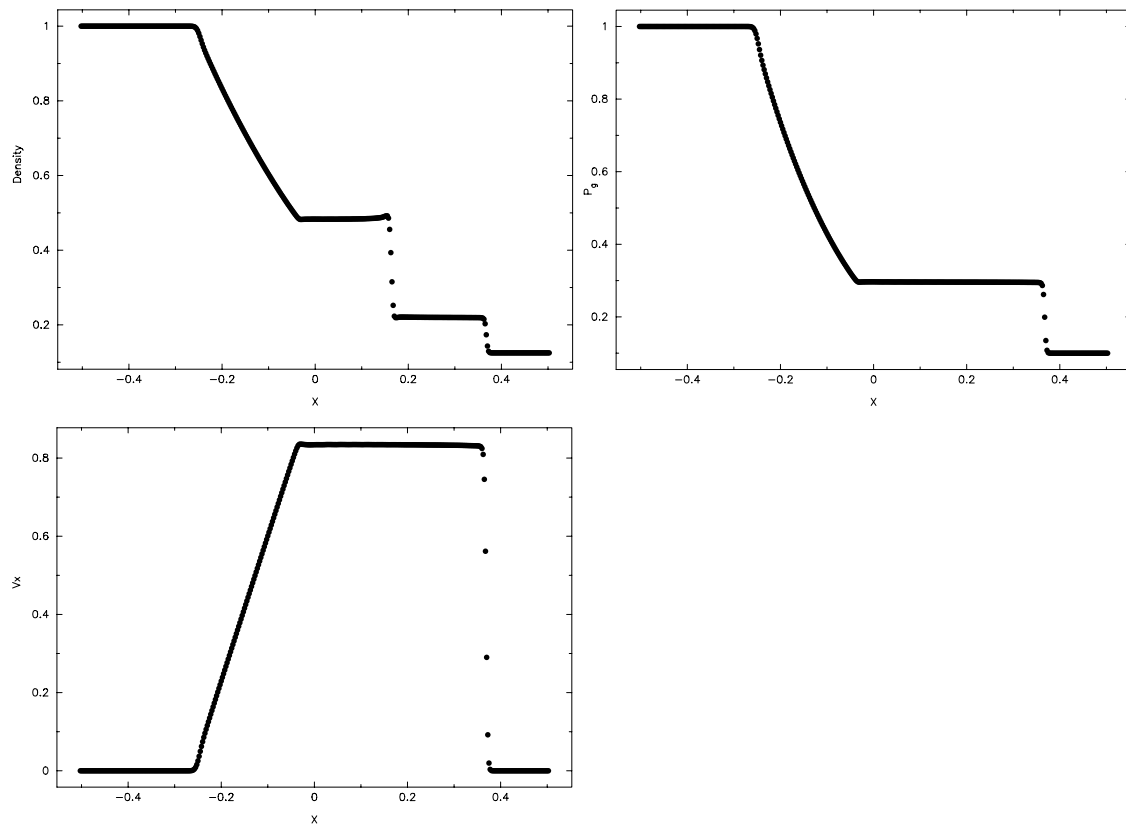


図 4.3: 通常の CIP 法で解いた結果。先に非移流相を、後に移流相を解くコードでの計算結果。左上が密度、右上が圧力、下は速度。

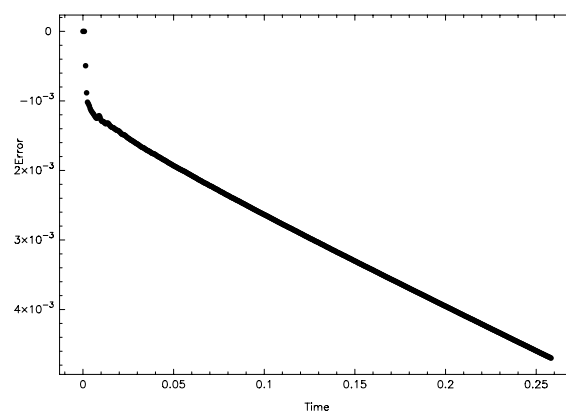


図 4.4: ρ_Σ のエラー時間変化。

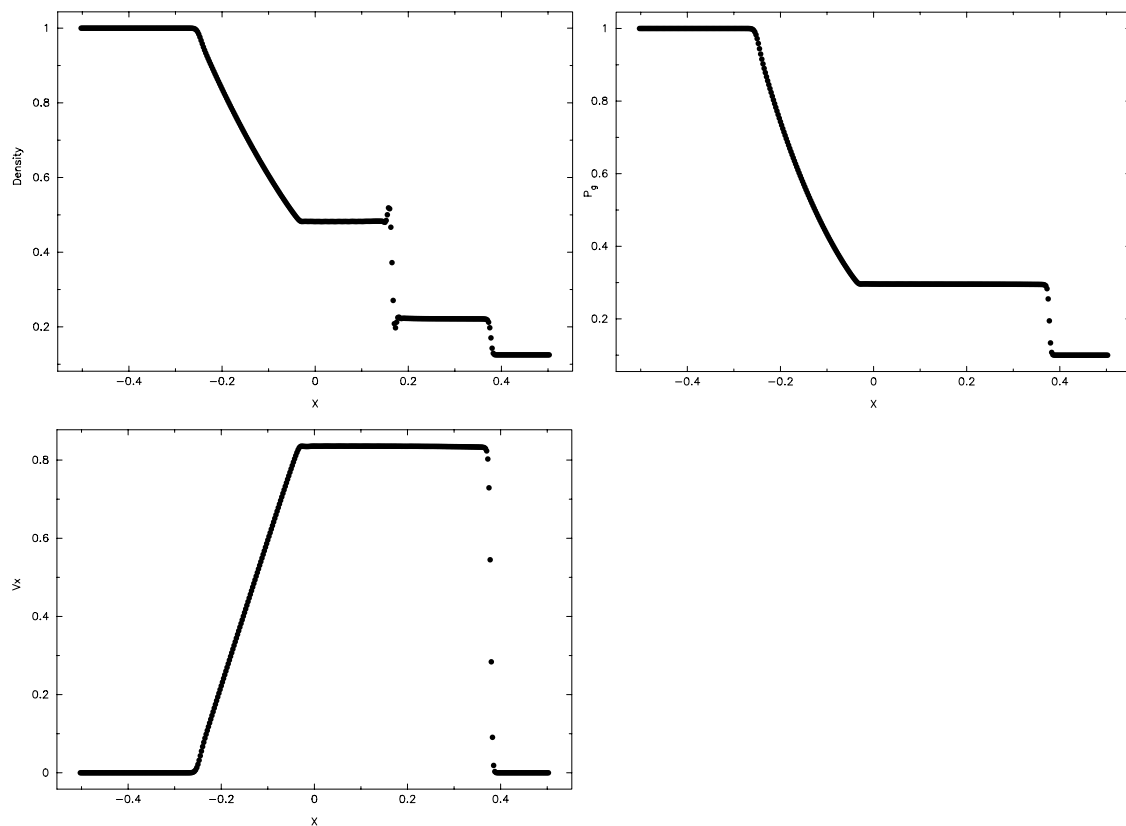


図 4.5: CIP-CSL2 法で解いた結果。先に移流相を、後に非移流相を解くコードでの計算結果。左上が密度、右上が圧力、下は速度。

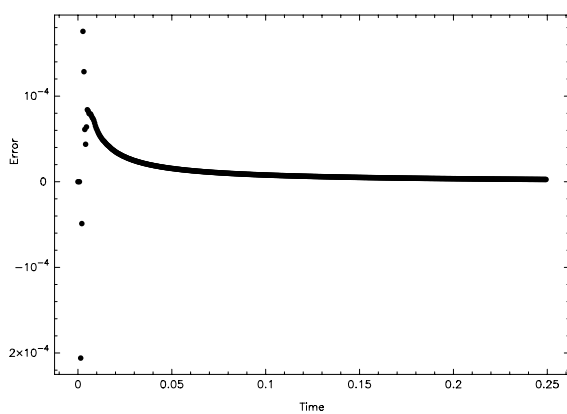


図 4.6: ρ_Σ のエラー時間変化。

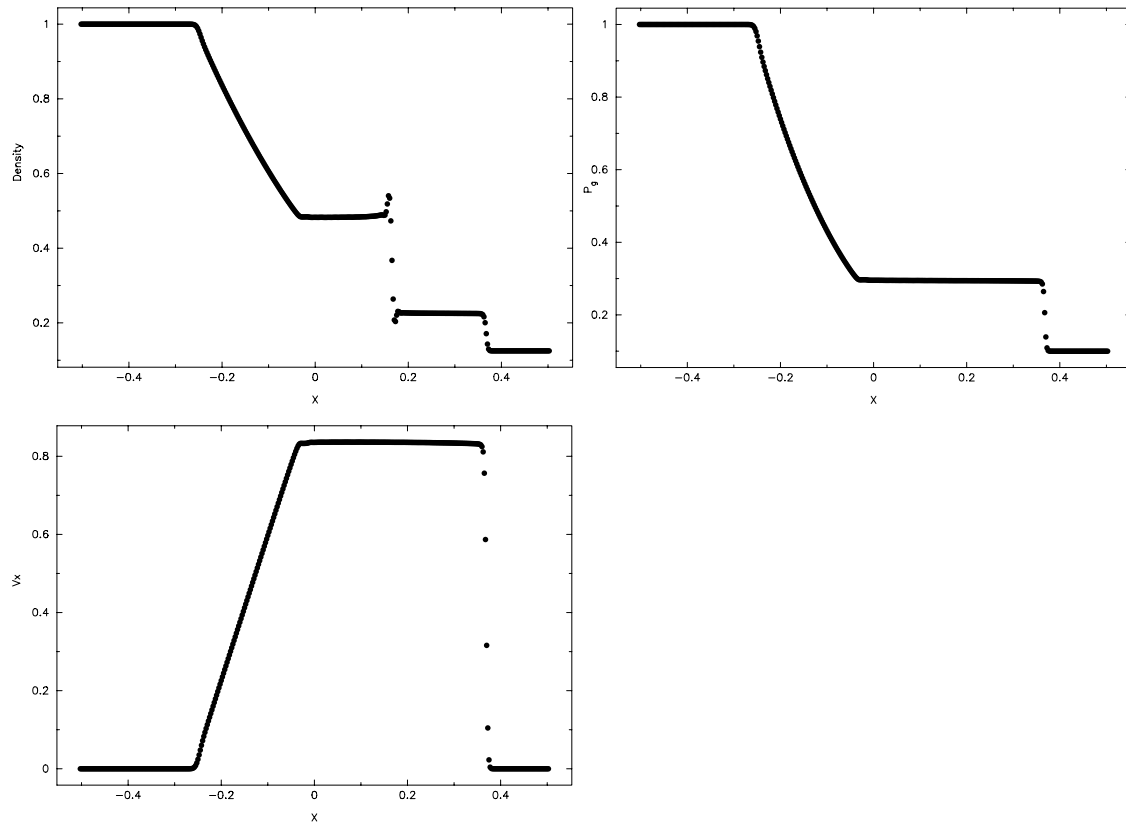


図 4.7: CIP-CSL2 法で解いた結果。先に非移流相を、後に移流相を解くコードでの計算結果。左上が密度、右上が圧力、下は速度。

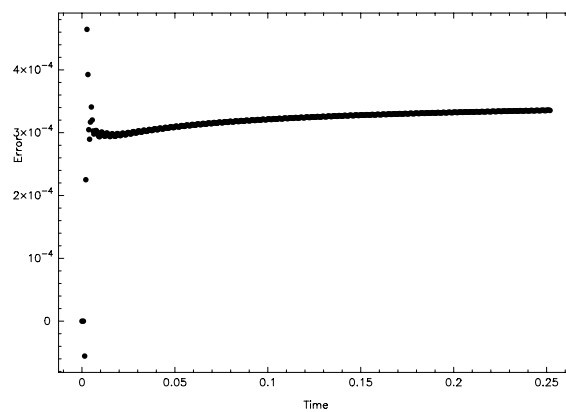


図 4.8: ρ_Σ のエラー時間変化。

4.4.2 ケルビンヘルムホルツ不安定性

次に CIP-CSL2 法を用いて 2 次元計算を行うための手法を述べる。いくつかの手段が提案されているが、最も簡単な directional splitting method による方法に関して述べる。それは、

$$\frac{\partial \rho}{\partial t} + \frac{\partial (u\rho)}{\partial x} + \frac{\partial (v\rho)}{\partial y} = 0 \quad (4.16)$$

という保存形の方程式を

$$\frac{\partial \rho}{\partial t} + \frac{\partial (u\rho)}{\partial x} = 0 \quad (4.17)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial (v\rho)}{\partial y} = 0 \quad (4.18)$$

という 2 つの式に分解して解くという方法である。積分量としては ρ を x 方向に積分した量 σ_x 、 y 方向に積分した量 σ_y 、 σ_x (σ_y) を y (x) 方向に積分した量 M が定義される。まず ρ と σ_x の値を x 方向に関して更新し、次に σ_y と M を x 方向に関して更新する (σ_y を x 方向に積分した量が M であるから)。この更新された値は時間方向に 1/2 ステップ進んだ量とすることができる。これら中間値を用いて、 ρ と σ_y を y 方向に、 σ_x と M を y 方向に関して更新することにより、時間 1 ステップ進んだ事になる。この時、先に x 方向に関して更新するか、 y 方向に更新するかは任意性がある。そのため、1 つ前の時間 1 ステップ進める過程で x 方向に関して先に更新したのであれば、このステップでは y 方向に関して先に更新する、とするのがよい。

通常の CIP 法 (directional splitting なし、先に非移流相を解く) と CIP-CSL2 法 (directional splitting、先に移流相を解く) によるケルビンヘルムホルツ不安定性の計算結果を示す。現在 CIP-CSL2 法では密度が小さくなり計算が進まなくなるという結果になっている。これはコードのバグなのか、1 次元計算の結果にあったような undershoot による結果なのか、今後調べる予定である。

参考文献

- [1] 矢部 孝, 内海 隆行, 尾形 陽一, 2003, CIP 法, 森北出版

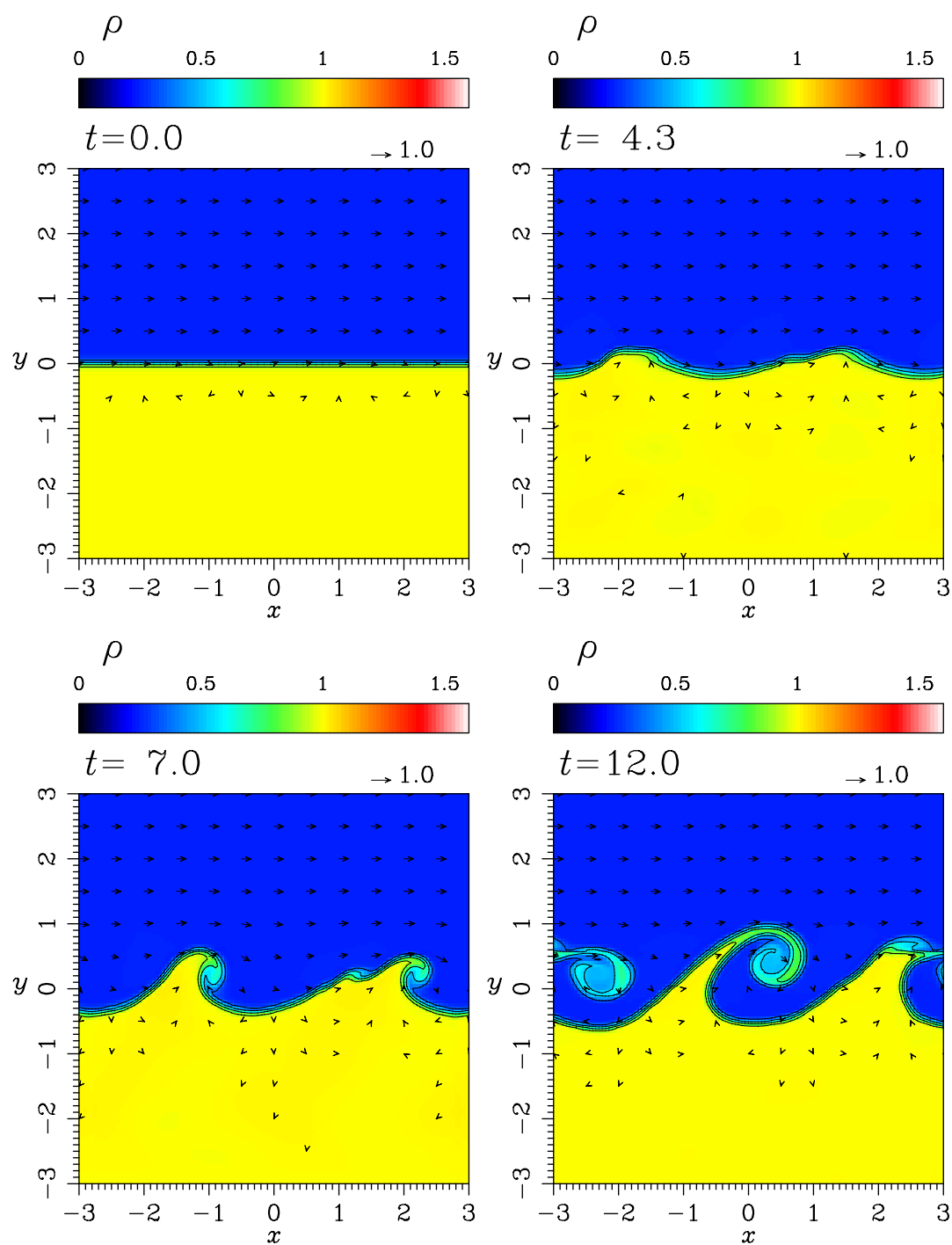


図 4.9: 通常の CIP 法を用いたケルビンヘルムホルツ不安定性の計算結果。グリッド数は 103×103 。クーラン数は 0.4、人工粘性係数は 0.6、比熱比 γ は 1.66666。

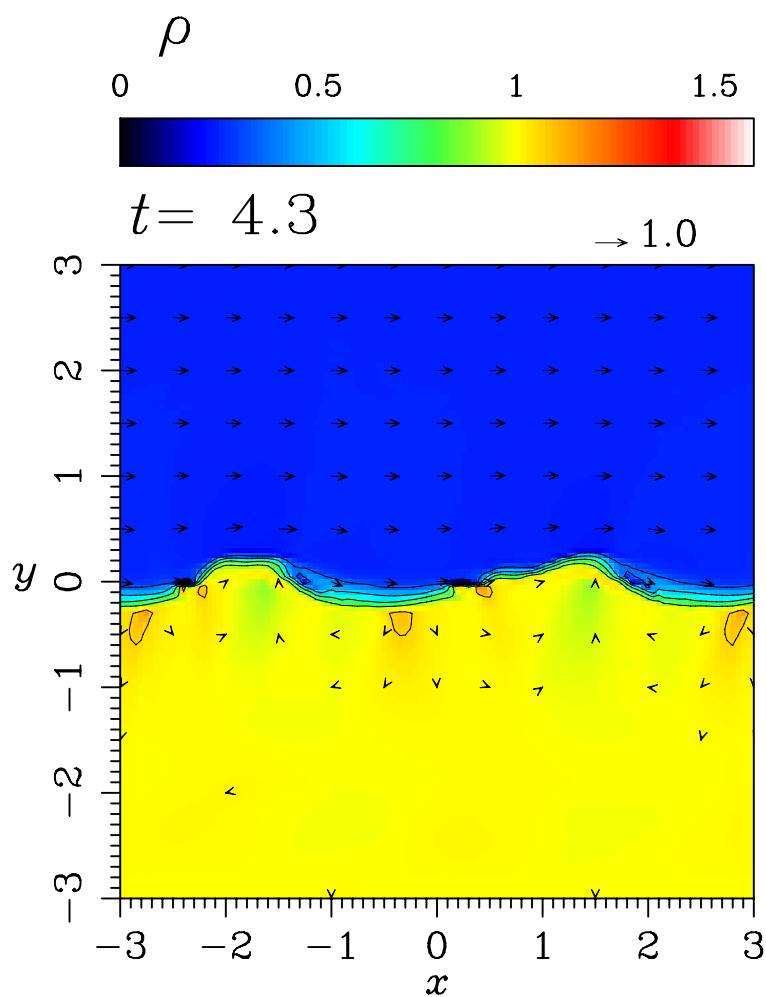


図 4.10: CIP-CSL2法を用いたケルビンヘルムホルツ不安定性の計算結果。パラメータは通常のCIP法での計算と同じ。波の前面部分で密度が小さい領域が出来て、時間刻み幅 Δt が小さくなったため、計算を打ち切った。

第 5 章

一般相対論的磁気流体力学の数値解法

京都大学 水野 陽介

5.1 相対論的磁気流体力学シミュレーションの必要性と現状

活動銀河核 (AGN)、X 線連星、ガンマ線バースト (GRB) などの天体で見られる様々な高エネルギー天体物理現象は全て強い重力源と電磁場を介したプラズマとの相互作用によって引き起こされていると考えられる。例えば活動銀河核 (AGN) から噴出する宇宙ジェットではパーセクスケールでの観測で超光速運動を見せるものがある。これを説明するためにはジェットが光速の 99.5% を超える (ローレンツ係数 $\gamma > 10$) 速度を持つ必要がある。また系内で見つかっているマイクロクエーサーと呼ばれる GRS1915+105 (Mirabel & Rodriguez 1994) や GROJ1655-40 (Tingay et al. 1995) でも $0.9c$ (c は光速を表す) の速度を持つジェットが見つかっている。このような天体現象を解明するためには相対論的な取り扱いが必要不可欠である。また同様にブラックホールのような強い重力源近くでの物理を厳密的に取り扱うには (一般) 相対論的な取り扱いが必要である。

相対論的流体力学 (RHD) シミュレーション、特に相対論的磁気流体力学 (RMHD) シミュレーションは近年多くの研究者によって精力的に開発が進められ、今最も注目されている研究分野である。RHD シミュレーションについては Marti & Müller (2003) と Font (2004) によるレビューがあるので参考にして頂きたい。一般相対論的磁気流体力学 (GRMHD) では、Willson (1977) が最初にカーブラックホール周りでの GRMHD 方程式を数値的に解いた。その後、Yokosawa (1993) まで GRMHD シミュレーションは行われなかった。一方で、特殊相対論的磁気流体力学 (SRMHD) に関しては van Putten (1993)、Koide, Nishikawa, & Mutel (1996)、Komissarov (1999)、Balsara (2001)、Koldoba, Kuznetsov, & Ustyugova (2002)、Del Zanna, Bucciantini, & Londrillo (2003) によってコードの開発が進められている。特に Komissarov (1999) や Del Zanna, Bucciantini, & Londrillo (2003) では近似リーマン解法を用いたゴドノフタイプの数値解法を適用している。GRMHD に関しては Koide, Shibata, & Kudoh (1998, 1999) によって初めてコード開発及び天体現象への応用が行われた。その後、このコードを用いて、カーブラックホール周りの降着円盤からのジェットの形成 (Koide et al. 2000)、カーブラックホールからのエネルギー引き抜き (Koide et al. 2002; Koide 2003)、ガンマ線バーストに関連した大質量星の重力崩壊 (Mizuno et al. 2004a, 2004b) に関してシミュレーションを行っている。

一方で、ここ数年では、De Villiers & Hawley (2003a) と Gammie, McKinney & Tóth (2003) によってお互い異なる数値解法を用いて GRMHD コードが形成され、同じテスト計算による比較に

よって同様の結果が得られることが示された。彼らはこの GRMHD コードを用いてブラックホール周りの降着円盤の進化やジェット形成 (De Villiers & Hawley 2003b; De Villiers, Hawley, & Krolik 2003; Hirose et al. 2004; De Villiers et al. 2004)、カーブラックホールの回転の進化 (Gammie, Shapiro, & McKinney 2004)、ブラックホールからのエネルギー引き抜き (McKinney & Gammie 2004) に関して精力的に研究を行っている。

ここでは、我々が使用している Koide et al. (2000) によって開発された GRMHD コードの基礎方程式及び数値解法について紹介する¹。第 2 節では、GRMHD の基礎方程式の 4 次元形式、そしてシミュレーションに適した 3+1 形式を紹介する。第 3 節ではその GRMHD 基礎方程式の数値解法を最終節では現状の問題点と今後の発展に向けたコメントを紹介する。それぞれについて、詳しいことは参考文献を見て頂きたい。

5.2 基礎方程式

5.2.1 GRMHD 方程式の 4 次元形式

一般相対論的磁気流体力学の基礎方程式は曲がった時空における粒子数保存、エネルギー運動量保存、マックスウェル方程式、そして電気抵抗なし (理想 MHD 条件) のオームの法則の一般相対論形式からなる。ここでは放射冷却、電気伝導度、粘性、自己重力を無視する。時空 $(x^0, x^1, x^2, x^3) = (ct, x^1, x^2, x^3)$ は、計量 $g_{\mu\nu}$ によって現される。ここで、 μ や ν のようなギリシャ文字の添え字は 0 から 3 までを取る。 c は光速である。4 次元時空における GRMHD の基礎方程式は

$$\nabla_\nu(\rho U^\nu) = \frac{1}{\sqrt{-||g||}} \frac{\partial}{\partial x^\nu} (\sqrt{-||g||} \rho U^\nu) = 0, \quad (5.1)$$

$$\nabla_\nu T^{\mu\nu} = \frac{1}{\sqrt{-||g||}} \frac{\partial}{\partial x^\nu} (\sqrt{-||g||} T^{\mu\nu}) + \Gamma_{\sigma\nu}^\mu T^{\sigma\nu} = 0, \quad (5.2)$$

$$\partial_\mu F_{\nu\lambda} + \partial_\nu F_{\lambda\mu} + \partial_\lambda F_{\mu\nu} = 0, \quad (5.3)$$

$$\nabla_\mu F^{\mu\nu} = -\mu_0 J^\nu, \quad (5.4)$$

ここで、 ∇_ν は共変微分、 $||g||$ は計量 $g_{\mu\nu}$ の行列式、 $\Gamma_{\mu\nu}^\lambda$ はクリストフェル記号、

$$\Gamma_{\mu\nu}^\lambda \equiv \frac{1}{2} g^{\lambda\sigma} \left(-\frac{\partial g_{\mu\nu}}{\partial x^\sigma} + \frac{\partial g_{\nu\sigma}}{\partial x^\mu} + \frac{\partial g_{\sigma\mu}}{\partial x^\nu} \right), \quad (5.5)$$

である。 U^ν と $J^\nu = (c\rho_e, J^1, J^2, J^3)$ はそれぞれ 4 次元速度、4 次元電流である (ρ_e は電荷密度)。一般相対論的エネルギー運動量テンソル $T^{\mu\nu}$ は以下のように与えられる。

$$T^{\mu\nu} = pg^{\mu\nu} + (e_{int} + p)U^\mu U^\nu + F_\sigma^\mu F^{\nu\sigma} - \frac{1}{4}g^{\mu\nu} F^{\lambda\kappa} F_{\lambda\kappa}, \quad (5.6)$$

ここで、 $F^{\mu\nu}$ は電磁場テンソル、

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu, \quad (5.7)$$

そして $A^\mu = (\phi_e/c, A^1, A^2, A^3)$ は 4 次元ベクトルポテンシャルである (ϕ_e は静電ポテンシャル)。電場 E_i と磁場 B_i はそれぞれ

$$E_i = cF_{i0} \quad (i = 1, 2, 3), \quad (5.8)$$

¹本章の内容の多くは Koide et al. (2000)、Koide (2003) を参考に書かれている。

$$B_1 = F_{23}, \quad B_2 = F_{31}, \quad B_3 = F_{12}, \quad (5.9)$$

で与えられる。スカラー量 ρ 、 p 、 e_{int} はそれぞれ密度、圧力、内部エネルギー密度である。プラズマガスは理想気体として表わされるとする。すなわち、

$$e_{int} = \rho c^2 + \frac{p}{\Gamma - 1}, \quad (5.10)$$

である。ここで Γ は比熱比である。これらの方程式に加えて、理想 MHD 条件を仮定する

$$F_{\mu\nu} U^\nu = 0. \quad (5.11)$$

この条件を使うことで方程式 (5.1)-(5.3) を閉じることができる。方程式 (5.4) は電流密度 J^μ を計算するときのみ使用する。

ここでは、計量は時間発展しないと仮定する。もし計量の非対角成分 $g_{\mu\nu}$ がゼロならば、

$$g_{ij} = 0 \quad (i \neq j) \quad (5.12)$$

である。ここでローマ字の添え字 (i, j) は 1 から 3 を取る。

$$g_{00} = -h_0^2, \quad g_{ii} = h_i^2, \quad (5.13)$$

$$g_{i0} = g_{0i} = -\frac{h_i^2 \omega_i}{c}, \quad (5.14)$$

これらを使って計量は以下のように書かれる。

$$\begin{aligned} (ds)^2 &= g_{\mu\nu} dx^\mu dx^\nu \\ &= -h_0^2 (cdt)^2 + \sum_{i=1}^3 [h_i^2 (dx^i)^2 - 2h_i^2 \omega_i dt dx^i]. \end{aligned} \quad (5.15)$$

5.2.2 GRMHD 方程式の 3+1 形式

一般に 4 次元形式で書かれた GRMHD 方程式は時間と空間がカップルしているため数値的に磁気流体の時間発展を解くのは困難である。そこでその困難を解決する方法である、空間と時間を分解した形式 (3+1 形式) を示す。GRMHD 方程式をこの 3+1 形式で書き下すことによって GRMHD 方程式の時間発展を数値的に解くことが可能となる。

まず観測される物理量を表わすために以下に示す幾つか異なる座標系を使う。

(1) 実験室系 (Laboratory frame)

これはブラックホールから遠く離れた観測者が見た座標系で、“star-fixed coordinate”あるいは“observer-at-infinity frame”と呼ばれる。カー時空ではこの系の座標は Boyer-Lindquist 座標で与えられる。ここから実験室系で反変ベクトルを a^μ として表わす。

(2) Fiducial observer (FIDO) frame

これは同じ時刻を取る 3 次元超表面 (hypersurface) の上に張られた座標 (空間 3 次元) に静止している観測者から見た座標系であり、局所慣性系である。この座標系 $(ct, \hat{x}^1, \hat{x}^2, \hat{x}^3)$ で、計量は

$$(ds)^2 = -(cd\hat{t})^2 + \sum_i (d\hat{x}^i)^2, \quad (5.16)$$

となる。ここで、

$$cd\hat{t} = \alpha cdt, \quad (5.17)$$

$$d\hat{x}^i = h_i dx^i - \beta^i \alpha cdt, \quad (5.18)$$

である。それぞれ α 、 β^i を “lapse function”、“shift velocity” と言う。

$$\alpha = \sqrt{h_0^2 + \sum_{i=1}^3 \left(\frac{h_i \omega_i}{c} \right)^2} \quad (5.19)$$

$$\beta^i = \frac{h_i \omega_i}{c\alpha}. \quad (5.20)$$

lapse function α は遠方の観測者が計ってる時間と FIDO frame での時間との間の経過時間の比を表し、shift velocity β^i に微小時間 dt を掛けた $\beta^i dt$ はある時刻 t の 3 次元超表面から次の時間 $t + dt$ での 3 次元超表面に移動した際の位置のずれを表している。また、この計量で行列式 $\|g\|$ は $\sqrt{-\|g\|} = \alpha h_1 h_2 h_3$ で与えられる。反変ベクトルは以下のように書かれる。

$$g^{00} = -\frac{1}{\alpha^2} \quad (5.21)$$

$$g^{i0} = g^{0i} = -\frac{1}{\alpha^2} \frac{\omega_i}{c} \quad (5.22)$$

$$g^{ij} = \frac{1}{h_i h_j} (\delta^{ij} - \beta^i \beta^j) \quad (5.23)$$

ここで δ^{ij} はクロネッカー記号の δ である。FIDO での反変ベクトル $\hat{a}^\mu$ は

$$\hat{a}^0 = \alpha a^0, \quad \hat{a}^i = h_i a^i - \alpha a^0 \beta^i \quad (5.24)$$

である。一方、共変ベクトルは

$$\hat{a}_0 = \frac{1}{\alpha} a_0 + \sum_i \beta^i \frac{1}{h_i} a_i, \quad \hat{a}_i = \frac{1}{h_i} a_i \quad (5.25)$$

である。これは計量がミンコフスキー時空と同じく $\hat{a}^0 = -\hat{a}_0$ そして $\hat{a}^i = \hat{a}_i$ が成り立つためである。

(3) 共動座標系 (Comoving frame)

この座標系では観測者がガスまたはプラズマに乗って全ての現象を局所的に見る。この座標系は圧力や密度などのスカラー量を測るのに使われる。

方程式 (5.24) と (5.25) を使って、FIDO frame で計られたベクトル、テンソルの成分は以下のように表される。

$$\hat{\gamma} = \alpha U^0, \quad (5.26)$$

$$\hat{D} = \hat{\gamma} \rho, \quad (5.27)$$

$$\hat{v}^i = \frac{h_i}{\hat{\gamma}} c U^i - c \beta^i, \quad (5.28)$$

$$\epsilon + Dc^2 = \hat{T}^{00} = \alpha^2 T^{00}, \quad (5.29)$$

$$\hat{P}^i = \frac{1}{c} \hat{T}^{i0} = \frac{\alpha h_i}{c} T^{i0} - \frac{\beta^i}{c} (\epsilon + Dc^2), \quad (5.30)$$

$$\hat{T}^{ij} = h_i h_j T^{ij} - \beta^i h_0 h_j T^{0j} - \beta^j h_i h_0 T^{i0} + \beta^i \beta^j \alpha^2 T^{00}, \quad (5.31)$$

$$\hat{F}_{i0} = -\hat{F}_{0i} = \frac{1}{\alpha h_i} F_{i0} + \sum_j \beta^j \frac{1}{h_i h_j} F_{ij}, \quad (5.32)$$

$$\hat{F}_{ij} = \frac{1}{h_i h_j} F_{ij}, \quad (5.33)$$

$$\hat{\rho}_e = \frac{1}{c} \alpha J^0, \quad (5.34)$$

$$\hat{J}^i = h_i J^i - \hat{\rho}_e c \beta^i, \quad (5.35)$$

ここで、 $\hat{\gamma}$ はローレンツ係数、 $\hat{v}^i$ は3次元速度、 ϵ はエネルギー密度、 $\hat{P}^i$ は運動量密度である。ここからはローレンツ係数 γ と電荷密度 ρ_e は $\gamma = \hat{\gamma}$ 、 $\rho_e = \hat{\rho}_e$ としてハットを取って記述する。

FIDO frame で測られた物理量の間の関係は特殊相対論的 MHD(SRMHD) の関係 (Koide, Nishikawa, & Mutel 1996; Koide 1997)² と同じである。ここにその関係をまとめておく。

$$D = \gamma \rho, \quad (5.36)$$

$$\gamma = \frac{1}{\sqrt{1 - \sum_{i=1}^3 (\hat{v}^i/c)^2}}, \quad (5.37)$$

$$\hat{P}^i = \frac{1}{c^2} h \gamma^2 \hat{v}^i + \frac{1}{c^2} (\hat{\mathbf{E}} \times \hat{\mathbf{B}})_i, \quad (5.38)$$

$$\hat{T}^{ij} = p \delta^{ij} + \frac{h}{c^2} \gamma^2 \hat{v}^i \hat{v}^j + \left(\frac{\hat{B}^2}{2} + \frac{\hat{E}^2}{2c^2} \right) \delta^{ij} - \hat{B}_i \hat{B}_j - \frac{\hat{E}_i \hat{E}_j}{c^2}, \quad (5.39)$$

$$\epsilon = h \gamma^2 - p - Dc^2 + \frac{\hat{B}^2}{2} + \frac{\hat{E}^2}{2c^2}, \quad (5.40)$$

ここで h は相対論的エンタルピー密度、

$$h = \rho c^2 + \frac{\Gamma p}{\Gamma - 1} = e_{int} + p, \quad (5.41)$$

である。また、磁場 $\hat{\mathbf{B}}$ と電場 $\hat{\mathbf{E}}$ は以下のように定義される。

$$\hat{B}_i = \sum_{j,k} \frac{1}{2} \epsilon^{ijk} \hat{F}_{jk}, \quad (5.42)$$

$$\hat{E}_i = c \hat{F}_{i0} \quad (5.43)$$

FIDO によって測られた変数を使用して、プラズマに関する一般相対論的保存則とマックスウェル方程式 (5.1)-(5.4)、(5.11) からなる GRMHD の基礎方程式を導く。

$$\frac{\partial D}{\partial t} = -\frac{1}{h_1 h_2 h_3} \sum_i \frac{\partial}{\partial x^i} \left[\frac{\alpha h_1 h_2 h_3}{h_i} D (\hat{v}^i + c \beta^i) \right], \quad (5.44)$$

²特殊相対論ではミンコフスキー時空を取るなので計量は $h_0 = -1$, $h_i = 1$ ($i = 1, 2, 3$) となる。

$$\begin{aligned} \frac{\partial \hat{P}^i}{\partial t} = & - \frac{1}{h_1 h_2 h_3} \sum_j \frac{\partial}{\partial x^j} \left[\frac{\alpha h_1 h_2 h_3}{h_j} (\hat{T}^{ij} + c \beta^j \hat{P}^i) \right] \\ & - (\epsilon + Dc^2) \frac{1}{h_i} \frac{\partial \alpha}{\partial x^i} + \alpha f_{curv}^i - \sum_j \hat{P}^j \sigma_{ji}, \end{aligned} \quad (5.45)$$

$$\begin{aligned} \frac{\partial \epsilon}{\partial t} = & - \frac{1}{h_1 h_2 h_3} \sum_i \frac{\partial}{\partial x^i} \left[\frac{\alpha h_1 h_2 h_3}{h_i} c^2 \left(\hat{P}^i - D \hat{v}^i + \frac{\beta^i}{c} \epsilon \right) \right] \\ & - \sum_i c^2 \hat{P}^i \frac{1}{h_i} \frac{\partial \alpha}{\partial x^i} - \sum_{i,j} \hat{T}^{ij} \sigma_{ji}, \end{aligned} \quad (5.46)$$

$$\hat{E}_i = - \sum_{j,k} \epsilon_{ijk} \hat{v}^j \hat{B}_k, \quad (5.47)$$

$$\frac{\partial \hat{B}_i}{\partial t} = \frac{-h_i}{h_1 h_2 h_3} \sum_{j,k} \epsilon^{ijk} \frac{\partial}{\partial x^j} \left[\alpha h_k \left(\hat{E}_k - \sum_{l,m} \epsilon^{klm} c \beta^l \hat{B}_m \right) \right], \quad (5.48)$$

$$\sum_i \frac{1}{h_1 h_2 h_3} \frac{\partial}{\partial x^i} \left(\frac{h_1 h_2 h_3}{h_i} \hat{B}_i \right) = 0, \quad (5.49)$$

$$\rho_e = \sum_{j,k} \frac{1}{c^2} \frac{1}{h_1 h_2 h_3} \frac{\partial}{\partial x^i} \left(\frac{h_1 h_2 h_3}{h_i} \hat{E}_i \right), \quad (5.50)$$

$$\alpha(\hat{J}^i + \rho_e c \beta^i) + \frac{1}{c^2} \frac{\partial \hat{E}_i}{\partial t} = \sum_{j,k} \frac{h_i}{h_1 h_2 h_3} \epsilon^{ijk} \frac{\partial}{\partial x^j} \left[\alpha h_k \left(\hat{B}_k + \sum_{l,m} \epsilon_{klm} \beta^l \frac{\hat{E}_k}{c} \right) \right], \quad (5.51)$$

ここで、

$$f_{curv}^i \equiv \sum_j (G_{ij} \hat{T}^{ij} - G_{ji} \hat{T}^{jj}), \quad (5.52)$$

$$G_{ij} \equiv - \frac{1}{h_i h_j} \left(\frac{\partial h_i}{\partial x^j} \right), \quad (5.53)$$

$$\sigma_{ij} \equiv \frac{h_i}{h_j} \left(\frac{\partial \omega_i}{\partial x^j} \right), \quad (5.54)$$

である。この形の方程式を 3+1 形式と呼ぶ。それは空間と時間に関する微分が完全に分けられた形だからである。3+1 形式に関する詳細は Thone, Price, & Macdonald (1986) の教科書を参考にしたい。実際のシミュレーションではこの 3+1 形式で書かれた GRMHD 方程式を差分法を用いて時間発展を解いていく³。数値解法については次章で述べる。

³時間発展を解いていく際、理想 MHD 条件を課しているため、電場 $\hat{E}$ の時間発展は差分法で直接解かない。式 (5.47) を式 (5.48) に代入して磁場 $\hat{B}$ の時間発展のみを解く。電場 $\hat{E}$ に関しては各時間での速度 $\hat{v}$ と磁場 $\hat{B}$ を使って式 (5.47) から求める。

5.2.3 GRMHD 方程式のベクトル形式

ここでは、前節で書き下した 3+1 形式での GRMHD 方程式を見通しよくするため、ベクトル形式での表記を行う。まず、ここで FIDO frame で測られた 3 次元ベクトル場 $\hat{\mathbf{a}}$ 、 $\hat{\mathbf{b}}$ そしてスカラー場 $\hat{\phi}$ の微分を導入する。

$$\hat{\nabla} \cdot \hat{\mathbf{a}} = \sum_i \frac{1}{h_1 h_2 h_3} \frac{\partial}{\partial x^i} \left(\frac{h_1 h_2 h_3}{h_i} \hat{a}^i \right), \quad (5.55)$$

$$(\hat{\nabla} \hat{\phi})_i = \frac{1}{h_i} \frac{\partial \hat{\phi}}{\partial x^i}, \quad (5.56)$$

$$(\hat{\nabla} \times \hat{\mathbf{a}})_i = \sum_{j,k} \frac{h_i}{h_1 h_2 h_3} \epsilon_{ijk} \frac{\partial}{\partial x^j} (h_k \hat{a}^k), \quad (5.57)$$

$$[(\hat{\mathbf{b}} \cdot \hat{\nabla}) \hat{\mathbf{a}}]_i = \sum_j \left[\frac{\hat{b}^j}{h_j} \frac{\partial \hat{a}^i}{\partial x^j} - G_{ij} \hat{a}^j \hat{b}^i + G_{ji} \hat{a}^j \hat{b}^j \right]. \quad (5.58)$$

これらの定義を使って、ほとんどのベクトル解析の公式を修正することなしに使用することができる。

$$(\hat{\nabla} \times \hat{\mathbf{a}}) \times \hat{\mathbf{b}} = (\hat{\mathbf{b}} \cdot \hat{\nabla}) \hat{\mathbf{a}} - (\nabla \hat{\mathbf{a}}) \cdot \hat{\mathbf{b}} + [\hat{\mathbf{a}}, \hat{\mathbf{b}}], \quad (5.59)$$

ここで、

$$[(\hat{\nabla} \hat{\mathbf{a}}) \cdot \hat{\mathbf{b}}]_i \equiv \sum_j \frac{1}{h_i} \frac{\partial \hat{a}^j}{\partial x^i} \hat{b}^j \quad (5.60)$$

$$[\hat{\mathbf{a}}, \hat{\mathbf{b}}]_i \equiv \sum_j G_{ij} (\hat{a}^j \hat{b}^i - \hat{a}^i \hat{b}^j) \quad (5.61)$$

である。さらに、

$$\hat{\nabla} \times (\hat{\mathbf{a}} \times \hat{\mathbf{b}}) = (\hat{\nabla} \cdot \hat{\mathbf{b}}) \hat{\mathbf{a}}^i - (\hat{\nabla} \cdot \hat{\mathbf{a}}) \hat{b}^i + \sum_j \frac{h_i}{h_j} \left[\hat{b}^j \frac{\partial}{\partial x^j} \left(\frac{\hat{a}^i}{h_i} \right) - \hat{a}^j \frac{\partial}{\partial x^j} \left(\frac{\hat{b}^i}{h_i} \right) \right]. \quad (5.62)$$

となる。

上記の 3 次元ベクトルとスカラー場の微分を使って GRMHD の基礎方程式 (5.44)-(5.51) は以下のような FIDO のベクトル形式で書き表すことができる。

$$\frac{\partial D}{\partial t} = -\hat{\nabla} \cdot [\alpha D (\hat{\mathbf{v}} + c\beta)], \quad (5.63)$$

$$\frac{\partial \hat{\mathbf{P}}}{\partial t} = -\hat{\nabla} \cdot [\alpha (\hat{\mathbf{T}} + c\beta \hat{\mathbf{P}})] - (\epsilon + Dc^2) \hat{\nabla} \alpha + \alpha \mathbf{f}_{curv} - \hat{\mathbf{P}} \cdot \boldsymbol{\sigma}, \quad (5.64)$$

$$\frac{\partial \epsilon}{\partial t} = -\hat{\nabla} \cdot [\alpha (c^2 \hat{\mathbf{P}} - Dc^2 \hat{\mathbf{v}} + \epsilon c\beta)] - (\hat{\nabla} \alpha) \cdot c^2 \hat{\mathbf{P}} - \hat{\mathbf{T}} : \boldsymbol{\sigma}, \quad (5.65)$$

$$\frac{\partial \hat{\mathbf{B}}}{\partial t} = -\hat{\nabla} \times [\alpha (\hat{\mathbf{E}} - c\beta \times \hat{\mathbf{B}})], \quad (5.66)$$

$$\alpha (\hat{\mathbf{J}} + \hat{\rho}_e c\beta) + \frac{1}{c^2} \frac{\partial \hat{\mathbf{E}}}{\partial t} = \hat{\nabla} \times \left[\alpha \left(\hat{\mathbf{B}} + \frac{1}{c} \beta \times \hat{\mathbf{E}} \right) \right], \quad (5.67)$$

$$\hat{\nabla} \cdot \hat{\mathbf{B}} = 0, \quad (5.68)$$

$$\hat{\rho}_e = \frac{1}{c^2} \hat{\nabla} \cdot \hat{\mathbf{E}}, \quad (5.69)$$

$$\hat{\mathbf{E}} + \hat{\mathbf{v}} \times \hat{\mathbf{B}} = 0, \quad (5.70)$$

ここで、 β は $\beta = (\beta_1, \beta_2, \beta_3)$ の成分を持つ 3 次元ベクトル β_i である。これらの方程式は幾何的な係数や lapse function α や shift velocity β^i を含む項を除いて非相対論 MHD 方程式の保存形と良く似ている⁴。また、 $-c^2 \hat{\nabla} \alpha$ は非相対論 MHD 方程式の重力に相当し、 $c^2 \alpha$ は重力ポテンシャルとみなすことができる。 $\alpha \mathbf{f}_{curv}$ の項は遠心力を含んでいる。 σ を含む項は空間自身の引きずりによるシアーから生じたものである。

5.2.4 ブラックホールの計量

ブラックホール周辺での天体現象を扱うためにはその曲がった時空構造を表す計量を使う必要がある⁵。ここでは回転しているブラックホール (カーブラックホール) の作る時空構造を考える。一般的にブラックホールは質量 M 、角運動量 J 、電荷 Q の 3 つの物理量を持つことができる。カーブラックホールはそのうちの 2 つ、質量 M と角運動量 J を持っている。ブラックホールの回転の度合いを表すパラメータとして回転パラメータ $a \equiv J/J_{max}$ を用いる。ここで $J_{max} = GM^2/c$ は質量 M を持つブラックホールが最大限に回転しているときに持つ角運動量である。Boyer-Lindquist 座標 $(x^0, x^1, x^2, x^3) = (ct, r, \theta, \phi)$ でカー時空の計量は

$$h_0 = \sqrt{1 - \frac{2r_g r}{\Sigma}}, \quad h_1 = \sqrt{\frac{\Sigma}{\Delta}}, \quad h_2 = \sqrt{\Sigma}, \quad h_3 = \sqrt{\frac{A}{\Sigma}} \sin \theta \quad (5.71)$$

と書かれる。ここで $r_g \equiv GM/c^2$ は重力半径、

$$\Delta = r^2 - 2r_g r + (ar_g)^2, \quad (5.72)$$

$$\Sigma = r^2 + (ar_g)^2 \cos \theta, \quad (5.73)$$

$$A = \{r^2 + (ar_g)^2\}^2 - \Delta (ar_g)^2 \sin^2 \theta, \quad (5.74)$$

である。この計量で lapse function は

$$\alpha = \sqrt{\Delta \Sigma / A}, \quad (5.75)$$

で与えられる。事象の地平面の半径は

$$r_H = r_g (1 + \sqrt{1 - a^2}), \quad (5.76)$$

である。これは $\alpha = 0$ を取るとき得られる。また、たまにブラックホールのシュバルツシルト半径 (Schwarzschild radius)、

$$r_S = 2GM/c^2 = 2r_g, \quad (5.77)$$

を用いることがある。

カーブラックホールでは、自身の回転のためにブラックホールの周りの時空を引きずる。これを時空の引きずり効果 (frame-dragging effect) と言う。この時空の引きずりによってブラックホールのすぐ外側にはエルゴ領域と呼ばれている特殊な領域ができる。この領域では、物質、エネルギー、情報

⁴ ここでの式は光速 c を含んだ形で書いてある。そのため $c \rightarrow \infty$ では非相対論 MHD の基礎方程式となる。

⁵ 平坦な時空 (ミンコフスキー時空) を扱うときは計量を $h_0 = -1, h_i = 1$ ($i = 1, 2, 3$) とすればよい。

のどれも全てブラックホールの回転と同じ方向に回転しなければならない。エルゴ領域の表面は $h_0 = 0$ 、即ち

$$r = r_g(1 + \sqrt{1 - a^2 \cos^2 \theta}), \quad (5.78)$$

で与えられる。エルゴ領域内では、shift velocity $c\beta^\phi$ は光速より速くなる。

ブラックホールが回転していない ($a = 0$) とき、カー時空の計量はシュバルツシルト時空の計量に変わる。このとき ブラックホールの時空の引きずり効果はなくなり、エルゴ領域もなくなる。また、shift velocity $c\beta^\phi$ はゼロになる。

5.3 GRMHD 方程式の数値解法

前の節で導出した GRMHD の基礎方程式を数値的に時間発展させる。数値的解法は様々なものがあり、第 4-1 節で紹介したように相対論的流体 (磁気流体) 力学においても様々な解法が用いられている。ここでは我々のグループが使用している simplified TVD method を簡単に紹介する。また時間発展によって得られる各時刻ごとの保存量から密度、圧力、速度といったそれぞれの物理量を導出するための方法を紹介する。

5.3.1 Simplified TVD Method

TVD とは total variation diminishing の略であり、時間発展において全変動 (total variation) が増加しないという条件である。その条件を満足する解法を TVD スキームと言う。TVD に関しては数値シミュレーションサマースクールテキスト第 1 章または藤井孝蔵の「流体力学の数値計算法」を参考にして頂きたい。ここでは Davis (1984) によって出された simplified TVD method について簡単に説明する。

まず 1 次元での保存形の方程式 $\partial u(x, t)/\partial t = -\partial w(u, x)/\partial x$ を考える。ここで u 、 w は一般に m 次元のベクトルである。simplified TVD method の手順は以下の通りである。

$$u_j^{(1)} = u_j^n - \kappa(w_j^n - w_{j-1}^n), \quad (5.79)$$

$$u_j^{(2)} = \frac{1}{2}[u_j^n + u_j^{(1)} - \kappa(w_{j+1}^{(1)} - w_j^{(1)})], \quad (5.80)$$

$$u_j^{n+1} = u_j^{(2)} + D_{j+1/2}^n - D_{j-1/2}^n, \quad (5.81)$$

ここで $\kappa \equiv \Delta t/\Delta x$ 、上付き添え字の n は時間ステップを下付き添え字の j はメッシュの位置を示している。最後の項の $D_{j+1/2}^n$ は以下のように定義される。

$$D_{j+1/2}^n = [\bar{K}_{j+1/2}^+(\nu, r_j^+) + \bar{K}_{j+1/2}^-(\nu, r_{j+1}^-)](u_{j+1}^n - u_j^n) \quad (5.82)$$

$$\nu = \max(|c_m|)\kappa \quad (5.83)$$

$$\bar{K}^\pm(\nu, r^\pm) = 0.5C(\nu)[1 - \phi(r^\pm)], \quad (5.84)$$

$$\phi(r) = \begin{cases} (2r, 1) & (r > 0) \\ 0 & (r \leq 0), \end{cases} \quad (5.85)$$

$$C(\nu) = \begin{cases} \nu(1 - \nu) & (\nu \leq 0.5) \\ 0.25 & (\nu > 0.5), \end{cases} \quad (5.86)$$

$$r_j^+ = (\Delta u_{j-1/2}^n, \Delta u_{j+1/2}^n) / (\Delta u_{j+1/2}^n, \Delta u_{j+1/2}^n), \quad (5.87)$$

$$r_j^- = (\Delta u_{j-1/2}^n, \Delta u_{j+1/2}^n) / (\Delta u_{j-1/2}^n, \Delta u_{j-1/2}^n) \quad (5.88)$$

$$\Delta u_{j+1/2}^n = u_{j+1}^n - u_j^n \quad (5.89)$$

c_j は各波の速度、 $(\mathbf{A}, \mathbf{B}) \equiv \mathbf{A} \cdot \mathbf{B}$ である。このスキームは Lax-Wendroff スキームに拡散項を付け加えたようなものとみなすことができる⁶。また、directional splitting method を用いてこのスキームを 2 次元に拡張したところ、弱い数値不安定性の発生が見られた。そこでこの不安定性を避けるため、我々のコードでは以下に示すように 2 つの係数 $\bar{K}^+$ と $\bar{K}^-$ を組み合わせて 1 つの係数 $\bar{K}$ として扱うように改良している。

$$\bar{K}(\nu, r^+, r^-) = 0.5C(\nu)[1 - \phi(r^+, r^-)] \quad (5.90)$$

$$\phi(r^+, r^-) = \max[0, (2r^+, r^-, 1), (2r^-, r^+, 1)] \quad (5.91)$$

5.3.2 保存量からの基本量の導出

保存形で記述された GRMHD の基礎方程式 (3+1 形式) を時間発展している間、各時間ステップごとに保存量 D 、 $\hat{\mathbf{P}}$ 、 ϵ 、 $\hat{\mathbf{B}}$ のみを得る。さらに時間発展を進めるためには 方程式 (5.36)-(5.38) と (5.40) を使ってこれら保存量 D 、 $\hat{\mathbf{P}}$ 、 ϵ 、 $\hat{\mathbf{B}}$ から基本量 γ 、 $\hat{\mathbf{v}}$ 、 p を計算しなければならない。これをするためには未知変数を $x \equiv \gamma - 1$ と $y \equiv \gamma(\hat{\mathbf{v}} \cdot \hat{\mathbf{B}})/c^2$ とする連立非線形代数方程式

$$\begin{aligned} & x(x+2) \left[\Gamma R x^2 + (2\Gamma R - d)x + \Gamma R - d + u + \frac{\Gamma}{2} y^2 \right]^2 \\ &= (\Gamma x^2 + 2\Gamma x + 1)^2 [f^2(x+1)^2 + 2\sigma y + 2\sigma x y + b^2 y^2], \end{aligned} \quad (5.92)$$

$$\begin{aligned} & \left[\Gamma(R - b^2)x^2 + (2\Gamma R - 2\Gamma b^2 - d)x + \Gamma R - d + u - b^2 + \frac{\Gamma}{2} y^2 \right] y \\ &= \sigma(x+1)(\Gamma x^2 + 2\Gamma x + 1), \end{aligned} \quad (5.93)$$

を解かなくてはならない。ここで $R = D + \epsilon/c^2$ 、 $d = (\Gamma - 1)D$ 、 $u = (1 - \Gamma/2)\hat{\mathbf{B}}^2/c^2$ 、 $f = \hat{P}/c$ 、 $b = \hat{B}/c$ 、 $\sigma = \hat{\mathbf{B}} \cdot \hat{\mathbf{P}}/c^2$ である。これらの代数方程式に含まれている磁場 $\hat{\mathbf{B}}$ を取ると、Duncan & Hughes (1994) によって導かれた相対論的流体力学のものになり、方程式 (5.93) は非常に簡単な式となる。これらの代数方程式を 2 変数の Newton-Raphson iteration method を用いて各 cell ごとに解く。そして各基本量は x 、 y 、 D 、 $\hat{\mathbf{P}}$ 、 ϵ 、 $\hat{\mathbf{B}}$ を使って

$$\gamma = 1 + x, \quad (5.94)$$

$$p = \frac{(\Gamma - 1)[\epsilon - x D c^2 - (2 - 1/\gamma^2)B^2/2 + (cy/\gamma)^2/2]}{[\Gamma x(x+2) + 1]} \quad (5.95)$$

$$\hat{\mathbf{v}} = \frac{\hat{\mathbf{P}} + (y/\gamma)\hat{\mathbf{B}}}{D + \{\epsilon + p + B^2/2\gamma^2 + (cy/\gamma)^2/2\}/c^2} \quad (5.96)$$

から簡単に求めることができる。この計算法は SRMHD シミュレーション (Koide, Nishikawa, & Mutel 1996; Koide 1997) で用いられているものと同じである。

⁶式 (5.79)-(5.81) を見れば分かるが、この simplified TVD method は左右非対称の数値解法になっている。そのため、衝撃波管のテスト計算などで、右向き (j が大きくなる向き) の衝撃波の伝播の計算と左向き (j が小さくなる向き) の衝撃波の伝播の計算とで計算限界に差が出るといった問題が発生している。即ち、右向きの衝撃波伝播の計算ができるのに、同じ問題 (同じ初期条件) で左向きの衝撃波伝播の計算ができないのである。これを解決するには、風上差分法のように流れに対する上流、下流を判別し、常に一定の方向で差分を取るようにすればよいであろう。

5.4 シミュレーションの計算例

5.4.1 宇宙ジェット

宇宙ジェットはジェット状に細く絞られたプラズマガスの噴流である。AGNで見られる宇宙ジェットは光速の95%以上の速度を持っている。このような速い速度を持つジェットがどのようにして形成され、加速されているのか大きな問題となっている。AGNで見られるような宇宙ジェットは中心のブラックホール近傍に渦巻いている降着円盤から噴き出していると考えられている。Koide, Shibata, & Kudoh (1998,1999)、Koide et al. (2000)、Aoki et al. (2004) ではこの宇宙ジェットの形成、加速に関して GRMHDコードを用いてシミュレーションしている。ここではカーブラックホール周りを回る降着円盤から形成される宇宙ジェットの GRMHD シミュレーションを行った Koide et al. (2000) を元にその初期条件と結果を簡単に紹介する。

初期は以下のような状況を考える。中心にカーブラックホールがあり、その周りをケプラー速度で回転する幾何学的に薄く、冷たいプラズマガスの降着円盤が回っているとする。このとき、ブラックホール周りのケプラー速度は

$$v_K = \frac{c}{\sqrt{2(r/r_s - 1)}}, \quad (5.97)$$

で与えられる⁷。降着円盤の周りには高温で希薄なプラズマガス (コロナ) がある。コロナはブラックホールに向かって自由落下している⁸。磁場は降着円盤を貫く、大局一様磁場を仮定する。この磁場の分布はカーブラックホール周りの一様磁場を表す Wald 解 (Wald 1974) を使用する。Wald 解のベクトルポテンシャルは

$$A_\mu = \frac{B_0}{2}(g_{\mu 3} + 2ar_g g_{\mu 0}), \quad (5.98)$$

である。ここで B_0 は磁場の強さを示すパラメータである。Boyer-Lindquist 座標では、これは

$$\hat{B}_r = B_0 \frac{\cos \theta}{\sqrt{A}} \left[\delta + \frac{2r_g r(r^4 - (ar_g)^4)}{\Sigma^2} \right], \quad (5.99)$$

$$\hat{B}_\theta = -B_0 \sqrt{\frac{\delta}{A}} \sin \theta \left[r - r_g + \frac{r_g}{\Sigma^2} \{ (r^2 + (ar_g)^2 \Sigma + 2(ar_g)^2 \cos^2 \theta (r^2 - (ar_g)^2) \} \right], \quad (5.100)$$

で与えられる。この計算では $B_0 = 0.3(\rho_0 c^2)^{-1/2}$ とする。 ρ_0 は $r = 3r_S = 6r_g$ での初期コロナの密度である。

シミュレーションでは、降着円盤がブラックホールと共回転 ($a = 0.95$) と反回転 ($a = -0.95$) している場合で行っている。シミュレーションの時間発展の様子は図 5.1 で示されている。ブラックホールと共回転のときはシュバルツシルトブラックホールのときとほぼ同じく、磁場によって角運動量を失った円盤のガスはブラックホールに向かって落ち込む。その降着流は遠心力バリアによって止められ、衝撃波を形成する。形成した衝撃波は高いガス圧を生み、落ち込んでくるガスをジェットとして外側に噴き出す。このようにシミュレーションではガス圧によって加速された速いジェットとその外側に磁気圧によって加速された遅いジェットが形成された (図 5.1(d))。一方で、反回転の場合、ガス圧ジェットの内側でより強い磁気圧ジェットの形成が見られた (図 5.1(c))。これは、降着流が遠心力バリアによって止められることなく、ブラックホールまで回転しながら落ち込み、エルゴ領域で

⁷非相対論ではケプラー速度は $v_K = \sqrt{GM/r}$ である

⁸ブラックホールへの自由落下の定常解は Bondi & Hoyle (1944) で詳しく解析されている。このシミュレーションでの自由落下の場合の密度、圧力、速度分布は Koide et al. (1999) を参照。

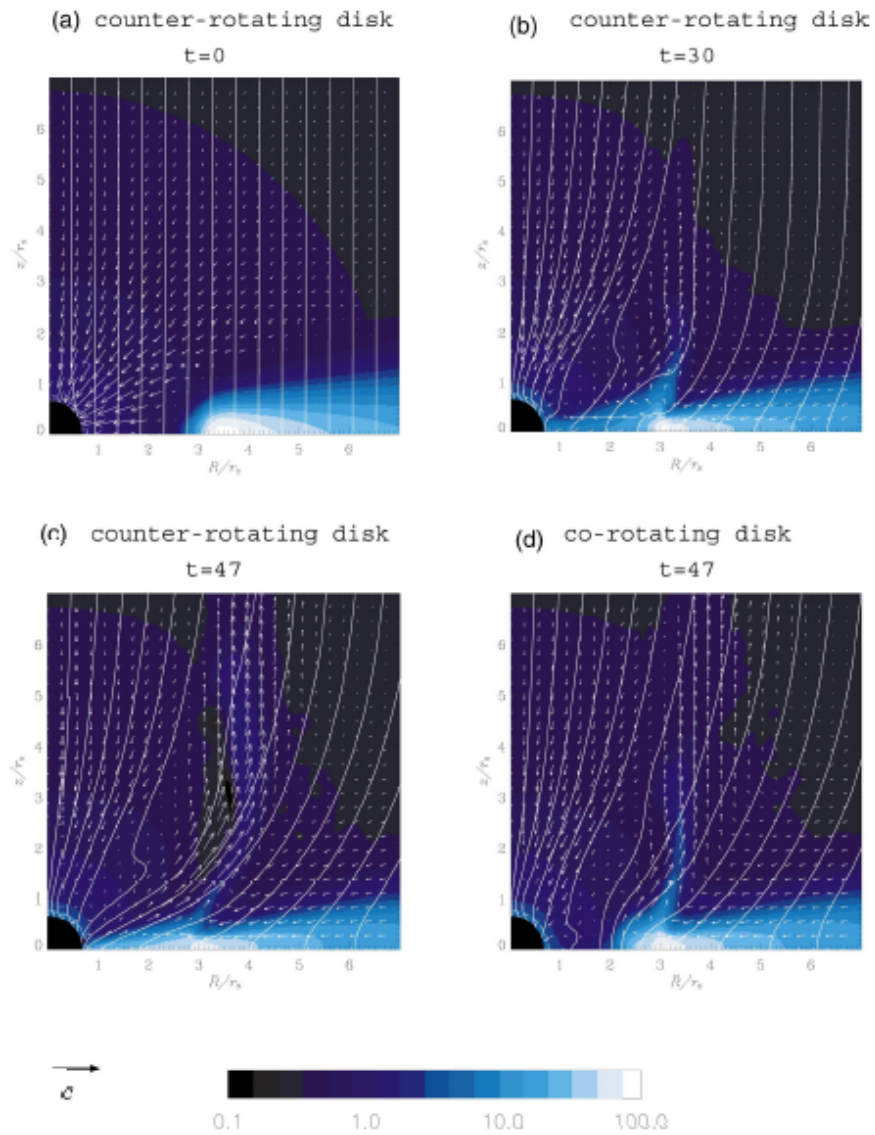


図 5.1: カーブラックホール周りを回る降着円盤からの宇宙ジェット的时间発展の様子。カラーは密度線は磁力線、矢印は速度を表している。磁場の効果によって降着円盤よりジェットは噴出している様子が分かる。(Koide et al. 2000)

の時空の引きずり効果によって強く捻られた磁場によってジェットとして噴出したものである。このように、降着円盤だけでなく、ブラックホール自身の回転の効果(時空の引きずりの効果)によってジェットが形成されるといった描像が GRMHD シミュレーションによって初めて示された。

5.4.2 ブラックホールからのエネルギーの引き抜き

前で示したように降着円盤から噴き出しているジェットのエネルギーは降着円盤の回転エネルギーである。それ以外の可能性としてブラックホールの回転エネルギーがある。このブラックホールの回転エネルギーを引き出しうることを初めて示したのは Penrose (1969) である。エルゴ領域で物質が逆方向の角運動量を持つ場合、物質のエネルギー (energy-at-infinity) は負になりうる。物質のエネルギーはカーブラックホールの周りの粒子の保存量の一つで、非相対論では静止質量エネルギー、運動エネルギー、重力エネルギーの和に相当する。エルゴ領域の外ではこの物質のエネルギーは必ず正となる。エルゴ領域内で正のエネルギーを持つ 2 つの粒子の角運動量を再配分することを考える。角運動量の再配分によって、一方の粒子に負のエネルギーを、もう一方の粒子に 2 つのエネルギーの和よりも大きい正のエネルギーを持つような状況を作る。その負のエネルギーを持つ粒子がブラックホールに落ち込み、正のエネルギーを持つ粒子がエルゴ領域から飛び出したとき、それはブラックホールの回転エネルギーが引き抜かれ、飛び去った粒子によって持ち出されたこととなる。このようにブラックホールの回転エネルギーを負の粒子を媒介にして引き抜く機構を“ペンローズ過程”と言う。また、Blandford & Znajek(1977) は Force-Free 条件下でのカーブラックホール周りでの磁場分布の定常軸対象の解を求め、ブラックホールからのエネルギー引き抜きの機構を示した。Koide et al.(2002)、Koide (2003) では GRMHD コードを用いてカーブラックホールからの磁気エネルギーの引き抜きについて数値的に調べ、“MHD ペンローズ過程”によるエネルギーの引き抜きの可能を示した。ここでは Koide (2003) を元に、その初期条件とシミュレーション結果について簡単に紹介する⁹。

初期条件として最大回転しているカーブラックホール ($a = 0.99995$) の周りに大局一様磁場と希薄なプラズマがある状況を考える。ブラックホール周りの希薄なプラズマは空間的に一様な密度 ρ_0 と圧力 $p_0 = 0.06\rho_0 c^2$ を持っている。大局一様磁場は Wald 解 (5.99)、(5.100) で与えられる。ここで $B_0 = 5.77\sqrt{\rho_0 c^2}$ とする。

図 5.2 はシミュレーションの時間発展の様子を示している。シミュレーション開始とともに、プラズマはブラックホールに向かって落ち込む (図 5.2A)。磁場は時空の引きずり効果によって捻られ、その捻れはアルフベン波としてプラズマの流れに逆らって磁力線に沿って外向きに伝播していく (図 5.2B)。特にブラックホールの地平面近くにできるエルゴ領域では磁場の捻れは強い (図 5.2C、D)。エルゴ領域内では磁場はブラックホールと同じ方向に曲げられる。曲げられた磁場は磁気張力によって元に戻ろうとするためエルゴ領域内のプラズマの角運動量を負にする (もしブラックホールの角運動量が正なら)。プラズマが失った角運動量は磁場によってエルゴ領域外に運ばれる。このように、磁場によるプラズマの角運動量の再配分によってペンローズ過程と同様にエルゴ領域内で負のエネルギー状態を取るプラズマが形成される可能性がある。負のエネルギーを持つプラズマがブラックホールに吸い込まれると、ブラックホールの全エネルギーが減少する。一方で、エルゴ領域から放出されたアルフベン波がプラズマの落下に逆らって外向きに伝播する。これはブラックホールのエネルギーが引き抜かれてアルフベン波として放出されたと見ることができる。このシミュレーションによって示された機構はペンローズ過程とよく似ている。それは、ペンローズ過程が粒子の角運動量の再配分によってエネルギーの引き抜きを行うのに対してこのシミュレーションで示されたのは磁気張力によ

⁹ この研究の日本語の解説は物理学会誌 2003 年 10 月号を参照

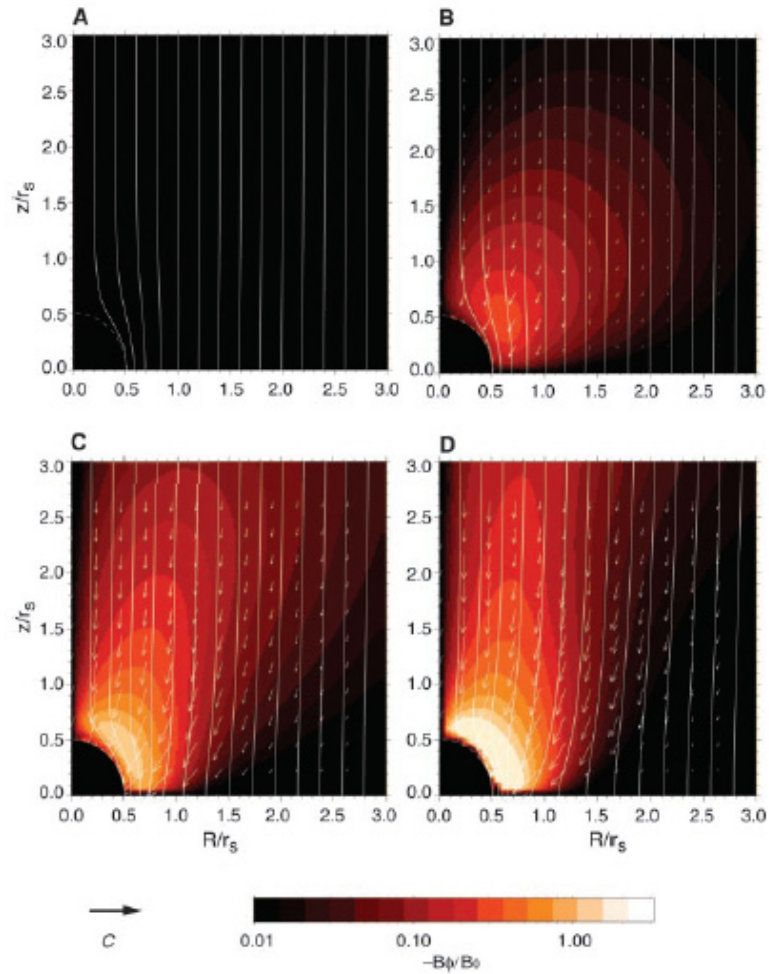


図 5.2: カーブラックホールとその周りに一様な磁場と希薄なプラズマのある系での時間発展の様子。矢印はポロイダル速度、実線は磁力線、カラーは $-B_\phi/B_0$ を表している。時間とともに中心付近で磁場が強く捻られている様子が分かる。(Koide et al. 2002)

て角運動量の再配分を行っている点である。このような磁氣的な引き抜き機構を“MHD ペンローズ過程”(Hirokuni et al. 1992) と言う。

Koide et al. (2002)、Koide (2003) では GRMHD シミュレーションによって、磁場によるブラックホールの回転エネルギーの引き抜き機構とその可能性について示した。しかし、残念ながらこのシミュレーションでは、Boyer-Lindquist 座標を用いているため、角運動量の輸送はブラックホールの地平面を横切っていない。そのため本当にブラックホールのエネルギーが引き抜かれたのか見ることができていない。地表面の見かけの特異点を避ける座標として Kerr-Schild 座標がある。今後、このような座標系での計算が必要であろう¹⁰。

5.5 現状の問題点と今後

前節で示したように我々のコードは今までに宇宙ジェット(Koide, Shibata, & Kudoh 1998, 1999; Koide et al. 2000; Aoki et al. 2004) やブラックホールからのエネルギーの引き抜き (Koide et al. 2002; Koide 2003) やガンマ線バースト(Mizuno et al. 2004a, b) など様々な天体物理現象に応用され、新たな画像を描き出してきた。ここではあえて我々のコードの直面している問題点と相対論的 MHD コードの今後の発展について書いていく。

我々のコードの最大の問題点は非常に狭いパラメータ範囲でしか長く時間発展を追うことができない点である。例えば、光速に近い速度 (ローレンツ係数 $\gamma > 3$) や強磁場中でのシミュレーションでは長く時間発展を追うことは難しいことが分かっている。我々のコードで行ったシミュレーションが止まるほとんどの原因は保存量から各基本量に戻すときに使用している Newton-Raphson iteration method で正しい解を見つけられない (収束しない) ためである。使用している基礎方程式が保存形である以上、保存量から各物理量への変換は避けては通れない問題である。現在我々は 2 変数の連立代数方程式まで式を組み合わせる Newton-Raphson iteration method を使用している点に問題があるのかもしれない。このことは次章で詳しく取り上げているが、我々の行っている 2 変数での連立代数方程式より Del Zanna et al. (2002) で用いられている 1 変数の代数方程式を Newton-Raphson iteration method で解く方法の方がより高いローレンツ係数での計算が可能であることが分かった。また Del Zanna et al. (2002) で用いている方法は、簡単なテスト計算ではローレンツ係数で $\gamma \sim 1000$ を超える計算も可能であることが分かった。

また、現在の問題を避ける方法として、基礎方程式の段階で保存形から非保存形へと基礎方程式を書き換えてしまうという方法がある。非保存形に式を書き下してしまえば、我々のコードで最も問題となっている保存量から各物理量への変換が必要なくなるのである。De Villiers & Hawley (2003) では、ZEUS コードを元に、このような非保存形の GRMHD 基礎方程式を用いた GRMHD コードを作り、ブラックホール周りの降着円盤の 3 次元シミュレーションで非常に長時間の計算を行っている。また非保存形でもっとも心配な点とされる物理量の保存精度に対しても De Villiers & Hawley (2003) では数 % 程度 (質量では 2% 程度) のエラーである。もちろん行うシミュレーションモデルや使用する数値解法によってこの保存精度は異なるが、保存形では解けなかった非常に光速に近い状況下での計算や強磁場での計算などが可能になるかもしれない。

(一般) 相対論的 MHD コードの今後の発展としては大きく 2 つ課題がある。1 つは理想 MHD 条件をはずすこと、即ち、電気抵抗を加えた相対論的 MHD コードの開発である。このようなシミュレーションは相対論的な状況下での磁気リコネクションの物理の解明など多くの研究において必要とされ

¹⁰Kerr-Schild 座標を用いたカーブラックホールからのエネルギーに引き抜きの GRMHD シミュレーションは McKinney & Gammie (2004) によって行われている。Kerr-Schild 座標での計量の与え方はこの論文を参考にして頂きたい。

ている。これをするためにはまずは基礎方程式の段階からの書き換えが必要である。特に今は磁場の時間発展だけを解いているが、理想 MHD 条件がなくなるため電場の時間発展も同時に解く必要がある。もう一つの課題は計量の時間発展を同時に解く GRMHD コードの開発である。現在行われているシミュレーションの多くはブラックホールの質量が周りの物質に対して非常に大きいためシミュレーションを行っている時間スケールではブラックホールの進化はほぼ無視できる状況である。しかしながら、例えば、大質量星の重力崩壊などで形成されたブラックホール周りでの物理を扱うときはシミュレーションを行う時間スケールでのブラックホールの進化を無視することは非現実的である。このような状況下でのシミュレーションを行うにはブラックホールの進化に伴う計量の時間発展も解く必要がある。これを行うためには計量の時間発展を表すアインシュタイン方程式を合わせて解かなければならない。これは非常に困難な課題である。ここでは詳しく述べることは控えるが、一般相対論的流体力学においてはここ 10 年で精力的に研究が行われ、大きな発展が見られる。特に日本では京大の中村卓史氏、東大の柴田大氏を中心に大きな成果を挙げてきている。それらに関しては日本語では 2002 年度理論懇シンポジウム集録を英語では Baumgarte & Shapiro (2003b) のレビューや Wilson & Mathews (2003) の教科書を参考にして頂きたい。また GRMHD においては非保存形ではあるが、Baumgarte & Shapiro (2003a) によってすでに計量の時間発展を含めた GRMHD 方程式が定式化されている。実際に GRMHD シミュレーションを行った結果が出てくるのは近い将来かもしれない。

この章の第 1 節でも述べたが、相対論的 MHD コードはまだ世界的にも数は少なく、特に一般相対論的 MHD コードにいたってはここ 5 年近く我々のコードが世界をリードしてきた。また RMHD コードの開発に関する論文は出ているが、その後実際に天体現象に応用した論文があまり出ていないものが多いように感じる。これは、テスト計算のような簡単な問題は解けるが、実際の天体現象のような複雑な問題ではうまく動いていない状況にあるのではないだろうか。先程も書いたように我々の使用しているコードには幾つか限界がある。しかし、実際の天体現象に応用され、現在もっとも成果をあげている GRMHD コードであるとも言える。我々の相対論的 MHD コードの方法がこれから相対論的 MHD コードを作ろうとしている人にとって参考になるものであることを願う。

参考文献

- [1] Balsara, D. 2001, ApJS, 132, 83
- [2] Baumgarte, T. W., & Shapiro, S. L. 2003b, ApJ, 585, 921
- [3] ———. 2003a, Phys. Rep. 376, 41
- [4] Blandford, R. D., & Znajek, R. 1977, MNRAS, 179, 433
- [5] Bondi, H., & Hoyle, F., 1944, MNRAS, 104, 273
- [6] Duncan, G. C., & Hughes, P. A. 1994, ApJ, 436, L119
- [7] Davis, S. F. 1984, NASA Contractor Rep. 172373, ICASE Rep., No. 84-20
- [8] Del Zanna, L., Bucciantini, N., & Londrillo, P. 2003, A&A, 400, 397
- [9] De Villiers, J. -P., & Hawley, J. F. 2003a, ApJ, 589, 458
- [10] ———. 2003b, ApJ, 592, 1060
- [11] De Villiers, J. -P., Hawley, J. F., & Krolik, J. H. 2003, ApJ, 599, 1238
- [12] De Villiers, J. -P., Hawley, J. F., Krolik, J. H., & Hirose, S. 2004, ApJ, in press (Astroph/0407092)
- [13] Font, J. A. 2003, LRR, 6, 4
- [14] Gammie, C., McKinney, J.C., & Tóth 2003, ApJ, 589, 444
- [15] Hirovani, K., Takahashi, M., Nitta, S-Y., & Tomimatsu, A. 1992, ApJ, 386, 455
- [16] Hirose, S., Krolik, J. H., De Villiers, J. -P., & Hawley, J. F. 2004, ApJ, 606, 1083
- [17] Koide, S. 1997, ApJ, 487, 66
- [18] Koide, S. 2003, Phys. Rev. D, 67, 104010
- [19] Koide, S., Meier, D. L., Shibata, K., & Kudoh, T. 2000, ApJ, 536, 668
- [20] Koide, S., Nishikawa, K.-I., & Mutel, R. L. 1996, ApJ, 463, L71
- [21] Koide, S., Shibata, K., & Kudoh, T. 1998, ApJ, 495, L63

- [22] ———. 1999, *ApJ*, 522, 727
- [23] Koide, S., Shibata, K., Kudoh, T., & Meier, D. L. 2002, *Science*, 295, 1688
- [24] Koldoba, A. V., Kuznetsov, O. A., & Ustyugova, G. V. 2002, *MNRAS*, 333, 932
- [25] Komissarov, S. S. 1999, *MNRAS*, 303, 343
- [26] Marti, J. M., & Müller, E. 2003, *LRR*, 6, 7
- [27] McKinney, J. C., & Gammie, C. F. 2004, *ApJ*, 611, 977
- [28] Mirabel, I. F., & Rodrigues, L. F. 1994, *Nature*, 371, 46
- [29] Mizuno, Y., Yamada, S., Shibata, K., & Koide, S. 2004a, *ApJ*, 606, 395
- [30] ———. 2004b, *ApJ*, in press
- [31] Penrose, R. 1969, *Nuovo Cimento I*, 252
- [32] Tingay, S. J., et al. 1995, *Nature*, 374, 141
- [33] Thorne, Kip S., Price, R. H., & Macdonald, D. A. 1986, *Membrane Paradigm* (Yale University Press, New Haven and London)
- [34] van Putten, M. H. P. M. 1993, *J. Comput. Phys.*, 105, 339
- [35] Wald, R. M., 1974, *Phys. Rev. D*, 10, 1680
- [36] Wilson, J. R. 1977, in *Proceedings of the Marcel Grossman Meeting*, ed. R. Ruffini (Amsterdam: North Holland), 393
- [37] Wilson, J. R. & Mathews, G. J. 2003, *Relativistic Numerical Hydrodynamics* (Cambridge University Press, Cambridge)
- [38] Yokosawa, M. 1993, *PASJ*, 45, 207
- [39] 宇宙シミュレーション・ネットラボラトリーシステム開発グループ 2004, 天体とスペースプラズマのシミュレーションサマースクール 講義テキスト
- [40] 小出真路 2003, *日本物理学会誌*, 58, 743
- [41] 柴田大 2002, 2002 年度理論懇シンポジウム集録
- [42] 藤井孝蔵 1994, *流体力学の数値計算法* (東大出版会, 東京)

第 6 章

相対論的磁気流体における保存量からの基本量の導出

千葉大学 富高 真

6.1 はじめに

相対論的 Jet のシミュレーション等のため、近年相対論的磁気流体 (RMHD) シミュレーションが行われるようになってきた。この章では基本変数として、エネルギー運動量等の保存変数を用いた RMHD コードについて紹介する。非相対論的な場合との大きな違いは、保存変数から圧力、速度などの基本変数に変換する際 iteration が必要になることであり、この部分の計算コストが最も大きくなる。また現在、Black Hole 付近の光速に非常に近い速度で動く磁気流体 ($\gamma \geq 10$) における iteration (反復法) の解が収束しないという問題に直面している。そこで現在我々の使用している 2-D Newton's method (Koide et al. : 1999) と、さらに収束が良いとされる 1-D Newton's method (Del Zanna et al. : 2002) についての収束範囲や最大収束回数について比較した結果を報告する。計算スキームとして、Lax-Wendroff 法と簡易化した Riemann solver を用いる方法 (HLL) を使用した。

6.2 相対論的磁気流体力学方程式(保存系)

相対論的に動く理想的な流体・磁気流体方程式は非相対論的な場合と同様、保存系で次の様に記される。(ここでは $c = 1$, $\mu_0 = \epsilon_0 = 1$, MKS 単位系とする) <<参考 : Koide (1999)>>

- 粒子数保存の法則

$$\nabla_\mu (\rho u^\mu) = 0 \quad (6.1)$$

- 運動量・エネルギー保存の法則

$$\nabla_\nu T^{\mu\nu} = 0 \quad (6.2)$$

- Maxwell 方程式

$$\partial_\mu F_{\nu\lambda} + \partial_\nu F_{\lambda\mu} + \partial_\lambda F_{\mu\nu} = 0 \quad (6.3)$$

$$\nabla_\mu F^{\mu\nu} = -J^\nu \quad (6.4)$$

ここで $g^{\mu\nu}$ (メトリック), $x^\nu = (t, x^i)$, $u^\nu = (\gamma, \gamma v^i)$ (四元ベクトル)
 $F_{\mu\nu}$ (電地場テンソル), J^ν (四元電流密度), ρ (固有密度), p_r (固有圧力)
 $e = \rho c^2 + p_r/(\Gamma - 1)$ (固有エネルギー), Γ (比熱比), γ (ローレンツ係数)
 ∇_μ (共変微分), $\mu = \nu = (0, 1, 2, 3)$, $i = (1, 2, 3)$ とする。

相対論的磁気流体の運動量エネルギーテンソル $T^{\mu\nu}$ は次のように記される。

$$T^{\mu\nu} = p_r g^{\mu\nu} + (e + p_r) u^\mu u^\nu + F_\sigma^\mu F^{\nu\sigma} - \frac{1}{4} g^{\mu\nu} F^{\lambda\kappa} F_{\lambda\kappa} = T^{\nu\mu} \quad (6.5)$$

簡単に考えるため特殊相対論 ($g^{\mu\nu} = \text{diag}[-1, 1, 1, 1] = \eta^{\mu\nu}$) で考える。
 $\omega = e + p_r$, $b^\mu = [\gamma(\mathbf{v} \cdot \mathbf{B}), \mathbf{B}/\gamma + \gamma(\mathbf{v} \cdot \mathbf{B})\mathbf{v}]$ とすると (6.1), (6.2), (6.3), (6.4) は次のように書き換えられる。<<参考 : Del Zanna (2002)>>

$$\partial_\mu(\rho u^\mu) = 0 \quad (6.6)$$

$$\partial_\mu[(\omega + |b|^2)u^\mu u^\nu - b^\mu b^\nu + (p_r + \frac{|b|^2}{2})\eta^{\mu\nu}] = 0 \quad (6.7)$$

$$\partial_\mu(u^\mu b^\nu - u^\nu b^\mu) = 0 \quad (6.8)$$

ここで

$$D = \gamma\rho \quad (6.9)$$

$$\mathbf{P} = \omega\gamma^2\mathbf{v} + \mathbf{E} \times \mathbf{B} \quad (6.10)$$

$$\varepsilon = \omega\gamma^2 - p_r - D + \frac{1}{2}(B^2 + E^2) \quad (6.11)$$

とすると保存量 $\mathbf{U} = (D, \mathbf{P}, \varepsilon, \mathbf{B})$ から

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial \mathbf{F}}{\partial x} \quad (6.12)$$

$$(6.13)$$

このように相対論磁気流体も保存系で数値解析することができる。

6.3 反復解法の必要性

なぜ相対論では iteration (反復解法による保存量からの基本量の導出) が必要であるのか？

まず、非相対論における保存量 $(\rho, \mathbf{P}, \epsilon, \mathbf{B})$ から基本量 $(p_r, \mathbf{v})$ の導出について考える。これは簡単で次の様に記される。

$$p_r = (\Gamma - 1)(E - \frac{1}{2}\rho v^2 - \frac{B^2}{2}) \quad (6.14)$$

$$\mathbf{v} = \frac{\mathbf{P}}{\rho} \quad (6.15)$$

ここで $\mathbf{P}$ は運動量ベクトル、 $E = \frac{v^2}{2} + \frac{p_r}{(\Gamma - 1)\rho}$ は全エネルギーとする。

次に相対論である場合、保存量 $\mathbf{U}$ を決定した後、次のように基本量 $(p_r, \mathbf{v}, \rho)$ を導出する。

$$p_r = \frac{\Gamma - 1}{\Gamma(\gamma^2 - 1) + 1} \left[\epsilon - (\gamma - 1)D - \frac{B^2}{2}(2 - \frac{1}{\gamma^2}) + \frac{(\mathbf{v} \cdot \mathbf{B})^2}{2} \right] \quad (6.16)$$

$$\mathbf{v} = \frac{\mathbf{P} + (\mathbf{v} \cdot \mathbf{B})\mathbf{B}}{D + \epsilon + p_r + \frac{B^2}{2\gamma^2} + \frac{(\mathbf{v} \cdot \mathbf{B})^2}{2}} \quad (6.17)$$

このように相対論の場合、簡単に速度と圧力が求まらないために iteration が必要となりその分の計算コストがかかってしまう。

6.4 2-D Newton's method (KO : Koide et al. 1999)

前に述べたように保存量から基本量を導出する際、相対論では (6.16), (6.17) で見られるように簡単には求めることが出来ない。そこで Koide(1999) の 2-D Newton's method について紹介する。

(6.16), (6.17) を $X = \gamma - 1$, $Y = \gamma(\mathbf{v} \cdot \mathbf{B})$ を使って書き換えると

$$p_r = \frac{\Gamma - 1}{\Gamma X(X + 2) + 1} \left[\epsilon - XD - \frac{B^2}{2}(2 - \frac{1}{(1 + X)^2}) + \frac{Y^2}{2(1 + X)^2} \right] \quad (6.18)$$

$$\mathbf{v} = \frac{\mathbf{P} + \frac{Y}{1 + X}\mathbf{B}}{D + \epsilon + p_r + \frac{B^2}{2(1 + X)^2} + \frac{Y^2}{2(1 + X)^2}} \quad (6.19)$$

となる。(6.19) に (6.18) を代入した $\mathbf{v}$ に $\mathbf{P}$ との内積 ($\mathbf{v} \cdot \mathbf{P}$)、 $\mathbf{B}$ との内積 ($\mathbf{v} \cdot \mathbf{B}$) と ($\mathbf{v} \cdot \mathbf{v}$) を使用すると

$$\begin{aligned} Y[\Gamma(a - \beta^2)X^2 + (2\Gamma a - 2\Gamma\beta^2 - d)X + \Gamma a - d + b - \beta^2 + \frac{\Gamma Y^2}{2}] \\ = (\Gamma X^2 + 2\Gamma X + 1)\sigma(X + 1) \end{aligned} \quad (6.20)$$

$$\begin{aligned} X(X + 2)[a\Gamma X^2 + (2\Gamma a - d)X + \Gamma a - d + b + \frac{\Gamma Y^2}{2}]^2 \\ = [\tau^2(1 + X)^2 + Y^2\beta^2 + 2\sigma(1 + X)Y](\Gamma X^2 + 2\Gamma X + 1)^2 \end{aligned} \quad (6.21)$$

ここでは次の変数を使った。

$$a = D + \varepsilon \quad d = (\Gamma - 1)D \quad b = (1 - \frac{\Gamma}{2})B^2 \quad \tau = |\mathbf{P}| \quad \beta = |\mathbf{B}| \quad \sigma = \mathbf{B} \cdot \mathbf{P}$$

(6.20), (6.21) より X と Y だけの関数になっていることがわかる。2-D Newton's method を使用し X と Y を求めることにより、(6.9), (6.18), (6.19) から基本量 (ρ , p_r , $\mathbf{v}$) を導出することができる。

ここで 2-D Newton's method とは関数 $f(X, Y) = 0$ $g(X, Y) = 0$ が存在するとき、

$$\begin{aligned} f(X + \delta X, Y + \delta Y) &= f(X_0, Y_0) + \frac{df(X_0, Y_0)}{dX} \delta X_1 + \frac{df(X_0, Y_0)}{dY} \delta Y_1 = 0 \\ g(X + \delta X, Y + \delta Y) &= g(X_0, Y_0) + \frac{dg(X_0, Y_0)}{dX} \delta X_1 + \frac{dg(X_0, Y_0)}{dY} \delta Y_1 = 0 \end{aligned}$$

$$\begin{bmatrix} \delta X_1 \\ \delta Y_1 \end{bmatrix} = - \begin{bmatrix} \frac{df}{dX} & \frac{df}{dY} \\ \frac{dg}{dX} & \frac{dg}{dY} \end{bmatrix}^{-1} \begin{bmatrix} f(X_0, Y_0) \\ g(X_0, Y_0) \end{bmatrix} \quad (6.22)$$

このようにして $X_1 = X_0 + \delta X_1$, $Y_1 = Y_0 + \delta Y_1$ を求め、さらに $f(X_1 + \delta X_2, Y_1 + \delta Y_2) = 0$, $g(X_1 + \delta X_2, Y_1 + \delta Y_2) = 0$ とし同様に X_2, Y_2 を求めていく。何度か繰り返し $\Delta = (\delta X_i^2 + \delta Y_i^2)^{\frac{1}{2}} \sim 0$ となるところで $X_{next} = X_0 + \delta X_1 + \dots + \delta X_i$, $Y_{next} = Y_0 + \delta Y_1 + \dots + \delta Y_i$ とし、この値を使用して以下のようにして基本量を導き出す。

$$\gamma_{next} = 1 + X \quad (6.23)$$

$$p_{r \ next} = \frac{\Gamma - 1}{\Gamma X(X + 2) + 1} \left[\varepsilon - DX - \frac{B^2}{2} \left(2 - \frac{1}{(X + 1)^2} \right) + \frac{1}{2} \left(\frac{Y}{X + 1} \right)^2 \right] \quad (6.24)$$

$$\mathbf{v}_{next} = \frac{\mathbf{P} + \frac{Y\mathbf{B}}{1+X}}{D + \varepsilon + p_r + \frac{B^2}{2(1+X)^2} + \frac{1}{2}\left(\frac{Y}{1+X}\right)^2} \quad (6.25)$$

6.5 1-D Newton's method (DZ : Del Zanna et al. 2002)

6.5.1 Del Zanna の 1-D Newton's method

この方法は前節同様、反復法 (Newton's method) を使うのだが、1 変数だけを使用する。Del Zanna et al. (2002) によると、この方法を使うと光速に非常に近い磁気流体 ($\gamma_{max} = 1000$) でも収束すると記されている。以下、この方法について紹介する。

(6.9),(6.10),(6.11) に注目しよう。 $\omega\gamma^2 = W$ とすると次のように書き下すことができる。

$$D = \gamma\rho \quad (6.26)$$

$$\mathbf{P} = (W + B^2)\mathbf{v} - (\mathbf{v} \cdot \mathbf{B})\mathbf{B} \quad (6.27)$$

$$\varepsilon = W - p_r - D + \frac{1}{2}[B^2 + v^2B^2 - (\mathbf{v} \cdot \mathbf{B})^2] \quad (6.28)$$

(6.27) の 2 乗を取ると

$$W^2v^2 + (2W + B^2)B^2\mathbf{v}_\perp^2 - P^2 = 0 \quad (6.29)$$

ここで

$$B^2v_\perp^2 = B^2v^2 - (\mathbf{v} \cdot \mathbf{B})^2 \quad (6.30)$$

とする。また (6.27) より

$$\begin{aligned} \mathbf{v} &= \frac{1}{W + B^2}[\mathbf{P} + (\mathbf{v} \cdot \mathbf{B})\mathbf{B}] \\ &= \frac{1}{W + B^2}[\mathbf{P} + \frac{S}{W}\mathbf{B}] \end{aligned} \quad (6.31)$$

ここで $S = (\mathbf{P} \cdot \mathbf{B}) = W(\mathbf{v} \cdot \mathbf{B})$ である。(6.31) を 2 乗を取ると

$$v^2 = \frac{1}{(W + B^2)^2} \left[P^2 + \frac{S^2}{W^2} B^2 + \frac{2S}{W} (\mathbf{P} \cdot \mathbf{B}) \right] \quad (6.32)$$

これを使って

$$\begin{aligned} B^2 v_\perp^2 &= \frac{B^2 P^2 - S^2}{(W + B^2)^2} \\ &= \frac{T^2}{(W + B^2)^2} \end{aligned} \quad (6.33)$$

ここで $T^2 = B^2 P^2 - S^2$ である。(6.28) について (6.33) を代入すると

$$W - p_r - D + \frac{1}{2} B^2 + \frac{1}{2} \frac{T^2}{(W + B^2)^2} - \varepsilon = 0 \quad (6.34)$$

$$\omega = \rho + \frac{\Gamma}{\Gamma - 1} p_r = \rho + \Gamma_1 p_r \Leftrightarrow p_r = \frac{1}{\Gamma_1} [(1 - v^2)W - \rho]$$

を使って

$$\left[\left(1 - \frac{1 - v^2}{\Gamma_1}\right)W + \frac{\rho}{\Gamma_1} - D + \frac{1}{2} B^2 - \varepsilon \right] (W + B^2)^2 + \frac{T^2}{2} = 0 \quad (6.35)$$

このことから W は v^2 だけの関数であることが分かる。すなわち、(6.29) に (6.33) を代入した

$$W^2 v^2 + (2W + B^2) \frac{T^2}{(W + B^2)^2} - P^2 = 0$$

に関して、 $v^2 = \xi$ とし

$$F(\xi) = W^2 \xi + (2W + B^2) \frac{T^2}{(W + B^2)^2} - P^2 \quad (6.36)$$

について、前節と同様 Newton' methodにより解くとよいことが分かる。変数が W と ξ の二つあるように思えるが、(6.35) について Cardano による三次方程式を解くことで W の値を保存量から決定することができる。 $F'(\xi)$ は簡単に求めることができる。

$$F'(\xi) = W^2 + 2WW' \left[\xi - \frac{T^2}{(W + B^2)^3} \right] \quad (6.37)$$

$$\delta \xi_1 = - \left(\frac{dF(\xi_0)}{d\xi} \right)^{-1} F(\xi_0) \quad (6.38)$$

前節と同様に反復させて ξ_{next} を求めることができる。

6.5.2 Cardano's method

<<参考 : internet page Cardano による三次方程式の解法>>
 ここで Cardano の三次方程式について触れておく。次の三次方程式を考える。

$$x^3 + ax^2 + bx + c = 0 \quad (6.39)$$

ここで

$$y = x + \frac{a}{3}$$

とおいて 2 次の項に落す。すると

$$y^3 + 3py + q = 0 \quad (6.40)$$

となる。ここで

$$p = -\frac{a^2}{9} + \frac{b}{3}, \quad q = \frac{2a^3}{27} - \frac{ab}{3} + c$$

である。いま、2 つの未知数 (u, v) を導入して

$$y = u + v$$

とおいて (6.40) に代入すれば

$$(u^3 + v^3 + q) + 3(u + v)(uv + p) = 0$$

となる。 u, v に関する連立方程式

$$u^3 + v^3 + q = 0, \quad (uv + p) = 0 \quad (6.41)$$

の解 (u, v) が求められれば、 $y = u + v$ は方程式の解となる。さらに 1 の 3 乗根 $w_1 = (-1 + \sqrt{3}i)/2$, $w_1^2 = w_2 = (-1 - \sqrt{3}i)/2$ をとれば、 (uw_1, vw_2) (uw_2, vw_1) も解となる。さらに $(y - (u + v))(y - (uw + vw^2))(y - (uw^2 + vw)) = y^3 + 3py + q$ となるから

$$\begin{cases} y_1 = u + v \\ y_2 = uw_1 + vw_2 = u\left(\frac{-1 + \sqrt{3}i}{2}\right) + v\left(\frac{-1 - \sqrt{3}i}{2}\right) \\ y_3 = uw_2 + vw_1 = u\left(\frac{-1 - \sqrt{3}i}{2}\right) + v\left(\frac{-1 + \sqrt{3}i}{2}\right) \end{cases}$$

(6.41) より $u^3 = \alpha$, $v^3 = \beta$ とおくと解くことができる、

$$u^3 = \frac{-q \pm \sqrt{q^2 + 4p^3}}{2} \quad (6.42)$$

$u_0 \neq 0$ ならば、 $v_0 = -\frac{p}{u_0}$ にすることができる。即ち、3 次方程式の解は

$$x_1 = u_0 + v_0 - \frac{a}{3}, \quad x_2 = u_0 w_1 + v_0 w_2 - \frac{a}{3}, \quad x_3 = u_0 w_2 + v_0 w_1 - \frac{a}{3} \quad (6.43)$$

6.5.3 Cardano 法に基づいて W を求める

(6.35) に注目する。 $\gamma\rho = D$ に注意する。

$$S_1 = 1 - \frac{1-v^2}{\Gamma_1}, \quad S_2 = \left(\frac{\sqrt{1-v^2}}{\Gamma_1} - 1\right)D + \frac{B^2}{2} - \varepsilon \quad (6.44)$$

とすると簡単になって

$$[S_1 W + S_2](W + B^2)^2 + \frac{T^2}{2} = 0 \quad (6.45)$$

ここで

$$S_S = \frac{S_2}{S_1}, \quad S_a = 2B^2 + S_S, \quad S_b = B^4 + 2B^2 S_S, \quad S_c = B^4 S_S + \frac{T^2}{2S_1} \quad (6.46)$$

とすると (6.45) は

$$W^3 + (S_a)W^2 + (S_b)W + (S_c) = 0 \quad (6.47)$$

と書き下すことができる。これは (6.39) と形が同じであることから Cardano's methodにより W を求めることができる。すなわち

$$S_p = -\frac{(S_a)^2}{9} + \frac{S_b}{3} \quad (6.48)$$

$$S_q = \frac{2}{27}(S_a)^3 - \frac{S_a S_b}{3} + S_c \quad (6.49)$$

$$(S_u)^3 = \frac{1}{2} \left[-S_q \pm \sqrt{(S_q)^2 + 4(S_p)^3} \right] \quad (6.50)$$

$$S_v = -\frac{S_p}{S_u} \quad (6.51)$$

$$W = \begin{cases} S_u + S_v - \frac{S_a}{3} \\ S_u w_1 + S_v w_2 - \frac{S_a}{3} \\ S_u w_2 + S_v w_1 - \frac{S_a}{3} \end{cases} \quad (6.52)$$

以上のようにして解が3個でてくる。(6.35)の $(W + B^2)^2$ に注目しよう。 $T = 0$ のとき、 W の解で $W = -B^2$ が求まるのがわかる。この値は W の最小値である。そのため解は3個のうち最大のものを取ればよい。

6.5.4 W' に関する記述

W を解く際、0割りが起きてしまう可能性がある点に気をつけなければならない。わかりやすいように $W = S_u + S_v - \frac{S_a}{3}$ が解のときを考えてみる。これを ξ で微分して

$$\frac{dW}{d\xi} = \frac{d(S_u)}{d\xi} + \frac{d(S_v)}{d\xi} - \frac{1}{3} \frac{d(S_a)}{d\xi} \quad (6.53)$$

$\frac{d(S_u)}{d\xi}$ に注目すると

$$\frac{d(S_u)}{d\xi} = \frac{d}{d\xi} \left[\sqrt[3]{\frac{1}{2}(-S_q + \sqrt{(S_q)^2 + 4(S_p)^3})} \right] \quad (6.54)$$

$$= \frac{d}{d\xi} \left[\sqrt[3]{S_{u3}} \right] = \frac{1}{3} (S_{u3})^{-\frac{2}{3}} \frac{d(S_{u3})}{d\xi} \quad (6.55)$$

また $\frac{d(S_{u3})}{d\xi}$ に注目すると

$$\frac{d(S_{u3})}{d\xi} = \frac{1}{2} \left[-\frac{d(S_q)}{d\xi} + \frac{d}{d\xi} [\sqrt{(S_q)^2 + 4(S_p)^3}] \right] \quad (6.56)$$

さらに $\frac{d(\sqrt{S_D})}{d\xi} = \frac{d}{d\xi} \left[\sqrt{(S_q)^2 + 4(S_p)^3} \right]$ は

$$\frac{d(\sqrt{S_D})}{d\xi} = \frac{1}{2\sqrt{S_D}} \frac{d(S_D)}{d\xi} \quad (6.57)$$

(6.55) と (6.57) をみれば一目瞭然であるとおりに、 $S_{u3} = 0$, $S_D = 0$ となると発散してしまう。 S_{u3} に関して展開して考えると

$$S_{u3} = 0 \Rightarrow (\sqrt{1-v^2} - \Gamma_1)D + \left(\frac{3}{2} - \Gamma_1\right)B^2 - \varepsilon = 0$$

このことから $S_{u3} = 0$ になることはあまり考えられない。

S_D に関して同様に

$$S_D = 0 \Rightarrow \frac{T^2}{2(S_1)} \left[\frac{T^2}{2(S_1)} - \frac{4}{27}(B^2 - S_S)^3 \right] = 0$$

$$\Leftrightarrow T^2 = B^2 P^2 - (\mathbf{P} \cdot \mathbf{B}) = 0$$

すなわち $\mathbf{P} = 0$, もしくは $\mathbf{B} = 0$ のときに $S_D = 0$ となる。この場合はよく存在するので気をつけなければならない。

6.5.5 変換後の基本量の値

1-D Newton's methodから導き出された ξ の値からの基本量を求める方法は次の通りである。

$$\rho_{next} = D\sqrt{1-\xi} \quad (6.58)$$

$$p_{r \ next} = \frac{(1-\xi)W - \rho_{next}}{\Gamma_1} \quad (6.59)$$

$$\mathbf{v}_{next} = \frac{\mathbf{P} + \frac{S}{W}\mathbf{B}}{W + B^2} \quad (6.60)$$

6.6 HLL scheme

計算スキームとして Lax-Wendroff scheme と HLL scheme を用いた。HLL scheme について少しだけ触れておく。

6.6.1 なぜ HLL scheme なのか？

HLL Scheme は Lax-Wendroff scheme のように shock での数値散逸が小さく、変化の少ないところでは正確な値を示す。しかし Roe 法に比べてなまりが大きく、Roe 法とのハイブリッドのコードが望ましい。しかし HLL scheme は Roe 法のように 8 つの固有値・固有ベクトルを導き出す必要が無い、また相対論では固有値・固有ベクトルはローレンツ係数の存在により容易に導き出すことができない。これらの点から現在は HLL scheme を使用している。

6.6.2 HLL Flux の導出

<<参考 : P.Janhunen (1999) >>
一次元の波を考える。

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} + S \quad (6.61)$$

ここで $U = U(x, t)$ は保存量、 $F = F(u)$ は流束、 $S = S(x, t)$ である。 $t = 0$ で U_L ($x < 0$), U_R ($x > 0$) とする。

また正の方向に最も速い波を b_R 、負の方向に最も遅い波を b_L とする。ある時間 τ において $L = (b_R - b_L)\tau$ とするとその空間での平均速度は

$$\bar{U}(t) \equiv \frac{1}{L} \int_{b_L\tau}^{b_R\tau} dx U(x, t) \quad (6.62)$$

(6.61) を積分すると次の式が得られる。

$$\bar{U}(\tau) = U_m + \Delta U_m \quad (6.63)$$

$$U_m = \frac{b_R U_R - b_L U_L - (F_R - F_L)}{b_R - b_L}, \quad \Delta U_m = \int_0^\tau dt \bar{S}(t) \quad (6.64)$$

一方

$$\bar{U}_L(\tau) = \frac{(\Delta x_L + b_L \tau) U_L - b_L \tau U_m}{\Delta x_L} = U_L - \frac{\tau}{\Delta x_L} (F_{HLL} - F_L) \quad (6.65)$$

ここで

$$F_{HLL} = F_L + b_L (U_m - U_L) \quad (6.66)$$

また

$$F_{HLL} = F_R + b_R (U_m - U_R) \quad (6.67)$$

(6.64) を (6.66), (6.67) に代入すると

$$F_{HLL} = \frac{b_L b_R (u_R - u_L) + b_R F_L - b_L F_R}{b_R - b_L} \quad (6.68)$$

これを新しい流束として適用する。すなわち

$$U(x, t+1) = U(x, t) - \frac{dt}{dx} [F_{HLL}(x) - F_{HLL}(x-1)] \quad (6.69)$$

1

6.7 テスト計算

CANSを元にして作ったコードを Balsara(2000) にそってコードテストした結果を報告する。

- scheme : 1 次元 2-step Lax-Wendroff 又は HLL scheme
- クーラン条件 : 光速の 20 %
- 人工粘性 : CANS の組み込み人工粘性を使用
- iteration : 2D Newton's method 又は 1-D Newton's method

以下、2-step Lax-Wendroff scheme $\Rightarrow$ LW, HLL scheme $\Rightarrow$ HLL
 2-D Newton's method (Koide 1999) $\Rightarrow$ KO
 1-D Newton's method (Del Zanna 2002) $\Rightarrow$ DZ で表す。

6.7.1 Alfvén Wave の伝播

Koide (2003) によると相対論における Alfvén Wave は次のように示される。

$$v_A = \frac{B_0}{\sqrt{\rho_0 + (\frac{\Gamma}{\Gamma-1} p_{r0} + B_0^2)/c^2}} \quad (6.70)$$

今、 $B_z = 0$ とおくと $B_0^2 = B_x^2 + B_y^2$ と書ける。Maxwell 方程式から、

$$\frac{\partial B_y}{\partial t} = -\frac{\partial}{\partial x}(v_y B_x) \quad (6.71)$$

また v_y に関して

$$v_y = A \cos \Omega \left(\frac{x}{v_A} + t \right) \quad (6.72)$$

¹HLL scheme の解説、プログラムについては共同研究者である浅野 栄治氏 (千葉大自然) にサポートをして頂きました。

とにおいて

$$\begin{aligned}\frac{\partial B_y}{\partial t} &= B_x \cdot \frac{A\Omega}{v_A} \sin \Omega \left(\frac{x}{v_A} + t \right) \\ \Leftrightarrow B_y &= -B_x \cdot \frac{A}{v_A} \cos \Omega \left(\frac{x}{v_A} + t \right)\end{aligned}$$

t=0 において

$$v_y = A \cos \Omega \left(\frac{x}{v_A} \right) = v_{al}$$

$$B_y = -B_x \cdot \frac{A}{v_A} \cos \left(\frac{\Omega x}{v_A} \right) = -B_x \cdot \frac{A}{v_A} \cos(\kappa x) = B_{al} \quad (6.73)$$

ここで $c = 1$, $\kappa = \frac{2\pi}{\text{全grid数}}$, $A = \text{const} \ll 1 \Rightarrow A = 0.01$ とする。

Test.1 Alfvén Wave の伝搬 ($\Gamma = 2.0$, $c = 1$, grid=807)

Test 1	ρ	v_x	v_y	v_z	p_r	B_x	B_y	B_z
<i>Alfven</i>	1.0	1.0	v_{al}	0.0	1.0	1.0	B_{al}	0.0

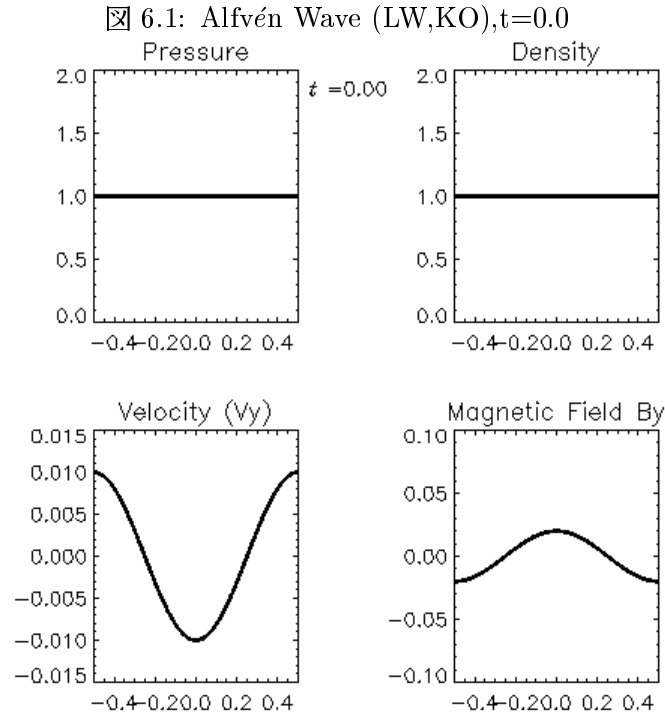


図 6.1, 6.3, 6.2, 6.4, からどちらも右から左へ Alfvén 波が伝わっているのが確認できた。
このテストは HD から MHD へ拡張させる際、必ず確認すべき Test 問題である。

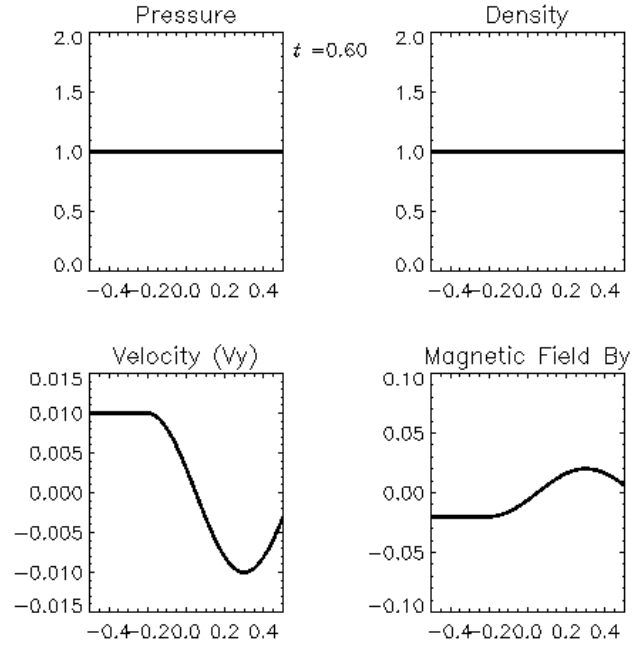
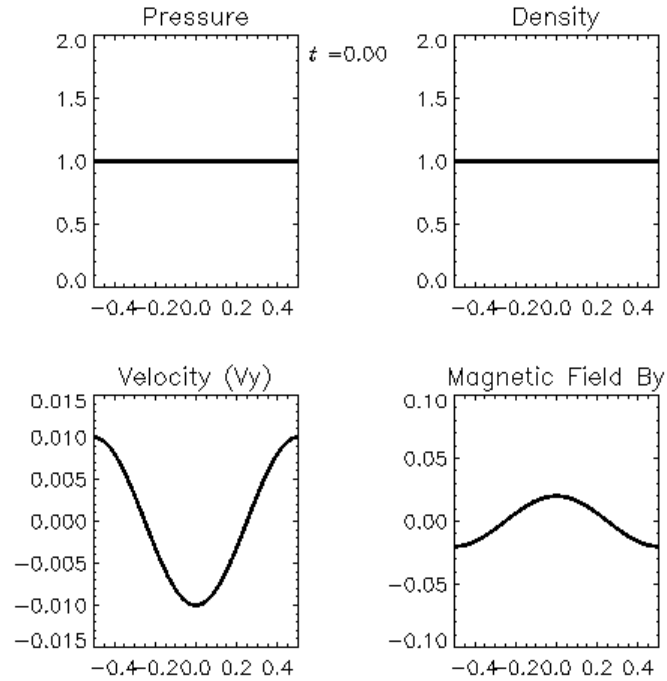
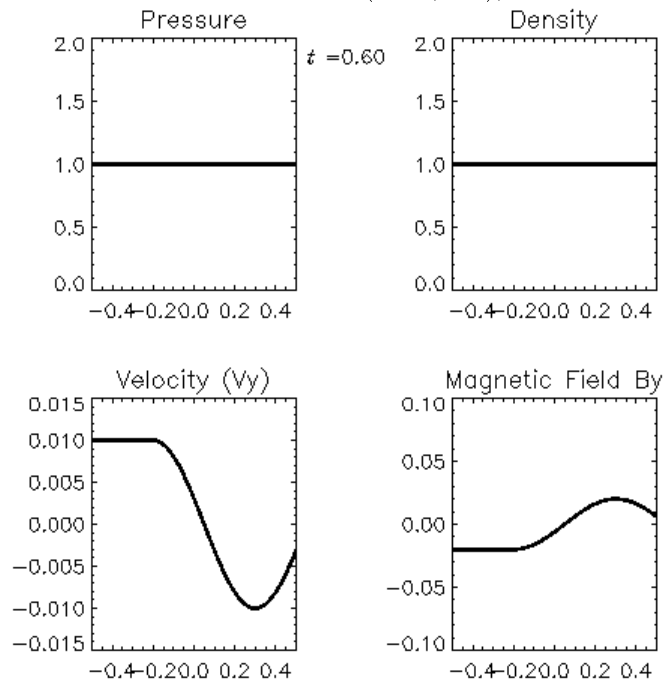
図 6.2: Alfvén Wave (LW,KO), $t=0.6$ 図 6.3: Alfvén Wave (HLL,KO), $t=0.0$ 

図 6.4: Alfvén Wave (HLL,KO), $t=0.6$ 

6.7.2 MHD Shock-tube Riemann problems

一番簡単な Test 問題として Brio & Wu (1988) を元にした Dinshaw Balsara(2000) の MHD Shock-tube 問題を解いた結果を記す。

Test.2 MHD Shock-tube 問題 ($\Gamma = 2.0$, $c = 1$, grid=807)

Test 2	ρ	v_x	v_y	v_z	p_r	B_x	B_y	B_z
Left	1.0	0.0	0.0	0.0	1.0	0.5	1.0	0.0
Right	0.125	0.0	0.0	0.0	0.1	0.5	-1.0	0.0

図 6.5, 図 6.6 の LW と HLL の違いに注目しよう。共に物理量の値、変化率といったものは Balsara (2000) と一致する。LW では特有の数値振動が見れるのに対し HLL では綺麗に解けている。また少ない grid 数で Shock をとらえている。DZ を用いて解いた場合では全く同じ結果がでたのでここでは省略する。

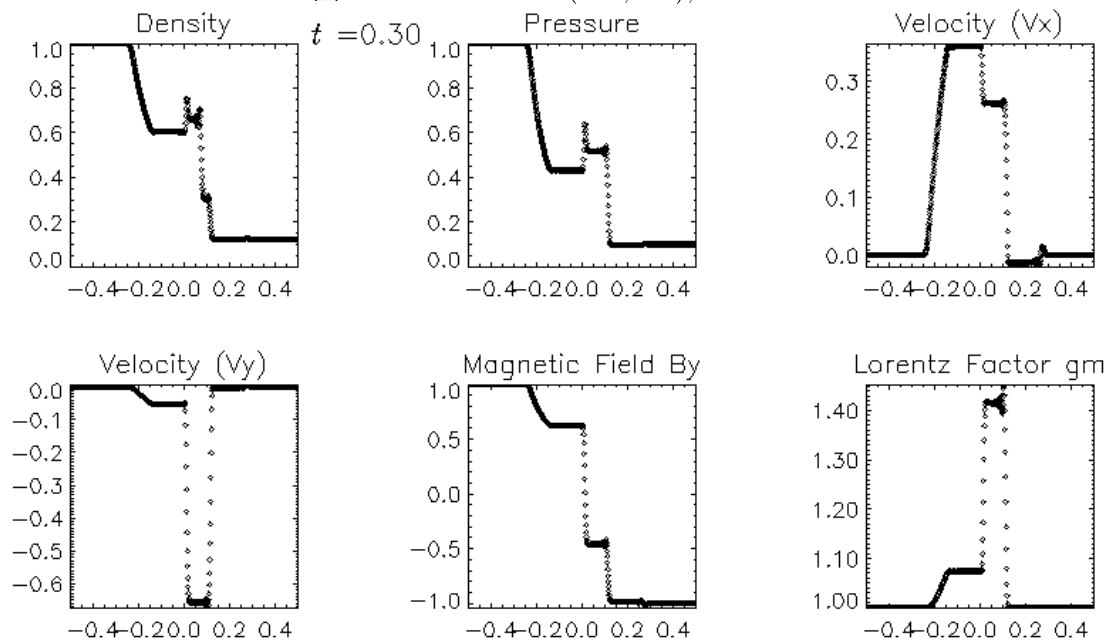
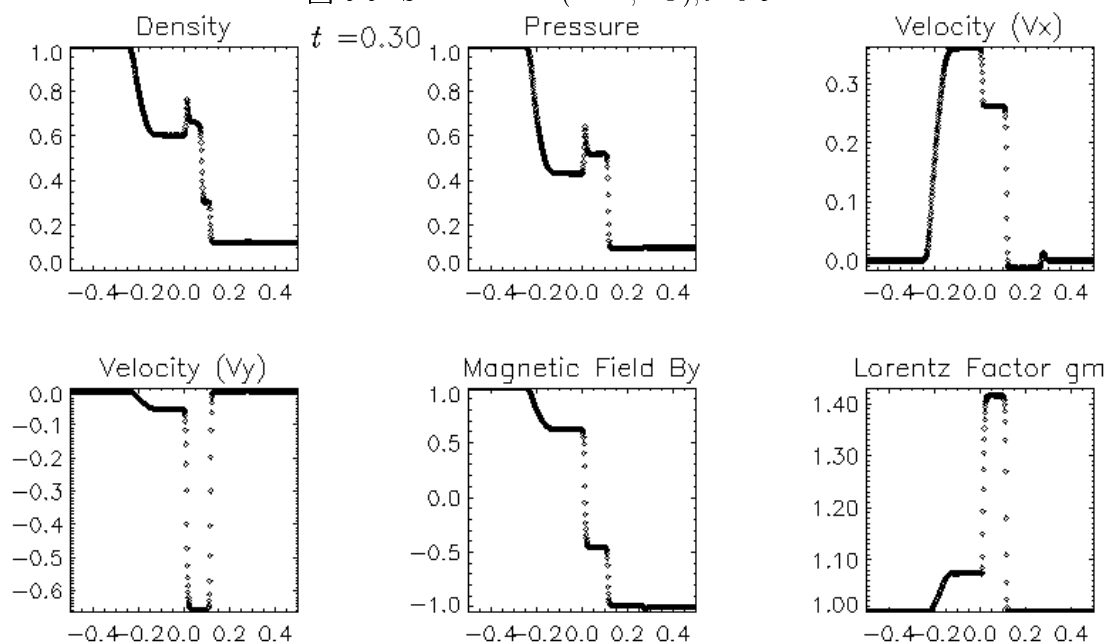
図 6.5: Shock Tube (LW,KO), $t=0.3$ 図 6.6: Shock Tube (HLL,KO), $t=0.3$ 

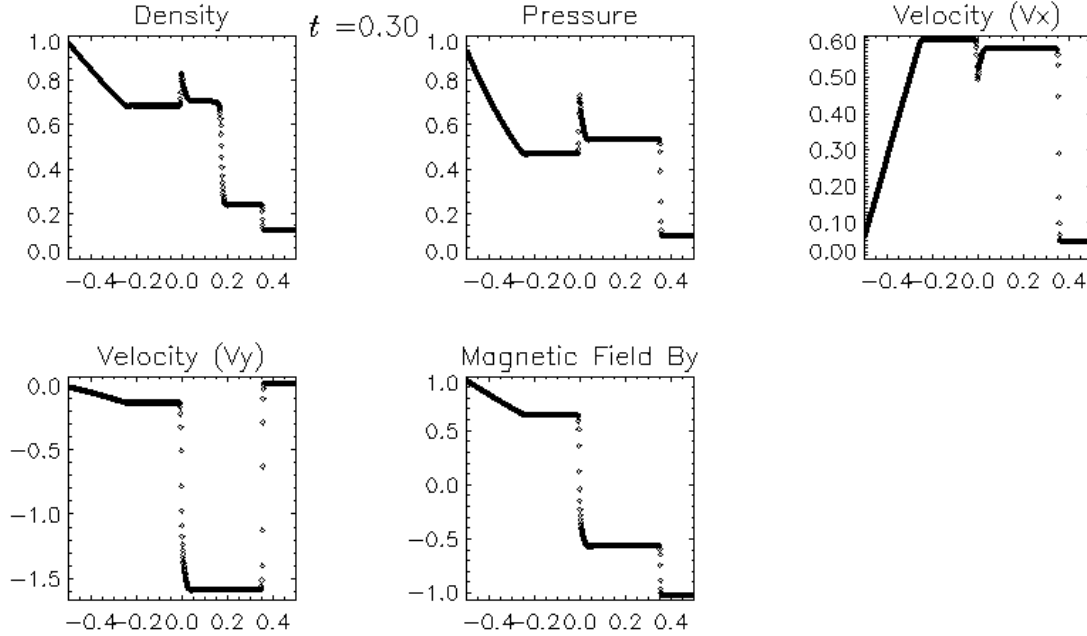
図 6.7: Non-relativistic Shock-tube (HLL), $t=0.3$ 

図 6.7はすべて同じ条件下での非相対論的 Shock-tube 問題を HLL で解いたものを表している。相対論的 Shock-tube 問題と比べると v_x の速度変化が大きくそれに伴って明らかに fast wave の速さが違うことがわかる。また X 方向の物理量の変化の幅も大きい。これは相対論的な効果、すなわち Lorentz 収縮の効果が現われたためである。ちなみにここでも LW 特性の数値振動はみられなく、LW に比べ HLL の方が少ない grid 数で Shock をとらえていた。

6.7.3 二種類の Blast Wave

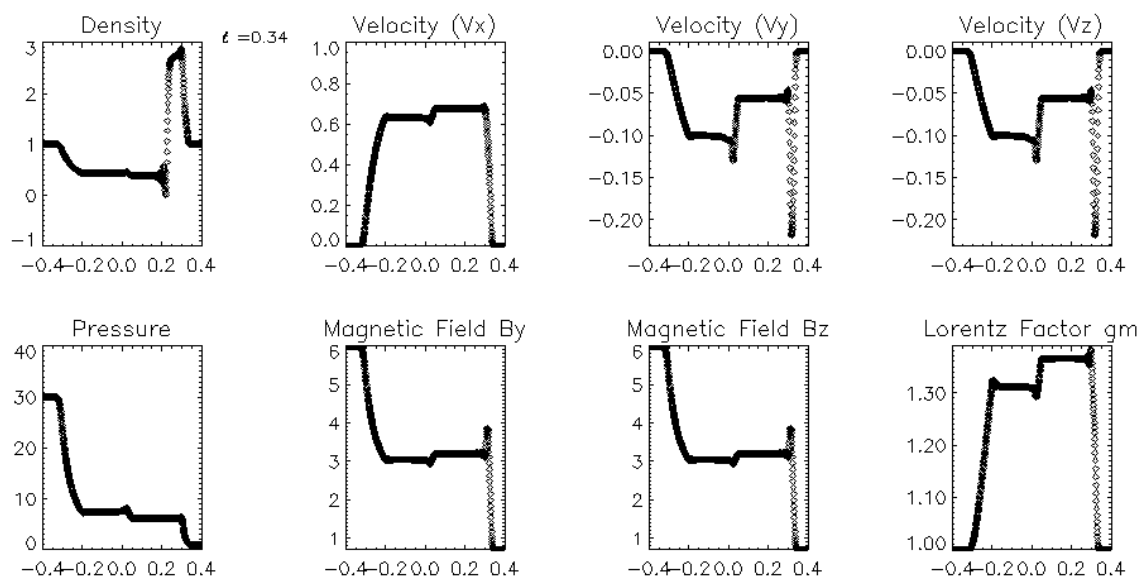
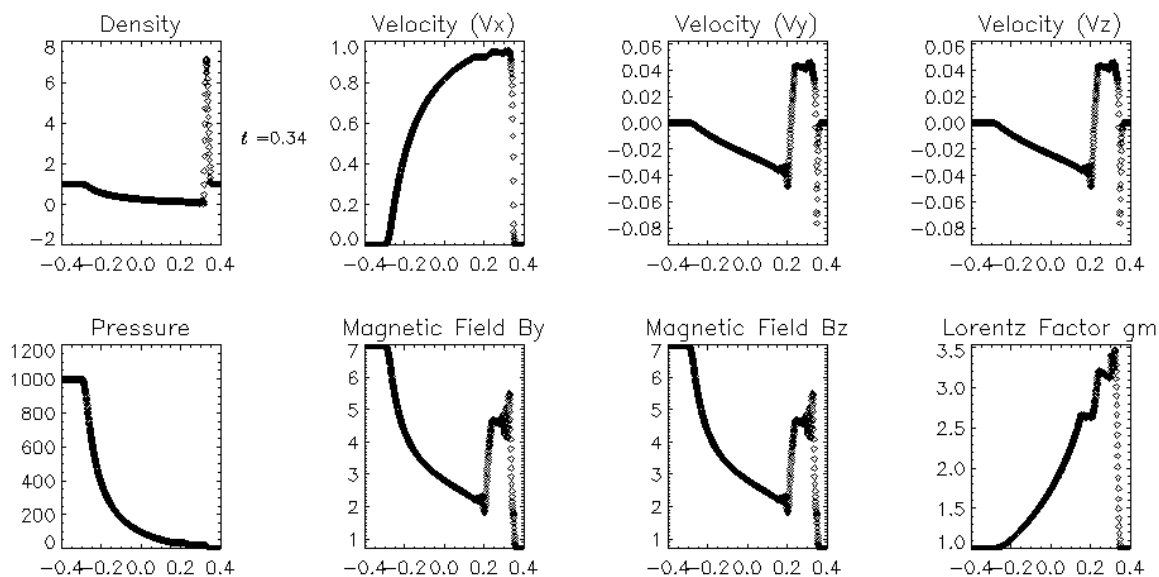
Balsara (2000) の Test 問題の二種類の Blast Wave を解いた結果を記す。ここでは LW scheme, DZ iteration を使用した。初期条件は以下の通りである。

Test.3 Blast Wave (moderate pressure) ($\Gamma = \frac{5}{3}$, $c=1$, grid=1001)

Test 3	ρ	v_x	v_y	v_z	p_r	B_x	B_y	B_z
Left	1.0	0.0	0.0	0.0	30.0	5.0	6.0	6.0
Right	0.125	0.0	0.0	0.0	1.0	5.0	0.7	0.7

Test.4 Blast Wave (strong pressure) ($\Gamma = \frac{5}{3}$, $c = 1$, grid=1001)

Test 4	ρ	v_x	v_y	v_z	p_r	B_x	B_y	B_z
Left	1.0	0.0	0.0	0.0	1000.0	10.0	7.0	7.0
Right	0.125	0.0	0.0	0.0	0.1	10.0	0.7	0.7

図 6.8: moderate pressure (LW,DZ), $t=0.3$ 図 6.9: strong pressure (LW,DZ), $t=0.3$ 

この二つの結果も Balsara (2000) と変化率、値ともに一致した。
 二つの密度の値を比べて見ると、strong pressure の fast wave と slow wave が潰れてしまっているのが分かる。それにともなって、Lorentz facor も急激に上昇し、速度も光速の 90 % になっている。これも先程同様、相対論的効果である Lorentz 収縮が顕著に見られるところである。

6.7.4 反復法(KO と DZ) の比較

KO と DZ の iteration を比較するために 5 つの Test を試みた。すべての Test において LW, HLL の scheme と KO, DZ の iteration を行った。初期条件である表 6.1 とシミュレーション結果である表 6.2 のグラフを見比べて欲しい。

表 6.1: 反復法のテスト ($\Gamma = \frac{5}{3}$, $c = 1$, grid=807)

		ρ	v_x	v_y	v_z	p_r	B_x	B_y	B_z
Test A	Left	1.0	0.9	0.0	0.0	1.0	0.5	1.0	0.0
Test A	Right	0.125	0.0	0.0	0.0	0.1	0.5	-1.0	0.0
Test B	Left	1.0	0.985	0.0	0.0	1.0	0.5	1.0	0.0
Test B	Right	0.125	0.0	0.0	0.0	0.1	0.5	-1.0	0.0
Test C	Left	1.0	0.9999995	0.0	0.0	1.0	0.5	1.0	0.0
Test C	Right	0.125	0.0	0.0	0.0	0.1	0.5	-1.0	0.0
Test D	Left	1.0	0.99999995	0.0	0.0	1.0	0.5	1.0	0.0
Test D	Right	1.0	0.99999995	0.0	0.0	1.0	0.5	1.0	0.0
Test E	Left	1.0	0.9	0.0	0.0	1.0	121.0	242.0	0.0
Test E	Right	1.0	0.9	0.0	0.0	1.0	121.0	1.0	0.0

表 6.2: 結果と反復法の最大収束回数 (単位 : 回)

	LW (KO)	LW (DE)	HLL(KO)	HLL(DE)
Test A	5	5	5	5
Test B	×	6	617	6
Test C	×	×	×	7
Test D	×	8	×	8
Test E	×	×	×	6

× は収束しない

- Test A

Test.2 の Shock-tube 問題の左側に初速、光速の 90 %を与えてみた。

図 6.10: $TestA(LW, KO), t = 0.1$

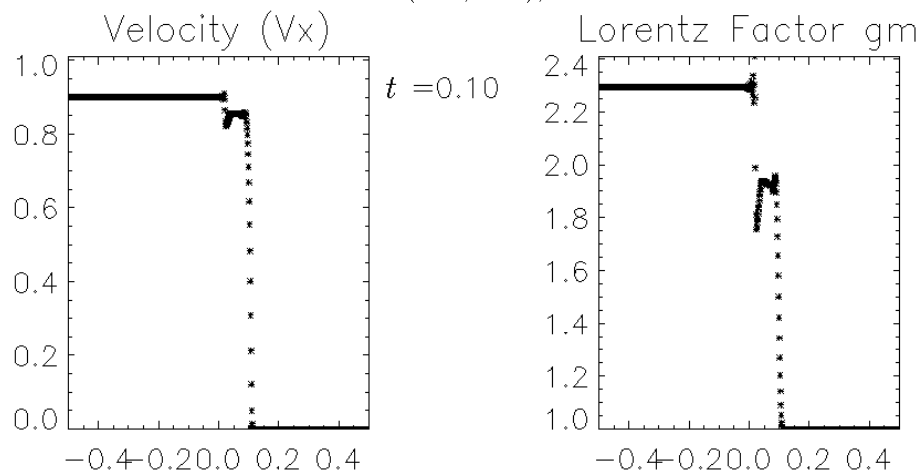
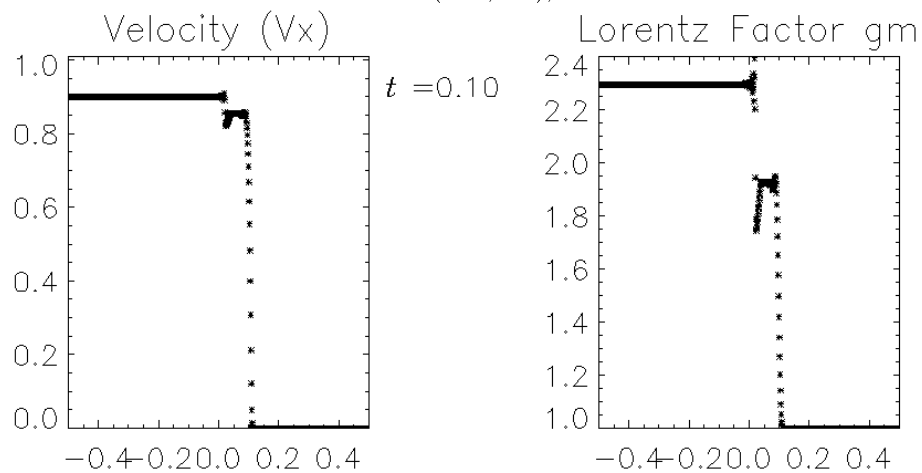


図 6.11: Test A (LW,DZ),t=0.1



この結果から見ても光速の 90 %程度であるなら、KO,DZ のどちらも正常な値を示す。しかし気になるのは LW 特有の数値振動である。もう少し光速に近づけた場合、どのような結果が出るのか？

- Test B

A の条件のまま左側の速度を光速に近づけてみた。Test B の初速の値 ($v_x=0.985$) は、(LW,DZ) でのこれ以上正常に作動しない限界速度である。

図 6.12: Test B (LW,KO), $t=0.05$

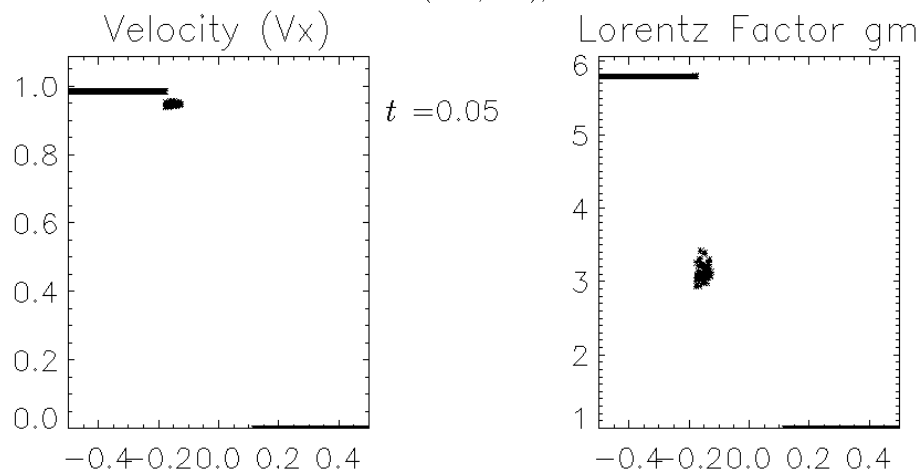
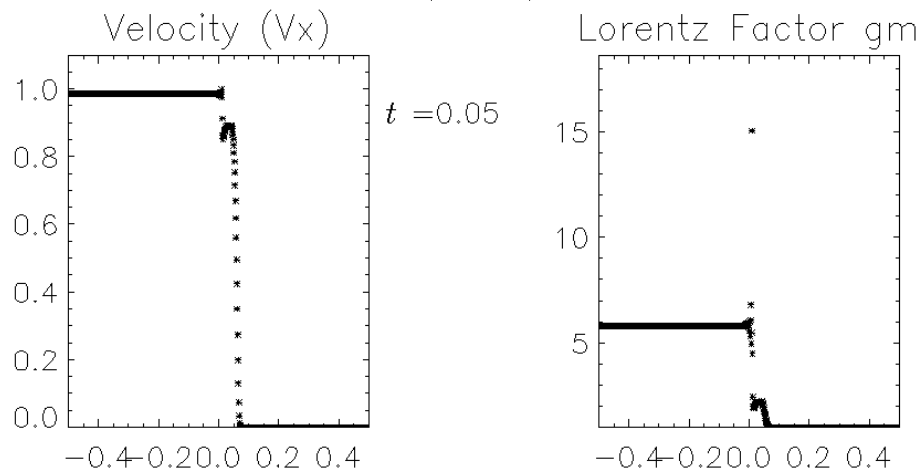


図 6.13: Test B (LW,DZ), $t=0.05$



(LW,KO) では、 $t=0.05$ のところで解が収束しなくなる。また、表から HLL でも DZ は綺麗に収束しているのに対し、KO の場合解を探す時その付近をよく振動していたため、最大収束回数が圧倒的に多くなっている。iteration としては最大でも 10 回前後で収束しなければ出力される値に信頼性がない。これらのことを踏まえ DZ の方がよく収束するのではないかという予測を立てた。

- Test C

今度は HLL の限界速度である速度 ($v_x=0.9999995$) を Shock-tube 問題の左側の初速の値として与えた。

図 6.14: Test C (HLL,KO), $t=0.1$

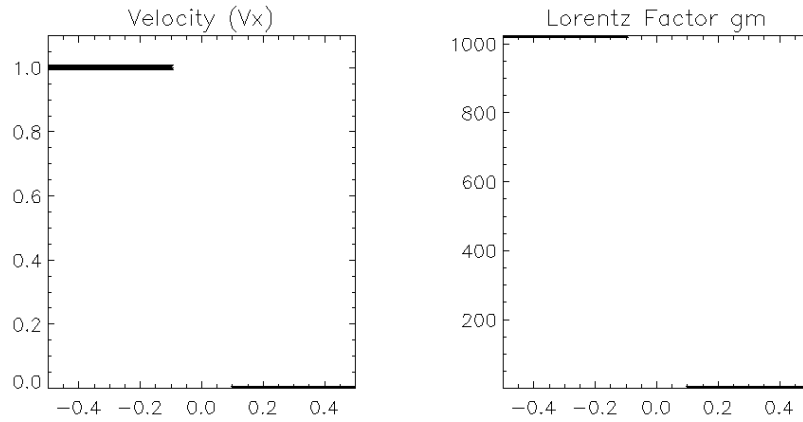
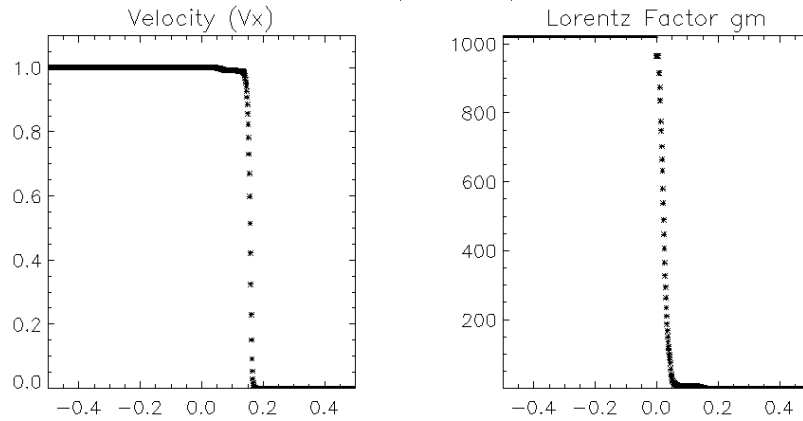


図 6.15: Test C (HLL,DZ), $t=0.1$



この結果からも DZの方がKOよりも収束していることが分かる。表からも10回以内に iteration は収束している。

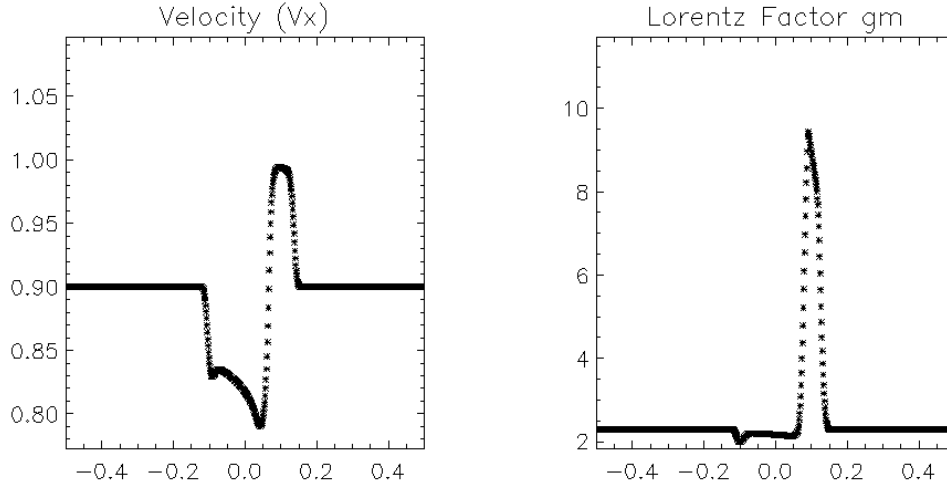
- Test D

この Test は定常状態においてどのくらいまで光速に近づけられるか示したものである。ここでもやはり DZの方が光速に近づけることが出来た。この時の γ factor は 2896.5 の値を示した。

- Test E

この Test では、強い磁場と iteration,scheme の関係を調べたものである。定常状態から Shock-tube 左側の初期値 B_y を強くしていき、(HLL,DZ) の収束する最大値を調べた。その後、他の scheme,iteration でも調べてみたものである。

図 6.16: Test E (HLL,DZ), $t=0.1$



定常状態にも関わらず、磁場が強くなると収束しない。また磁場を強くした場合でも、やはり (HLL,DZ) の収束が良いことがわかる。

6.8 考察

なぜ、DZの方が解が収束しやすいのか？

この問題は推測するに、DZはiterateする前に3次方程式により $W = \omega\gamma^2$ を決定するため解が見つけやすくなっていることが要因ではないかと考える。

また、このテスト計算から収束はschemeの精度によっても変わることが分かる。光速に限りなく近いシミュレーションをするためにはLax-Wendroffなどの数値振動が激しいschemeを使わず、Roe-typeのschemeなどの精度の高いschemeを使うことが必然となってくる。しかしながら、光速の30～40%程度の計算であればLax-Wendroff schemeや2-D Newton's methodでも十分解けることがわかる。さらに、2-D Newton's methodの方が3次方程式を使わない分計算コストが小さくなることが分かっている。しかし、1-D Newton's methodにおいても3次方程式の部分を改良すれば変わらない計算スピードを出せるのではないかと考える。

参考文献

- [1] Balsara,D.2000,ApJ,132,83
- [2] Gammie,C.F.,McKinney,J.C., & Tóth,G.2002,ApJ,589,444
- [3] Koide,S.,Shibata,K., & Kudoh,T.1999,ApJ,522,727
- [4] Koide,S.2003,Physical Review,67,104010
- [5] P.Janhunen.1999,J.comput.phys.,160,649
- [6] Zanna.L.D., & Bucciantini,N.2002,A & A,390,1177

第 7 章

Vlasov 方程式の数値解法

東京大学 天野 孝伸

宇宙プラズマのマイクロダイナミクスの研究は従来より粒子法と呼ばれる数値計算法が好んで用いられてきた。これは超粒子と呼ばれる質量、電荷共に実際の粒子よりも大きな仮想的な粒子の運動を追う手法であり、少ない粒子数でもプラズマの運動を正しく記述する結果が得られたことから広く受け入れられてきた。一方で粒子法は、実際の粒子数よりもはるかに少ない粒子しか扱うことができないため数値ノイズに悩まされ、また分布関数の裾野や非熱的粒子の扱いは不得手であった。ここではこれらの問題点を解決する可能性を秘めた Vlasov 方程式の直接解法を紹介する。

7.1 イントロダクション

宇宙空間に存在するプラズマの大部分は非常に希薄で、クーロン衝突の平均自由行程が特徴的な空間スケールに対して非常に大きな「無衝突」プラズマと考えられる。この無衝突プラズマのダイナミクスを支配する方程式として良く知られた方程式が以下の Vlasov 方程式 (7.1) である。

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{x}} + \mathbf{F}_s \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0 \quad (7.1)$$

ここで $f_s = f_s(\mathbf{x}, \mathbf{v}, t)$ は粒子種 s の分布関数、 $\mathbf{F}_s$ は粒子に働く力で、プラズマ中では電磁場 $\mathbf{E}, \mathbf{B}$ から

$$\mathbf{F}_s = \frac{q_s}{m_s} \left(\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B} \right) \quad (7.2)$$

なる力がはたらく。ここで現れる電磁場は Maxwell 方程式

$$\frac{1}{c} \frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B} - \frac{4\pi}{c} \mathbf{J} \quad (7.3)$$

$$\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \quad (7.4)$$

$$\nabla \cdot \mathbf{E} = 4\pi\rho \quad (7.5)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (7.6)$$

によって決定される。またここで

$$\rho = \sum_s \int f_s(x, v) dv \quad (7.7)$$

$$\mathbf{J} = \sum_s \int \mathbf{v} f_s(x, v) dv \quad (7.8)$$

によってプラズマから電磁場へフィードバックがかかる。従来はこの Vlasov 方程式を直接解くことはあきらめて、分布関数 $f(\mathbf{x}, \mathbf{v})$ を有限個の粒子の足し合わせによって表していたが¹ ここでは分布関数 f を 6 次元の関数とみなし、直接その時間発展を追いかける。しかし、現実的には現在のスーパーコンピュータを用いても 6 次元の計算によって物理的に意味のある結果を得ることは容易ではない² ので以下では空間 1 次元の問題を考える。

7.2 2 次元 Vlasov-Poisson System

背景磁場が 0 で静電近似の Vlasov 方程式を考える。速度空間は $v = v_x$ のみ考えれば十分で、この時の支配方程式は次の式で表される。

$$\frac{\partial f_s}{\partial t} + v \frac{\partial f_s}{\partial x} + \frac{q_s}{m_s} E \frac{\partial f_s}{\partial v} = 0 \quad (7.9)$$

$$\nabla^2 \phi = -4\pi\rho \quad (7.10)$$

ただし ϕ は静電ポテンシャルであり

$$E = -\frac{\partial \phi}{\partial x} \quad (7.11)$$

と表される。

7.2.1 The Splitting Scheme

目的は (7.9) と (7.10) を連立させて解くことだがここでは Cheng & Knorr(1976) によって提唱された Splitting Scheme を用いる。これは 2 次元の Vlasov 方程式 (7.9) を以下の 2 つの方程式に分離し、交互に解くものである。ただし、ここでは簡単のためイオンは背景に静止していると仮定し、電子の分布関数のみを対象として粒子種を表すインデックスを省略している (すなわち $f = f_e$)。

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} = 0 \quad (7.12)$$

$$\frac{\partial f}{\partial t} - \frac{e}{m} E \frac{\partial f}{\partial v} = 0 \quad (7.13)$$

(7.12) は移流速度 v が時間、空間に依存しない単純な線形移流方程式である。代入してみればすぐ分かるようにこの方程式の解析解は

$$f(x, v, t + \Delta t) = f(x - v\Delta t, v, t) \quad (7.14)$$

と表される。一方 (7.13) も同様に線形の移流方程式であることが分かる。線形の移流方程式の解を求めるスキームはさかんに研究されているが、(7.12) と (7.13) をどのような順序で解けばよいかは自明ではない。

Cheng & Knorr(1976) はこの問題を以下のようにして解くことで 2 次精度となることを示した。す

¹プラズマ物理で用いられる粒子法は Particle-In-Cell(PIC) と呼ばれ重力多体問題とは異なり、粒子間の相互作用は無視する。電磁場の値はグリッド上でのみ定義され粒子位置とグリッド位置で互いに物理量を補間し合う手法である

²最近では 5 次元、6 次元の計算も行なわれるようになってきてはいるが長時間発展を追うことはできず、テスト問題や線形解析の延長程度に留まっている

なわち

$$f^*(x, v) = f^n(x - v\Delta t/2, v) \quad (7.15)$$

$$f^{**}(x, v) = f^*(x, v + \frac{e}{m}E^*(x)\Delta t) \quad (7.16)$$

$$f^{n+1}(x, v) = f^{**}(x - v\Delta t/2, v) \quad (7.17)$$

の3ステップを順に解く。ただし、(7.16)における $E(x)^*$ は $f^*(x, v)$ を用いて Poisson 方程式を解いて得られたものである。この Splitting Scheme の3次元以上、またはより高次精度への一般化は §7.3.1で紹介する Symplectic Integrator という概念によって与えられる。

7.2.2 CIP 法

CIP 法の解説は日本語の教科書 (矢部、内海、尾形) やサマースクールのテキストにも記述があるが、復習も兼ねて簡単に述べる。CIP 法の応用範囲は極めて広いが、基本は次の線形移流方程式 (7.18) の高精度解法の一つである。

$$\frac{\partial f}{\partial t} + c \frac{\partial f}{\partial x} = 0 \quad (c > 0) \quad (7.18)$$

ここで格子間 $(i, i+1)$ を3次関数

$$f(x) = aX^3 + bX^2 + cX + d \quad (7.19)$$

(ただし $X = x - x_i$)

を用いて補間する。ここで係数 a, b, c, d を決定するには4つの条件式が必要となるが、CIP 法では f_i, f_{i+1} の他に格子点 $i, i+1$ における微分値 (これを g_i, g_{i+1} とする) を用いる。すなわち

$$f_i = d \quad (7.20)$$

$$f_{i+1} = a\Delta x^3 + b\Delta x^2 + c\Delta x + d \quad (7.21)$$

$$g_i = c \quad (7.22)$$

$$g_{i+1} = 3a\Delta x^2 + 2b\Delta x + c \quad (7.23)$$

なる条件を用いて a, b, c, d を決定する。

$$a = \frac{g_i + g_{i+1}}{\Delta x^2} + \frac{2(f_i - f_{i+1})}{\Delta x^3} \quad (7.24)$$

$$b = \frac{3(f_{i+1} - f_i)}{\Delta x^2} - \frac{2g_i + g_{i+1}}{\Delta x} \quad (7.25)$$

$$c = g_i \quad (7.26)$$

$$d = f_i \quad (7.27)$$

これを用いて Δt 後の f_{i+1}, g_{i+1} は

$$f_{i+1} = a\xi^3 + b\xi^2 + c\xi + d \quad (7.28)$$

$$g_{i+1} = 3a\xi^2 + 2b\xi + c \quad (7.29)$$

によって求められる。(ただし $\xi = c\Delta t$)

従って CIP 法では必要な物理量だけでなくその微分値も同時に解き進める必要がある。これは記憶容量の面で不利になると思われがちだが、それでも多くの問題に適用され成功をおさめている。

7.2.3 CIP 法の Vlasov 方程式への適用

§7.2.1によって 2 次元の Vlasov-Poisson System の解は 1 次元の線形移流方程式を解くことに帰着した。Cheng & Knorr(1976) はこの解法に 3 次スプライン補間や Fourier 級数展開による補間を用いた。これらの方法はスプライン補間をするためには行列反転が必要となり、Fourier 級数による補間では Fourier 変換が必要となるってしまう。ところが、ここで用いる CIP 法はそれらの手間が省ける分高速な計算が可能である。またスプラインや Fourier 変換は並列計算に向けた手法とは言えないが、CIP 法は完全にローカルに補間関数を構築できるため並列計算にも有効であると考えられる。その CIP 法による Vlasov 方程式の解法は Nakamura & Yabe(1999) によって与えられ、より少ない格子点数でも高精度解が求まることを示された。ここではこの方法を用いることとする。前述の通り、CIP 法では物理量だけでなくその微分値も同時解き進める必要がある。今扱う 2 次元の Vlasov 方程式の場合分布関数 f の他に、 $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial v}$ が必要となる。しかし、例えば (7.15) を解く際には CIP 法で求めることができるのは $(f, \partial f / \partial x)$ だけである。従って (7.15) によって

$$(f, \partial f / \partial x) \rightarrow (f^*, \partial f^* / \partial x) \quad (7.30)$$

が求まるように $\partial f^* / \partial v$ も何らかの方法によって求める必要がある³。Nakamura & Yabe(1999) は (7.15) を v で微分することによって求められる

$$\frac{\partial}{\partial t} \left(\frac{\partial f}{\partial v} \right) + \frac{\partial}{\partial v} \left(v \frac{\partial f}{\partial x} \right) = 0 \quad (7.31)$$

を以下のように差分化した

$$\frac{\partial f_{i,j}^*}{\partial v} = \frac{\partial f_{i,j}}{\partial v} - \frac{\Delta t}{4\Delta v} \left[v_{j+1} \left(\frac{\partial f_{i,j+1}^*}{\partial x} + \frac{\partial f_{i,j+1}}{\partial x} \right) - v_{j-1} \left(\frac{\partial f_{i,j-1}^*}{\partial x} + \frac{\partial f_{i,j-1}}{\partial x} \right) \right] \quad (7.32)$$

ただし、 i, j はそれぞれ x, v の格子点を表すインデックスである。この式 (7.32) は右辺が $\partial f_{i,j}^* / \partial x$ と $\partial f_{i,j} / \partial x$ の平均を中央差分した形になっているため、形式的には半陰解法となっているが $\partial f^* / \partial x$ は CIP によって求まっているため陽的に計算することができる。(7.16) も同様に CIP 法によって求めた $\partial f^{**} / \partial v$ を用いて

$$\frac{\partial f_{i,j}^{**}}{\partial x} = \frac{\partial f_{i,j}^*}{\partial x} + \frac{\Delta t}{4\Delta x} \left[\frac{e}{m} E_{i+1} \left(\frac{\partial f_{i+1,j}^{**}}{\partial v} + \frac{\partial f_{i+1,j}^*}{\partial v} \right) - \frac{e}{m} E_{i-1} \left(\frac{\partial f_{i-1,j}^{**}}{\partial v} + \frac{\partial f_{i-1,j}^*}{\partial v} \right) \right] \quad (7.33)$$

と書ける。Nakamura & Yabe(1999) はこのスキームによってグリッドの幅が一定の場合においては質量が保存することを示した。

7.2.4 解法のまとめ

ここで CIP 法による (7.30) と (7.32) によって Δt だけ時間を進める演算をそれぞれ次のように書くことにする。

$$(f^*, \partial f^* / \partial x) = \text{CIP}(f, \partial f / \partial x, v, \Delta t) \quad (7.34)$$

$$\partial f^* / \partial v = \text{FDM}(\partial f / \partial v, \partial f / \partial x, \partial f^* / \partial x, v, \Delta t) \quad (7.35)$$

³さらに $\frac{\partial^2 f}{\partial x \partial v}$ もメモリに保持して同時に解き進めることも可能であるが、より高次元への適用を考えた時には現実的ではない

(7.34) は CIP 法を用いて分割した次元方向に Δt だけ値 f とその方向の微分値を更新することを表す。また (7.35) は (7.32) や (7.33) の差分 (Finite Difference Method) によって分割方向以外の次元の微係数を更新することを表す。

この記法を用いて 2 次元の Vlasov-Poisson System の解法をまとめると

1. Step1

$$(f^*, \partial f^* / \partial x) = \text{CIP}(f^n, \partial f^n / \partial x, v, \frac{\Delta t}{2}) \quad (7.36)$$

$$\partial f^* / \partial v = \text{FDM}(\partial f / \partial v, \partial f^n / \partial x, \partial f^* / \partial x, v, \frac{\Delta t}{2}) \quad (7.37)$$

2. Step2

$$(f^{**}, \partial f^{**} / \partial v) = \text{CIP}(f^*, \partial f^* / \partial v, E^*, \Delta t) \quad (7.38)$$

$$\partial f^{**} / \partial x = \text{FDM}(\partial f^* / \partial x, \partial f^* v / \partial, \partial f^{**} / \partial v, E^*, \Delta t) \quad (7.39)$$

3. Step3

$$(f^{n+1}, \partial f^{n+1} / \partial x) = \text{CIP}(f^{**}, \partial f^{**} / \partial x, v, \frac{\Delta t}{2}) \quad (7.40)$$

$$\partial f^{n+1} / \partial v = \text{FDM}(\partial f^{**} / \partial v, \partial f^{**} / \partial x, \partial f^{n+1} / \partial x, v, \frac{\Delta t}{2}) \quad (7.41)$$

と表される。ただし、Step1 と Step2 の間で Poisson 方程式を解き E_x^* を求める。

7.2.5 線形ランダウ減衰

ここでは線形ランダウ減衰のテスト問題を示す。この問題は多くの著者によってコードのテストとして用いられる標準問題である (e.g. Cheng & Knorr (1976), Nakamura & Yabe (1999), Filbet, Sonnendrücker, & Bertrand (2001))。分布関数の初期値は

$$f_0(x, v) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right) (1 + A \cos kx) \quad (7.42)$$

で表される。ここでは $A = 0.01, k = 0.5$ で、系の大きさを $L = 4\pi$ とし、速度空間を $-4.0 < v < 4.0$ と取った時の計算結果を示す (図 7.1)。ただし、ここでは $n_e = 1, |q_e| = m_e = 1/4\pi$ すなわち $\omega_{pe} = 1$ 。また $v_{th} = \lambda_D = 1$ に規格化している。計算結果と線形解析による減衰率 $\gamma = 0.1533$ が非常に良く一致していることが分かる。

また、同様に線形解析からこの場合の振動数は $\omega = 1.4156$ と求まっているが、これから波と共鳴する粒子は $v_{resonance} = \omega/k \simeq 2.83$ の速度を持った粒子であることが分かる。つまりこの問題は分布関数の裾野の粒子と場がエネルギー交換を行なうため従来の粒子法では解きにくい問題であるが、Vlasov 方程式を直接解くことで非常に精度良く解けていることが分かる。また Nakamura & Yabe (1999) によって格子点数を半分に減らしても誤差が小さい範囲で収まることが示されている。

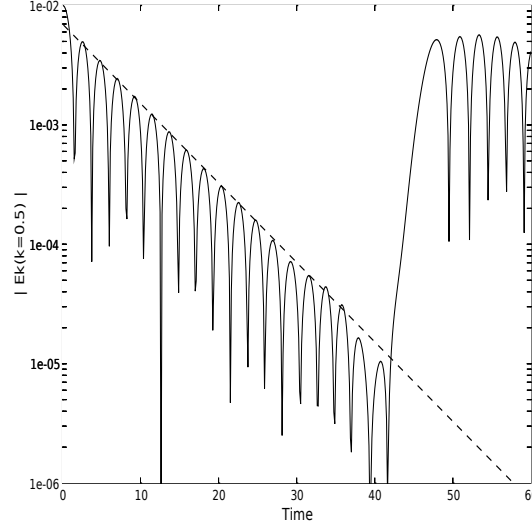


図 7.1: 線形ランダウ減衰 線形ランダウ減衰のテスト ($N_x = N_v = 32, A = 0.01, k = 0.5$)。実線が計算により求められた電場の振幅、破線が線形解析により求められた減衰率 $\gamma = 0.1533$ を表す。

7.3 3次元 Vlasov-Maxwell System

§7.2によって示された解法を磁場を含めた Vlasov-Maxwell System へと拡張する。ここでは実空間を §7.2と同じく x とし、背景磁場の方向を z 方向にとる。背景磁場が x, y 成分を含まない時、粒子のジャイロ運動は (v_x, v_y) 空間の中に閉じるので v_z 方向は独立となり解く必要がなくなる。この時解くべき方程式は

$$\frac{\partial f_s}{\partial t} + v_x \frac{\partial f_s}{\partial x} + \frac{q_s}{m_s} \left(E_x + \frac{v_y}{c} B_z \right) \frac{\partial f_s}{\partial v_x} + \frac{q_s}{m_s} \left(E_y - \frac{v_x}{c} B_z \right) \frac{\partial f_s}{\partial v_y} = 0 \quad (7.43)$$

$$\frac{\partial E_y}{\partial t} = -c \frac{\partial B_z}{\partial x} - 4\pi J_y \quad (7.44)$$

$$\frac{\partial B_z}{\partial t} = -c \frac{\partial E_y}{\partial x} \quad (7.45)$$

と Poisson 方程式 (7.10) である。ここで述べる方法は 3 次元の速度関数にも直ちに拡張可能な方法である。

7.3.1 Symplectic Integrator

§7.2.1で導入した Splitting Scheme をより高次元へと拡張するためにここで Symplectic Integrator⁴ と呼ばれる概念を導入する。簡単のため再び 2 次元の Vlasov-Poisson System へ戻る。粒子の Hamiltonian が次のように表されるとき

$$H = \frac{v^2}{2} + \phi(x) \quad (7.46)$$

⁴Symplectic Integrator の概念について分かり易い日本語の説明が Web [3] にある。また Vlasov 方程式への適用は Mangeney et al.(2002) を参照

Vlasov 方程式はこれを用いて

$$\frac{\partial f}{\partial t} = [H, f] \equiv \Lambda f \quad (7.47)$$

と書ける。ただしポアソン括弧

$$[H, f] \equiv \{(\partial H / \partial x)(\partial f / \partial v) - (\partial H / \partial v)(\partial f / \partial x)\}$$

を用い、この演算を形式的に Λ と定義した。この演算子 Λ を用いて、この方程式の形式的な解は

$$f(x, v, t + \tau) = \exp(\Lambda \tau) f(x, v, t) \quad (7.48)$$

と書かれる。つまり $\exp(\Lambda \tau)$ を評価することは2次元の移流方程式を解くことに値する。しかしこれをそのまま解くのは難しいので何らかの方法で近似的に評価することを考える。今、Hamiltonian H は次のように二つの部分に分離可能である。

$$H = H_1(v) + H_2(x) \quad (H_1(v) = \frac{v^2}{2}, H_2(x) = \phi(x)) \quad (7.49)$$

ここで H_1, H_2 に対応する演算子をそれぞれ Λ_1, Λ_2 と定義する

$$\Lambda_i = [H_i, f] \quad (i = 1, 2) \quad (7.50)$$

これは §7.2.1 で次元ごとに分割したことに対応し、

$$\exp(\Lambda_1 \tau) f(x, v) = f(x - v\tau, v) \quad (7.51)$$

$$\exp(\Lambda_2 \tau) f(x, v) = f(x, v + \frac{e}{m} E(x)\tau) \quad (7.52)$$

であることが分かる。これらはそれぞれ線形の移流方程式を解くスキームによって解を求めることにし、この $\exp(\Lambda_1 \tau), \exp(\Lambda_2 \tau)$ の組を用いて $\exp(\Lambda \tau) = \exp((\Lambda_1 + \Lambda_2)\tau)$ を評価する。単純にこの2つの演算子を順番に作用させたとき

$$\exp(\Lambda \tau) = \exp(\Lambda_1 \tau) \exp(\Lambda_2 \tau) + O(\tau^2) \quad (7.53)$$

であることが分かり、これは $\exp(\Lambda \tau)$ を1次精度で近似している。これは粒子の運動方程式を解く際の leap-frog スキームに対応する。

より高次の近似を求めるために $\tau \ll 1$ の時 Taylor 展開

$$\exp(\Lambda \tau) = 1 + \Lambda \tau + \frac{(\Lambda \tau)^2}{2} + O(\tau^3) \quad (7.54)$$

を用いると

$$\begin{aligned} & \exp\left(\frac{\Lambda_1 \tau}{2}\right) \exp(\Lambda_2 \tau) \exp\left(\frac{\Lambda_1 \tau}{2}\right) \\ & \simeq 1 + \Lambda_1 \tau + \Lambda_2 \tau + \frac{(\Lambda_1 \tau)^2 + (\Lambda_2 \tau)^2 + \Lambda_1 \Lambda_2 \tau + \Lambda_2 \Lambda_1 \tau}{2} + O(\tau^3) \\ & = 1 + (\Lambda_1 + \Lambda_2) \tau + \frac{((\Lambda_1 + \Lambda_2) \tau)^2}{2} + O(\tau^3) \end{aligned}$$

となる。つまり §7.2.1 は $\exp(\Lambda\tau)$ の評価に

$$\exp(\Lambda\tau) = \exp\left(\frac{\Lambda_1\tau}{2}\right) \exp(\Lambda_2\tau) \exp\left(\frac{\Lambda_1\tau}{2}\right) + O(\tau^3) \quad (7.55)$$

なる 2 次精度の近似を用いたことに相当する。

このように分割した微分作用素を組み合わせることで高次精度のスキームを構築する方法を Symplectic Integrator と呼ぶ。数学的には指数関数の有限積は一つの指数関数に必ずまとめられることが知られている。すなわち

$$\prod_{i=1}^k \exp(u_i \Lambda_1 \tau) \exp(v_i \Lambda_2 \tau) = \exp(\Lambda' \tau) \quad (7.56)$$

これにより Symplectic Integrator によって積分することは Λ の代わりに Λ' によって正準変換したことに相当し、 Λ と Λ' の誤差の範囲でエネルギーが保存する。また (7.55) では 2 次精度を用いているが、この演算子の組合せにより任意精度が可能であることが知られている。例として次に 4 次の近似式を与えておく。

$$\exp(\Lambda\tau) = S_2(\alpha\tau, \Lambda_1, \Lambda_2) S_2((1-2\alpha)\tau, \Lambda_1, \Lambda_2) S_2(\alpha\tau, \Lambda_1, \Lambda_2) + O(\tau^5) \quad (7.57)$$

ただし

$$\alpha = \frac{1}{2 - \sqrt[3]{2}} \quad (7.58)$$

$$S_2(\tau, \Lambda_1, \Lambda_2) = \exp\left(\frac{\Lambda_1\tau}{2}\right) \exp(\Lambda_2\tau) \exp\left(\frac{\Lambda_1\tau}{2}\right) \quad (7.59)$$

7.3.2 3 次元 Vlasov 方程式の解法

§7.3.1 の Symplectic Integrator を 3 次元の Vlasov 方程式に適用するために

$$\Lambda_1 \equiv v_x \frac{\partial}{\partial x} \quad (7.60)$$

$$\Lambda_2 \equiv \frac{q}{m} \left(E + \frac{v_y}{c} B_z \right) \frac{\partial}{\partial v_x} \quad (7.61)$$

$$\Lambda_3 \equiv \frac{q}{m} \left(E - \frac{v_x}{c} B_z \right) \frac{\partial}{\partial v_y} \quad (7.62)$$

と演算子 $\Lambda_1, \Lambda_2, \Lambda_3$ を定義するとこれらの作用素を組み合わせると ((7.55) を用いると)

$$\begin{aligned} \exp(\Lambda\tau) &\equiv \exp((\Lambda_1 + \Lambda_2 + \Lambda_3)\tau) \\ &= \exp\left(\frac{\Lambda_1\tau}{2}\right) \exp((\Lambda_2 + \Lambda_3)\tau) \exp\left(\frac{\Lambda_1\tau}{2}\right) + O(\tau^3) \\ &= \exp\left(\frac{\Lambda_1\tau}{2}\right) \left[\exp\left(\frac{\Lambda_2\tau}{2}\right) \exp(\Lambda_3\tau) \exp\left(\frac{\Lambda_2\tau}{2}\right) \right] \exp\left(\frac{\Lambda_1\tau}{2}\right) + O(\tau^3) \end{aligned}$$

のように 5 ステップで 2 次精度となることが分かる。

それぞれの移流方程式は CIP 法によって解き、§7.2.4 と同様に CIP 法によって更新できない微分値は差分によって求めればよい。

7.3.3 Maxwell 方程式の解法

ここでは Vlasov 方程式と連立させて解く Maxwell 方程式 (7.44), (7.45) の解法を述べる。

電磁波の伝播

まず電流 $J_y = 0$ の場合（真空状態）での電磁波の伝播⁵ を考える。(7.44) + (7.45), (7.44) - (7.45) よりそれぞれ次の式を得る。

$$\frac{\partial(E_y + B_z)}{\partial t} + c \frac{\partial(E_y + B_z)}{\partial x} = 0 \quad (7.63)$$

$$\frac{\partial(E_y - B_z)}{\partial t} - c \frac{\partial(E_y - B_z)}{\partial x} = 0 \quad (7.64)$$

これはそれぞれ速度 $c, -c$ の線形移流方程式の形をしているため CIP 法などのスキームを用いて解くことができる。注目すべきはこの変形によって電磁波を右に進む波と左に進む波に分離したことであり、これによって自由境界が自然に実現することである。例えば $x = 0$ の境界において $E_y(+x) = E_y(-x), B_z(+x) = B_z(-x)$ (計算では $E_{y0} = E_{y1}, B_{z0} = B_{z1}$ とすれば良い) とするだけで簡単に自由境界が実現する。図 7.2 の左端はこの境界条件となっているが、非物理的な波の反射は見られない。また右端は $E_{yN} = -E_{yN+1}, B_{zN} = B_{zN+1}$ としてある。これによって E_y は固定端、 B_z は自由端として反射している。

電流項

CIP 法では方程式を移流相と非移流相に分離して解くが、ここでもその方法を用いることにする。つまり、(7.44) を

$$\frac{\partial E_y}{\partial t} = -4\pi J_y \quad (7.65)$$

$$\frac{\partial E_y}{\partial t} = -c \frac{\partial B_z}{\partial x} \quad (7.66)$$

と分離し、(7.65) は

$$E_y^{n+\frac{1}{2}} = E_y^{n-\frac{1}{2}} - 4\pi J_y^n \Delta t \quad (7.67)$$

によって解けば良い。ただし、CIP 法を用いる際には微分値 $\partial E_y / \partial x$ も必要なため、

$$\left(\frac{\partial E_y}{\partial x} \right)^{n+\frac{1}{2}} = \left(\frac{\partial E_y}{\partial x} \right)^{n-\frac{1}{2}} - 4\pi \left(\frac{\partial J_y}{\partial x} \right)^n \Delta t \quad (7.68)$$

によって $\partial E_y / \partial x$ も更新する必要がある。

ここで電場の定義時間を半整数ステップとしたのは時間差分を中央差分とするためである。

7.3.4 解法のまとめ

以上の解法をまとめると次のようになる（図 7.3 参照）。記号 CIP, FDM の意味は §7.2.4 と同じである。また

$$F_x = \frac{q}{m} \left(E_x + \frac{v_y}{c} B_z \right) \quad (7.69)$$

⁵ より詳しい解説は CIP 法の教科書（矢部、内海、尾形）参照

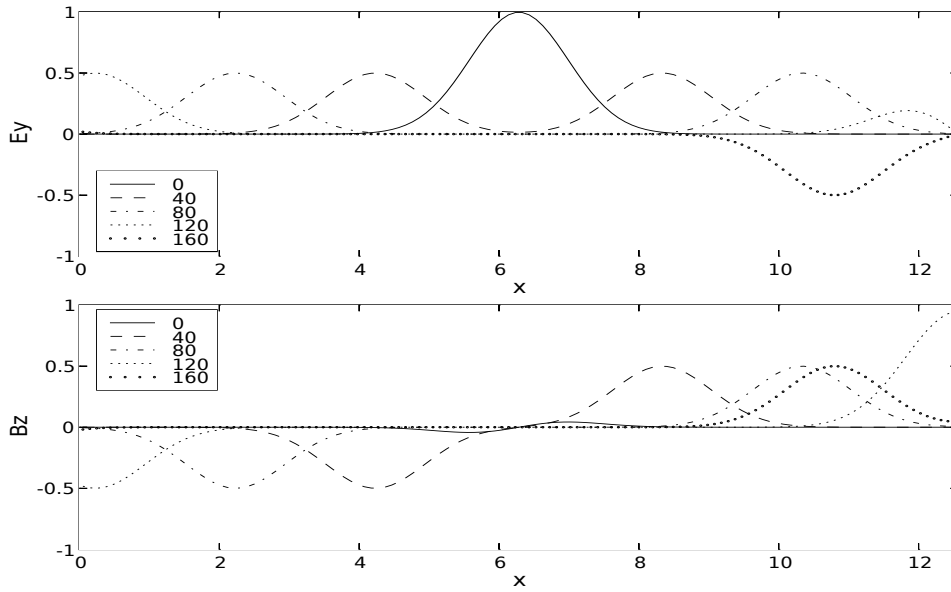


図 7.2: 電磁波の伝播 ($N = 128, L = 4\pi, c = 1.0$)。左は自由境界、右は $E_y = 0$ の固定端 B_z が自由端。それぞれ 0, 40, 80, 120, 160 ステップの解の様子を示している。移流方程式の解法には CIP 法を用いた。

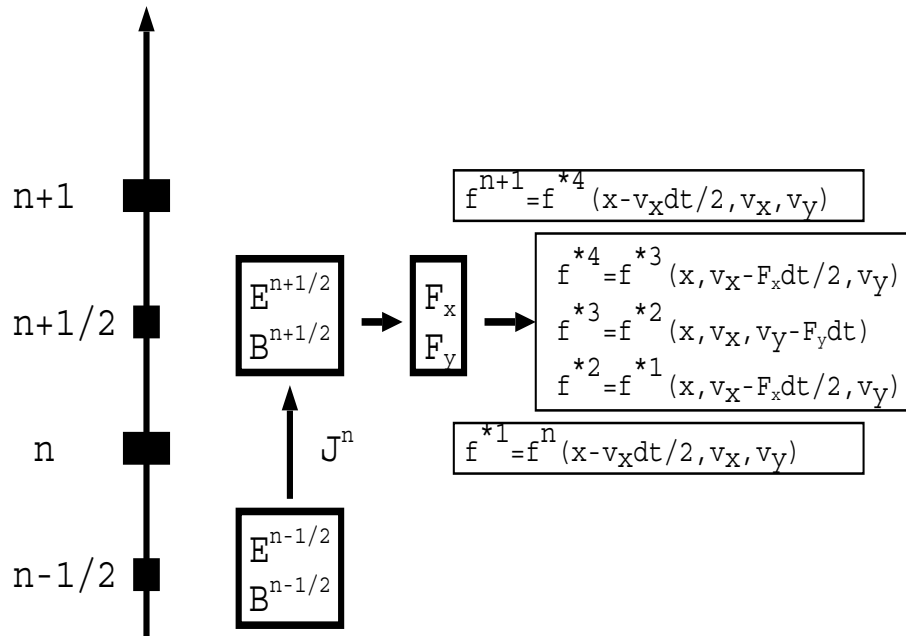


図 7.3: 3 次元の Vlasov-Maxwell System の時間ステップ更新方法

$$F_y = \frac{q}{m} \left(E_y - \frac{v_x}{c} B_z \right) \quad (7.70)$$

とする。

Step1

$$(f^{*1}, \partial f^{*1} / \partial x) = \text{CIP}(f^n, \partial f^n / \partial x, v_x, \Delta t / 2) \quad (7.71)$$

$$\partial f^{*1} / \partial v_x = \text{FDM}(\partial f^n / \partial v_x, \partial f^n / \partial x, \partial f^{*1} / \partial x, v_x, \Delta t / 2) \quad (7.72)$$

$$\partial f^{*1} / \partial v_y = \text{FDM}(\partial f^n / \partial v_y, \partial f^n / \partial x, \partial f^{*1} / \partial x, v_x, \Delta t / 2) \quad (7.73)$$

Step2

$$(f^{*2}, \partial f^{*2} / \partial v_x) = \text{CIP}(f^{*1}, \partial f^{*1} / \partial v_x, F_x, \Delta t / 2) \quad (7.74)$$

$$\partial f^{*2} / \partial x = \text{FDM}(\partial f^{*1} / \partial x, \partial f^{*1} / \partial v_x, \partial f^{*2} / \partial v_x, F_x, \Delta t / 2) \quad (7.75)$$

$$\partial f^{*2} / \partial v_y = \text{FDM}(\partial f^{*1} / \partial v_y, \partial f^{*1} / \partial v_x, \partial f^{*2} / \partial v_x, F_x, \Delta t / 2) \quad (7.76)$$

Step3

$$(f^{*3}, \partial f^{*3} / \partial v_y) = \text{CIP}(f^{*2}, \partial f^{*2} / \partial v_y, F_y, \Delta t) \quad (7.77)$$

$$\partial f^{*3} / \partial x = \text{FDM}(\partial f^{*2} / \partial x, \partial f^{*2} / \partial v_y, \partial f^{*3} / \partial v_y, F_y, \Delta t) \quad (7.78)$$

$$\partial f^{*3} / \partial v_x = \text{FDM}(\partial f^{*2} / \partial v_x, \partial f^{*2} / \partial v_y, \partial f^{*3} / \partial v_y, F_y, \Delta t) \quad (7.79)$$

Step4

$$(f^{*4}, \partial f^{*4} / \partial v_x) = \text{CIP}(f^{*3}, \partial f^{*3} / \partial v_x, F_x, \Delta t / 2) \quad (7.80)$$

$$\partial f^{*4} / \partial x = \text{FDM}(\partial f^{*3} / \partial x, \partial f^{*3} / \partial v_x, \partial f^{*4} / \partial v_x, F_x, \Delta t / 2) \quad (7.81)$$

$$\partial f^{*4} / \partial v_y = \text{FDM}(\partial f^{*3} / \partial v_y, \partial f^{*3} / \partial v_x, \partial f^{*4} / \partial v_x, F_x, \Delta t / 2) \quad (7.82)$$

Step5

$$(f^{n+1}, \partial f^{n+1} / \partial x) = \text{CIP}(f^{*4}, \partial f^{*4} / \partial x, v_x, \Delta t / 2) \quad (7.83)$$

$$\partial f^{n+1} / \partial v_x = \text{FDM}(\partial f^{*4} / \partial v_x, \partial f^{*4} / \partial x, \partial f^{n+1} / \partial x, v_x, \Delta t / 2) \quad (7.84)$$

$$\partial f^{n+1} / \partial v_y = \text{FDM}(\partial f^{*4} / \partial v_y, \partial f^{*4} / \partial x, \partial f^{n+1} / \partial x, v_x, \Delta t / 2) \quad (7.85)$$

ただし E_x は Step1 と Step2 の間で Poisson 方程式を解いて求め、 E_y, B_z は f^n の値から電流を計算し、これを用いて Maxwell 方程式を解いて求める。

また注意として、安定に解き進めるためには Maxwell 方程式を解く際に電流の寄与（非移流項）を計算する (7.67), (7.68) を電磁波の伝播（移流項）よりも先に計算すると良い⁶。

⁶ 著者の経験では移流項を先に計算すると電磁場のエネルギーが単調に増えてしまったため移流項を後に計算するようにしている

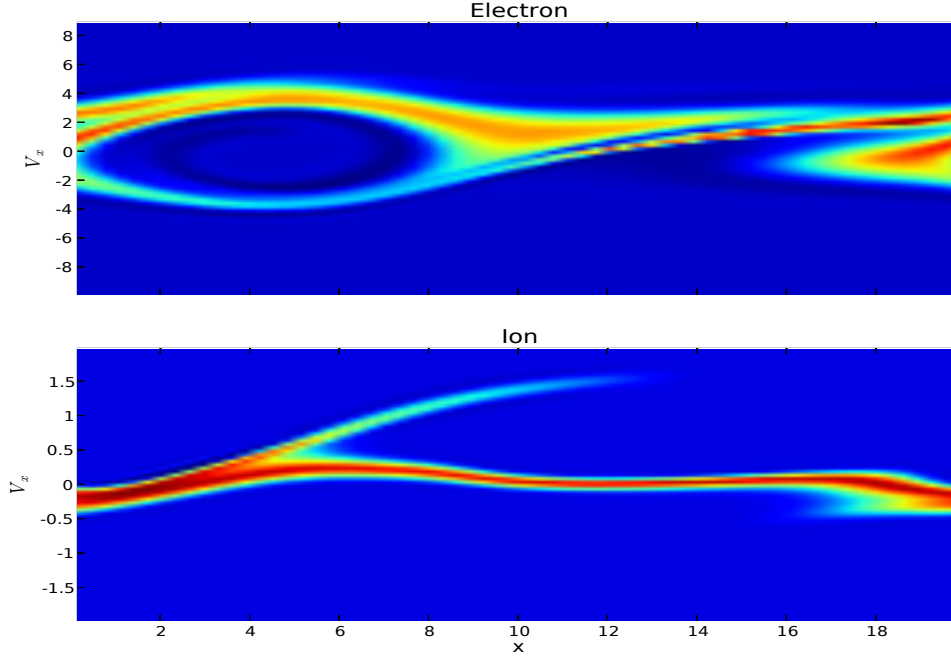


図 7.4: Buneman 不安定性の計算例 ($N_x = 64, N_{V_x} = N_{V_y} = 80, \omega_{pe}/\Omega_e = 10, m_i/m_e = 25$) $\omega_{pet} = 35.0$ における電子 (上)、イオン (下) の位相空間図。横軸が x , 縦軸が v_x でカラーで位相空間密度を表している。

7.3.5 計算例: Buneman 不安定

電子とイオンが相対速度を持ち運動している (すなわち電流が流れている) 時にその相対速度が電子の熱速度よりも大きい場合 Buneman 不安定と呼ばれる不安定により静電的な波が発生し、その結果電子の位相空間に hole 構造 (electron hole) が形成されることが知られている。ここでは 3 次元の Vlasov 方程式に $\omega_{pe}/\Omega_e = 10$ 程度の弱い磁場を入れた場合の計算例を図 7.4 に示す。ただし初期状態ではイオンが静止、電子は $V_0/c = 0.4$ でドリフトし、それぞれ熱速度は $v_e/c = 0.05, v_i/c = 0.01$ ($T_e = T_i$) としている。

参考文献

- [1] Cheng, C.Z. & Knorr, G., 1976, *J. Comput. Phys.*, **22**, 330
- [2] Nakamura, T. & Yabe, T., 1999, *Comput. Phys. Comm.*, **120**, 122
- [3] <http://www-cms.phys.s.u-tokyo.ac.jp/~naoki/CIPINTRO/SYMP/>
- [4] Filbet, F., Sonnendrücker, E. & Bertrand, P., 2001, *J. Comput. Phys.*, **172**, 166
- [5] Mangeney, A. et al., 2002, *J. Comput. Phys.*, **179**, 495
- [6] 矢部孝 内海隆行 尾形陽一, 2003, CIP 法 (森北出版)

第 II 部

数値シミュレーションの可視化

第 8 章

IDL を使った可視化

名古屋大学 越智 康浩

本章では、数値計算の結果を出力した 2 次元のサンプルデータを読み込み、可視化する手法の一例を紹介する。今回は Fortran プログラムで出力した 2 次元のサンプルデータを利用し、このデータを可視化する方法について述べる。IDL は 1 次元、3 次元データも扱うことができるが、ここではデータ解析の際に最も頻繁に利用する 2 次元画像の作成方法に焦点を絞る（なお AVS を使った 3 次元データの可視化については次章を参照のこと）。本章で使用するサンプルデータの作成の仕方については §8.7 を参照のこと。本章では、まず必要なメインプログラムやプロシージャについて §8.1 で確認する。次に §8.2 でサンプルデータを読み込む方法について説明し、そして §8.3 で読み込んだデータをディスプレイやファイルに出力する方法について説明する。ファイルに出力する際のカラーテーブルについては §8.5 で紹介する。

8.1 準備

本節では、§8.2 以降で説明する可視化に必要なメインプログラムや設定についてまとめる。§8.1.1 では、§8.2 以降の操作に必要なメインプログラムやプロシージャそしてサンプルデータなどを一覧を示す。§8.1.2 ではこれらのメインプログラムを利用するための設定について述べる。

8.1.1 使用するファイル

本節では §8.2 以降の操作に必要なメインプログラムやプロシージャの一覧を示す。表 8.1 は、§8.2, 8.3 で説明するメインプログラムの一覧である。ただしこれらのメインプログラムは全て 2 次元データ専用である。

表 8.2 は、表 8.1 のメインプログラムを実行する場合に必要なプロシージャである¹。

表 8.3 は §8.2, 8.3 で使用するサンプルデータの一覧である。なおこれらのサンプルデータの詳細については §8.7 を参照のこと。

これらのメインプログラムとプロシージャ、そしてサンプルデータは付属 CD-ROM の *Chap8* というディレクトリの中に収録されている。ディレクトリ *Chap8* の中はメインプログラムとプロシージャを納めた *idl* ディレクトリと、サンプルデータを納めた *data* ディレクトリに分かれている。

¹これらのプロシージャは 横山 央明 氏 (東京大学), 松本 倫明 氏 (法政大学), 杉本 香葉子 氏 (名古屋大学), そして著者が作成または編集した。

メインプログラム	用途
<i>read2dss.pro</i>	データを IDL に読み込ませる
<i>tvcn2dss.pro</i>	カラースケール、コントア、矢印を重ねてプロットさせる
<i>tvcn2dclosess.pro</i>	<i>tvcn2dss.pro</i> と同じだが任意の領域だけプロットさせる
<i>tvcn2dSLss.pro</i>	<i>tvcn2dss.pro</i> と同じだが流線もプロットさせる
<i>tvcn2d4pss.pro</i>	<i>tvcn2dss.pro</i> と同じだが 4 つの画像をプロットさせる
<i>tvcn2d6pss.pro</i>	<i>tvcn2dss.pro</i> と同じだが 6 つの画像をプロットさせる
<i>tvcn2dss.png.pro</i>	<i>tvcn2dss.pro</i> と同じだが PNG ファイルを出力する
<i>tvcn2dss.eps.pro</i>	<i>tvcn2dss.pro</i> と同じだが EPS ファイルを出力する
<i>write2dss.png.pro</i>	<i>tvcn2dss.png.pro</i> と同じだが自動的に出力する

表 8.1: 本章で使用、あるいは説明するメインプログラムの一覧。ただしこれらのメインプログラムは全て 2 次元データ専用である。これらのメインプログラムは全て付属 CD-ROM の *Chap8* ディレクトリ中の *idl* ディレクトリの中に収録されている。

プロシージャ	用途
<i>beginp.pro</i>	画面上で画を描く領域を指定する
<i>endp.pro</i>	<i>beginp.pro</i> で変更されたシステム変数を元に戻す
<i>tvcn.pro</i>	カラースケールとコントアを描く
<i>tvcn2ve.pro</i>	カラースケール、コントア、矢印を描く
<i>tvcn2ve2.pro</i>	<i>tvcn2ve.pro</i> と同じだが、矢印の数が 2 倍多い。
<i>vfield.pro</i>	ベクトル場を描く
<i>c_scale.pro</i>	カラースケールを描く
<i>ps1.pro</i>	device を “X” から “PS” に切替え、PS ファイルに画を出力する。
<i>ps1_col.pro</i>	“X” からカラーテーブルを取り込み、“PS” に内挿する。
<i>ps2.pro</i>	<i>ps1.pro</i> で変更されたシステム変数を元に戻す。
<i>ps_ajust_offset.pro</i>	画が紙面の中心になるように位置を調整する。
<i>ps_filename.pro</i>	command line から out put file 名を読む。
<i>ps_readfile.pro</i>	指定したファイルから変数のラベルや文字列を読む。
<i>ps_size.pro</i>	command line やファイルからから画の大きさや位置を読む。。
<i>cont_level.pro</i>	コントアのレベルを適切に調節する。
<i>bw.pro</i>	白黒の EPS ファイル作成のためのカラーテーブル
<i>rainbowf.pro</i>	カラーの EPS ファイル作成のためのカラーテーブル
<i>idl_startup.pro</i>	IDL の初期化時に実行される環境設定のプログラム

表 8.2: 表 8.1 のメインプログラムを実行するために必要なプロシージャの一覧。これらのプロシージャは全て付属 CD-ROM の *Chap8* ディレクトリ中の *idl* ディレクトリの中に収録されている。

サンプルデータ	用途
<i>tsample.D</i>	§8.2, 8.3 で主に使用するバイナリデータ
<i>tsample.d</i>	<i>tsample.D</i> のヘッダファイル
<i>sample0.D</i>	§8.3.7, 8.4.3 で使用するバイナリデータ
<i>sample0.d</i>	<i>sample0.D</i> のヘッダファイル
<i>sample1.D</i>	§8.3.7, 8.4.3 で使用するバイナリデータ
<i>sample1.d</i>	<i>sample1.D</i> のヘッダファイル
<i>sample2.D</i>	§8.3.7, 8.4.3 で使用するバイナリデータ
<i>sample2.d</i>	<i>sample2.D</i> のヘッダファイル
<i>sample3.D</i>	§8.3.7, 8.4.3 で使用するバイナリデータ
<i>sample3.d</i>	<i>sample3.D</i> のヘッダファイル
<i>sample1fps.mov</i>	§8.4.4 の方法で作成したサンプルムービー

表 8.3: 本章で使用使用するサンプルデータ。ただしこれらのデータは全て 2 次元である。これらのサンプルデータは全て付属 CD-ROM の *Chap8* ディレクトリ中の *data* ディレクトリの中に収録されている。

8.1.2 設定

前節では使用するメインプログラムやそのために必要なプロシージャの一覧を見た。本節ではこれらのメインプログラムを使用するための設定について述べる。以降は UNIX または Linux マシンで IDL を利用する場合を想定して説明を進める。なお Windows で IDL を利用する場合の説明は割愛した。

作業は以下の 3 つである。

1. ディレクトリの作成とメインプログラム、プロシージャの保存
2. パスの設定
3. デバイスの設定

メインプログラム、プロシージャの保存

まずメインプログラム用のディレクトリと、データ用のディレクトリ 2 つを作成する。メインプログラム用のディレクトリとは 表 8.1, 8.2 のメインプログラム、プロシージャを保存するためのものである。すでに IDL を利用したことがあり、IDL のメインプログラムを保存したディレクトリがある場合には特に作成する必要はなく、そのディレクトリに表 8.1, 8.2 のメインプログラム、プロシージャを保存すればよい。ディレクトリがない場合はホームディレクトリに *idl* という名前のディレクトリを作成する。同様にサンプルデータを保存するためのディレクトリ (ここでは *data* とする。) も作成する。例えば国立天文台共同利用 WS 群で使用する場合、WS にログインした後、下記のように作成する。

(ディレクトリの作成)

```
r01{ochiys}1: cd ~
/home2/ochiys
r01{ochiys}2: mkdir idl
r01{ochiys}3: mkdir data
```

ディレクトリが作成できれば、その中に表 8.1, 8.2 のメインプログラム、プロシージャを *idl* に、サンプルデータを *data* に保存する。これらのメインプログラムとプロシージャ、そしてサンプルデータは付属 CD-ROM の *Chap8* というディレクトリに収録されている。ディレクトリ *Chap8* の中はメインプログラムとプロシージャを納めた *idl* ディレクトリと、サンプルデータを納めた *data* ディレクトリに分かれている。表 8.1, 8.2, 8.3 のメインプログラム、プロシージャ、データの合計容量は約 660KB である。念のために全てのメインプログラムとプロシージャが保存できたかどうか確認する(プロシージャが足りないと、メインプログラム実行の際エラーが生じる可能性がある)。全て保存できれば以下のように表示される。

(保存できたかどうか確認)

```
r01{ochiys}3: cd ~/idl
/home2/ochiys/idl
r01{ochiys}4: ls
README          ps1.pro          rainbowf.pro     tvcn2dss.pro
beginp.pro       ps1_col.pro      read2dss.pro     tvcn2dss_eps.pro
bw.pro           ps2.pro          tvcn.pro         tvcn2dss_png.pro
c_scale.pro      ps_ajust_offset.pro tvcn2d4pss.pro   tvcn2ve.pro
cont_level.pro   ps_filename.pro   tvcn2d6pss.pro   tvcn2ve2.pro
endp.pro         ps_readfile.pro   tvcn2dSLss.pro   vfield.pro
idl_startup.pro  ps_size_offset.pro tvcn2dclosess.pro write2dss_png.pro
r01{ochiys}5: cd ~/data
/home2/ochiys/data
r01{ochiys}6: ls
README          sample1.D        sample2.D        sample3.D        tsample.d
sample0.D        sample1.d        sample2.d        sample3.d
sample0.d        sample1fps.mov   sample2d.f        tsample.D
r01{ochiys}7:
```

全て保存できていれば、次にパスの設定を行う。

パスの設定

ホームディレクトリ下の、*.cshrc* に下記のような *IDL_PATH* を設定する。環境変数 *IDL_PATH* は、ユーザが使用するメインプログラムやプロシージャ、関数のファイル (拡張子 *.pro* がつくファイル) を検索するためのディレクトリのリストである。詳細は Introduction to IDL version 5.1 (アダムネット株式会社 1998) p 31 を参照のこと。

(IDL_PATH の設定)

```
r01{ochiys}1: cat ~/.cshrc
setenv IDL_PATH \~/idl:\+$IDL_DIR/lib:\+$IDL_DIR/examples
```

環境変数 `IDL_DIR` は、IDL メインディレクトリの場所である。ここには文字フォント、システムカラーテーブルファイルなどの初期化ファイルが保存されている。上記の設定により IDL を利用する場合、どのディレクトリで作業していても

```
~/idl, $IDL_DIR/lib, $IDL_DIR/example
```

の場所までファイルを探しに行く。これら以外の場所にプロシージャや関数のファイルがある場合にはその場所も追加する。

デバイスの設定

ここではデバイス (X ウィンド) の設定について述べる。X ウィンドの設定が不適切な場合、カラーの画像が表示されないという深刻なトラブルが生じる可能性があるので、X ウィンドの設定は重要である。

まず、IDL の環境設定のためのプログラム `idl_startup.pro` が IDL の初期化時に実行されるように設定する。この `idl_startup.pro` は §8.1.2 でメインプログラムやプロシージャを保存した際に、一緒に `~/idl` の中に保存されている。まず、§8.1.2 で編集したホームディレクトリ下の、`~/cshrc` に下記のような環境変数 `IDL_STARTUP` の設定を追加する。

(IDL_STARTUP の設定)

```
r01{ochiys}2: cat ~/.cshrc
setenv IDL_PATH \+~/idl:\+$IDL_DIR/lib:\+$IDL_DIR/examples
setenv IDL_STARTUP ~/idl/idl_startup.pro
```

追加した後も下記のように設定を適用しなければならないので注意。

(設定の適用)

```
r01{ochiys}3: source ~/.cshrc
```

これにより、IDL を起動した際、`idl_startup.pro` に記述した設定が自動的に反映される。プログラムファイル `idl_startup.pro` の中は以下のようにになっている。

(環境設定プログラム `idl_startup.pro`)

```
r01{ochiys}4: cat ~/idl/idl_startup.pro
DEVICE,DECOMPOSED=0,TRUE_COLOR=24,RETAIN=2
```

上記プログラムの記述について簡単に説明する。プロシージャ「`DEVICE`」は IDL に代わってユーザーがデバイスを制御するものである。プロシージャ「`DEVICE`」の詳細についてはオンラインガイド (起動方法は IDL を起動した状態で “?” を入力) を参照のこと。キーワード「`DECOMPOSED=0`」はカラーインデックスを指定する際、Red, Green, Blue のカラーベクタごとに個別にインデックスを割り当てることを意味する。最近、広く普及している 24 bit ディスプレイでは、このような方法で Red, Green, Blue のカラーベクタにそれぞれ 8 bit ずつインデックスを割り当て

ているので 16,700,000 種類の色を同時に表示させることができる。このような色の表示のさせ方を True Color と呼ぶ。True Color の詳細については Introduction to IDL version 5.1 (アダムネット株式会社 1998) p 37 を参照のこと。キーワード「TRUE_COLOR=24」は True Color ビジュアルを選択する際に指定し、値は使用しているディスプレイの 1 ピクセル当たりの bit 数 (色数: color depth) を選ぶ。ここでは国立天文台共同利用 WS 群、同“可視化システム”高速画像処理機器群で IDL を利用することを想定し、それらのディスプレイの色数は 24 であったので、上記のように設定した²。これら 2 つのキーワードが設定されていないと、画像がカラーで表示されない可能性もあるので忘れずに設定して欲しい。最後のキーワード「RETAIN=2」は Backing store の設定である。値を「2」と指定することで、IDL のウィンドが他のウィンドと重なって消えた部分を再描画するためのデータを IDL に保存させる。このように設定しておかないと、ウィンドが例えばターミナルと重なる毎に再描画させなければならないので大変効率が悪い。

8.2 データの読み込み

本節ではサンプルデータを IDL に読み込ませる方法について述べる。ここで想定しているのは UNIX マシン (ex. 国立天文台共同利用 WS 群, VPP5000) で出力したサンプルデータを UNIX マシン (ex. 同 WS 群, “可視化システム” 高速画像処理機器群) で可視化する場合である。Linux マシンまたは UNIX マシンで出力したデータを Linux マシンで可視化する場合もほとんど同じである³。

本節ではサンプルデータは与えられたものとして、それを IDL に読み込ませる所から見ていく。なお本節で用いるサンプルデータを作成するためのソースプログラムや生成されるファイルについては §8.7.1 を参照のこと。

まずサンプルデータがあるディレクトリ (*data*) に移動し、IDL を起動する。そして *.r read2dss* と入力する。

(IDL の起動と *read2dss.pro* の実行)

```
r01{ochiys}1: cd ~/data
r01{ochiys}2: idl
IDL Version 5.5, Solaris (sunos sparc). (c) 2001, Research Systems, Inc.
Installation number: 70630.
Licensed for use by: NAO

IDL> .r read2dss
% Compiled module: $MAIN$.
Input file name to read =
:
```

すると、ファイル名を入力するように聞かれるので「*tsample*」と入力する
 「(「*tsample.d*」や「*tsample.D*」のように拡張子をつけないことに注意)。もし *.r read2dss* と入力した時に「No such file or directory」というメッセージが表示された場合にはパスが通っていない

²リモートログインして使用する場合についても、最近では、多くのパソコンのディスプレイも色数が 24 以上に対応しているので上記の色数の設定でも恐らく問題ないと考えた

³ただし UNIX マシンで出力したバイナリデータを Linux マシンで可視化する場合、バイナリデータを読み込む際にオブションが必要である。詳細は §8.6.2 を参照

可能性があるのでパスを通す必要がある。詳細は §8.1.2、または §8.6.1 を参照。

(ファイル名の入力)

```
IDL> .r read2dss
% Compiled module: $MAIN$.
Input file name to read =
:tsample [RET]
t=      0.00000
IDL>
```

これで IDL にデータが読み込まれた。 *read2dss.pro* の中は下記のようにになっている。

(*read2dss.pro*)

```
; read2d.pro
;   to read a data file generated by 2D code
;-
;
fn='file name'
print, 'Input file name to read = '
read, fn                                ; ファイル名の読み込み
;
ctitle='title'
precision='FLOAT'
timei=0.
;
OPENR, Unit1, fn+'.d', /GET_LUN          ;tsample.d を開く
READF, Unit1, ctitle, FORMAT='(A11)'    ; ファイルタイトル、
READF, Unit1, ixs, ix, iys, iye, FORMAT='(4I5)' ; 配列の範囲 (x 方向、
READF, Unit1, timei, FORMAT='(/,/,/,/,7X,F15.7)' ; y 方向)、時刻
READF, Unit1, precision, FORMAT='(A6)'   ; データ型の読み込み
FREE_LUN, Unit1
;
case precision of
'FLOAT ': begin
;print,'data type: ',precision
x = FLTARR(ix-ixs+1)                    ; 配列の定義 (float 型の場合)
y = FLTARR(iy-iys+1)
rho = FLTARR(ix-ixs+1, iy-iys+1)
vx = FLTARR(ix-ixs+1, iy-iys+1)
vy = FLTARR(ix-ixs+1, iy-iys+1)
bx = FLTARR(ix-ixs+1, iy-iys+1)
```

```

by = FLTARR(ixe-ixs+1, iye-iys+1)
end
else: begin
;print,'data type: ',precision           ; 配列の定義 (float 型以外の場合)
x = DBLARR(ixe-ixs+1)
y = DBLARR(iye-iys+1)
rho = DBLARR(ixe-ixs+1, iye-iys+1)
vx = DBLARR(ixe-ixs+1, iye-iys+1)
vy = DBLARR(ixe-ixs+1, iye-iys+1)
bx = DBLARR(ixe-ixs+1, iye-iys+1)
by = DBLARR(ixe-ixs+1, iye-iys+1)
end
endcase
;                                           ; tsample.D から
OPENR, Unit2, fn+'.D', /F77_UNFORMATTED, /GET_LUN ; 配列の読み込み
READU, Unit2, x
READU, Unit2, y
READU, Unit2, rho
READU, Unit2, vx
READU, Unit2, vy
READU, Unit2, bx
READU, Unit2, by
FREE_LUN, Unit2
;
print,'t=',timei
END

```

8.3 tvcn による図の作成

前節では、IDL に 2 次元のサンプルデータを読み込ませた。本節では、このデータを画像として出力する方法について述べる。ここではカラースケール、コントラスト、矢印を重ねた図 8.1 を作成する方法を見ていく。使用するメインプログラムは *tvcn2dss.pro* である。

8.3.1 表示したい物理量の選択

まず、前節のようにサンプルデータを読み込ませた後、*tvcn2dss.pro* を起動する。カラーで表示させるためには、*tvcn2dss.pro* を起動する前にカラーテーブルをロードしておく。

(*tvcn2dss* の起動)

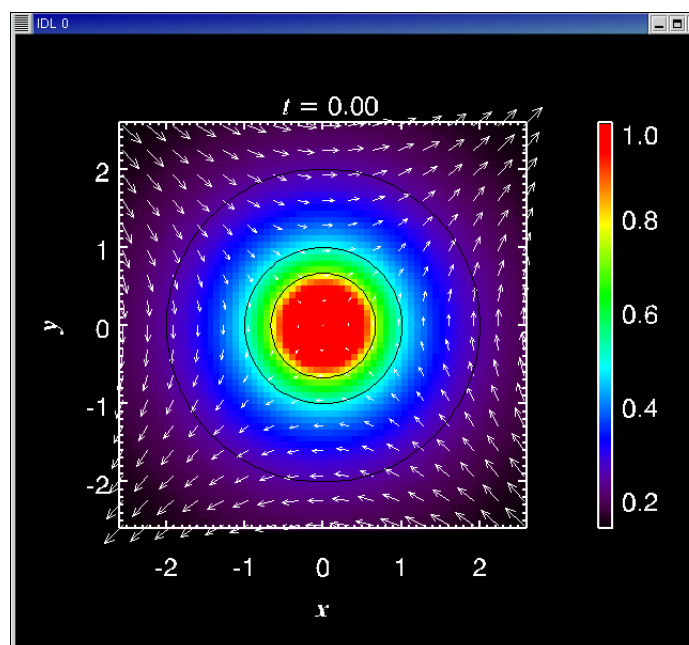


図 8.1: カラースケール、コントア、矢印を重ねた図

```

IDL> .r read2dss
% Compiled module: $MAIN$.
Input file name to read =
: tsample
t=      0.00000
IDL> loadct,39
% Compiled module: LOADCT.
% Compiled module: FILEPATH.
% Compiled module: PATH_SEP.
% LOADCT: Loading table Rainbow + white
IDL> .r tvcn2dss
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
for color scale,:

```

カラーテーブルは、もともと IDL に 41 個 (0 - 40 番) 添付されている。ここでは 39 番 (“Rainbow + white”) を使用する。メインプログラム *tvcn2dss.pro* を実行すると、まず最初にカラースケールに用いる物理量を選択するように聞かれるので、カラースケールに用いたい物理量を *rho*, *vx*, *vy*, *bx*, *by* などの中から選択する。選択できる物理量は *tvcn2dss.pro* の 30 行目付近に定義されている。

(*tvcn2dss* 30 行目付近)

```

phys=rho
print,'Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"'
read,' for color scale,:',parameter
case parameter of
    'rho':phys=rho                ; 選択可能な物理量
    'vx':phys=vx                  ; とその定義
    'vy':phys=vy                  ;
    'bx':phys=bx                  ;
    'by':phys=by                  ;
    'lrho':phys=log10(rho)        ;
    'vt':phys=sqrt(vx^2+vy^2)     ;
    'bt':phys=sqrt(bx^2+by^2)     ;
    'em':phys=(bx^2+by^2)/(8*pi)  ;
    'va':phys=sqrt(bx^2+by^2)/(4*pi*rho) ;
    '' :phys=physoldt            ;
else:begin
    print,'"lrho" will be used!'
    phys=log10(rho)
end
endcase
physoldt=phys

```

これを見ると分かる通り、*rho* を選択すると、*phys* という配列に *rho* が格納される。もし *rho*, *vx*, *vy*, *bx*, *by*, *lrho*, *vt*, *bt*, *em*, *va* または空白以外を入力すると、*rho* の常用対数が *phys* に格納される。もし表示させたい物理量が上記以外のものであれば、各自で *tvcn2dss.pro* の中に追加してもらいたい。

カラスケールに用いる物理量を入力すると、次にコントアに用いる。物理量を選択するように聞かれるのでカラスケールの場合と同様に選択する。コントアに使用できる物理量もカラスケールに使用できるものと同じである。ここではカラスケール、コントアともに *rho* を選択する。

(カラスケール、コントアの選択)

```

IDL> .r tvcn2dss
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho          ; カラスケールの選択
  for contour.:rho              ; コントアの選択
Input "v" or "b"
  for vector.:

```

次に vector (矢印) を選択するように聞かれるので、*v* または *b* を選択する。*v* は x 成分に *vx*, y 成分に *vy* を用いる。*b* は x 成分に *bx*, y 成分に *by* を用いる。ここでは *v* を選択する。

(矢印の選択)

```
IDL> .r tvcn2dss
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho
  for contour,:rho
Input "v" or "b"
  for vector.:v          ; 矢印の選択
x-y slice
velocity scale=      56.2501
IDL>
```

するとウィンドが開き、図 8.1 のような画像が表示されるはずである。

8.3.2 カラースケール範囲の指定

§8.3.1 では、カラースケールで表示させた物理量 (ρ) はその最小値から最大値まですべて表した。しかし、実際には特定のスケールだけ詳しく見たい (例えば高密度の部分だけ細かく見る) 場合も考えられる。その場合にはカラースケールの範囲を指定することで、特定のスケールを細かく表示させることが可能である。カラースケールの範囲は *tvcn2dss.pro* の 76 行目付近の「t_min」と「t_max」で指定されている。

(*tvcn2dss.pro* 76 行目付近)

```
; parameters for tv
t_max=max(phys)
t_min=min(phys)
```

上記の設定では、カラースケールの最大値、最小値は配列 *phys* (ここでは ρ) の最大値、最小値に一致している。特定のスケールレンジを細かく表示させるには、例えば下記のように書き換える。

(カラースケールの最大値、最小値の指定)

```
; parameters for tv
t_max=1.0
t_min=0.5
```

このように指定すると図 8.2 のように特定のスケールレンジ ($0.5 \leq \rho \leq 1.0$) を細かく表示できる。

8.3.3 コントアや矢印の指定

§8.3.2 では、カラーで表示する物理量のスケールレンジを指定した。ここではコントアや矢印を指定する方法を述べる。まず初めに、コントアレベルを指定する方法から見ていく。コントアレベルは *tvcn2dss.pro* の 82 行目付近で指定されている。

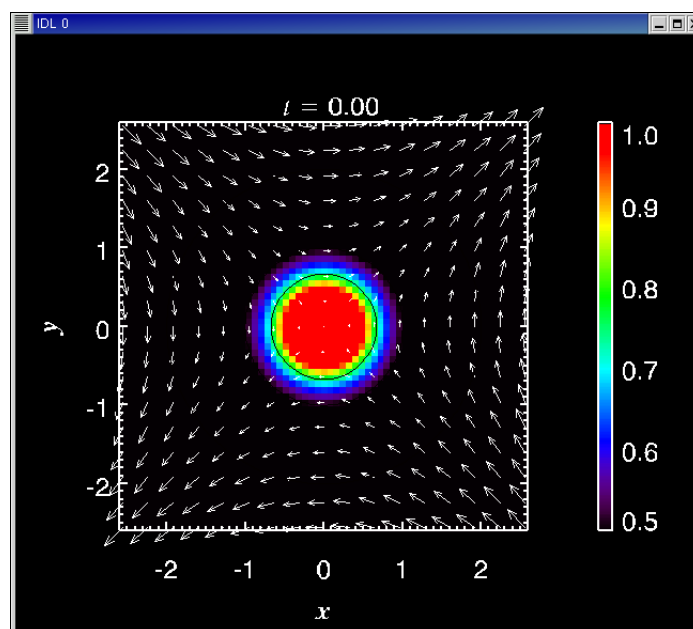


図 8.2: 図 8.1と同じであるが特定のスケールレンジ ($0.5 \leq \rho \leq 1.0$) を細かく表示させた。

(*tvcn2dss.pro* 82 行目付近)

```
; parameters for contour
c_lev=findgen(4)/4
c2_lev=findgen(16)/4-4
```

上の設定は図 8.1を作成したときの設定である。コントアレベルの指定には 2 つの変数 (*c_lev* と *c2_lev*) が用いられている。従って、例えば一方は間隔を細かくし、もう一方は間隔を広くするという使い方もできる。上の設定では一方のコントアレベル (*c_lev*) は 0 (ゼロ) から 0.75 までの範囲を 0.25 きざみで区切り、もう一方のコントアレベル (*c2_lev*) は 4 から 0 までの範囲を 0.25 きざみで区切っている。例えば以下のように書き換えると、図 8.3のようになる。

(コントアレベルの指定)

```
; parameters for contour
c_lev=findgen(10)*0.05+0.5
c2_lev=findgen(10)*0.05
```

ちなみに、コントアの色も「*c_lev*」と「*c2_lev*」で別々に設定することができる。コントアレベル「*c_lev*」と「*c2_lev*」の色の指定は *tvcn2dss.pro* の 127 行目付近にある「*c_col*」と「*c2_col*」である。下のように指定すると「*c_lev*」で指定したコントアは黒、「*c2_lev*」で指定したコントアは白になる。

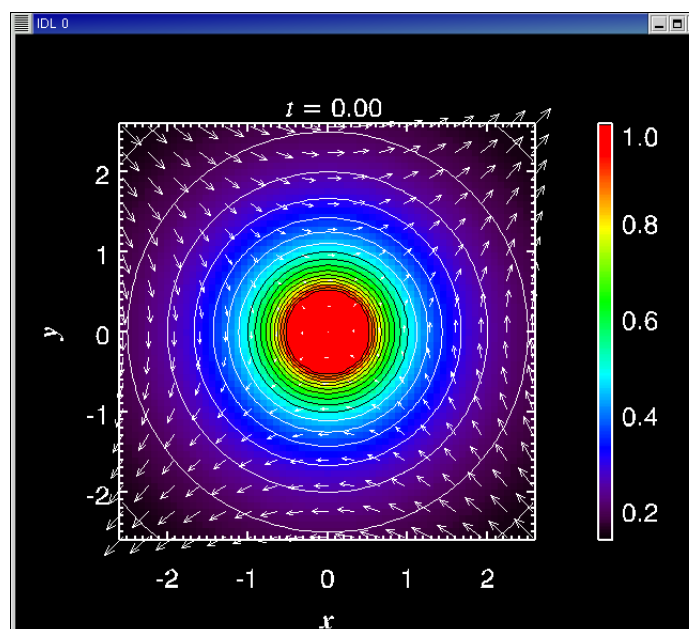


図 8.3: 図 8.1と同じであるがコントアの間隔をより狭くした。

(コントアの色)

```
levels=c_lev,c_col=1,$
l2levels=c2_lev,c2_col=0
```

ちなみに上記の 1 行目の最後にある「\$」記号はプログラム行を継続させる、という意味である。次に矢印について見ていく。矢印の間隔の指定は *tvcn2dss.pro* の 87 行目付近で指定されている。

(*tvcn2dss.pro* 87 行目付近)

```
; parameters for vector
i_skip=ixe/16
j_skip=iy/16
```

変数「*i_skip*」、「*j_skip*」はそれぞれ *x*, *y* 方向で、矢印の始点の間隔 (スキップ) を表す。変数「*ixe*」、「*iy*」はそれぞれ *x*, *y* 方向のメッシュの数 (今読み込んでいるデータ の場合 64) であるからこの場合、*i_skip* = *j_skip* = 4 (メッシュ) となる。従って、矢印の始点は 4 メッシュごとに並んでいる。言い替えると、*x*, *y* 方向それぞれに 17 個の矢印が表示される。下記のように書き換えると、矢印は *x*, *y* 方向それぞれに 26 個ずつ配置され、それは図 8.4 のようになる。

(矢印間隔の指定)

```

; parameters for vector
i_skip=ixe/25
j_skip=iyj/25

```

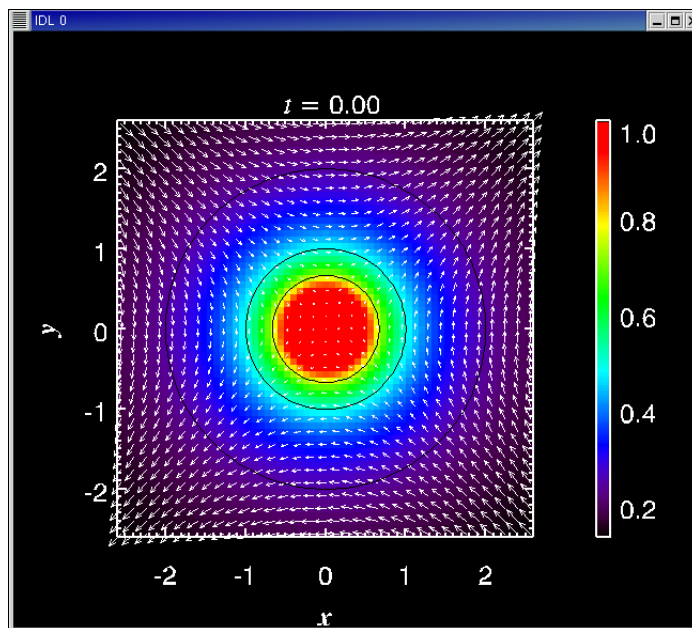


図 8.4: 図 8.1と同じであるが、矢印の数を増やした。

次に、矢印の長さについて述べる。矢印の長さは *tvcn2dss.pro* 113 行目付近に指定されている。

(*tvcn2dss.pro* 113 行目付近)

```

v_scale=long(max(sqrt(sv1^2+sv2^2)))/((x(1)-x(0))*i_skip)
if (v_scale le 0.) then $
  v_scale=max(sqrt(sv1^2+sv2^2)))/((x(1)-x(0))*i_skip)

```

上の設定では $v = \sqrt{v_x^2 + v_y^2}$ の最大値を基にスケールされている (今の場合 $sv1 = v_x$, $sv2 = v_y$ である)。現在の「v_scale」の値は「*tvcn2dss.pro*」を実行した際、コマンドライン上に、「velocity scale= ****」という形式で表示される。

(v_scale の値)

```

IDL> .r tvcn2dss
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho
  for contour,:rho
Input "v" or "b"
  for vector.:v
x-y slice
velocity scale=      56.2501    ; v_scale の値
IDL>

```

もし任意の速度スケールを固定したい場合は、例えば下記のように変数「v_scale」に定数を代入し、他の「v_scale」についての記述はコメントアウトする。代入する値は表示される「v_scale」の値を参考にすると良い。ただし、矢印の長さは代入した「v_scale」の値に反比例することに注意。

(矢印の長さの指定)

```
v_scale=30.0
```

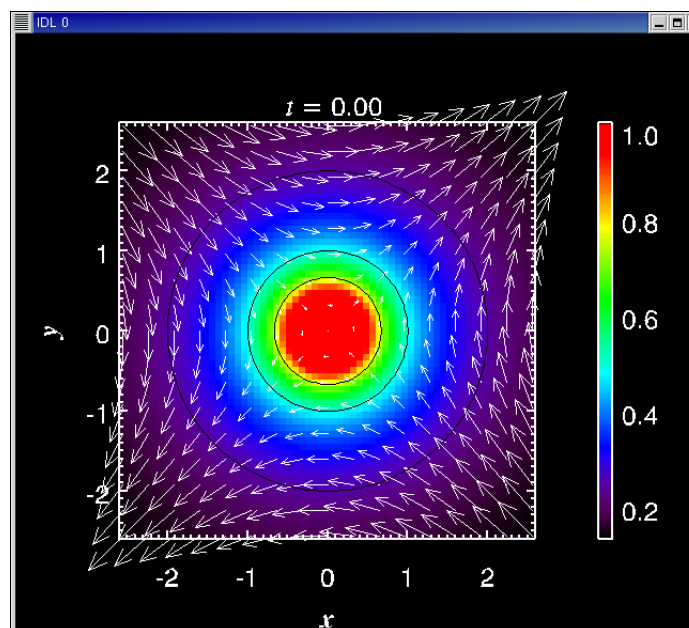


図 8.5: 図 8.1と同じであるが、矢印を長くした。

矢印の色も指定することができる。例えば、矢印の色とカラースケールの色が近く、区別が付きにくい場合には適宜変更してもらいたい。矢印の色の指定は *tvcn2dss.pro* の 124 行目にある「v_col」に 0 から 255 の値を代入することでできる。もし「v_col=-1」とすると、矢印の色は白色になる。下記は図 8.1 を作成した時の設定である。

(*tcn2dss.pro* 124 行目: 矢印の色)

```
iskip=i_skip,jskip=j_skip,v_col=-1,scale=v_scale,$
```

8.3.4 タイトルの指定

前節までに表示させる物理量を選択したり、カラスケール、コントラ、矢印を調節する方法について見てきた。本節では画面に表示させるタイトルの指定方法について見ていく。

図 8.1 では、時間をタイトルとした。タイトルは *tcn2dss.pro* の 129 行目付近で指定している。

(*tcn2dss.pro* 129 行目付近)

```
; parameters for title
stime=STRING(FORMAT='(F5.2)',timei)
xyouts,-max(x)*0.2,max(y)*1.05,/data,'!8t!3 =' + stime
```

「stime」には時刻「timei」の値を文字列に置き換えたものが入る。関数「STRING」は引数を文字列に変換する。プロシージャ「xyouts」は文字列を任意の位置に表示させる。ここでは「/data」によりデータ座標系で位置を指定している（データ座標の詳細は Introduction to IDL version 5.1 (アダムネット株式会社 1998) p94 を参照）。インデックス「!8」や「!3」は文字の種類を指定する。インデックス「!8」は「Complex Italian」、「!3」は「Simplex Roman」を表す。その他の種類については Introduction to IDL version 5.1 (アダムネット株式会社 1998) の p109 表 7 を参照。

文字列以外にも、任意の長さの線を表示させることも可能である。例えば、基準の速度の大きさを矢印で表し、それとデータ領域内の矢印と比較することができる。

(タイトルに矢印を追加する方法)

```
v0=5.0
arrow,max(x)*0.6,max(y)*1.10,max(x)*0.6+(v0/v_scale),max(y)*1.10,/data,$
hsize=30.*v0/v_scale
xyouts,max(x)*0.65+(v0/v_scale),max(y)*1.09,'!8v=1.0 v!D0!N!3',/data
```

図 8.1 を作成する時、矢印に v (速度) を選択した。ここでは基準速度を想定し、それを「 v_0 」と書くことにする。そして、この「 v_0 」の大きさを持つ矢印をタイトルの位置に表示させることを考える。図 8.6 はこのプログラムを使って矢印をタイトルの位置に表示させた例である。基準速度「 v_0 」の大きさは利用時に適宜変更してもらいたい。ここでは仮に「5.0」と置くことにする。プロシージャ「arrow」は頭のついた矢印を表示するものである。ここでは長さが「 v_0 」の矢印を表示させるように指示している（値「 v_0 」を「v_scale」でスケールしている理由は、データ領域内の矢印も同じ「v_scale」でスケールしているから）。プロシージャ「arrow」の詳細については IDL Reference Guide A-M version 5.3 (Research Systems, Inc. 1999) p 51 を参照のこと。上記プログラムの最後の行は「 $v = 1.0 v_0$ 」という文字列を、上の行で出力した矢印の隣に配置するように指示している。最後の行の中で「!D」はそれに続く「0」を下付添字に指定している。インデックス「!N」は下

付 (または上付) 添字から元に戻すための指示である。ちなみに上付添字は「!U」で指定できる。

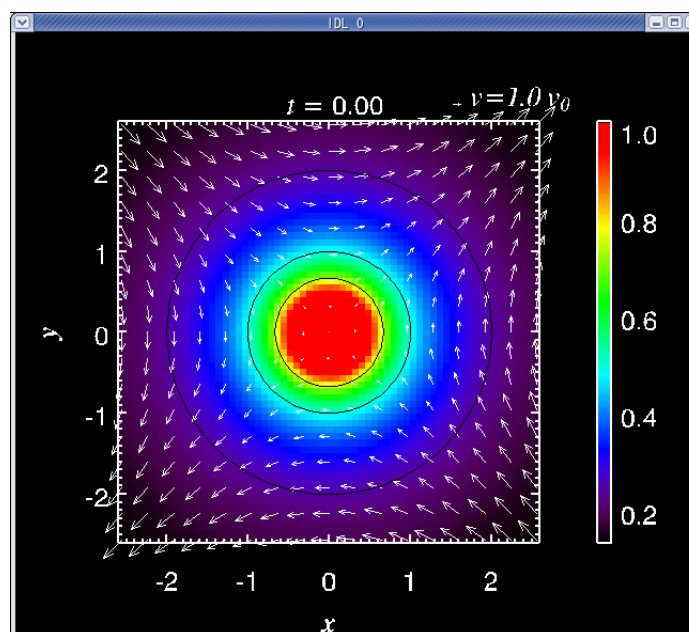


図 8.6: 図 8.1と同じであるが、タイトルの位置に基準速度「 v_0 」と同じ大きさをもつ矢印 (図右上) を追加した。

8.3.5 文字の指定

図中の文字の大きさや線の太さは *tvcn2dss.pro* の 17 -22 行目に指定されている。

(*tvcn2dss.pro* 17 - 22 行目)

```
!p.font=1           ; graphics text font system を指定
!p.charsize=3       ; 文字の大きさの指定
!x.thick=2.5        ; x 方向に沿った線の太さ
!y.thick=2.5        ; y 方向に沿った線の太さ
!z.thick=2.5        ; z 方向に沿った線の太さ
!p.thick=1.5        ; コントアや矢印の太さ
```

最初の *!p.font* は graphics text font を指定するもので、上記のように整数「1」を代入すると、True Type font systemが使用される。True Type font system では、文字は多角形の集合として表される (ほとんど画像として扱われる)。もし *!p.font* に値「0」を代入すると文字は device font (hardware font) として扱われる。device font では文字がディスプレイなどのデバイスから直接与えられる (デバイスには依存するが、文字は文字のまま扱われる)。どちらのタイプの font を選択するかについては、もし画像をポストスクリプトファイルとして出力して後からイラストレータなどで文字を編集することが想定される場合には、device font を選択することを勧める。なお graphics

text font の詳細については IDL Reference Guide (Objects & Appendixes) version 5.3 (Research Systems, Inc. 1999) p 2196 以降を参照のこと。

最後の *!p.thick* はコントラストや矢印などの線の太さを指定する。記号 *!p* に続く変数はプロットシステム変数と呼ばれ、全てのプログラムユニットが使用できる定義済の変数である。上記のように定義することによって、このプログラムの中では線の太さは *thick=1.5* に指定される。システム変数の詳細については Introduction to IDL version 5.1 (アダムネット株式会社 1998) p 35 -37, IDL Reference Guide (Objects & Appendixes) version 5.3 (Research Systems, Inc. 1999) p 2157 以降を参照のこと。

8.3.6 一部の領域だけ拡大表示させる方法

ここまでは IDL に読み込ませた配列の全領域を表示させてきた。しかし、場合によっては一部の領域だけ可視化すれば良いこともある。そこで本節では読み込んだ配列の内、一部の領域だけ拡大し、可視化する方法について見ていく。

一部の領域だけ拡大するために、まず *read2dss.pro* を使って IDL に配列全てを読み込ませ、次に各配列の必要な部分だけを格納した新たな配列を作成し、その新たな配列を可視化するという方法をとる。新たな配列を作成し、それを基に可視化するメインプログラムが *tvcn2dclosess.pro* である。この *tvcn2dclosess.pro* を使って、図 8.6 の中心部分だけ拡大したのが、図 8.7 である。

(*tvcn2dclosess.pro* の実行)

```
IDL> .r read2dss                ; データの読み込み
% Compiled module: $MAIN$.
Input file name to read =
: tsample
t=      0.00000
IDL> .r tvcn2dclosess           ; tvcn2dclosess の実行
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
for color scale,:rho           ; カラースケールの選択
for contour.:rho               ; コントラストの選択
Input "v" or "b"
for vector.:v                  ; 矢印の選択
x-y slice
velocity scale=      36.9231
IDL>
```

全領域を可視化する *tvcn2dss.pro* の大きな違いは、可視化する部分の情報だけを持った配列を新たに作成することである。下記のメインプログラムは *tvcn2dclosess.pro* の中で、可視化する部分の情報だけを持った配列を新たに作成している部分である。

(*tvcn2dclosess.pro* 24 行目以降)

```
; array subscript           ; 元々の配列の内、どの部分を切り抜くかを指定。
ixss=(ixe-ixs+1)/4         ; ここでは中心領域（全領域の 1/4 の面積）を指定。
ixes=(ixe-ixs+1)*3/4
```

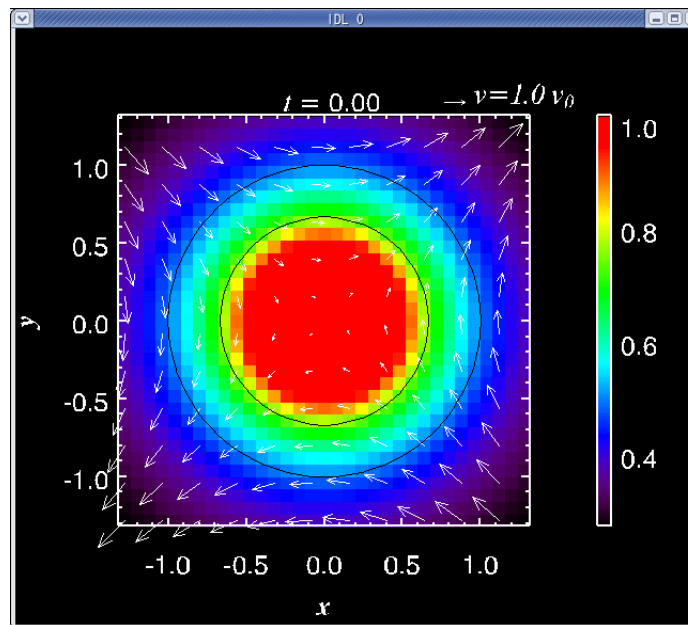


図 8.7: 図 8.6と同じであるが、中心部分だけ拡大した。右端のカラーバーの範囲が図 8.6よりも狭くなった理由は、中心領域だけ見た場合の ρ の最小値が、全領域で見た場合の最小値と比べて大きいからである。今の場合、カラーバーの下限値は ρ の下限値に合わせている。

```

iyss=(iye-iys+1)/4
iyes=(iye-iys+1)*3/4
;
; array definition
case precision of
'FLOAT ': begin
;print,'data type: ',precision
xs = FLTARR(ixes-ixss+1)           ; 中心領域のデータを格納する
ys = FLTARR(iyes-iyss+1)           ; 配列の定義 (データの精度が
rhos = FLTARR(ixes-ixss+1, iyes-iyss+1) ;float 型の場合)
vxs = FLTARR(ixes-ixss+1, iyes-iyss+1)
vys = FLTARR(ixes-ixss+1, iyes-iyss+1)
bxs = FLTARR(ixes-ixss+1, iyes-iyss+1)
bys = FLTARR(ixes-ixss+1, iyes-iyss+1)
;
end
else: begin
;print,'data type: ',precision
xs = DBLARR(ixes-ixss+1)           ; 中心領域のデータを格納する
ys = DBLARR(iyes-iyss+1)           ; 配列の定義 (データの精度が

```

```

rhos = DBLARR(ixes-ixss+1, iyes-iyss+1) ;float 型以外の場合)
vxs = DBLARR(ixes-ixss+1, iyes-iyss+1)
vys = DBLARR(ixes-ixss+1, iyes-iyss+1)
bxs = DBLARR(ixes-ixss+1, iyes-iyss+1)
bys = DBLARR(ixes-ixss+1, iyes-iyss+1)
end
endcase
;
;substitution ;データを新しい配列 (xs) に格納
for i=ixss,ixes do begin
xs(i-ixss)= x(i)
endfor
;
for j=iyss,iyes do begin
ys(j-iyss)= y(j) ;データを新しい配列 (ys) に格納
endfor
;
for j=iyss,iyes do begin
  for i=ixss,ixes do begin
rhos(i-ixss,j-iyss)=rho(i,j);データを新しい配列 (rhos) に格納
vxs(i-ixss,j-iyss)=vx(i,j) ;データを新しい配列 (vxs) に格納
vys(i-ixss,j-iyss)=vy(i,j) ;データを新しい配列 (vys) に格納
bxs(i-ixss,j-iyss)=bx(i,j) ;データを新しい配列 (bxs) に格納
bys(i-ixss,j-iyss)=by(i,j) ;データを新しい配列 (bys) に格納
  endfor
endfor

```

上記プログラム以外の部分は *tvcn2dss.pro* と *tvcn2dclosess.pro* はほとんど変わらない (変数 x が xs に置き換わる程度)。

8.3.7 複数の図を表示する方法

前節までは一つの画面に一つのデータを表示させる方法を見てきた。実際には一つの画面に異なるデータを並べて表示させて、データを比較する場面もある。本節では複数のデータを読み込んで、一つの画面に表示させる方法について見ていく。

図は時刻 $tstep$ の異なる 4 つのデータ (*sample0*, *sample1*, *sample2*, そして *sample3*) を一つの画面に並べて表示させたものである。

一つの画面に並べて表示させるためのメインプログラムは *tvcn2d4pss.pro* である。メインプログラム *tvcn2d4pss.pro* だけではデータファイルを読み込み、可視化の両方を行うので、*read2dss.pro* を実行する必要はない。

(*tvcn2d4pss.pro* の実行)

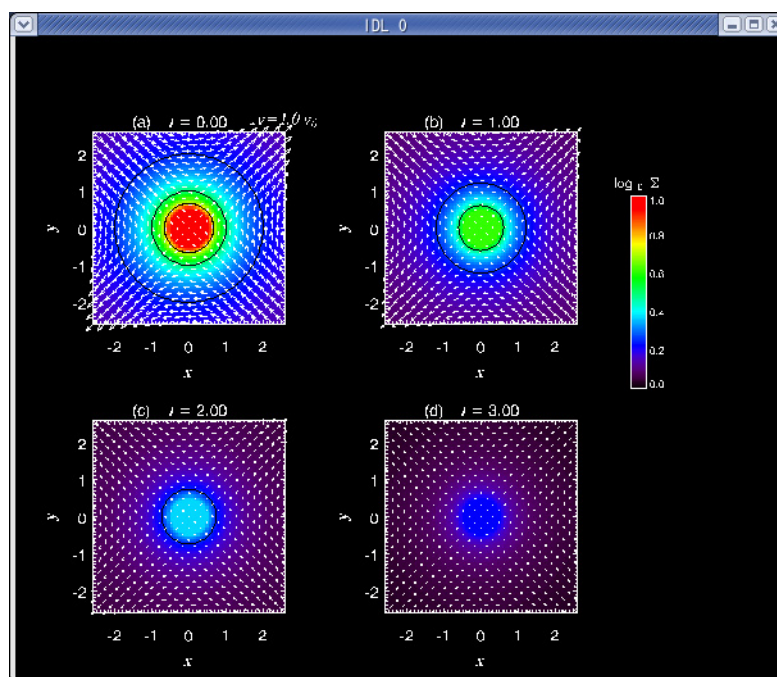


図 8.8: 異なる 4 つの時刻 ($tstep$) のデータを一つの画面に並べた。

```
IDL> .r tvcn2d4pss
% Compiled module: $MAIN$.
t=      0.00000
velocity scale=    56.2501
velocity scale=    56.2501
t=      1.00000
velocity scale=    56.2501
velocity scale=    56.2501
t=      2.00000
velocity scale=    56.2501
velocity scale=    56.2501
t=      3.00000
velocity scale=    56.2501
velocity scale=    56.2501
IDL>
```

データファイル名や画面上の位置、そして図を識別する記号は *tvcn2d4pss.pro* 中の 17 - 22 行目に指定している。下記の例ではデータファイルに (*sample0*, *sample1*, *sample2*, そして *sample3*) を指定している。

(*tvcn2d4pss.pro* 17 -22 行目; ファイル、位置、記号の指定)

```
fn=['sample0','sample1','sample2','sample3'] ; ファイル名の指定
px1=[0.10,0.48,0.10,0.48] ; 各図の左下の x 座標
py1=[0.55,0.55,0.10,0.10] ; 各図の左下の y 座標
```

```

px2=[0.40,0.78,0.40,0.78]      ; 各図の右上の x 座標
py2=[0.85,0.85,0.40,0.40]      ; 各図の右上の y 座標
panel_n=['(a)','(b)','(c)','(d)'] ; 各図につける記号の指定

```

一つの画面に 6 つの図を並べて表示したい場合は、上記のやり方でファイル名、各図の座標や記号を 6 つ指定すればよい。ただし、本節では 4 つの図を並べて表示させる *tvcn2d4pss.pro* について説明する (6 つの図を表示させるメインプログラム *tvcn2d6pss.pro* についても基本的なやり方は同じである)。

表示させるカラースケール、コントア、そして矢印の指定は *tvcn2d4pss.pro* の中の 153 行目から 158 行目の間に直接書き込む。下の例ではカラースケールに rho, コントアに rho, 矢印に速度 v (sv1, sv2 はそれぞれ矢印の x, y 成分である) を選択している。

(カラースケール、コントア、そして矢印の指定)

```

; color & countour & vector ;;;;;;;;;;;;;;
;color scale
phys=rho
;contour
physc=rho
;vector
sv1=vx
sv2=vy
;;;;;;;;;;;;;

```

もし、カラースケール、コントアに rho の常用対数、矢印に磁場 b を指定する場合は以下のようにすればよい。

(カラースケール、コントア、そして矢印の変更)

```

; color & countour & vector ;;;;;;;;;;;;;;
;color scale
phys=alog10(rho)
;contour
physc=alog10(rho)
;vector
sv1=bx
sv2=by
;;;;;;;;;;;;;

```

また、矢印の間隔、長さ、カラースケールの最大値、最小値、そしてコントアの間隔を指定する場合は *tvcn2d4pss.pro* の 74 行目から 87 行目の部分を §8.3.2, §8.3.3 と同じやり方で編集すればよい。ここでの指定は 4 つの図に共通して適用される。

(*tvcn2d4pss.pro* 74 - 87 行目)

```

; parameters for vector
i_skip=ixe/16
j_skip=iye/16
v_scale=long(max(sqrt(vx^2+vy^2)))/((x(1)-x(0))*i_skip)
if (v_scale le 0.) then $
    v_scale=max(sqrt(vx^2+vy^2))/((x(1)-x(0))*i_skip)
;
; parameters for tv
t_max=1.0
t_min=0.
; contour levels
;
c_top=!d.table_size
c_lev=findgen(4)/4
c2_lev=findgen(16)/4-4

```

カラーバーの位置と向きも自由に変えることができる。位置と向きは *tvcn2d4pss.pro* の 191 行目以降で指定している。下記は図 8.8 を作成したときの設定である。

(*tvcn2d4pss.pro* 191 行目付近: カラーバーの位置と向き)

```

; portrait
beginp, [0.80, 0.45, 0.82, 0.75]
    c_scale, phys, top=c_top-3+1, max=t_max, min=t_min, bottom=1, $
/portrait, yaxis=1, title='log!D10!N !9 S !3'
; landscape
;beginp, [0.400, 0.960, 0.670, 0.975]
; c_scale, phys, top=c_top-3+1, max=t_max, min=t_min, bottom=1, xaxis=0, $
;title='log!D10!N !9 S !3'

```

プロシージャ *beginp* は画面上で画を描く領域を指定するものであり、この宣言の後にカラーバーを表示したい座標を指定する。座標は「始点の x 座標、始点の y 座標、終点の x 座標、終点の y 座標」の順番である。キーワード「title」はカラーバーのタイトルである。もし、カラーバーを横向きに表示させたい場合は、上記の「; portrait」から「; landscape」の間をコメントアウトして、「; landscape」以下の部分をコメントインすればよい。

8.3.8 流線を表示させる方法

本節では流線を表示させる方法について見ていく。流線は、矢印だけでは分かりにくいガスの流れを理解するための有効な手段である。流線を表示させるメインプログラムは *tvcn2dSLss.pro* である。図 8.9 は図 8.6 の上に流線を重ねた結果である。作成方法は下記の通りである。

(*tvcn2dSLss.pro* の実行)

```

IDL> .r read2dss                      ; データの読み込み
% Compiled module: $MAIN$.
Input file name to read =
: tsample
t=      0.00000
IDL> .r tvcn2dSLss                    ; tvcn2dSLss の実行
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho                ; カラースケールの選択
  for contour.:rho                    ; コントアの選択
Input "v" or "b"
  for vector.:v                       ; 矢印の選択
x-y slice
% Compiled module: TVCN2VE.
velocity scale=      56.2501
IDL>

```

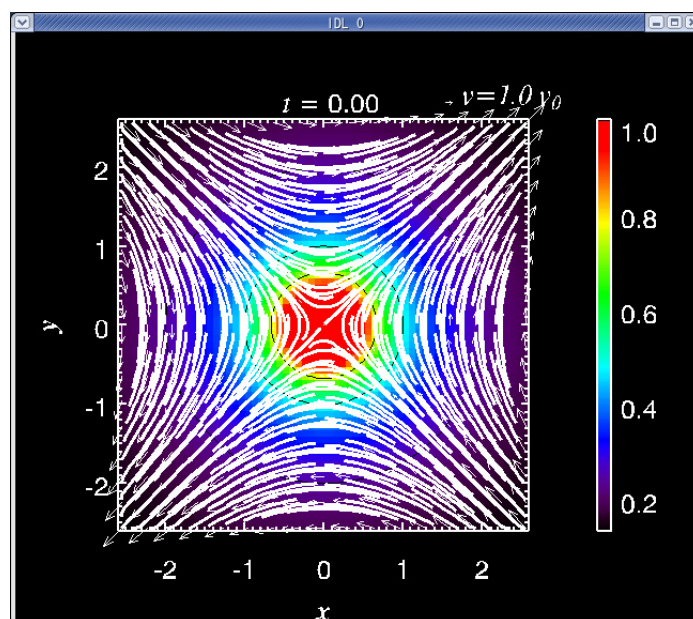


図 8.9: 図 8.6に流線を重ねた図

このメインプログラムの中での流線の求め方は、流線の式

$$\frac{dx}{ds} = U_x \quad (8.1)$$

$$\frac{dy}{ds} = U_y \quad (8.2)$$

をデータの中心から外側に向かって積分（一次精度）することである。ここで ds は流線に沿った微小長さ、 U_x , U_y はそれぞれ

$$U_x = v_x / \sqrt{v_x^2 + v_y^2 + \epsilon} \quad (8.3)$$

$$U_y = v_y / \sqrt{v_x^2 + v_y^2 + \epsilon} \quad (8.4)$$

で定義される (ϵ は微小量)。この積分は、データの全領域を幾つかの領域に分割して、それぞれの領域ごとに個別に行っている。分割した領域が大き過ぎると積分のとき誤差が大きくなり、正しい流線が得られない。逆に分割した領域が小さ過ぎると、隣り合う流線の間隔が狭くなり、見づらくなる。従って適切な大きさに分割しなければならない。分割の大きさを指定している変数は *iskip2*(x 方向) と *jskip2*(y 方向) である。

(*tvcn2dSLss.pro* 75 - 76 行目)

```
iskip2=(nmax-1)/20
```

```
jskip2=(nmax-1)/20
```

上記の例では x, y 方向ともにデータの全領域を 20 個に分割している (*nmax* は x または y 方向のメッシュの数である)。

8.4 ファイルの出力方法

前節までは、作成した画像をディスプレイに表示させてきた。実際には、作成した画像をプレゼンテーションや L^AT_EX の文章に挿入したりするためにファイルの形で残したい場合がある。本節では作成した画像をファイルの形で出力する方法について見ていく。

8.4.1 PNGファイルの出力

まず最初に、PNG 形式の画像を作成するメインプログラム *tvcn2dss_png.pro* について見ていく。PNG 形式の画像はパワーポイントなどのプレゼンテーションツールに使用することができる。メインプログラム *tvcn2dss_png.pro* の使用方法は §8.3.1 から §8.3.4 で説明した *tvcn2dss.pro* の使用方法とほとんど同じである。

(*tvcn2dss_png.pro* の使用方法)

```

IDL> .r read2dss                ; データの読み込み
% Compiled module: $MAIN$.
Input file name to read =
: tsample
t=      0.00000
IDL> .r tvcn2dss_png            ; tvcn2dss_png の実行
% Compiled module: $MAIN$.
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho           ; カラースケールの選択
  for contour.:rho              ; コントアの選択
Input "v" or "b"
  for vector.:v                 ; 矢印の選択
x-y slice
velocity scale=      56.2501
IDL>

```

メインプログラム *tvcn2dss.pro* と異なり、ディスプレイに画像は表示されない。

図 8.10 は *tvcn2dss_png.pro* を使って作成した PNG ファイル (L^AT_EX に挿入するために PNG 形式から EPS 形式に変換したが...) である。

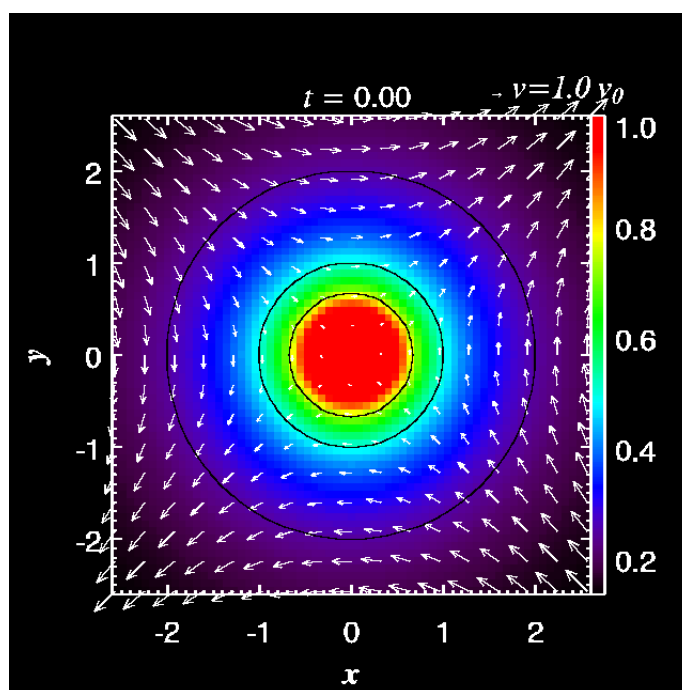


図 8.10: 図 8.6 と同じであるが、PNG 形式のファイルに書き出した (L^AT_EX に挿入する都合上 PNG 形式から EPS 形式に変換した)。

出力されるファイル名は *tvcn2dss_png.pro* の 143 行目、プロシージャ *write_png* の行に指定されている。下記の例では *ochif10.png* である。

(ファイル名の指定; *tvcn2dss.png.pro* 143 行目)

```
write_png,'ochif10.png', TVRD(),r,g,b
```

画像の大きさは *tvcn2dss.png.pro* の 18 行目 で指定する。下記の例では縦、横ともに 640 ピクセルである。

(画像の大きさ指定; *tvcn2dss.png.pro* 18 行目)

```
xsize=640 & ysize=640
```

8.4.2 EPS ファイルの出力

次に EPS ファイルを出力する方法について見ていく。使用するメインプログラムは *tvcn2dss.eps.pro* であり、実行の仕方は *tvcn2dss.pro* とほぼ同じである。

(*tvcn2dss.eps.pro* の使用方法)

```
IDL> loadct,40          ; カラーテーブルのロード
% LOADCT: Loading table Rainbow + black
IDL> .r read2dss        ; データの読み込み
% Compiled module: $MAIN$.
Input file name to read =
: tsample
t=      0.00000
IDL> .r tvcn2dss_eps    ; tvcn2dss.eps の実行
% Compiled module: $MAIN$.
  xsize = 12.7cm,  ysize = 12.7cm
xoffset =  3.7cm, yoffset =  7.8cm
Input "rho,vx,vy,bx,by,lrho,vt,bt,em,va" or "return"
  for color scale,:rho    ; カラースケールの選択
  for contour.:rho       ; コントアの選択
Input "v" or "b"
  for vector.:v          ; 矢印の選択
x-y slice
velocity scale=      56.2501
Postscript device is closed
IDL>
```

メインプログラム *tvcn2dss.png.pro* を実行したときと同様、画像はディスプレイに表示されず、ファイルに出力される。図 8.11 はその出力された EPS ファイルである。

出力されるファイル名と大きさは *tvcn2dss.eps.pro* の 17 行目で指定している。下記の例では、ファイル名が *ochif11.eps*、縦、横の長さが 12.7 cm となる。

(ファイル名と大きさの指定; *tvcn2dss.eps.pro* 17 行目)

```
ps1,/col,/in,psfile='ochif11.eps',xsize=12.7,ysize=12.7
```

上記プログラム最初の *ps1* は、device を “X” から “PS” に切替え、Postscript ファイルに画を出力するプロシージャ (*ps1.pro*) を呼び出す。キーワード */col* はカラーの作画を描く際にセットする。

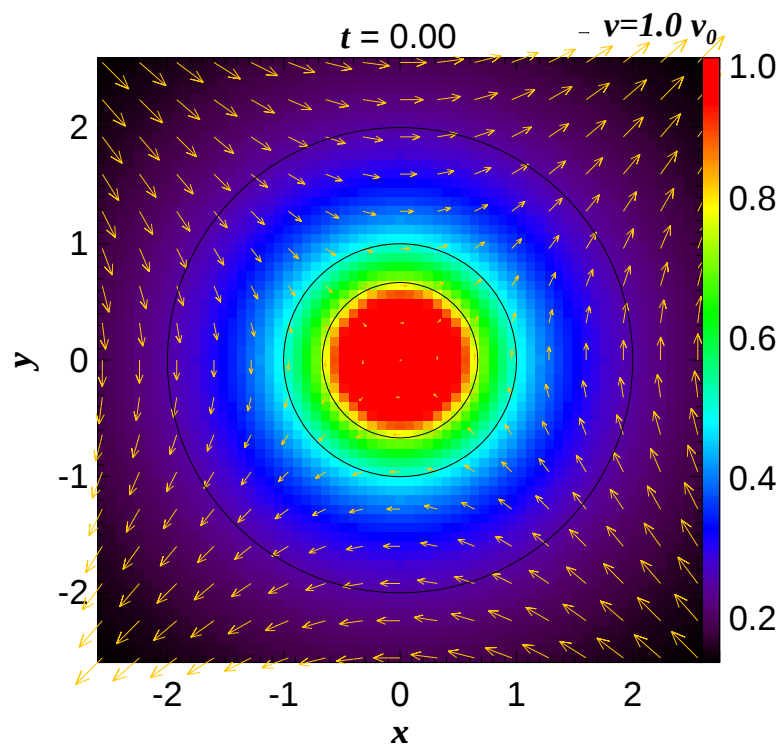


図 8.11: 図 8.6と同じであるが、EPS 形式のファイルに書き出した。ただし、矢印は黄色にしている。矢印の色の指定については §8.3.3 を参照のこと。

キーワード */in* は “X” のカラーインデックスを “Postscript” のカラーインデックスに interpolation することを指示する。

8.4.3 連続スナップショットの作成方法

前節までは、基本的には一つのファイルを読み込ませて、一つの画像を出力するという方法について見てきた。しかし時間発展 (ex. 原始連星にガスが降着していく過程など) を追跡したい場合など、多数のバイナリデータを同じような形式で可視化する場面もある。その場合、一つ一つのデータを手動で「読み込み」、「画像出力」を行うのは非常に面倒である。そこで本節では自動でデータの「読み込み」、「画像出力」を行う方法について見ていく。

使用するメインプログラムは *write2dpng.pro* である。このメインプログラムはムービー（「パラパラ漫画」）を作成するためのプログラムであり、たくさんの指定したバイナリデータを自動で読み込み、PNG 形式で画像を出力する。ただしここで作成できるのは各時間ステップごとの静止画だけであり、動画にするためには動画作成ソフト (例えば Quick Time Pro) が必要である。なお、ここで作成した静止画をムービーにする方法については §8.4.4 で述べる。図 8.12 は *write2dpng.pro* を実行して作成された図である。データは §8.3.7 で使用したのと同じである (*sample0*, *sample1*, *sample2*, そして *sample3*)。

(*write2dpng.pro* の実行)

```
IDL> .r write2dss_png
% Compiled module: $MAIN$.
Input "v" or "b"
for vector.:v                ; 矢印の選択。ここでは ‘v’ を選択。
t=      0.00000
x-y slice
velocity scale=      56.0000
% Loaded DLM: PNG.
t=      1.00000
x-y slice
velocity scale=      56.0000
t=      2.00000
x-y slice
velocity scale=      56.0000
t=      3.00000
x-y slice
velocity scale=      56.0000
IDL>
```

カラスケールやコントラに用いる物理量の選択、カラスケールの最大値、最小値の指定、コントラや矢印の間隔の指定は *write2dpng.pro* の 91 - 108 行目で行う。指定の仕方は §8.3.7 で述べたやり方と同じである。

(*write2dpng.pro* 91 - 108 行目; カラスケールやコントラなどの指定)

```
phys=rho                ; カラスケールの選択
physc=rho              ; コントラの選択
;
```

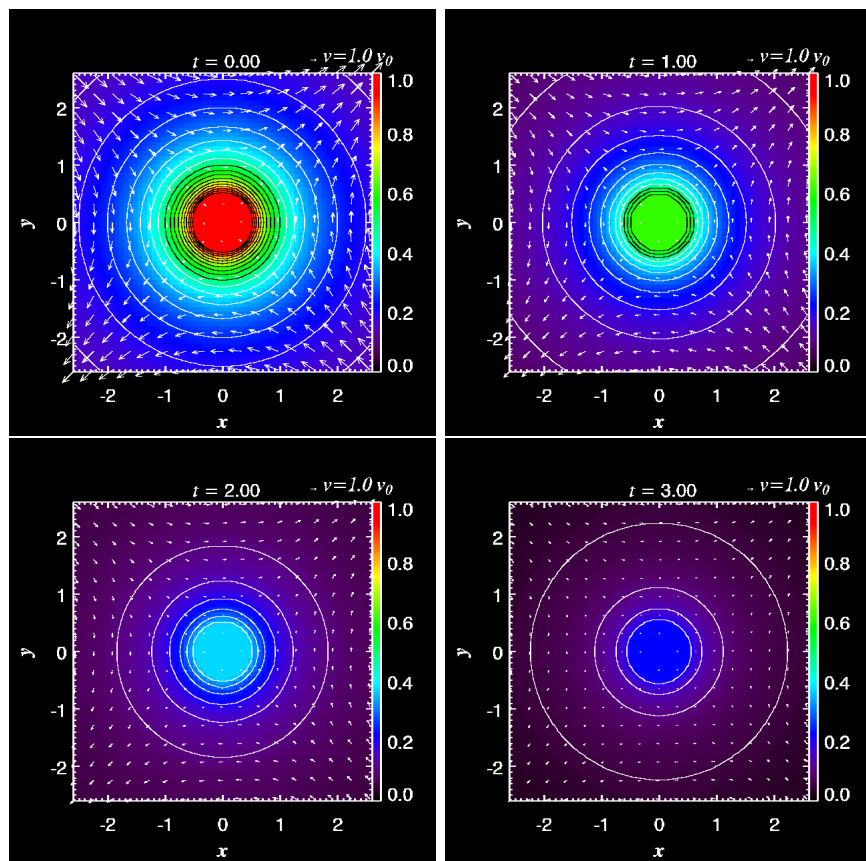


図 8.12: 異なる 4 つの時刻 ($tstep$) のデータ ($sample0$, $sample1$, $sample2$, そして $sample3$) を自動で読み込み、PNG 形式でファイルに出力させた (L^AT_EX に挿入する都合上、後から EPS 形式に変換している)。

```

; parameters for tv
t_max=1.0          ; カラースケールの最大値の指定
t_min=0.           ; カラースケールの最小値の指定
;
;
;
; parameters for contour
. . .
c_lev=findgen(10)*0.05+0.5 ; コントア間隔の指定
c2_lev=findgen(10)*0.05    ; コントア間隔の指定
;
; parameters for vector
i_skip=ixe/16          ; 矢印間隔の指定
j_skip=iye/16          ; 矢印間隔の指定

```

読み込みむファイル名と出力ファイル名の指定

メインプログラム *write2dss_png.pro* では、下記のやり方で読み込むデータを指定し、PNG ファイルを出力している。

(*write2dss_png.pro* 35 - 41 行目)

```

fn1=findfile('sample*.d') ; 読み込むファイルを探す
fn=strmid(fn1,0,7)        ; 部分文字列の抽出
;
its=0                     ; 繰り返しの始値
ite=size(fn,/n_elements)-1 ; 繰り返しの終値
;
for l=its,ite do begin    ; 繰り返しの始まり

```

上記のメインプログラムの主な点について説明する。関数 *findfile* はカレントディレクトリにある *sample*.d* (* はワイルドカード) という名前のファイルを探し、その名前を文字列型の配列 *fn1* に代入する。文字列の中に数字がある場合は、小さい数字を含む名前から先に格納される。関数 *strmid* は文字列から文字を抽出する。上記の指定では文字列の左端 (0 番目) から数えて 7 個の文字を抽出する。関数 *size* は上記の場合、文字列型配列 *fn* の要素数がいくつであるかを返し、それから 1 を引いて変数 *ite* に代入する。例えばカレントディレクトリに (*sample0.d*, *sample1.d*, *sample2.d*, そして *sample3.d*) というファイルがある場合、それぞれの関数が返す結果は以下のようになる。

(各関数の返す値)

```

IDL> $!s sample*.d
sample0.d sample1.d sample2.d sample3.d
IDL> fn1=findfile('sample*.d')
IDL> print,fn1
sample0.d sample1.d sample2.d sample3.d
IDL> fn=strmid(fn1,0,7)
IDL> print,fn
sample0 sample1 sample2 sample3
IDL> ite=size(fn,/n_elements)-1
IDL> print,ite
      3
IDL>

```

(*write2dss_png.pro* 158 -160 行目)

```

tv!ct,r,g,b,/get      ; RGB color system から color table を読み込む
write_png,'sample'+strmid(fn(1),6,1)+'.png', TVRD(),r,g,b ;PNG file へ出力
endfor                ; 繰り返しの終り

```

上記は PNG ファイルの画像を出力している部分である。PNG ファイル名は、*sample?.png* であり、? には読み込んだデータファイル名と同じ数字が入る (ex. 読み込んだデータファイル名が *sample2.d* であれば、出力される PNG ファイル名は *sample2.png* である)。プロシージャ *tv!ct* は指定した変数からカラーテーブルを読み込む。ここではキーワード *get* により現在のカラーテーブルのベクトルを変数、*r,g,b* に取り込ませている。プロシージャ *write_png* は画像を PNG ファイルに書き込む。関数 *TVRD* は device の指定された部分のコンテンツ (各ピクセルの情報; 0 または 1 で表される) を返す。これらの関数やプロシージャの詳細については IDL Reference Guide A-M version 5.3, IDL Reference Guide N-Z version 5.3 (Research Systems, Inc. 1999) を参照のこと。

8.4.4 ムービーの作成

前節では自動的に静止画を作成する方法について見てきた。本節ではこれらの静止画を使ってムービー (「パラパラ漫画」) を作成する方法の一つを紹介する。ここでは Windows にインストールされた Quick Time Pro を使って作成する方法について説明する (ただし Quick Time Pro はシェアウェアである)。

まず、前節で作成した、時間ステップ毎にファイル名のついた静止画 (イメージシーケンス) を Windows 上のフォルダに保存する。次に Quick Time Pro を起動し、

「ファイル」→「イメージシーケンスを開く (Q)」

をクリックし、イメージシーケンスのあるフォルダを開く。ここで注意すべき事は

1. イメージシーケンスの順番が正しいかどうか (時間ステップ通り並んでいるかどうか)
2. ムービーにしたいイメージファイル以外のファイルが混ざっていないかどうか (混ざっているとそれもムービーの 1 コマになってしまう可能性があるから)

である。そしてムービーにしたいイメージシーケンスの先頭のイメージを選択し、「開く (O)」ボタンをクリックする。すると「イメージシーケンスの設定」というウィンドが現れるのでフレームレートを選択する。「OK」ボタンを押すとムービーが作成され、画面に表示されるの

で再生を行い、問題が無ければ「ファイル」→「書き出し」をクリックし、名前を付けて保存する。なお図 8.12 の 4 枚の PNG ファイルを使って作成したムービー (*sample1fps.mov*; 1 フレーム / 秒) を付属 CD-ROM の *Chap8* ディレクトリの中の *data* ディレクトリの中に収録したのでムービーを作成する際の参考にして欲しい。

8.5 カラーの選択

前節まででデータを可視化するための基本的な手順について見てきた。本節では可視化する際に使用できるオリジナルのカラーテーブルの幾つかを紹介する。IDL にはもともと 41 個のカラーテーブル (0 - 40 番) が添付されている。しかしこれらでは十分対応できない場合がある。その場合には、下記に紹介するオリジナルのカラーテーブルを使用すればよい。

8.5.1 白黒の EPS ファイルを作成する場合

(使用するプロシージャ) :: *bw.pro*

白黒のカラーテーブルは IDL に添付されている (“B-W LINEAR”)。しかし、§8.4 で説明したメインプログラム *tvcn2dss_eps.pro* を使用したとき、カラースケールに選択した物理量の値が小さければ小さい程、その場所は黒くなり見にくい。カラースケールに選択した物理量の値が小さければ小さい程、その場所が白くなる方が見やすい。そのようなカラーテーブルが *bw.pro* である。図 8.13 は同じデータを、(a) IDL に添付されたカラーテーブル (“B-W LINEAR”), (b) *bw.pro* をロードして作成した画像である。

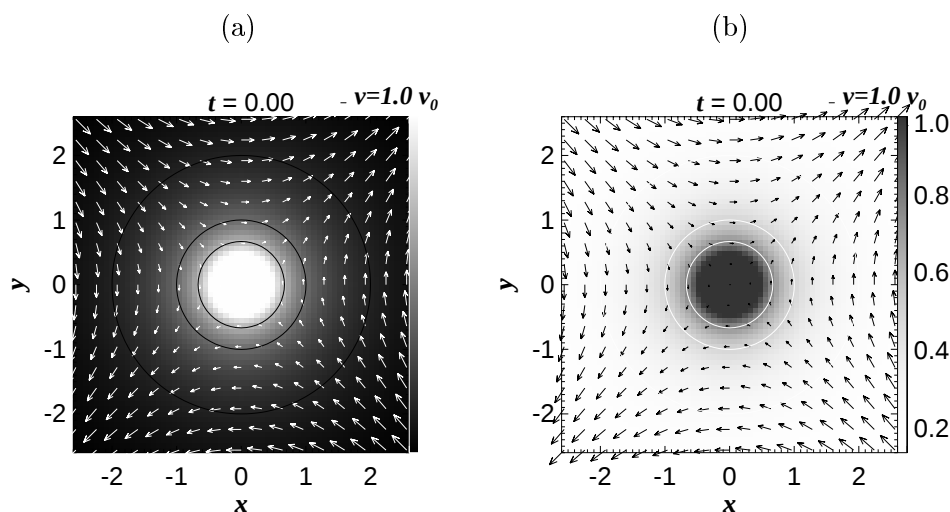


図 8.13: (a) IDL に添付されたカラーテーブル (“B-W LINEAR”) をロードし、メインプログラム *tvcn2dss_eps.pro* を使用して作成した画像。(b) カラーテーブル *bw.pro* をロードして同様に作成した画像。

カラーテーブル *bw.pro* のロードは下記の通りである。

(*bw.pro* のロード)

```
IDL> bw
% Compiled module: BW.
IDL>
```

8.5.2 カラーの EPS ファイルを作成する場合

(使用するプロシージャ) :: *rainbowf.pro*

IDL に添付されているカラーテーブルは §8.4 で説明した *tvcn2dss_eps.pro* を使用すると、カラースケールに選択した物理量の値が小さければ小さい程、その場所が濃い色になるものが多い。しかし、作成した EPS ファイルを L^AT_EX に挿入して白い紙に印刷すると、その濃い色の部分がかえって目立つこともある。そこで §8.4.1 で述べたようにカラースケールに選択した物理量の値が小さければ小さい程、白色に近づく方が見やすい。そのようなカラーテーブルが *rainbowf.pro* である。図 8.14 は同じデータを、(a) IDL に添付されたカラーテーブル (“Rainbow + black”)、(b) *rainbowf.pro* をロードして作成した画像である。

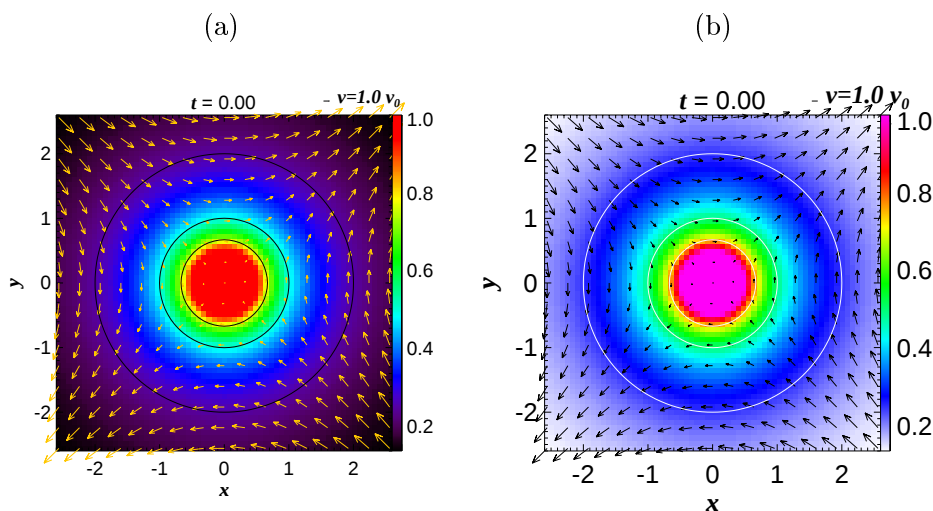


図 8.14: (a) IDL に添付されたカラーテーブル (“Rainbow + black”) をロードし、メインプログラム *tvcn2dss_eps.pro* を使用して作成した画像。 (b) カラーテーブル *bw.pro* をロードして同様に作成した画像。

カラーテーブル *rainbowf.pro* のロードは下記の通りである。

(*bw.pro* のロード)

```
IDL> rainbowf
% Compiled module: RAINBOWF.
IDL>
```

8.6 トラブルシューティング

本節では §8.2、8.3、そして 8.4 の操作を実行する際に発生するかも知れないトラブルとその対処方法について述べる。

8.6.1 “No such file or directory”

IDL のメインプログラム (ex. *read2dss.pro*) を実行した際に下記のようなエラーメッセージが表示されることがある。

(エラーメッセージ)

```
IDL> .r read2dss
% Error opening file. File: read2dss
  No such file or directory
IDL>
```

これはコマンドのパスが通っていない場合に表示されるので、§8.1.2で述べた方法にしたがってホームディレクトリ下の、*.cshrc* に下記のような *IDL_PATH* を設定する。

(IDL_PATH の設定)

```
r01{ochiys}1: cat ~/.cshrc
setenv IDL_PATH \+~/idl:\+${IDL_DIR}/lib:\+${IDL_DIR}/examples
```

これにより *~/idl*, *\${IDL_DIR}/lib*, *\${IDL_DIR}/example* の場所を検索する。これら以外の場所にコマンドがある場合にはその場所も指定しなければならない。追加した後も下記のように設定を適用しなければならないので注意。

(設定の適用)

```
r01{ochiys}2: source ~/.cshrc
```

8.6.2 “Corrupted f77 unformatted file detected.”

UNIX マシンで出力したバイナリデータを Linux マシンで読み込もうとすると、下記のようなエラーメッセージが表示されることがある。

(エラーメッセージ)

```
IDL> .r read2dss
% Compiled module: $MAIN$.
Input file name to read =
: tsample
% READU: Corrupted f77 unformatted file detected. Unit: 100, File: tsample.D
% Execution halted at: $MAIN$      45 /home/aplab/yasuhiro/idl/read2dss.pro
IDL>
```

この原因は UNIX と Linux ではデータの形式が異なるからである。解決策はメインプログラム *read2dss.pro* の 45 行目に「/SWAP_IF_LITTLE_ENDIAN」を追加する。

(*read2dss.pro* 45 行目)

```
OPENR, Unit2, fn+'.D', /F77_UNFORMATTED, /GET_LUN, /SWAP_IF_LITTLE_ENDIAN
```

8.7 ソースファイル

8.7.1 サンプルデータ用のソースファイル

本節では §8.2, 8.3, 8.5 で使用したバイナリデータを作成したソースプログラムについて補足する。ソースプログラム (*sample2d.f*) は *rho*, *vx*, *vy*, *bx*, *by* という 2 次元配列を定義して、値を代入し、それをヘッダーファイル (*sample.d*; 配列の大きさなどの情報) とバイナリデータ (*sample.D*; 配列の値の情報) を出力するための簡単なプログラムである。

(*sample2d.f*)

```
IMPLICIT REAL*8(A-H,O-Z)
integer, parameter :: nmin=0, nmax=2**6
real rho(nmin:nmax,nmin:nmax),
$   vx(nmin:nmax,nmin:nmax),
$   vy(nmin:nmax,nmin:nmax),
$   bx(nmin:nmax,nmin:nmax),
$   by(nmin:nmax,nmin:nmax),
REAL x(nmin:nmax),y(nmin:nmax)
real :: tstep
integer i,j,ixs,ixe,iys,iye
tstep=0.
ixs=nmin
ixe=nmax
iys=nmin
iye=nmax
C write header file "sample.d"
```

```

        OPEN(2,FILE="sample.d")
        WRITE(2,100) "sample data"
100  FORMAT(A11)
        WRITE(2,101) ixS,ixe,iys,iye
101  FORMAT(OP,4I5)
        WRITE(2,105)
105  FORMAT('*****')
        WRITE(2,104)
104  FORMAT(2X,'ixS',2X,'ixe',2X,'iys',2X,'iye')
        WRITE(2,105)
        WRITE(2,102) "sample.D"
        WRITE(2,106) TSTEP
        WRITE(2,107)
102  FORMAT('FILENAME=',A10)
106  FORMAT('TIME=',F15.7)
107  FORMAT('FLOAT ')
        CLOSE(UNIT=2)
C initial condition
        do i=ixS,ixe
            x(i)=0.08*(i-32)
        end do
        do j=iys,iye
            y(j)=0.08*(j-32)
        end do
        DO j=iys,iye
            DO i=ixS,ixe
                RI=MAX(sqrt(x(i)**2+y(j)**2),1.D-8)
                rho(i,j)=0.5/RI
                vx(i,j)=5.*y(j)
                vy(i,j)=5*x(i)
                bx(i,j)=x(i)/sqrt(x(i)**2+y(j)**2)
                by(i,j)=y(j)/sqrt(x(i)**2+y(j)**2)
            enddo
        enddo
C write binary "sample.D" in single precision
        OPEN(12,FILE="sample.D",FORM='UNFORMATTED')
        write(12) x
        write(12) y
        write(12) rho
        write(12) vx
        write(12) vy

```

```

write(12) bx
write(12) by
CLOSE(UNIT=12)
END

```

この *sample2d.f* では、あらかじめ配列の大きさ (e.g. *ixs*, *ixe*, *iys*, *iye*) や物理量 *rho*, *vx*, *vy*, *bx*, *by* を与えているが、実際には利用者が使用する配列の大きさや、シミュレーションを行って得られた物理量のデータを使用すればよい。

上記のプログラム *sample2d.f* をコンパイル、実行するとヘッダファイル (ファイル名 *sample.d*) とバイナリデータ (ファイル名 *sample.D*) が生成される。

(コンパイル、実行)

```

k15{ochiys}1: frt sample2d.f
k15{ochiys}2: ./a.out

```

ファイル *sample.d* は配列の大きさやバイナリデータが書き込まれたファイルの名前、時間ステップ、データ型などの情報を持ったヘッダーファイル (テキスト形式) である。 *sample.D* は配列の値の情報を持つバイナリデータである。 *sample.d* の中は、下記のようにになっている。

(*sample.d*)

```

k15{ochiys}3: cat sample.d
sample data
  0  64  0  64
*****
  ixs ixe iys iye
*****
FILENAME=  sample.D
TIME=      0.0000000
FLOAT
k15{ochiys}4:

```

データの書き出し方について、上記プログラム *sample2d.f* のようにヘッダファイル (*sample.d*) とバイナリデータ (*sample.D*) の両方を書き出しておく、§8.2 で扱った *read2dss.pro* をそのまま使って IDL にデータを読み込ませることができる。また *read2dss.pro* に配列の大きさやバイナリデータが書き込まれたファイルの名前、時間ステップ、データ型などをあらかじめ書き込んでおけば、ヘッダーファイルを作成する必要はなくなり、バイナリデータだけを読み込めばよい。どの方法

を用いるかは利用者の目的に応じて選んでもらいたい。

参考文献

- [1] Research Systems 1999, IDL Reference Guide A-M (version 5.3: Research Systems, Inc.)
- [2] Research Systems 1999, IDL Reference Guide N-Z (version 5.3: Research Systems, Inc.)
- [3] Research Systems 1999, IDL Reference Guide (Objects & Appendixes) (version 5.3: Research Systems, Inc.)
- [4] Research Systems 2003, IDL Online Guide (version 6.0: Research Systems, Inc.)
- [5] アダムネット株式会社 1998, Introduction to IDL (version 5.1: アダムネット株式会社)

第 9 章

AVS を使った可視化

千葉大学 錦織 弘充

本章では、3 次元の数値データを可視化し、視覚的に理解できる形にする方法について説明する。3 次元データの可視化ソフトウェアとして、天文分野では現在「AVS/Express (Application Visualization System)」がよく利用されている。しかし、解説書の類がほとんど無かったり、付属のマニュアルだけでは解説が不十分であったりという理由から、初心者がその使い方に慣れるまでに時間がかかってしまうのが現状である。(著者自信も、研究室の経験者、使い方講習会でアドバイスをもらいながらトライ & エラーを繰り返し、徐々に覚えたことを思い出す。)

本章は、初心者が理解しやすいよう、チュートリアル形式で書かれている。ユーザ自身が付属 CD-ROM に集録されている計算データを使って、実際に可視化実習を行いながら、最終的には論文等に掲載できる程度の画像、動画を出力するところまで解説してある。

§9.1で AVS/Expressの特徴、簡単な使い方の紹介する。§9.2では 1 タイムステップの、§9.3では netCDF 形式で出力された計算データを、チュートリアル方式で解説する。§9.4は、動画の作り方など、応用の内容を盛り込んだ。

また、データファイル、field データファイル (*.fld ファイル)、モジュールネットワーク (*.v ファイル)(それぞれ後述) など実習に必要なファイル類は、全て付属 CD-ROM/Chap9(約 300MB) に用意したので是非利用して頂きたい。

なお、2 次元データの可視化については、第 8 章、“IDL を使った可視化”を参照されたい。

9.1 可視化ソフトウェア AVS とは

本節では、可視化ソフトの中でも、天文分野でよく使われている「AVS/Express」の特徴、使用法の概要等について述べる。

9.1.1 AVS/Express の特徴

「AVS/Express」の特徴は、読み込みや画像の処理、設定等を、固有の機能を持った“モジュール”と呼ばれるパーツを組み合わせることで、コマンドライン等によるプログラミングをすること無く、視覚的に行うことが出来る点である。動作環境も、Unix 版、Linux 版をはじめ Windows 版や Mac 版も用意されている。ちなみに今回は、以下の環境の下で行った。

CPU: Intel Pentium4 3.2GHz
OS: Red Hat Professional Workstation
RAM: 2.0Gbyte
AVS: AVS/Express ver. 6.3 Linux 用
インストール先: /usr/local/express/bin/linux

9.1.2 AVS の画面

作業を始める前に、パッケージを任意のディレクトリにコピーする (約 300MB あるので、quota 等に注意して頂きたい)。今後の作業は、このディレクトリ内で行うこととする。下はユーザのホームディレクトリにコピーし作業している例である。そこで AVS/Express を起動する。

```
% cd  
% cp -r [付属 CD-ROM]/Chap9/ .  
% cd Chap9  
% vxp
```

これで起動しない場合、パスの設定が通っていない可能性がある。以下の設定を行い、その後再ログインすることでパスが通る。

```
% PATH=$PATH:/usr/local/express/bin/linux/  
% source
```

まず AVS を起動すると、3つのウィンドウが現われる。(図 9.1)
今回はより使いやすいように、以下の手順を行う。

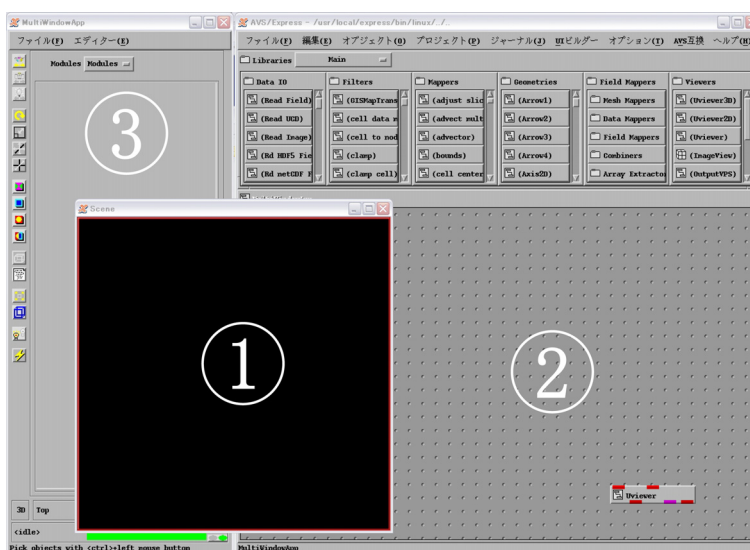


図 9.1: 起動画面

2 のウィンドウで

[ファイル] → [アプリケーションの削除]

→ 2 つウィンドウが消える

[ファイル] → [新しいアプリケーション]

[アプリケーション・タイプの選択] 画面

[アプリケーション・タイプ] → シングルウィンドウ・ビュー

[ビューワー・タイプ] → 3D

→ 一つウィンドウが現われ、合計 2 つのウィンドウが表示される。

次にそれぞれのウィンドウの説明を行う。

ネットワークエディタ

実際にモジュールネットワークを作るウィンドウがこの「ネットワークエディタ」(図 9.2)であり、AVS/Express のメインウィンドウである。上半分にあるモジュール群から、行いたい処理に適切なモジュールを下半分のワークスペースと呼ばれる領域にドラッグ & ドロップ (今後、この処理を“インスタンス”すると表記する。) することで、視覚的にプログラミングを進めて行くことが出来る。

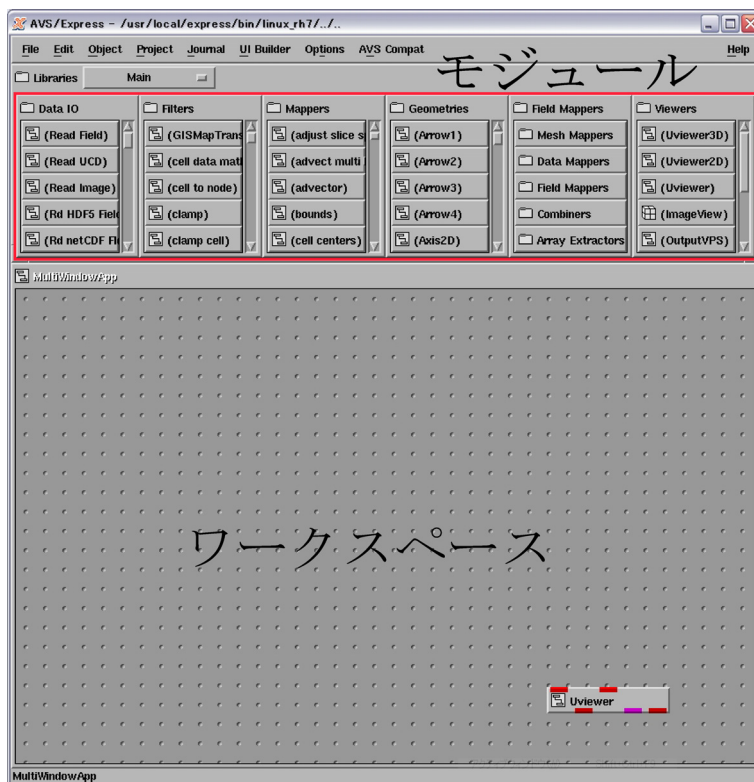


図 9.2: ネットワークエディタ

コントロールパネル／ビューーウィンドウ

9.1.2で行った処理によって図 9.1の 1、3 のウィンドウが一つにまとまったウィンドウが表示される (図 9.3)。図 9.3の青い網掛け部分をビューーウィンドウといい、可視化された画像はここに出力される。また、黄色い網掛け部分をコントロールパネルとよび、ビューーウィンドウの操作やインスタンスしたモジュールの設定を行う。ツールバーの使い方は、マニュアル等を参照されたい。

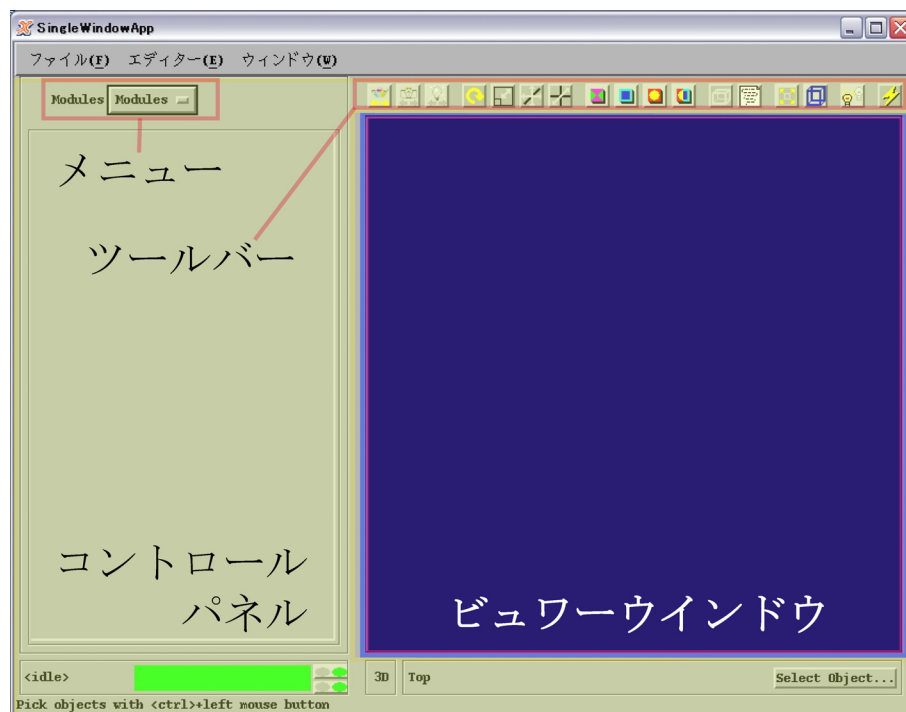


図 9.3: シングルウィンドウビュー

9.1.3 モジュール

可視化は、機能別にいくつも用意されているモジュールを組み合わせ、一つの“モジュールネットワーク”を作って行う。モジュール (図 9.4) の使い方の原則は、「同じ色の出力ポートと入力ポートを接続するという作業を繰り返して大きなモジュールネットワークを作る」である。

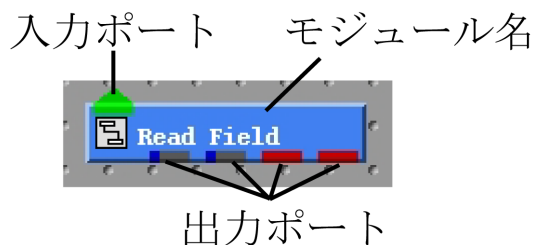


図 9.4: モジュール詳細

9.1.4 モジュールネットワークファイルの読み込み・保存

構築したモジュールネットワークは “.v” という拡張子のファイルとして保存することができる。
メインウィンドウのメニューから

[ファイル] → [アプリケーション保存]

を選択し、パス、ファイル名を指定した後、‘了解’を押せば保存される。
また、読み込みは、同じくメインウィンドウのメニューから

[ファイル] → [アプリケーションの読み込み]

を選択し、目当てのモジュールネットワークファイルを指定することで読み込める。
不要になったモジュールネットワークは

[ファイル] → [アプリケーション消去]

で消すことが出来る。

9.1.5 AVS/Express の終了

メインウィンドウのメニューから

[ファイル] → [終了]

を選択する。

9.1.6 付属 CD-ROM の集録内容

```
% ls -R
.:
Data.dir/  avs.n1.v  avs.t3b1.v  avs.t4a.v  avs.t5a2.v  sample001000.fld
avs.a1.v  avs.t1a.v  avs.t3b2.v  avs.t4b1.v  avs.t5b.v   sample001100.fld
avs.a2a.v  avs.t2a.v  avs.t3b3.v  avs.t4b2.v  avs.t7.v    sample1.fld
avs.a2b.v  avs.t3a.v  avs.t3c.v  avs.t5a1.v  sample000900.fld sample2.fld

./Data.dir:
1000/  1100/  900/  out.cdf*

./Data.dir/1000:
bx  by  bz  coord1  coord2  coord3  pr  ro  vx  vy  vz

./Data.dir/1100:
bx  by  bz  coord1  coord2  coord3  pr  ro  vx  vy  vz

./Data.dir/900:
bx  by  bz  coord1  coord2  coord3  pr  ro  vx  vy  vz
```

付属 CD-ROM には、各節で使うモジュールネットワークファイル (*.v)、field データファイル (*.fld)、§9.2、§9.4で使う、タイムステップごとの計算データ (Data.dir/900, 1000, 1100)、§9.3、§9.4で使う、netCDF 形式で出力された計算データ (Data.dir/out.cdf) を集録した。

9.2 1 time step の 3 次元データの可視化

本節では、AVS/Expressの操作法を 1 タイムステップの 3 次元データを用い、チュートリアル方式で説明する。field データファイルの作成、モジュールをインスタンスする所から始まり、スカラー、ベクトルの可視化、最終的には画像の出力までを行う。

なお、この先は 各自の環境にコピーされた、パッケージの 'Chap9' ディレクトリに入って作業していることを前提とし、使用しているモジュールネットワークファイル内のファイル名指定欄は 全て相対パスで記述してある (§9.1.2参照)。必要に応じて絶対パスに書き換えて欲しい。

9.2.1 field データファイルの作成

まず、「あるタイムステップの 3 次元データ」を可視化する場合について。最初に、AVS/Express で読み込めるようにしたデータについて、以下の点を確認しておく。

- ・各成分のグリッド数 (例: 直交座標系、 $121 \times 121 \times 121$)
- ・データの形式 (ascii/binary) (例: 単精度 binary)
- ・データの数及び内容、順番 (例: 8 個、密度 (ro), 速度 (V_x, V_y, V_z), 磁場 (b_x, b_y, b_z), 圧力 (pr))
- ・読み込みデータのある場所 (例: [付属 CD-ROM]/Chap9/)

データを読み込むためには、その情報を記した「field データファイル」¹と呼ばれるファイルを作る必要があり、その拡張子は“.fld”としなければならない。“sample1.fld”ファイルを用いて説明していく²。

```
# AVS field file
#
# This is the sample field file for demonstration of AVS manual.
# Written by NISHIKORI, Hiromitsu.

ndim=3
dim1=121
dim2=121
dim3=121
nspace=3
veclen=8
data=float
field=rectilinear
label=ro, vx, vy, vz, bx, by, bz, pr

variable 1 file=./Data.dir/900/ro filetype=unformatted skip=0 stride=1
variable 2 file=./Data.dir/900/vx filetype=unformatted skip=0 stride=1
variable 3 file=./Data.dir/900/vy filetype=unformatted skip=0 stride=1
variable 4 file=./Data.dir/900/vz filetype=unformatted skip=0 stride=1
variable 5 file=./Data.dir/900/bx filetype=unformatted skip=0 stride=1
variable 6 file=./Data.dir/900/by filetype=unformatted skip=0 stride=1
variable 7 file=./Data.dir/900/bz filetype=unformatted skip=0 stride=1
variable 8 file=./Data.dir/900/pr filetype=unformatted skip=0 stride=1
coord 1 file=./Data.dir/900/coord1 filetype=ascii skip=0 stride=1
coord 2 file=./Data.dir/900/coord2 filetype=ascii skip=0 stride=1
coord 3 file=./Data.dir/900/coord3 filetype=ascii skip=0 stride=1
```

¹field データファイルの記述の詳細については、マニュアル参照のこと。

²この章で使用している計算結果のデータについては、Nishikori et al. (2002) および Nishikori (2003) 参照のこと。

#AVS field データファイルの最初にはこの文字列が必要。このあとにコメントなど、任意の文字を書くことが出来る。また、2 行目以降で、# で始まる行はコメントアウトされているとみなされる。

ndim 読み込むデータの次元数。例では 3 次元データを扱っているので “ndim=3” と記述している。

dim1, dim2, dim3 各成分のグリッド数を記述する。例では、dim1、dim2、dim3 はそれぞれ 121、121、121 となる。

nspac データがある空間の次元数を定義する。例では “nspac=3”。

veclen 各グリッドに存在するデータの個数を定義。例では各格子点に 8 個ずつデータがあるので “veclen=8” と記述する。

data データ成分の型。ここでは “data=float” と指定する。

field 使用している座標について定義する。例では、直交不等間隔格子として定義する、“rectilinear” を使う。詳細はマニュアルで確認のこと。他に、間隔 1 の直交等間隔格子を表す “uniform”、ばらばらの格子点の座標を定義する “irregular” がある。

label 入力したデータにラベルをつける。読み込んだ順番にカンマもしくはスペースで区切り記述する。

variable n file= 文字 filetype= 文字 skip= 数字 stride= 数字

n データを区別する値で 1 から始まる整数値を記述する。例では 8 個のデータを読み込むので 1 から 8 まで。

file 読み込むファイル名を指定する。例では n=1~8 に、ro、vx、vy、vz、bx、by、bz、pr のファイルを読み込んでいる。絶対パス、相対パスどちらでも良い。例は相対パスで記述している。

filetype 読み込むファイルタイプを指定する。例は FORTRAN の UNFORMATTED で出力したデータであるので、“filetype=unformatted” と記述。他に、“ascii”(ascii 形式)、“binary”(C のプログラムでの出力形式) が選択出来る。

skip 各データを読み込む際、最初に読み飛ばすデータ量を指定する。filetype が ascii の場合には最初に読み飛ばす行数を、binary、unformatted の場合は最初に読み飛ばすバイト数を記述する。例で使用しているデータは「単精度バイナリ」であるので、“skip=0” は、各データに付いて最初に 0 バイトずつ飛ばす、という意味である。(0 のときは書かなくても良い。)

stride データをどれだけごとに読み込むかを指定。ascii の場合は読み込む項数を、binary、unformatted の場合は読み込む要素数を表す。例のように “stride=1” と記述すると

$$(1 \text{ データサイズ}) \times 1[\text{byte}] = 4\text{byte} \times 1 = 4[\text{byte}]$$

ごとにデータを込むことを意味する。

coord n file= 文字 filetype= 文字 skip= 数字 stride= 数字 coord で始まった行は座標情報を表しており、“nspace” で記述した数だけ必要である。n の値で座標軸を区別し、n=1、2、3 は直交座標系の場合、それぞれ x、y、z 軸についての情報であるということを意味する。例では x 軸に関する情報を coord1 から、y 軸に関する情報を coord2 から、z 軸に関する情報を coord3 から読み込むことを意味している。

9.2.2 表示領域の設定

ここから、実際に可視化をするという作業に入る。本文や‘可視化結果’で示されたネットワーク図等を見ながらモジュールネットワークを組んでほしい。また、各節で指定された‘使用ネットワーク’は、その節で組むモジュールネットワークの完成形である。最初に §9.1.4 の手順で読み込むことで、1 からモジュールネットワークを組む手間を省くことができる。なお、今後モジュール名を「 」で、各モジュールのコントロールパネル等での設定の記述を‘ ’で記す。

インスタンスするモジュールは少ない方がハードウェアに負担がかからず、可視化速度が速い。不必要なモジュールはなるべくインスタンスしないようにするとよい。

field データファイルの読み込み -Read Field

使用モジュール	: Read Field、Uviewer3D
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.tla.v

まず、9.2.1 で作った field データファイルを読み込む。field データファイルを読み込む機能をもったモジュールが「Read Field」である。

[Main] → [Data IO] → Read Field

にあるので、ワークスペースにインスタンスする³。すると、コントロールパネルのメニューが Modules[Read Field] となり、Read Field モジュールが選択されていることが確認できる。その下のメインコントロールパネル部で、field データファイルを指定する⁴。

```
% ./sample1.fld
```

例では、current directory にある ‘sample1.fld’ ファイルを読み込んでいる。エラーを示すウインド

³モジュールの場所が分からないときは、場所を検索することが出来る。[オブジェクト] → [全てのライブラリを検索] とするとオブジェクト検索ウインドウが現われる。検索パターン欄に探しているモジュール名を入れ Return(Enter) キーをおすと、オブジェクト欄に検索結果が現われる。目当てのモジュールを選択し、‘表示’ ボタンを押すと該当するモジュールが青く表示され、場所が分かる。

⁴AVS/Express は、値やファイル名を変更した後、その場で Return(Enter) キーを押さなければ変更が認識されないの
で注意して欲しい。

ウが現われなければ field データファイルの読み込み、及びデータの読み込みは完了である。

境界の表示 -bounds / 座標軸の表示 -Axis3D

使用モジュール	: bounds、Axis3D
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t2a.v
可視化結果	: 図 9.5

次に可視化領域の境界、及び coord 1、2、3 で読み込んだ座標データを可視化する。まず、

[Main] → [Mappers] → bounds

をインスタンスする。次に「Read Field」の青出力ポートと「bounds」の青入力ポートを、「bounds」の赤出力ポートと「Uviewer3D」の赤入力ポートを接続⁵する。するとビューウインドウに白い枠が表示される。同様にして

[Main] → [Geometries] → Axis3D

をインスタンスし、「bounds」と同様に接続する (図 9.5)。ここでビューウインドウ上でマウスをドラッグすると、可視化された画像の視点を変えることが出来る。「Read Field」モジュールのコントロールパネルを表示させたのと同じ要領で、コントロールパネル部に「Axis3D」の設定画面を表示させる (メニューから「Axis3D」を選択)。ここで各軸の名前や値の範囲、主線・副線の間隔や線種を変更することが出来る。ここでは、見やすいように、「X step」、「Y step」、「Z step」を 10 に設定しておく。

9.2.3 スカラーデータの可視化

ここでは密度などのスカラーデータの可視化を行う。

等値面の表示 -isosurface

使用モジュール	: isosurface、crop
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t3a.v
可視化結果	: 図 9.6

3 次元空間で、あるデータの等数値面を表示するモジュールが「isosurface」である。これは密度等、スカラー量の可視化を行う場合に有効である。ここでは、表示する領域を指定する、「crop」モジュールとともに説明する。まず、「isosurface」、「crop」をインスタンスする。

[Main] → [Filters] → crop

[Main] → [Mappers] → isosurface

「isosurface」は、あるスカラー量の 3 次元空間における等値面を可視化するモジュールである。コ

⁵マウスカーソルを接続したいモジュールの出力ポートに置き、マウス左ボタンを押したままカーソルを接続先のモジュールの入力ポートの方へ動かす。目的のモジュール同士を結ぶ細い白線が表示されたら、マウス左ボタンを離す。

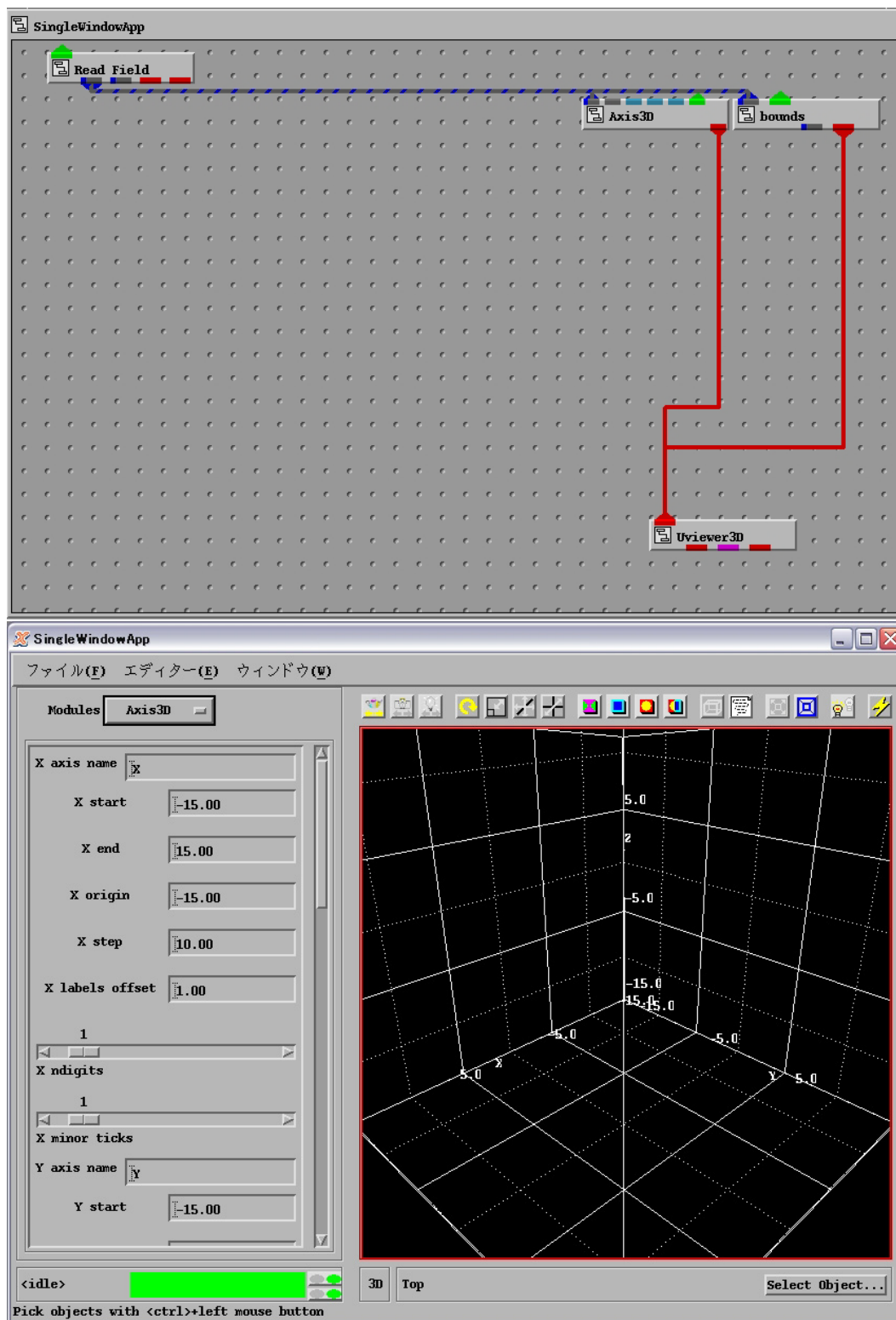


図 9.5: 境界、座標軸の表示

ントロールパネルの設定画面の 'iso component' で、表示するパラメータを選択する。また、'iso level' を動かすことで、どの値の等値面を可視化するか指定する。つまり、例は「isolevel=0.04 の等密度面」を可視化していることになる。

また、「crop」はデータの一部領域を表示する場合に有効なモジュールである。設定画面では、表示領域を指定することができる。I、J、K それぞれをスライダを動かすことによって指定することが出来る。例では「I min=0, I max=60, J min=0, J max=120, K min=0, K max=120」と指定して出力した例である。切り口を見てみるとよく分かるが、等密度面を可視化しているため、中が空洞になっている。

等値ボリュームの表示 -isovolume

使用モジュール	: isovolume
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t3b1.v
可視化結果	: 図 9.7

ここでは「isovolume」モジュールを使い、等値ボリュームの可視化を行う。入力データ成分に対して計算を行う「data math」モジュール、データの最大値と最小値を指定する「set minmax」モジュールも付せて説明する。

[Main] → [Filters] → crop
 [Main] → [Mappers] → isovolume

まず、「crop」→「isovolume」→「Uviewer3D」と接続されている、avs.t3b1.vを読み込む。前項で紹介した、「isosurface」が“ある値の面”を可視化していたのに対し、「isovolume」は等値面の内側のボリュームを可視化するモジュールである。「isovolume」のコントロールパネルでは、どの物理量を (iso component)、どの等値面の内側について (iso level) 可視化するかを指定する⁶。例では、“密度について、0.04 より小さい値について”可視化している。

データの最大値、最小値の設定 -set minmax

使用モジュール	: isovolume、set minmax
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t3b2.v
可視化結果	: 図 9.8

[Main] → [Filters] → set minmax

「crop」→「isovolume」→「set minmax」→「Uviewer3D」と接続されている、avs.t3b2.vを読み込む。「set minmax」は、カラーコントラの最小値、最大値を指定するモジュールである。このモジュールを使用しない場合、「iso volume」等で使われるカラーコントラは入力されたデータ領域

⁶コントロールパネルの 'Above' を解除すると、等値面の外側を可視化する。一般に、ボリュームを可視化 (ボリュームレンダリング) するとき、ハードウェアに大きな負担をかける。iso level が内側を可視化するのに適した値であっても、外側についても適しているとは限らない。値によってはハードウェアに大きな負担をかけることとなり、可視化に非常に時間がかかったり、場合によっては AVS/Express のソフトが異常終了してしまう事になってしまう。iso level の指定は、内側の可視化時は低めの、外側の可視化時は高めの値から始めるとよい。

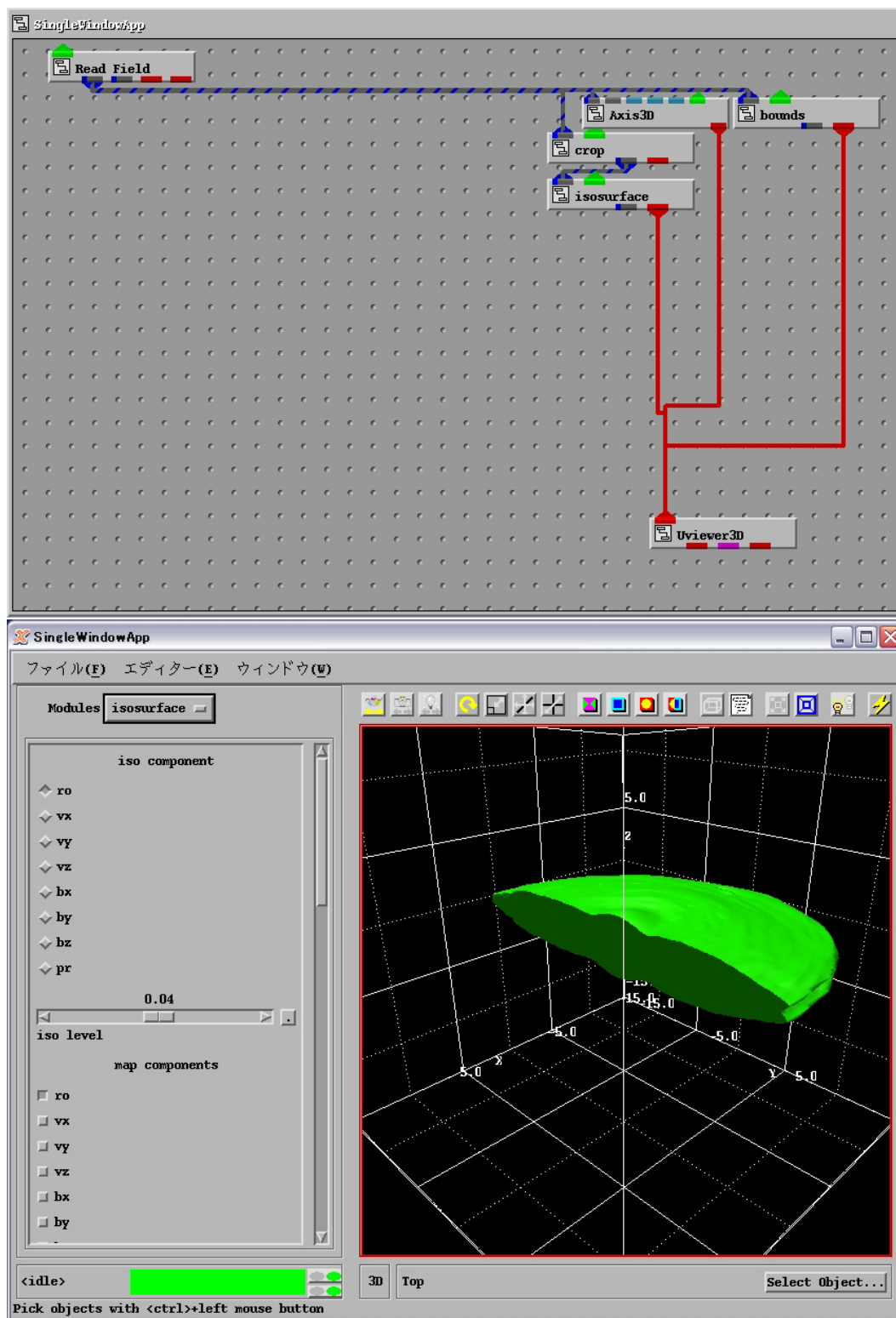


図 9.6: 等数値面の表示

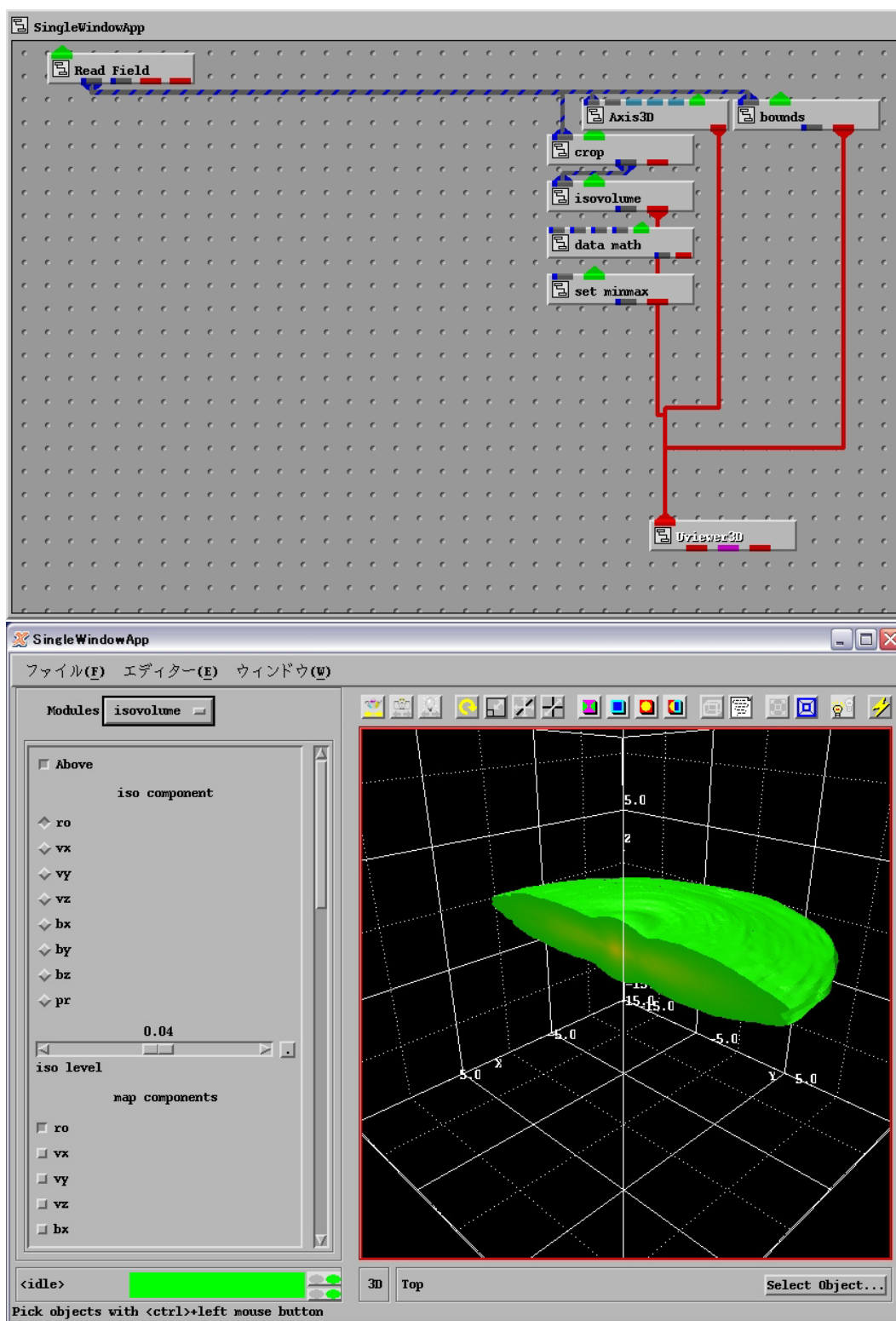


図 9.7: 等値ボリュームの表示

によって、その都度、自動的に作成される。このため、カラーコントラを、時系列でもしくは様々なデータ間で統一することが出来ない。「set minmax」モジュールを使用することで、入力データによって最大値、最小値が変化した場合でも、同じカラーコントラを利用することが出来るようになる。コントロールパネルで最小値 (min value)、最大値 (max value) を設定する。例は、“入力データ ‘ro’ について、 $0.04 < ro < 0.4$ の範囲” でカラーマップを作成している。このとき、「isovolume」の iso level で設定した値に最小値⁷を合わせておくと、可視化された領域全てを「set minmax」で設定した範囲に入れることができる。

数式演算処理を行う -data math

使用モジュール	: isovolume、set minmax、data math
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t3b3.v
可視化結果	: 図 9.9

[Main] → [Filters] → data math

次に「crop」→「isovolume」→「data math」→「set minmax」→「Uviewer3D」と接続されている、avs.t3b3.vを読み込む。「data math」は、4つの青い入力ポートからそれぞれ違うデータを読み込み、算術演算するモジュールである。コントロールパネルで data type、行う演算式を入力する。このとき、入力データは左のポートから入って来た順に「#1, #2, #3, #4」と表す。例の場合、

$$[\text{出力データ}] = \log_{10} \#1 \quad (9.1)$$

の計算を行い、値を常用対数で表すように演算している⁸。ビューワーウィンドウには、「data math」を通さなかった場合よりも、密度差がよく分かる図が可視化されているはずである。また、「set minmax」の min value、max value も、指数部分を指定するように変わっていることも確認して欲しい。

格子に沿った断面の密度コントラの表示 -orthoslice

使用モジュール	: orthoslice
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t3c.v
可視化結果	: 図 9.10

本節では、「orthoslice」モジュールを使って、格子に沿ってある断面を切り出し、その面のスカラー量変化を可視化する。

[Main] → [Mappers] → orthoslice

コントロールパネルの axis で、どの軸に垂直な面について切り出すかを、plane でその位置を指定する。例は axis=2、つまり z 軸について垂直で、plane=60 の面を切り出して可視化している。こ

⁷ Above を off にした場合は最大値。

⁸ 計算式を記述する場合、データや数字、計算部分 (#1 * #2 等) は必ず括弧 () で囲うようにする。これを怠ると、誤って計算されてしまう可能性がある。試しに、例を括弧を外した場合 ($\log_{10} \#1$) と比べてみて欲しい。

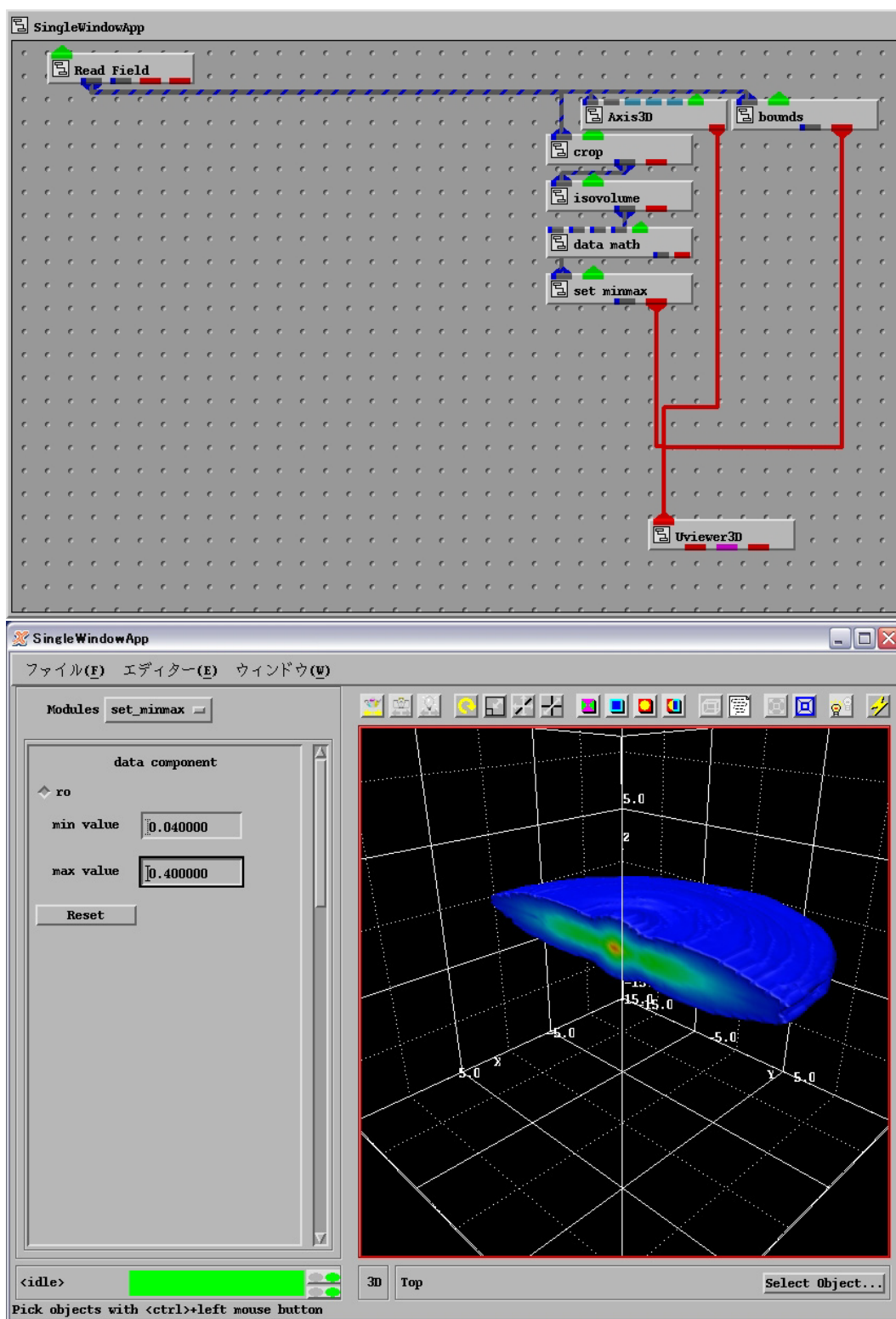


図 9.8: データの最大値、最小値の設定

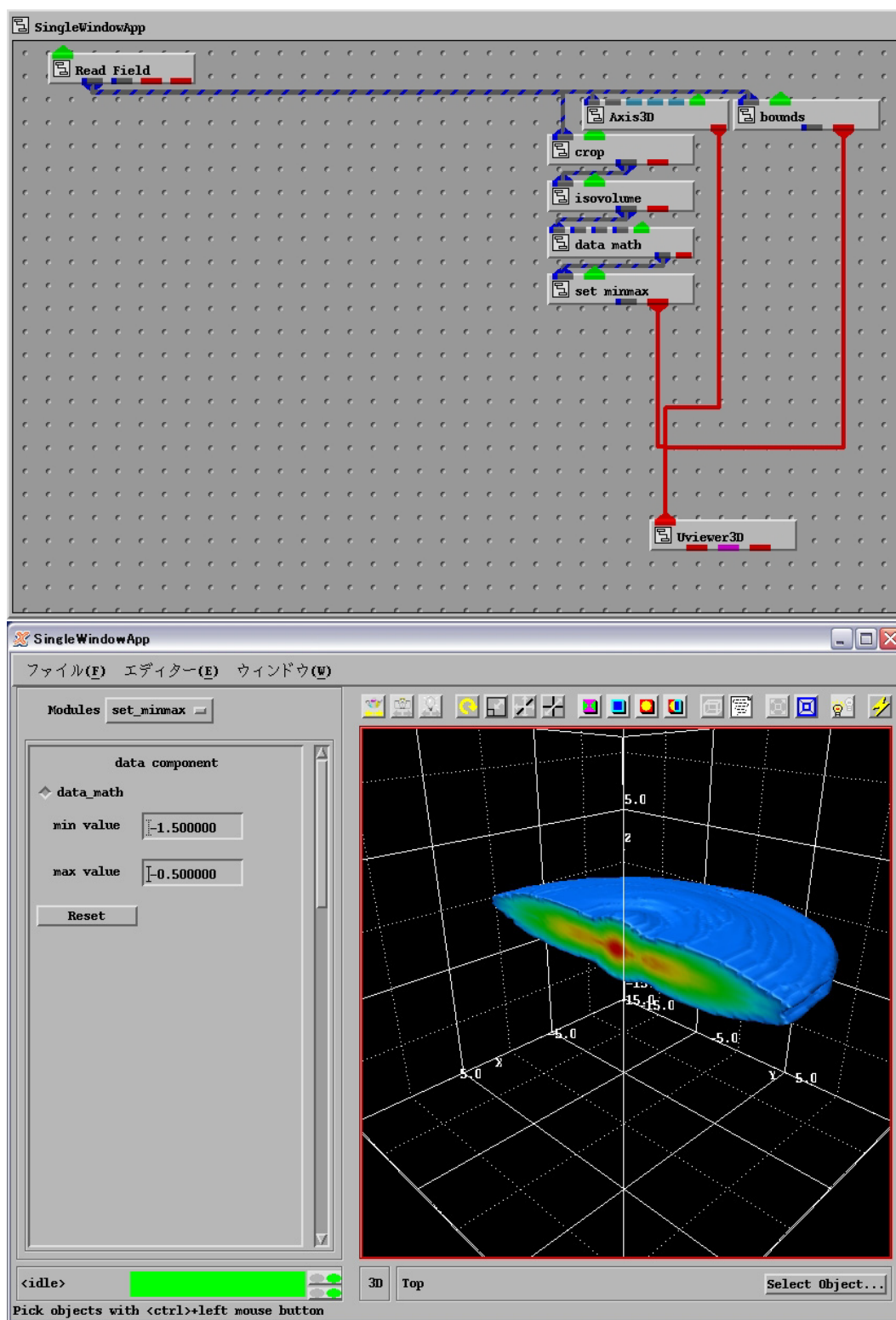


図 9.9: 数式演算処理を行う

のとき、「set minmax」の最大値最小値を前々節で設定した値にし、カラーコントラストをあわせておく。

同じモジュールを複数インスタンスすると、2つ目、3つ目…のモジュールには自動的に「(モジュール名)#1」、「(モジュール名)#2」…と、名前が付けられる。使用ネットワーク‘avs.t3c.v’では、「data math」、「set minmax」に関して、この決まりが適用されている。

9.2.4 ベクトルデータの可視化

流線の表示 -streamlines

使用モジュール	: combine vect、streamlines、FPlane
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t4a.v
可視化結果	: 図 9.11、図 9.12

ここでは、磁力線や速度ベクトルなど、流線の可視化を行う。

[Main] → [Filters] → combine vect
 [Main] → [Geometries] → FPlane
 [Main] → [Mappers] → streamlines × 2

可視化の流れとしては、

1. 入力された複数のスカラーデータからベクトルデータを作成 (「combine vect」)
2. 流線の始点の設定 (「FPlane」)
3. 流線の可視化 (「stream lines」)

となる。

「**combine vect**」 まず「combine vect」モジュールを用いて、ベクトルデータを作成する。コントロールパネルの‘veclen’でベクトルにする成分数を、‘vector component’で用いるスカラーデータを指定する。例は“bx, by, bz の3成分を用いてベクトルデータをつくる”設定である⁹。

「**FPlane**」 「FPlane」は、流線の始点を置く面を指定するモジュールである。コントロールパネルで‘Transformation Editor’のラジオボタンを押すと、別に始点面の設定画面が現われる。ここで、「FPlane」の赤出力と「3DViewer」の赤入力ポートを接続すると、現在指定されている始点面が白く表示され、設定がしやすい。この状態で設定変更を行う。

‘X Rotation’、‘Y Rotation’、‘Z Rotation’は、それぞれどの向きに何度傾けるかを、‘Scale’は、指定した面の格子の大きさに対する割合を、‘X Tran’、‘Y Tran’、‘Z Tran’は、各成分について、どのくらい位置をずらすかを指定する。例では、‘z=0.5にある、大きさが格子の大きさの1倍の面’を指定している¹⁰。

⁹ラジオボタンで‘vector components’を選択する際、一時的に‘veclen’で指定した数より選択した数のほうが少なくなってしまう。このときエラーウインドウが現われるが、気にせずに veclen 数と同じ数の vector components を選択し、最後にエラーウインドウを閉じるようにするとよい。

¹⁰「FPlane」の代わりに「crop」を用いることができる。この場合も、「crop」の赤出力と「3DViewer」の赤入力ポートを接続して設定すると便利である。

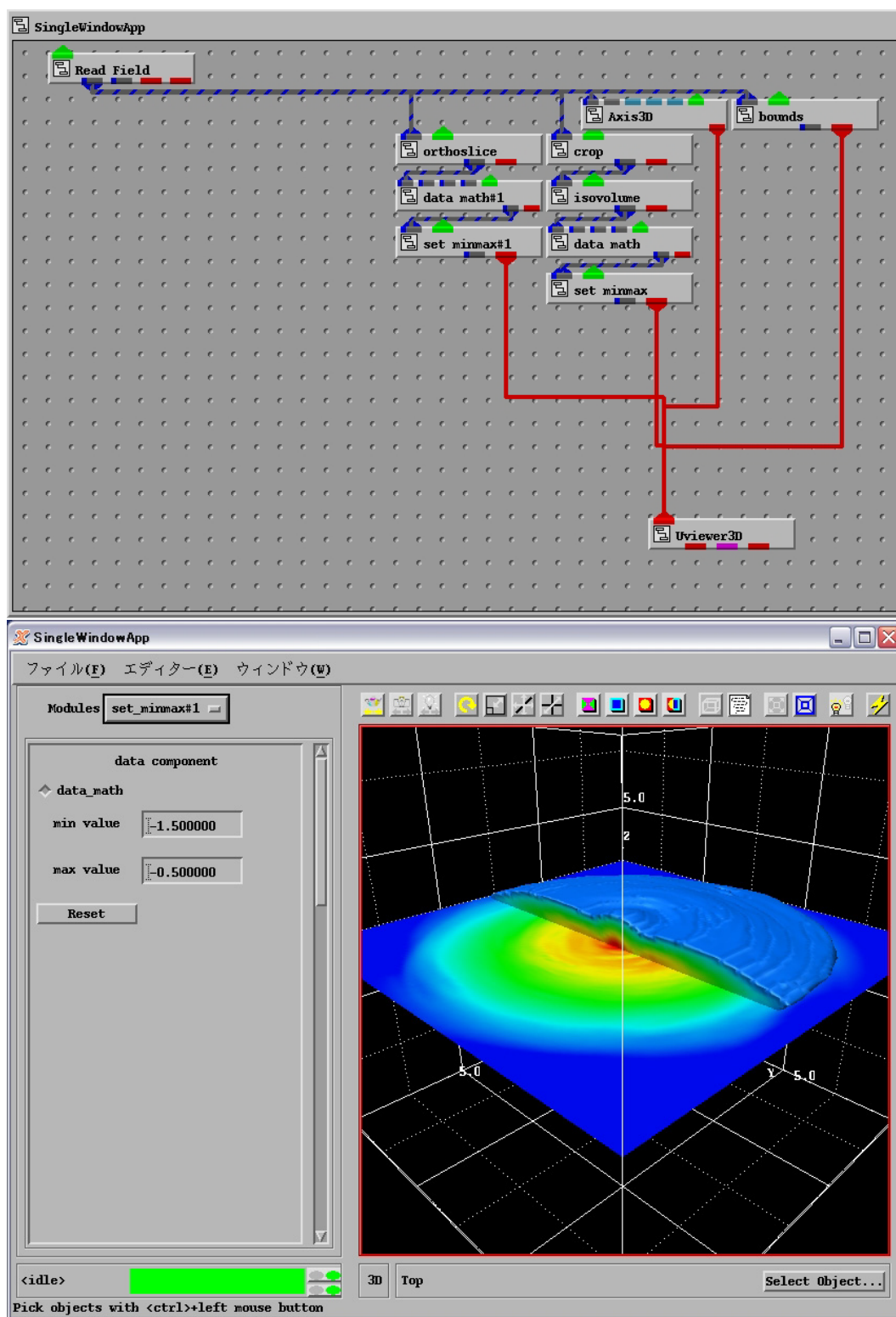


図 9.10: 格子に沿った断面の密度コントアの表示

コントロールパネルにもどる。‘x-dimension’、‘y-dimension’は‘Transformation Editor’で設定した面を x、y 方向に何分割するかを指定するスライダである。つまり、流線の始点の数を、それぞれの方向について決める値で、これが大きい程格子点の数も多くなり、結果たくさんの流線が表示される。流線が多すぎてもみづらく、また可視化時にハードウェアに負担もかける。例の様な場合では、それぞれ 8 程度が適当であろう。

「**streamlines**」上で指定したベクトルデータ、始点面からの流線を「streamlines」モジュールで可視化する。説明は後ほど行うが、「streamlines」モジュールは 2 つインスタンスする。「streamlines」モジュールのコントロールパネルを開き、まず、「combine vect」の‘velocity vector’が‘combine_vector’となっていることを確認する。これで「combine vect」で設定したベクトルデータを受け取っていることが分かる。次の‘backward’、‘forward’で、各始点から延ばす流線の方向を設定する。例では「streamlines」で‘forward’を、「streamlines#1」で‘backward’を選択しており、このために「streamlines」を 2 つインスタンスした。‘nsegments’は、各グリッドから出る流線の単位長さを指定するパラメータである。この値が大きい程単位長さが小さくなるが、より細かい変化も可視化できる。‘max segments’で各グリッドからの流線計算の積分ステップを最大 10000 まで指定することができる。この値が大きいほど、一つのグリッドから出た流線の長さが伸びることになる。‘nsegments’、‘max segments’は可視化するデータによって適した値を選ぶ必要がある。‘order’で積分定数の指定、つまり精度の指定を行う。この値は、特に理由がない限り、最大値である 4 に設定しておく。‘color’で色の有無を指定する。On にした場合、ベクトルの大きさに応じて色が付けられる¹¹ (図 9.11)。逆に、Off にした場合は後述するカラーマップの編集の項で指定した色で表示される (図 9.12)。例では白に設定されている。

矢印の表示 - Arrow1

使用モジュール	: Arrow1、downsize、glyph
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t4b1.v, avs.t4b2.v
可視化結果	: 図 9.13、図 9.14

この項では、流線の方向を示す矢印を可視化する方法を紹介する。まず、「orthoslice」で面を指定し、そこに矢印を出力する方法から説明する。avs.t4b1.v を読み込む。

```
[Main] → [Geometries] → Arrow1
[Main] → [Mappers] → glyph
[Main] → [Filters] → downsize
[Main] → [Mappers] → orthoslice
```

データは「combine vect」→「orthoslice」の過程で、ある面内のベクトルデータに限定される。

「**downsize**」 「downsize」は、各軸方向の格子数を間引くモジュールである。コントロールパネルの‘I downsize factor’、‘J downsize factor’¹²で、間引きの係数を指定する。各成分について

¹¹ベクトルの大きさを計算するために「magnitude」モジュールも用意されている。

¹²例では「orthoslice」と接続しているので「downsize」には 2 次元平面のデータが入力されている。そのため、I、J の 2 成分のみが表示されているが、後述する crop を用いた場合など、入力データが 3 次元の場合は、‘K downsize factor’

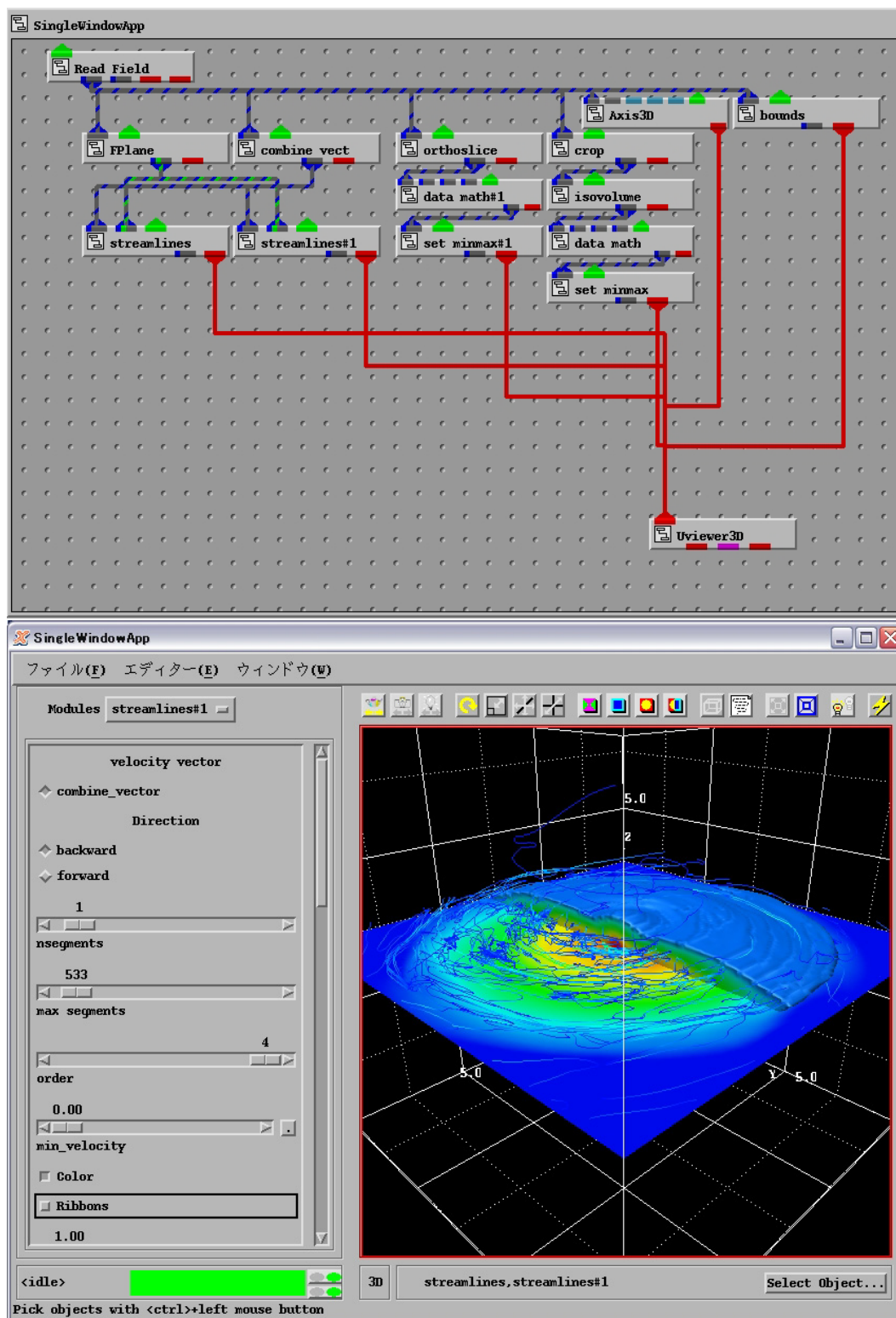


図 9.11: 流線の表示 1

も現われる。

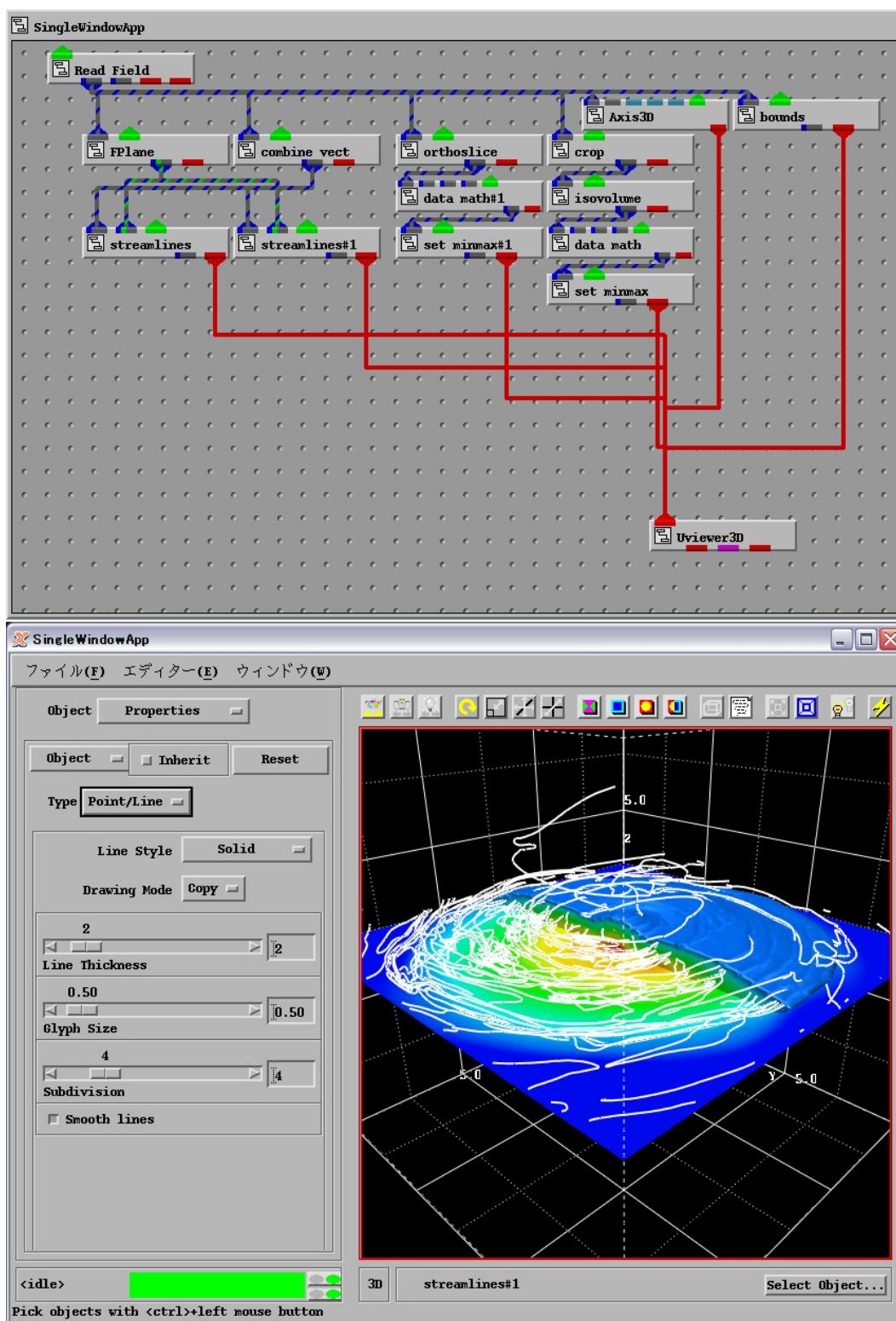


図 9.12: 流線の表示 2

$$[\text{downsize後の格子数}] = \frac{\text{downsize前の格子数}}{\text{downsize factorの値}} \quad (9.2)$$

の計算式を用いて変更されるので、‘N downsize factor’ の値を大きくするほど格子数は小さくなり、結果矢印の個数は少なくなる。

「**glyph**」 次に「glyph」モジュールで矢印の大きさを設定する。‘Mode’ で表示方法を選択する。ここでは‘vector’を選ぶ¹³。‘Normalize’ の ON、OFF で矢印の大きさを、各格子点でのデータの大きさに依存させるか、‘scale’ で設定した値でそろえるかを選択する。スケーリングは、‘Scale X’、‘Scale Y’、‘Scale Z’、から選択された成分について行われる。

「**Arrow1**」 最後に矢印を表示するモジュール、「Arrow1」を接続する。「Arrow2」、「Arrow3」、「Arrow4」も矢印の形が異なるだけで機能は同じであるが、もっとも見やすいのは「Arrow1」ではないか、ということで例では採用している。これは各自の好みで選択してもらいたい。

‘avs.t4b2.v’ のネットワークは、「orthoslice」の代わりに「crop」を使用し、ある空間にのみ矢印を表示させている例である (図 9.14)。

9.2.5 補助的なデータの可視化

カラーバーの表示 -LegendHoriz/LegendVert

使用モジュール	: LegendHoriz、LegendVert
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t5a1、avs.t5a2
可視化結果	: 図 9.15、図 9.16

ここでは、可視化したデータの、色とデータ値の対応表である、カラーバーの表示について説明する。カラーバーは、データを画像として出力し、論文や研究会などの場で発表するときに必要な。AVS/Express では、横置きと縦置きの 2 種類が用意されており、それぞれ「Legend Horiz」、「Legend Vert」である。これらは置き方以外の機能は全く同じであるので、可視化するデータによってより見やすい方を選択することができる。‘avs.t5a.v’ は「LegendHoriz」を、‘avs.t5b.v’ は「LegendVert」を用いたネットワーク例である。

[Main] → [Geometries] → LegendHoriz
or
[Main] → [Geometries] → LegendVert

カラーマップを表示したいモジュールの出力ポートに接続する。例では §9.2.3 で設定した、「set minmax」と接続することで、「isovolume」のカラーマップを表示するようにしている。ここで、「set minmax」のコントロールパネルで、‘min value=-1.5’、‘max value=-0.5’ が、カラーバーのデータラベル (-1.50~-0.5) と一致していることを確認する。さらに、‘min value’、‘max value’ の値を適当な値に変え、カラーバーのデータラベルが一緒に変化することも見てみるとよい。

¹³‘scalar’、‘component’についてはマニュアル参照のこと。

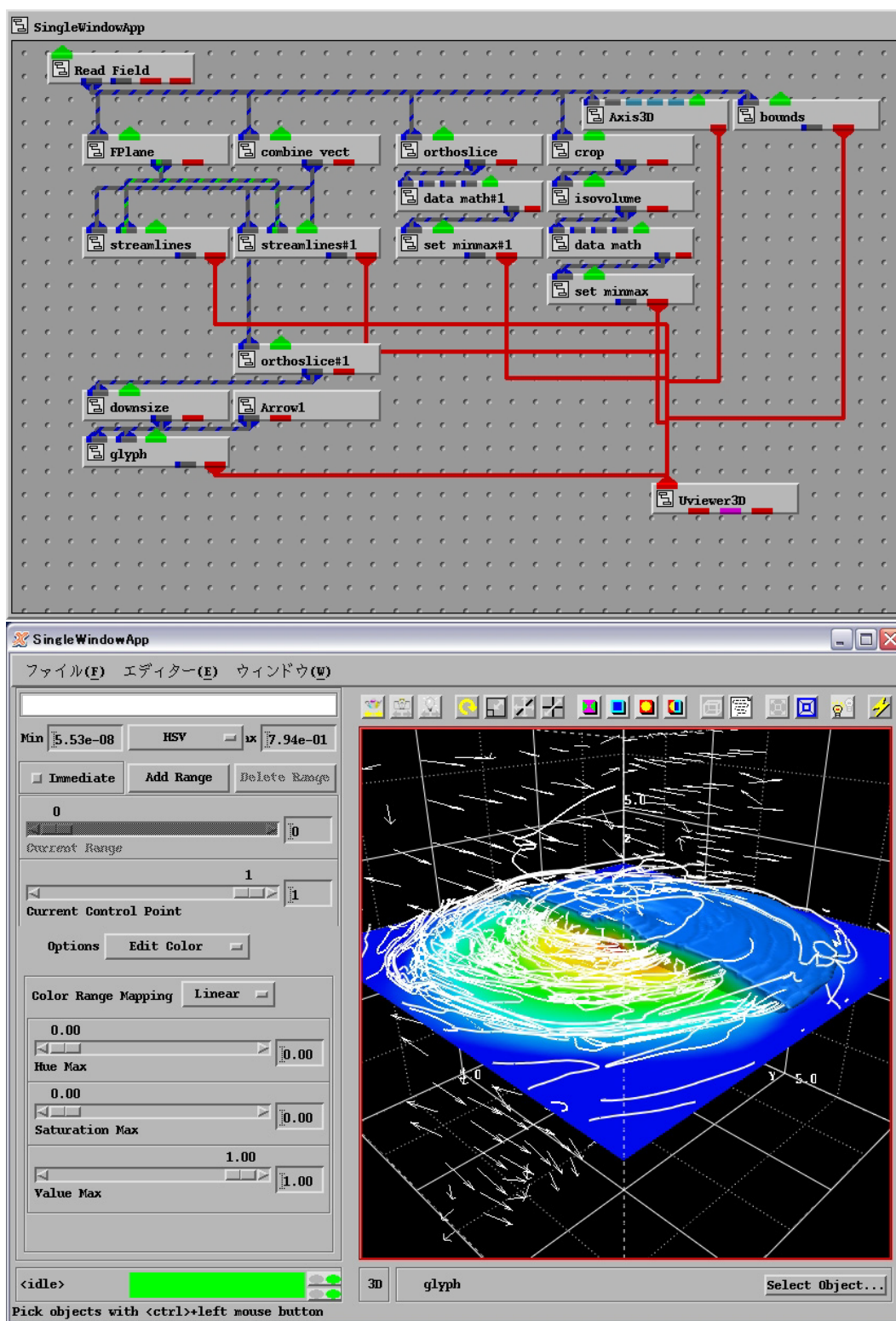


図 9.13: 矢印の表示 orthoslice 利用

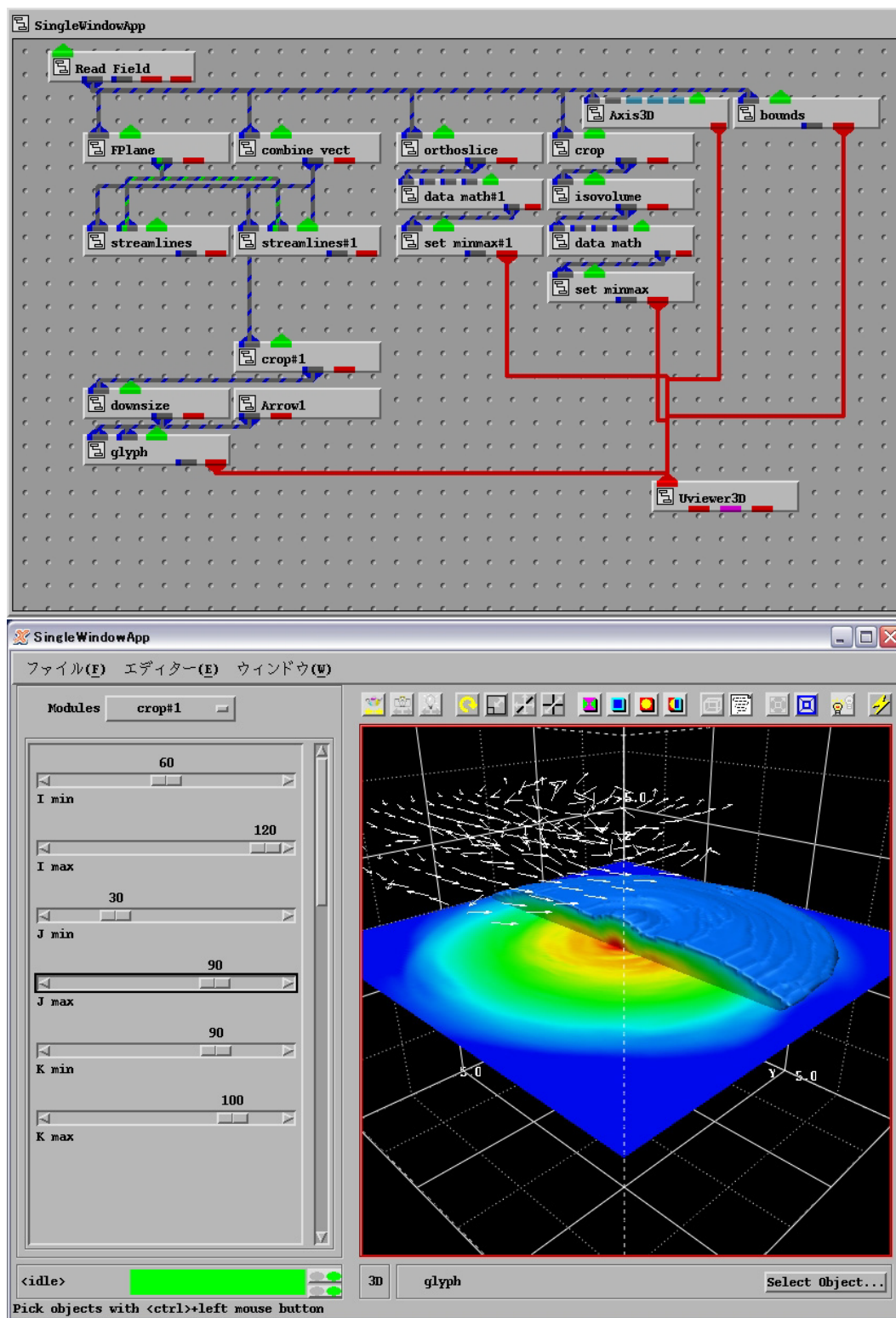


図 9.14: 矢印の表示 crop 利用

コントロールパネルの、‘X Min’、‘X Max’、‘Y Min’、‘Y Max’は各成分の端の位置を指定しており、これでカラーバーのサイズを指定する。また、‘Z Value’でz方向の位置を決めている。‘Interval’で表示するデータラベルの数を、‘Label Offset’で、データラベルのカラーバーからの相対位置を指定する。‘Format’でデータラベルの書式を設定する。これはFORTRAN等での書式と同じであるのでそちらを参照されたい。例は‘全部で3桁、小数部分2桁で小数で表示’を表している。また、‘Outline’、‘Labels’、‘Ticks’で、それぞれカラーバーの枠、データラベル、目盛の表示、非表示を選択する。これも各自の好みで設定してもらいたい。

出力に際して、注意したいのは‘Font’の設定である。例では以下のようにになっている¹⁴。

```
Font    -adobe-helvetica-medium-r-*-15-*-*-*-*
```

ここで‘15’は文字サイズであるが、このままだと、可視化データを画像として出力する際、そのやり方によっては、小さすぎて見にくくなってしまうことがある。状況に応じて変更することをおすすめする¹⁵。例ではカラーコントラを有効に接続できるのは「isovolume」→「set minmax」、「orthoslice」→「set minmax#1」である。今回は二つの「set minmax」の‘min value’、‘max value’を同値にしているので、どちらに接続しても「LegendHoriz」は同じカラーバーを出力する¹⁶。

タイトルの表示 -TextTitle

使用モジュール	: TextTitle
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t5b.v
可視化結果	: 図 9.17

本節では「TextTitle」モジュールを使い、タイトルを表示する方法について紹介する。

[Main] → [Geometries] → TextTitle ¹⁷

コントロールパネルの‘String’にタイトル名を入力する。‘Background’を On にしておくで見やすい。また、‘Offset’のラジオボタンを On にし、‘X Offset’、‘Y Offset’、‘Z Offset’を指定すると、表示する位置を指定することができる。詳細は AVS/Express のマニュアルを参照されたい。

9.2.6 モジュールごとの表示の設定

ここでは、モジュールごとに線の幅や色、表面の特徴などを設定する方法を紹介する。これによって、より効果的で分かりやすく可視化することが出来る。

編集するモジュールの選択 コントロールパネルの下部に

¹⁴フォント、フォーマットはUNIXのX Window形式で指定する。

¹⁵‘Font’内のこの値は、図の出力結果を見ながら最適な値を見つけて欲しい。後に詳しく述べるが、500×500の画像の時は15、1000×1000の画像のときは25程度がよいようである。

¹⁶オブジェクトごとに複数のカラーバーを接続することもできる。コントロールパネルでバーやデータラベルの位置や幅を調整し、うまく表示させる。

¹⁷似た機能のモジュールに「TextString3D」がある。このモジュールで指定した文字列は、ビューウィンドウでオブジェクトの回転などの操作を行ったときに「TextTitle」で指定した文字列が、画面に固定されているのとは違い、一緒に動く。

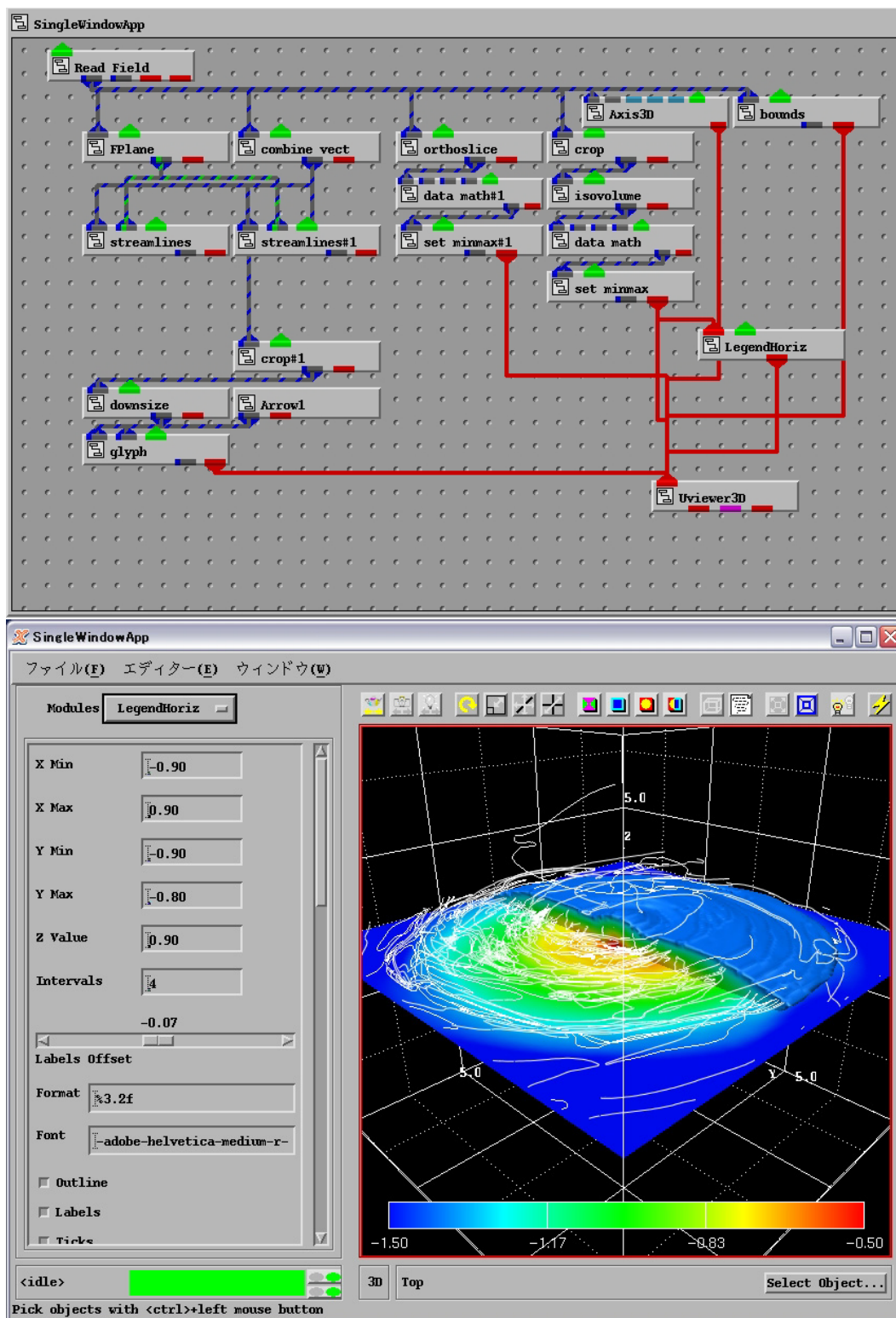


図 9.15: カラーバーの表示 LegendHoriz 利用

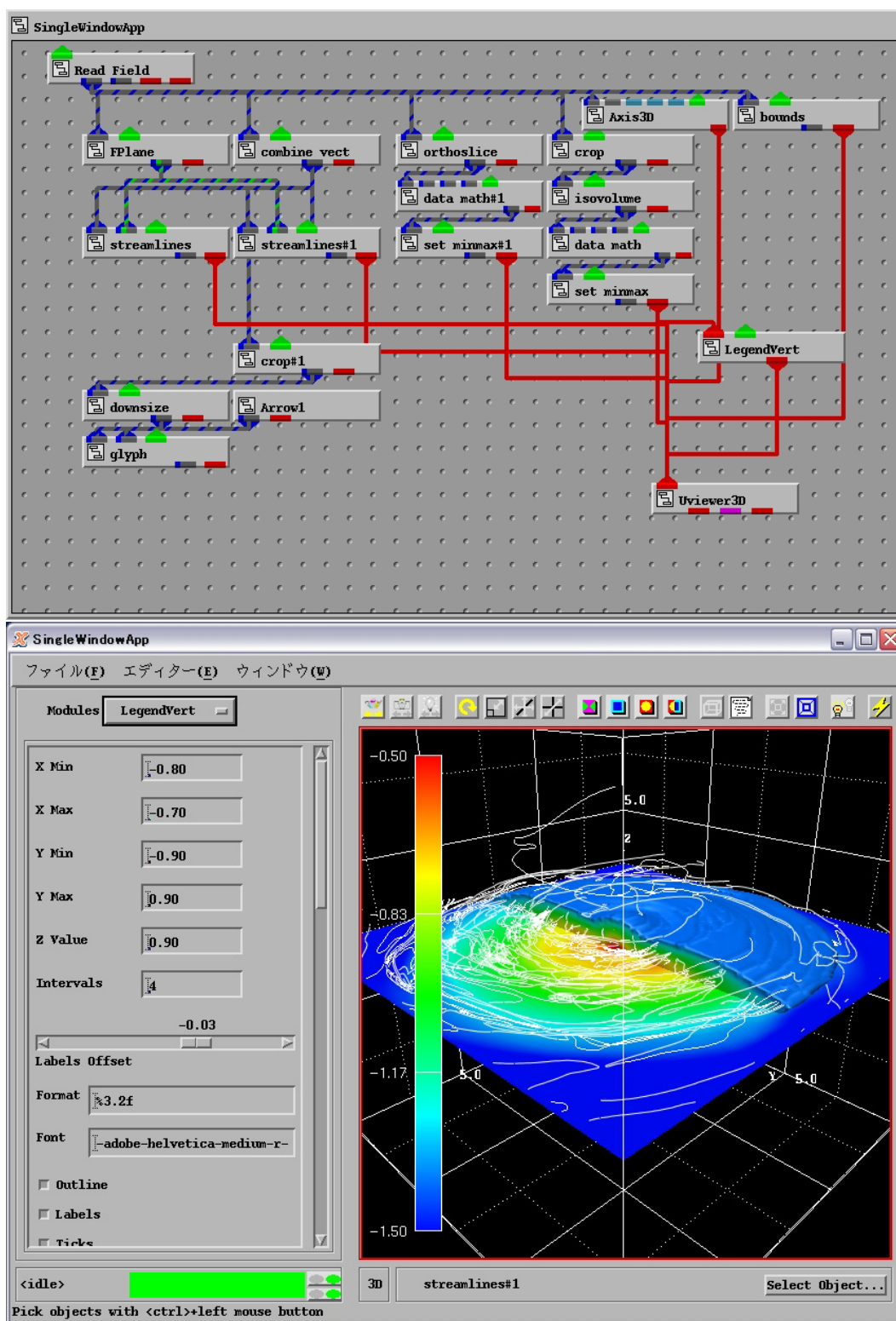


図 9.16: カラーバーの表示 LegendVert 利用

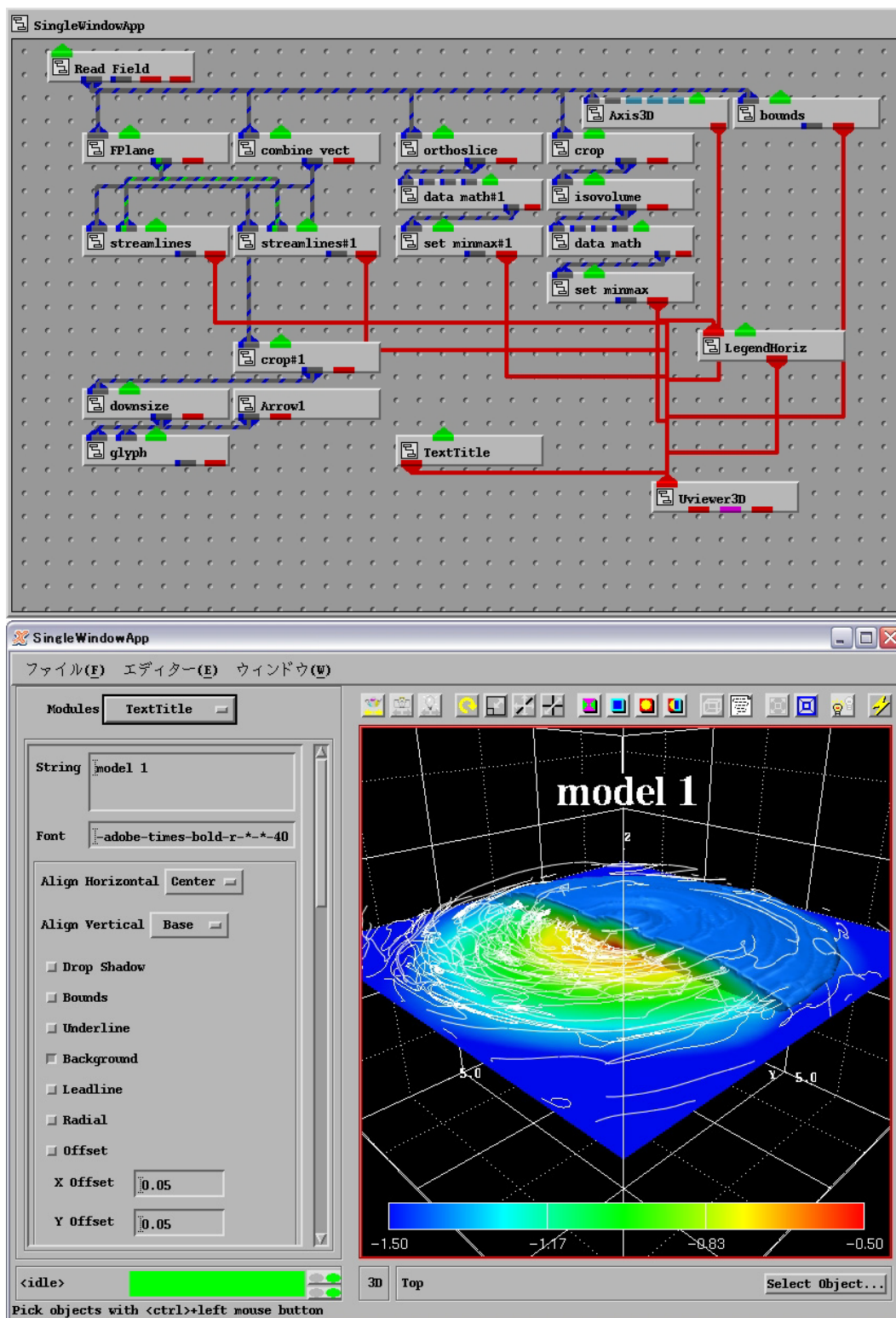


図 9.17: タイトルの表示

3D Top

[select Object...]

の部分がある。‘Select Object...’をクリックすると、Object Selector ウィンドウが現われる。(図 9.18上) ここには、「Uviewer3D」モジュールにつながっているモジュールが表示されている。この画面で設定を変更するモジュール、もしくは最終的に「Uviewer3D」と接続されているモジュールを選択する。例えば、カラーバーのラベルの表示を設定したい場合は、‘LegendHoriz Labels’を、「Orthoslice」の表示を設定したい場合は、‘Set minmax #1’を選択すればよい。また、初期に選択されている‘Top’を選ぶと、「Uviewer3D」に接続されている全てのモジュールを選択する¹⁸。ここでは、「Orthoslice」を変更するために‘Set minmax#1’を選択しておく。

表示方法の編集 次のようにしてコントロールパネルメニューから、表示方法の編集画面を開く (図 9.18下左)。

[エディター] → [Object]、Object[Properties]

Type[General] 単色のオブジェクトの色を決める。

Type[Point/Line] 線の太さ、線種等を選択する。「Stream lines」や「Axis3D」、「bounds」等、線の出力を行うモジュールについて有効。

Type[Surface] オブジェクト全般に有効で、表面の設定を行う。特に‘Opacity’(透過度)の変更が有効である。例では‘Opacity=0.8’にし、少し向こうが透けるようにしている。

カラーマップの編集 次のようにしてコントロールパネルメニューから、カラーマップの編集画面を開く (図 9.18下右)。

[エディター] → [Datamap]、Object[Properties]

‘Current Control Point’が0の場合、最小値の、1の場合最大値の色の編集を行う。編集法の詳細はマニュアル参照のこと。

¹⁸‘Object Selector’は、要は、ビューウィンドウに表示されている、どのオブジェクトを変更するかである。あるItemを選んでいて状態でビューウィンドウを回転させたりすると、選択したオブジェクトのみが回転してしまい、他のオブジェクトとずれてしまう。この失敗はよくやりがちである。視点などを変えるときは、必ずItemsが‘Top’になっていることを確認すること。

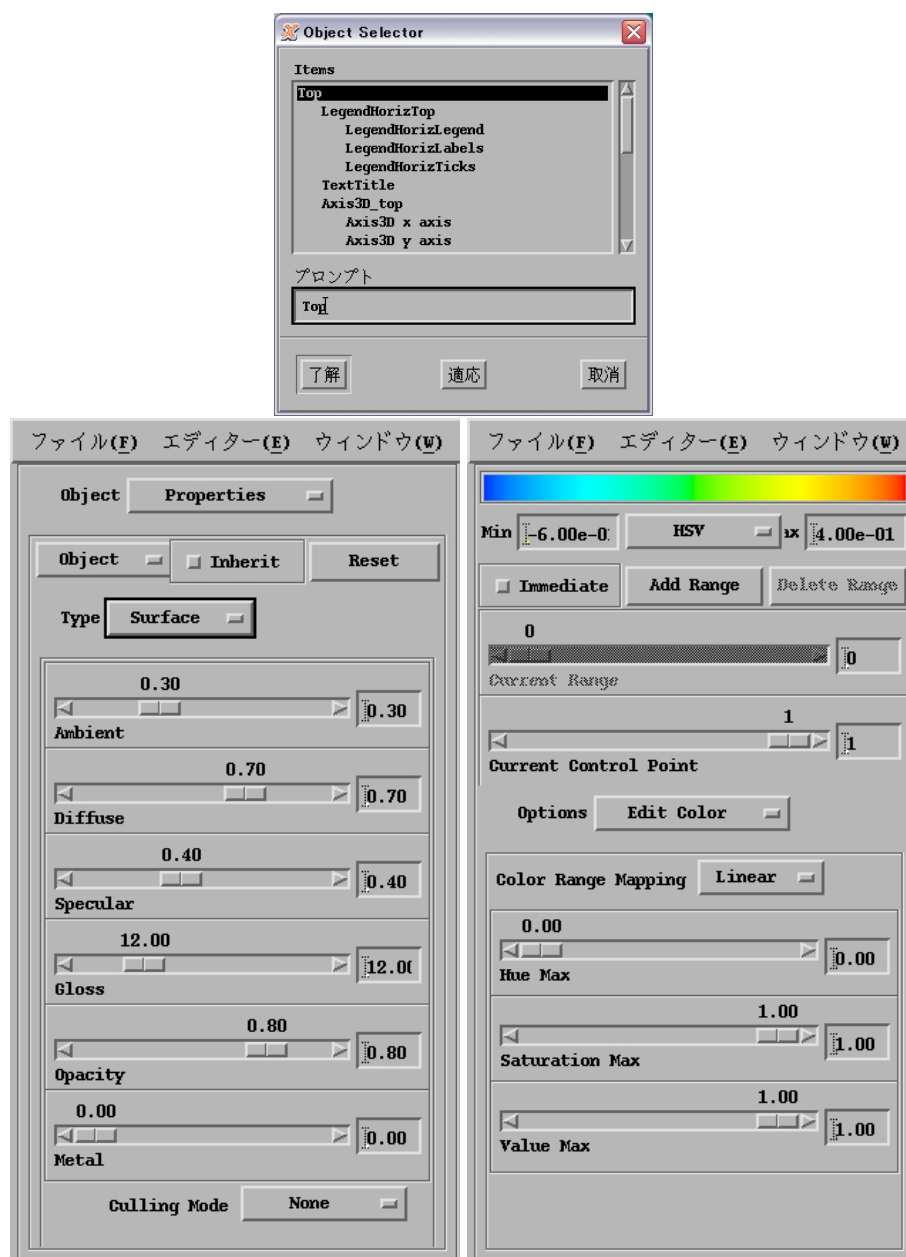


図 9.18: モジュールごとの表示の設定

9.2.7 画像の出力

ビューワーウィンドウの出力 1 - 画像として保存 -OutputField/Write Image

使用モジュール	: Output field、Write Image
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.t7.v
可視化結果	: 図 9.19

ここではビューワーウィンドウに可視化された図を、画像データファイルとして出力する操作について

て説明する。以下の 2 つのモジュールをインスタンスする。

[Main] → [Viewers] → OutputField

[Main] → [Data IO] → Write Image

まず、「Write Image」で出力する画像データファイルのフォーマット等を指定する。コントロールパネルを表示する。画像として出力する場合、「File Format」を「TIFF」にすると高画質の画像が出力されるので推奨する。「Image Filename」でディスクに保存するファイル名を選択する。「Browse」ボタンを押すとファイルダイアログが開くので、適当なファイル名を選択、もしくは入力する。入力欄に直接フルパスを入力してもよい。「Overwrite」の On/Off で、「Write File」ボタンを押してファイルに上書きするとき、上書きの確認メッセージを表示するか否かを指定する。「Flip」は画像の上下反転を行うか否かの選択である。

次に「OutputField」モジュールのコントロールパネルを開く。「Width」、「Height」で出力する画像のサイズを指定する。後にその画像を使用する際に拡大や縮小の処理を行うことがよくある。そのとき、元の画像が大きく高詳細であるほど、加工後の画質がよい。そのためにも、ここで指定する画像サイズはなるべく大きいものにしておくとよい。例では 1000×1000 で出力することにする。「Dynamic」を On にしておくと、データが更新される度に画像を出力するので便利である。

「Write Image」、「OutputField」の設定ができたなら「OutputField」の「Output」ボタンをクリックし画像を出力する。

ビューワーウィンドウの出力 2 -PostScript ファイルとして保存

ここでは可視化された画像を PostScript ファイルとして出力する方法を紹介する。

「Uviewer3D」モジュールがあれば他に必要なモジュールは無く、そのときビューワーウィンドウに表示されている画像が出力されることになる。次のようにして出力の設定画面を開く (図 9.20)。

[エディター] → [Print]

各種設定を行った後、「Create Print File」ボタンを押すと、「File」で指定した場所、名前で PostScript ファイル画像が出力される。

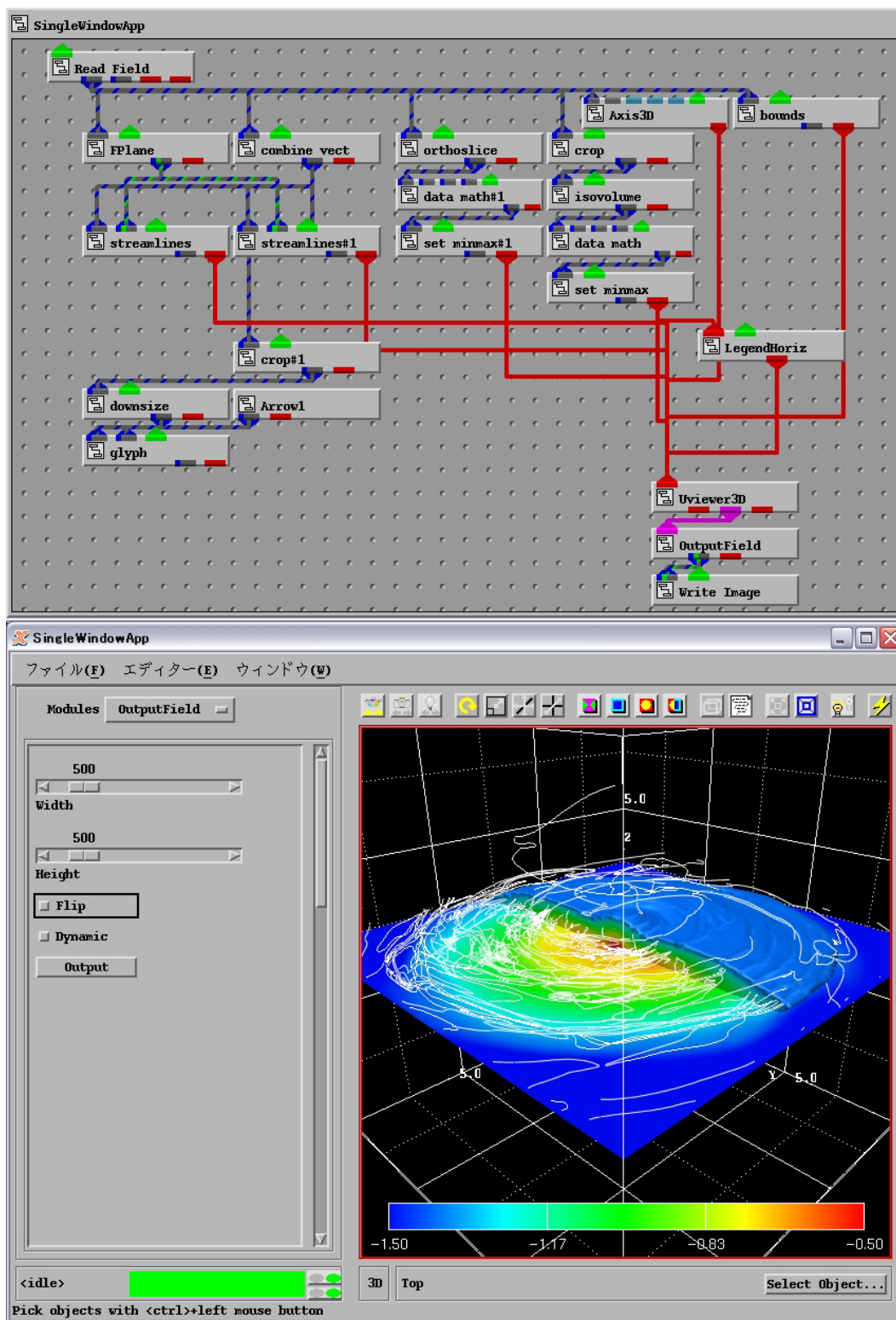


図 9.19: 画像の出力

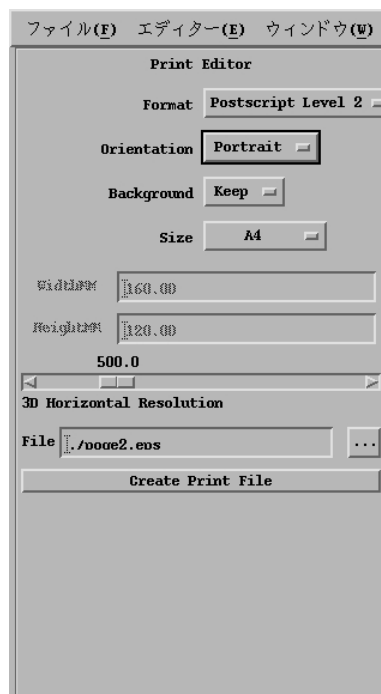


図 9.20: ビューワーウィンドウの出力 2

9.3 netCDFのデータの可視化

この章では、“CANS (Coordinated Astronomical Numerical Software) ¹⁹ ” で数値計算を行った場合のデータ出力形式のひとつでもある、“netCDF(network Common Data Format)” ファイルからデータを読み込み、可視化する方法を説明する。

使用モジュール	: Rd netCDF Fld
使用ファイル	: Chap9/Data.dir/out.cdf
使用ネットワーク	: avs.n1.v
可視化結果	: 図 9.21

まず、「Rd netCDF Fld」という、§9.2の「Read Field」に相当するモジュールをインスタンスする。このモジュールで netCDF データを読み込む。

[Main] → [Data IO] → Rd netCDF Fld

「Rd netCDF Fld」のコントロールパネルの設定を行う。‘netCDF filename’で読み込む netCDF ファイル (out.cdf) ²⁰を読み込むと、‘Select netCDF variables’に、ファイルに含まれていたデータが表示される。ここで使用するデータを選ぶのだが、多く選べばそれだけ処理も重くなる。使用しないデータは Off にしておく。(例では ro、bx、by、bz を使用する設定である。) ‘Field Type’ は Unspecified にしておく ²¹。‘Extract time step’ を On にすると、複数のデータからあるタイムステップを読み出すことができる。例は 5 タイムステップのデータなので On にしておく。‘Read File’ ボタンを押し、netCDF データファイルを読み込む。その後は ‘Time Step’ のスライダを動かすことで、取り出すタイムステップのデータを指定する。

データを読み込んだあとは、§9.2と同じように、モジュールを使用することができる。

¹⁹宇宙磁気流体現象のシミュレーションを支援するソフトウェアライブラリ (第 1 章参照)。

²⁰千葉大学大学院自然科学科 浅井直樹氏 提供。

²¹Unspecified にすると、読み込まれた netCDF ファイルに応じて、自動的にフィールドデータを作る。つまり、ユーザは標準の netCDF ファイルを「Rd netCDF Fld」を使って読み込むだけで可視化をスタートさせることができる。ここが非常に便利な点である。用いた例は、CANS を用いて計算した、直交座表系 (94×94×94) の計算データである。

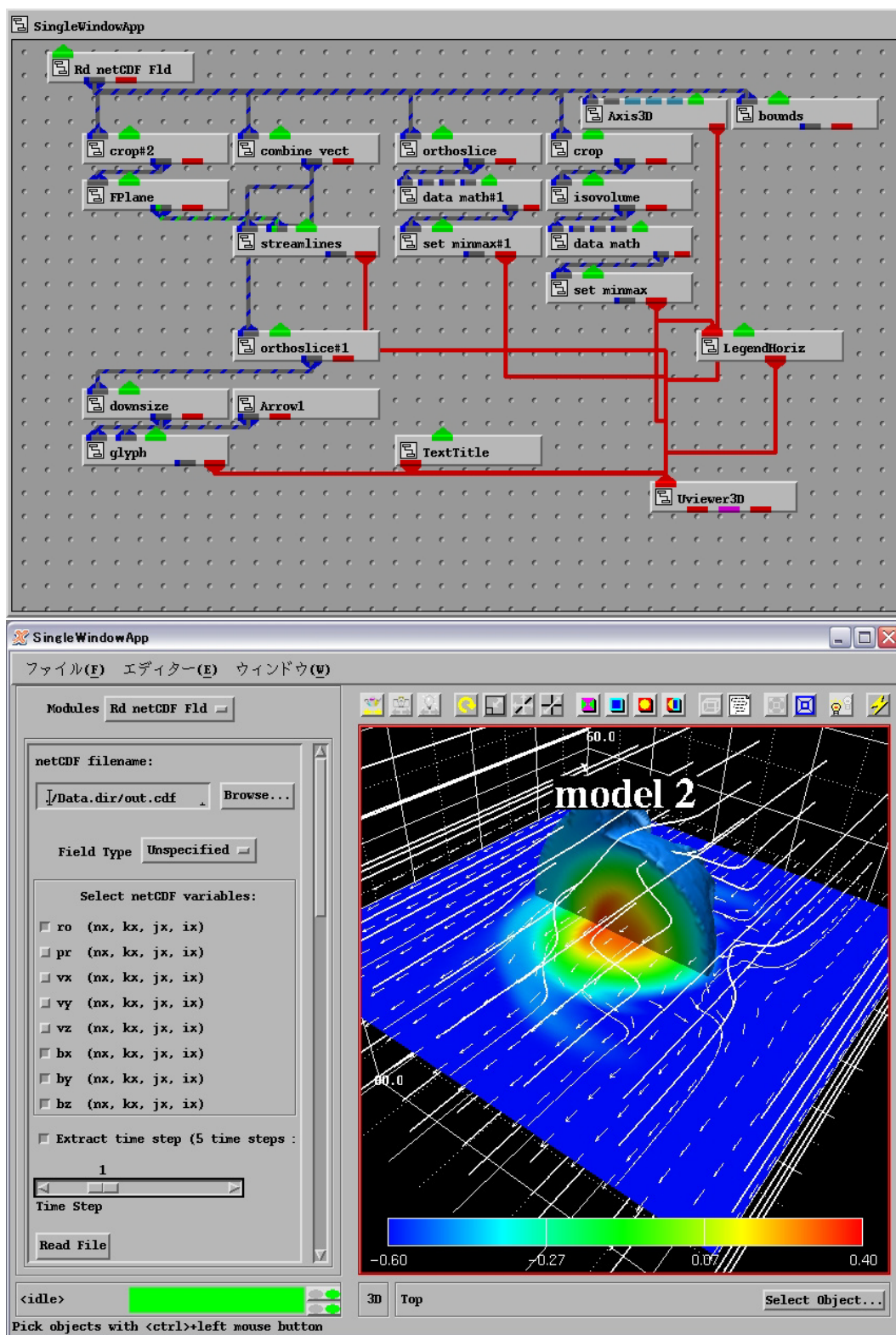


図 9.21: netCDF のデータの可視化

9.4 応用編

本節では、複数タイムステップの field データファイルを読み込む方法、§9.2、§9.3で実習した、基本的な可視化画像を用いた、動画の作成方法を紹介する。

9.4.1 連続の field データファイルの読み込み

使用モジュール	: Read Field
使用 field データファイル	: sample2.fld
使用ネットワーク	: avs.a1.v
可視化結果	: 図 9.22

ここでは §9.2 で 1 タイムステップを読み込んで可視化していたのに対し、一度に複数タイムステップを読み込む方法を説明する。具体的には、‘field データファイル’の書式を変更するだけでよい。‘sample2.fld’ファイルを見ながら説明する。

```
# AVS field file
#
# This is the sample field file for demonstration of AVS manual.
# Written by NISHIKORI, Hiromitsu.

ndim=3
dim1=121
dim2=121
dim3=121
nspace=3
veclen=8
data=float
field=rectilinear
label=ro, vx, vy, vz, bx, by, bz, pr

#time step 1
time value=10001
variable 1 file=./Data.dir/900/ro filetype=unformatted skip=0 stride=1
variable 2 file=./Data.dir/900/vx filetype=unformatted skip=0 stride=1
variable 3 file=./Data.dir/900/vy filetype=unformatted skip=0 stride=1
variable 4 file=./Data.dir/900/vz filetype=unformatted skip=0 stride=1
variable 5 file=./Data.dir/900/bx filetype=unformatted skip=0 stride=1
variable 6 file=./Data.dir/900/by filetype=unformatted skip=0 stride=1
variable 7 file=./Data.dir/900/bz filetype=unformatted skip=0 stride=1
variable 8 file=./Data.dir/900/pr filetype=unformatted skip=0 stride=1
coord 1 file=./Data.dir/900/coord1 filetype=ascii skip=0 stride=1
coord 2 file=./Data.dir/900/coord2 filetype=ascii skip=0 stride=1
coord 3 file=./Data.dir/900/coord3 filetype=ascii skip=0 stride=1
EOT

#time step 2
time value=10002
```

timestep 各ステップについて、時刻やステップ数などの情報を記述する²²。

EOT 各タイムステップの記述の最後に記述しなければ成らない。最終ステップの後にも必要である。

field データファイルを読むモジュールは、§9.2と同様、「Read Field」である。

[Main] → Read Field

複数のタイムステップが記述された field データファイルを読み込むと、§9.2の場合には出て来なかったパラメータが現われる。ここでは‘Store All Steps’を On にし、全てのステップを読み込むようにしておく。総ステップ数が‘Total Steps’と一致していることを確認する。‘Current Steps’は、‘Step Forward’、‘Step Backward’ ボタンをクリックすると上下し、現在出力されているステップ数を表示する。また、直接値を入力することによって、目当てのタイムステップ数を指定することもできる。‘On-time’、‘Continuous’、‘Bounce’ は、連続出力の方法の選択肢で、それぞれ、1 回連続出力、永久ループ 1 (最終ステップ→最初に戻る)、永久ループ 2(最終ステップ→最初の方に逆再生)の設定になる。

9.4.2 アニメーションの作成

本節では、研究会などの発表の場で、シミュレーション結果を効果的に示すことができる、動画の出力について解説する。動画の原理はみなさんも御存知の通り、何枚もの絵を連続で見せることであたかも動いているように見せる、いわゆる“パラパラ漫画”である。動画作成に関しては、AVS/Express には、大きく

1. 動画ファイルをつくり、出力する機能

2. 画像を変化に応じ、連続的に出力する機能 (他ソフトウェアで動画への加工が必要)

がある。ここでは、後の利用や加工のしやすさを考慮し、2. の方法で動画をつくる方針を採用し、AVS/Express で「動画の素材」をつくる手順を紹介する。なお、以降、これらの「動画の素材」を、便宜上「動画ファイル」と呼ぶことにする。2 種類の動画の作成について述べる。

時間変化の動画

使用モジュール	: Read Field、Loop、String、OutputField、Write Image
使用 field データファイル	: sample1.fld
使用ネットワーク	: avs.a2a.v
可視化結果	: 図 9.24, 図 9.25

この項では、

「視点：固定」・「オブジェクト：固定」・「時間：変化」

の設定で動画ファイルを出力する方法について説明する。なお、使用ネットワークの‘avs.a2a.v’で

²²例のように直接記述する方法の他に、‘time file’文を使ってファイルからコメントを読み込む方法もある。これについてはマニュアル参照のこと。

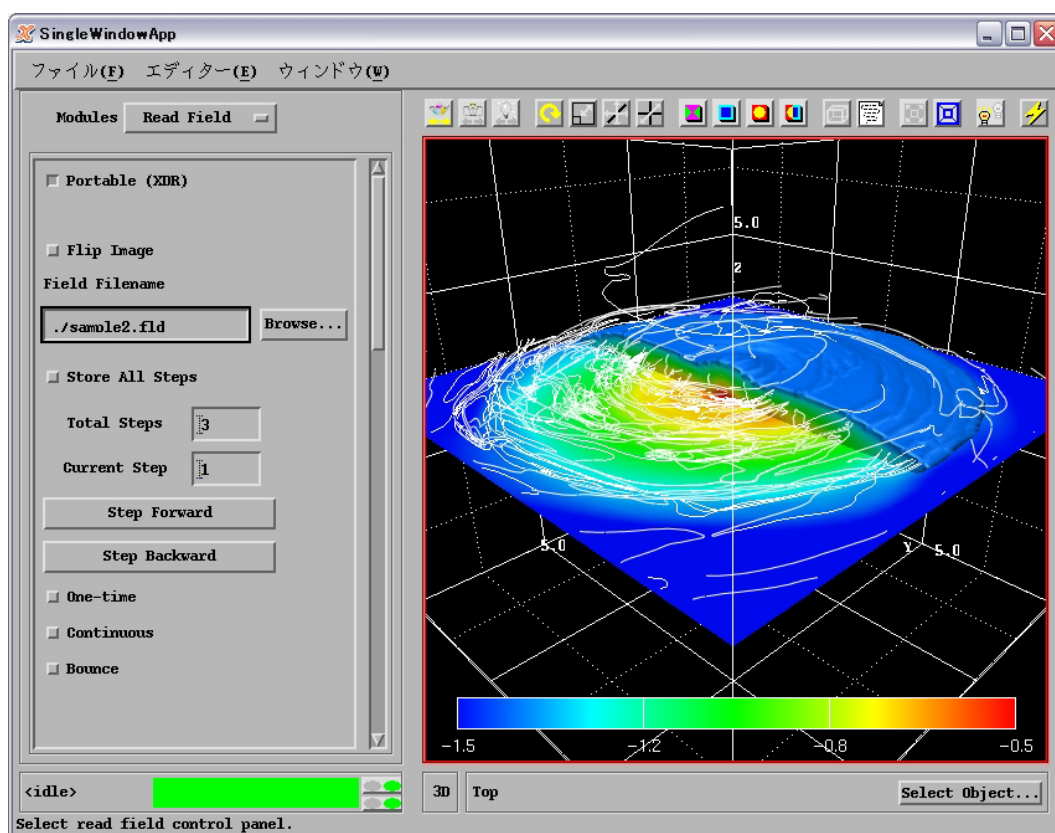


図 9.22: 連続の field データファイルの読み込み

は、読み込んだときに画像ファイルが勝手に書き出されないよう、「Write Image」モジュールの「Overwrite」を Off にしてある。後述するが、連続書き出しをする際は、ここを On にしてから始める。

実際の行程に入る前に、準備が必要である。§9.2で、作った“1 タイムステップのデータを読むための field データファイル”を、動画ファイルを作るために使用するデータの分、それぞれ別の名前で作る。名前の付け方は、

[文字列 (ファイル名の固定部分)] + [タイムステップ (ファイル名の数字部分)]

としておくと非常に便利である。例では、900 ステップ目、1000 ステップ目、1100 ステップ目を読む field データファイルを、それぞれ、‘sample000900.fld’、‘sample001000.fld’、‘sample001100.fld’としている。ここで、タイムステップの桁数をそろえておかなければならない点に注意すること。

ここから AVS/Express を使用し、動画ファイルの作成段階に移る。新しく出て来た次のモジュールをインスタンスする。

[Main] → [Data IO] → Loop

[Standard Object] → [Parameters] → string×2

「String」はファイルの名前を指定するモジュール、「Loop」はファイル名の数字部分を周期的に変化させるモジュールである。流れとしては、「String」モジュールを「Read Field」、「Write Im-

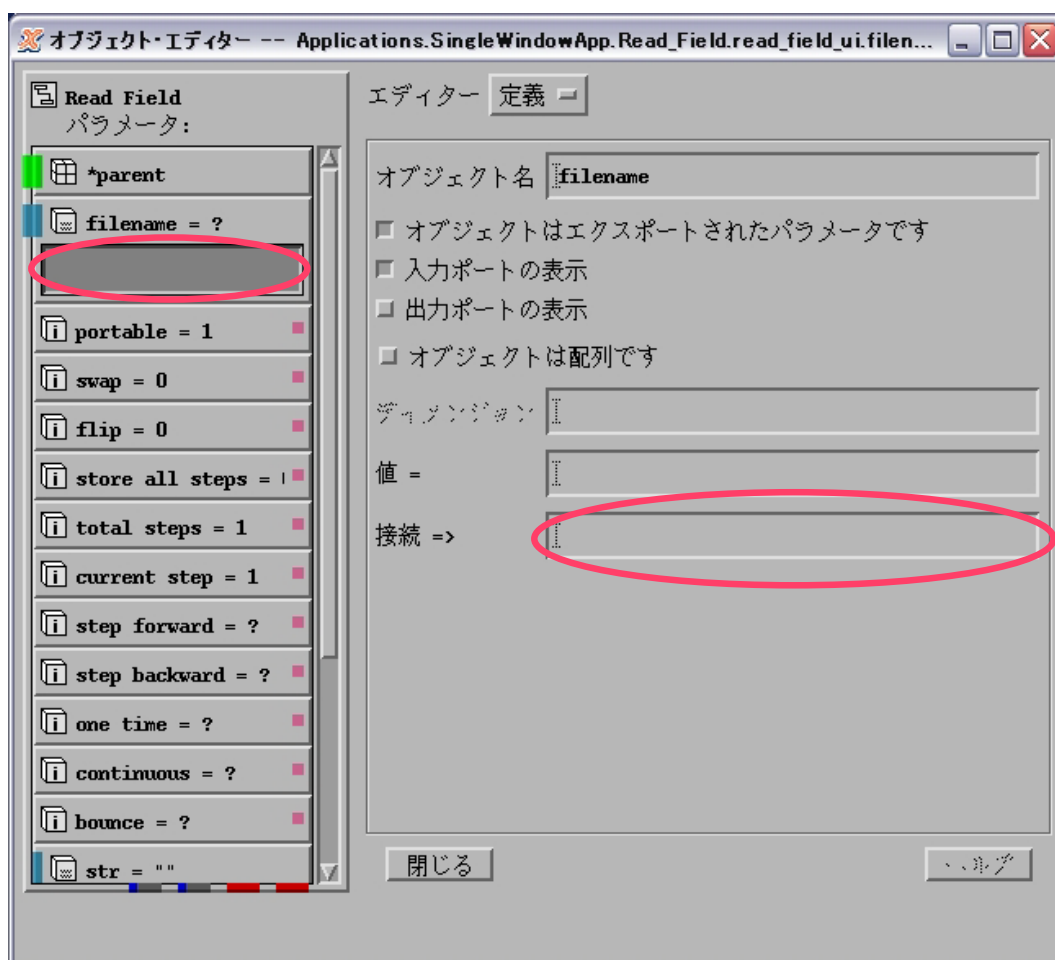


図 9.23: 時間変化の動画「Read Field」モジュールのオブジェクトエディターでの設定

「String」と「Read Field」の再接続 先程確認したように、「String」の水色出力ポートと「Read Field」の水色入力ポートを再接続する²⁴。

「String#1」と「Write Image」の接続 「String #1」には、動画ファイルの出力の情報を記述する。以下は、カレントディレクトリに書き出すよう、相対パスで設定した例である。

```
=> ((' ./image' + str_format('%06d',.Loop.count)) + '.tiff')
```

ここで、'' はダブルクォーテーションである。

「OutputField」、「Write Image」の設定 最後に、出力のためのモジュールの設定を確認しておく。「OutputField」で出力サイズを指定し、「Dynamic」を On にしておく。「Write Image」

²⁴このあたりの手順をあやまると、先程オブジェクトエディター内で空欄にした、「Read Field」や「Write Image」の「filename」欄に、何かが書かれてしまう場合がある。この場合は再度空欄にするところから戻って設定をやり直す必要がある。

の‘File Format’を‘TIFF’にし、‘Overwrite’を On にする²⁵。これらのモジュールについての詳細は §9.2.7 を参照のこと。

動画ファイルの出力の開始 「Loop」モジュールのコントロールパネルを開き、‘Run’をクリックすれば、連続入出力が始まる。field データファイルを読み込む度に‘Loop count’が増えていく。

動画ファイルの出力の終了、確認 ‘Loop count’が‘End Value’の値になると、動画ファイル出力が終了する。早速ファイルが出来ているか確認する。

```
% du -kh ./image*
740K    image000900.tiff
740K    image001000.tiff
740K    image001100.tiff
```

740KB のデータが「Loop」で指定したように出力されていることが確認できた²⁶(図 9.25)。

スライス面が変化する動画

使用モジュール	: Loop、String
使用ファイル	: Chap9/Data.dir/out.cdf
使用ネットワーク	: avs.a2b.v
可視化結果	: 図 9.27、図 9.28

「Loop」を利用して特定のモジュールの変数を変化させることを利用すると、ボリューム表示させる値²⁷や領域²⁸、磁力線の始点等が連続的に変化する動画ファイルを作ることが出来る。ここでは、一つの例として、スライス面を変化させる動画ファイルを出力してみる。

「視点：固定」・「オブジェクト：変化」・「時間：固定」

なお、使用ネットワークの‘avs.a2b.v’では、読み込んだときに画像ファイルが勝手に書き出されないよう、「Write Image」モジュールの‘Overwrite’を Off にしてある。連続書き出しをする際は、ここを On にしてから始める。

次のモジュールをインスタンスする。

```
[Main] → [Data IO] → Loop
[Standard Object] → [Parameters] → string
```

「Loop」と「Orthoslice」の接続 前の“時間変化の動画”では、「Loop」モジュールはインスタンスした後、他のモジュールと接続する必要は無かった。（「String」モジュールで‘Loop count’を

²⁵他の‘File Format’でも、動画ファイルをつくる方法は変わらない。

²⁶このとき、失敗していたりすると、異常に容量が小さいファイルができていたりする。最も実行中にエラーウィンドウが現われるので、不都合があった場合はすぐに解るであろう。また、上下さかさまな図が出力されていた場合は、「OutputField」、「Write Image」の‘Flip’ボタンを操作してみる。

²⁷「Loop」モジュールで‘isovolume」モジュールの‘iso level’を連続的に変化させる。

²⁸「Loop」モジュールで‘crop」モジュールで指定する領域を連続的に変化させる。

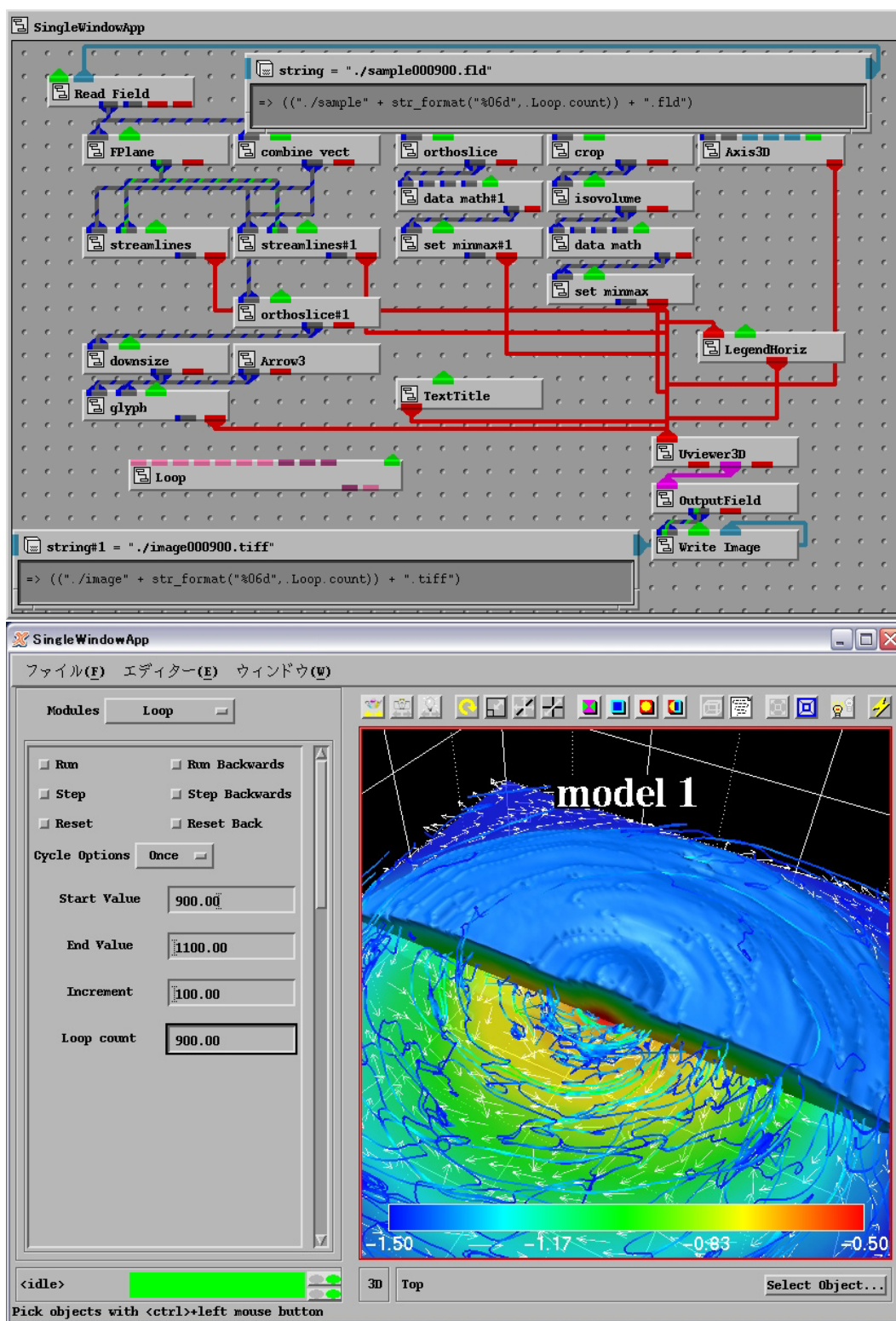


図 9.24: アニメーションの作成 - 時間変化の動画

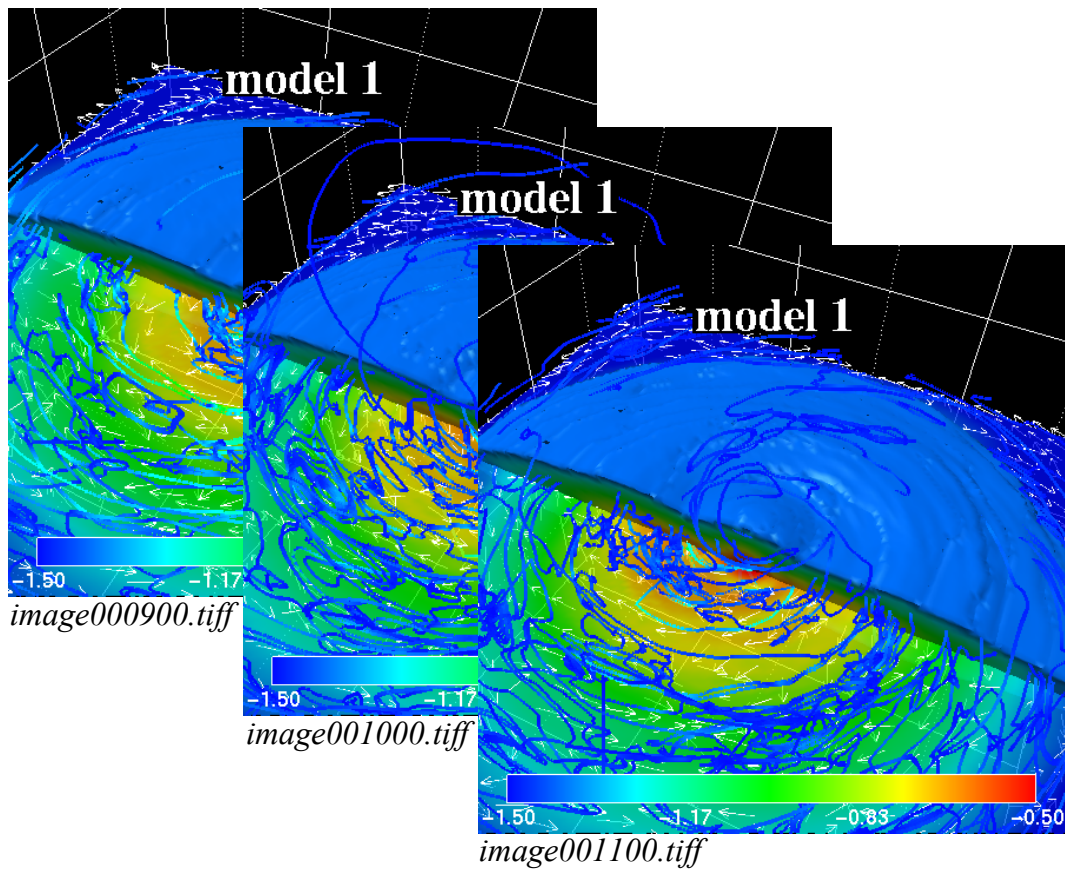


図 9.25: アニメーションの作成 - 時間変化の動画 (出力結果)

読みに行くように設定していた。) 今回は「Loop」を「Orthoslice」に直接接続して、‘plane’ の値を変化させる。まず、「Orthoslice」のオブジェクトエディターを開き、‘plane’ パラメーターを選択する。‘値’、‘接続’ の入力欄を消去し、‘入力ポートの表示’ のラジオボタンを On にした後ウィンドウを閉じる (図 9.26)。すると「Orthoslice」にピンク入力ポートが現われるので、「Loop」の紫出力ポート (カウントデータ出力ポート) と接続する²⁹。

「String」の設定 「String」モジュールが単独であることを確認し、「Loop」からカウントを読み込むように、次のように記述する³⁰。

```
=> (('./ortho_image' + str_format('%03d',..Loop.count)) + '.tiff')
```

’ ’ はダブルクォーテーションである。

²⁹ 接続出来ない場合は、もう一度「Orthoslice」のオブジェクトエディターを開き、‘値’、‘接続’ 欄に文字が入力されていないかを確認する。ここに記入が残っていると、ピンク入力ポートが現われていても接続が出来ない。

³⁰ 詳細は、“時間変化の動画” の「String」と「Read Field」、「Write Image」の接続確認、‘String’ の設定’ を参照のこと。

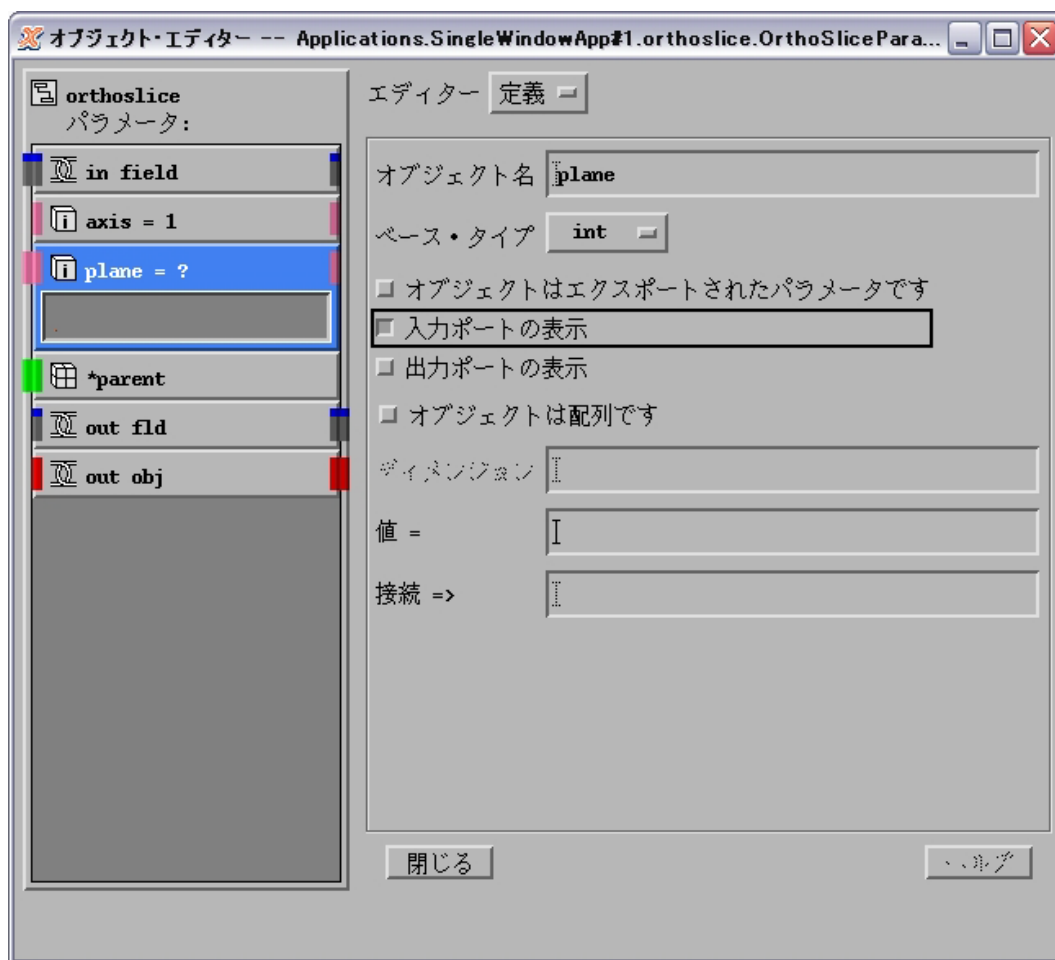


図 9.26: スライス面の時間変化の動画 Orthoslice のオブジェクトエディター設定

「String」、「Write Image」の接続 「String」の水色出力ポートと「Write Image」の水色入力ポートを接続する。

「Loop」の設定 コントロールパネルを開き、「Orthoslice」の「plane」を変化させる範囲、増分を指定する³¹。例は、20 から 74 まで 2 ずつ増加させる設定になっている。

「OutputField」、「Write Image」の設定 最後に、出力のためのモジュールの設定を確認しておく。「OutputField」で出力サイズを指定し、「Dynamic」を On にしておく。「Write Image」の「File Format」を「TIFF」にし、「Overwrite」を On にする。

動画ファイルの出力 設定を確認し、「Loop」モジュールのコントロールパネルで「Run」を押し、動画ファイルの出力を開始する。ここで「Start Value」<「End Value」となっていない場合、「Run」ボタンを押しても、中間の値を無視し、「Current Value」から「End Value」へ一気に飛んでしまうので

³¹詳細は、「時間変化の動画」の「Loop」の設定 参照。

注意。大きい値から小さい値へ変化させたいときは、‘Run Backward’を使う（順番が逆になるだけで、結局同じファイルが出力される）。

動画ファイルの出力の終了、確認 ‘Loop count’ が ‘End Value’ の値になると、動画ファイル出力が終了する。出力ファイルを確認する。

```
% du -kh orthoimage*
740K   ortho_image020.tiff
740K   ortho_image022.tiff
740K   ortho_image024.tiff
740K   ortho_image026.tiff
740K   ortho_image028.tiff
740K   ortho_image030.tiff
740K   ortho_image032.tiff
740K   ortho_image034.tiff
740K   ortho_image036.tiff
740K   ortho_image038.tiff
740K   ortho_image040.tiff
740K   ortho_image042.tiff
740K   ortho_image044.tiff
740K   ortho_image046.tiff
740K   ortho_image048.tiff
740K   ortho_image050.tiff
740K   ortho_image052.tiff
740K   ortho_image054.tiff
740K   ortho_image056.tiff
740K   ortho_image058.tiff
740K   ortho_image060.tiff
740K   ortho_image062.tiff
740K   ortho_image064.tiff
740K   ortho_image066.tiff
740K   ortho_image068.tiff
740K   ortho_image070.tiff
740K   ortho_image072.tiff
740K   ortho_image074.tiff
```

動画ファイルを使ったムービーの作成

出力された動画ファイルなどの静止画を使って動画を作るソフトウェアは、Media Studio pro (canopus)、Quick Time Pro (Apple)をはじめ、Windows用のものが多い。これらのソフトウェアは、読み込んだ複数の静止画を1つのファイルに順に書き出していく、ムービー（パラパラ漫画）を作る機能を持っている。フレームレートなども設定できるので非常に便利である。第8章で Quick Time Pro を使った動画の作り方が紹介されているので、参照して欲しい。

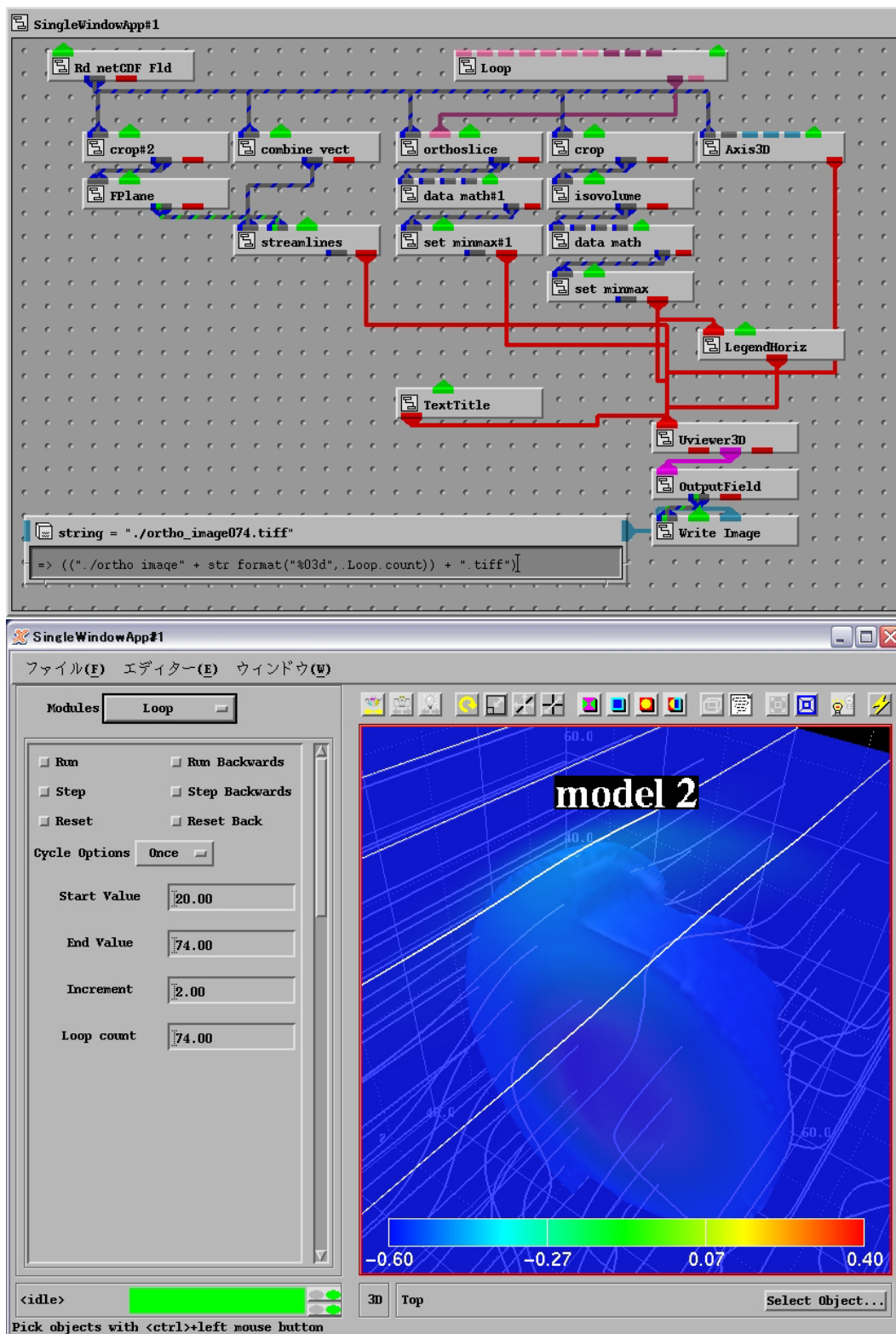


図 9.27: スライス面の時間変化

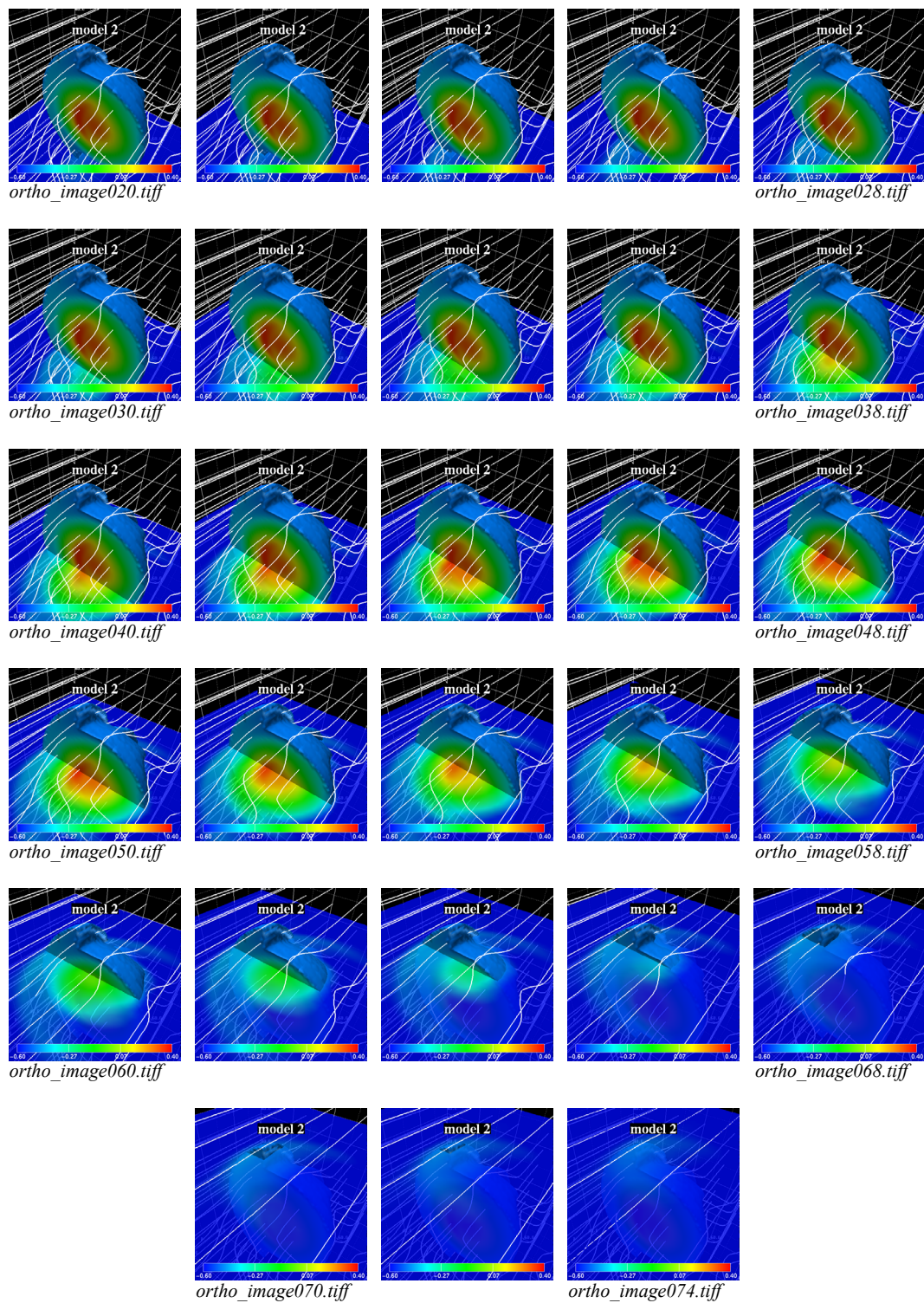


図 9.28: スライス面の時間変化 (出力結果)

9.5 付録

付属 CD-ROM に用意したモジュールネットワークファイル名と、それぞれで利用しているモジュールの一覧表を、付録として添付する。なお、ワークスペース上にあってもビューワーウインドウに表示されていない(「Uviewer3D」に接続されていない)モジュールは除いている。

モジュール	§9.2					
	avs.t1a.v	avs.t2a.v	avs.t3a.v	avs.t3b1.v	avs.t3b2.v	avs.t3b3.v
Arrow1						
Axis3D		○	○	○	○	○
bounds		○	○	○	○	○
crop			○	○	○	○
combine vect						
data math						○
downsize						
FPlane						
glyph						
isosurface			○			
isovolume				○	○	○
LegendHoriz						
LegendVert						
Loop						
orthoslice						
OutputField						
Rd netCDF Fld						
Read Field	○	○	○	○	○	○
set minmax					○	○
streamlines						
string						
TextTitle						
Uviewer3D	○	○	○	○	○	○
Write Image						

表 9.1: モジュールネットワークファイル・モジュール対応表 1

モジュール	§9.2					
	avs.t3c.v	avs.t4a.v	avs.t4b1.v	avs.t4b2.v	avs.t5a1.v	avs.t5a2.v
Arrow1			○	○		
Axis3D	○	○	○	○	○	○
bounds	○	○	○	○	○	○
crop	○	○	○	○	○	○
combine vect		○	○	○	○	○
data math	○	○	○	○	○	○
downsize			○	○		
FPlane			○	○		
glyph			○	○		
isosurface						
isovolume	○	○	○	○	○	○
LegendHoriz					○	
LegendVert						○
Loop						
orthoslice	○	○	○	○	○	○
OutputField						
Rd netCDF Fld						
Read Field	○	○	○	○	○	○
set minmax	○	○	○	○	○	○
streamlines		○	○		○	○
string						
TextTitle						
Uviewer3D	○	○	○	○	○	○
Write Image						

表 9.2: モジュールネットワークファイル・モジュール対応表 2

モジュール	§9.2		§9.3	§9.4		
	avs.t5b.v	avs.t7.v	avs.n1.v	avs.a1.v	avs.a2a.v	abs.a2b.v
Arrow1			○		○	
Axis3D	○	○	○		○	○
bounds	○	○				
crop	○	○	○		○	○
combine vect	○	○	○		○	○
data math	○	○	○		○	○
downsize			○		○	
FPlane			○		○	
glyph			○		○	
isosurface						
isovolume	○	○	○		○	○
LegendHoriz	○	○	○		○	○
LegendVert						
Loop					○	○
orthoslice	○	○	○		○	○
OutputField		○			○	○
Rd netCDFn Fld			○			○
Read Field	○	○			○	
set minmax	○	○	○		○	○
streamlines	○	○	○		○	○
string					○	○
TextTitle	○		○		○	○
Uviewer3D	○	○	○	○	○	○
Write Image		○			○	○

表 9.3: モジュールネットワークファイル・モジュール対応表 3

参考文献

- [1] Nishikori, H., Machida, M. & Matsumoto, R. 2002, The Proceedings of IAU 8th Asian-Pacific Regional Meeting Volume II, 187
- [2] Nishikori, H. 2003, “Global Three-Dimensional MHD Simulations of Galactic Gaseous Disks” (Master’s Thesis)
- [3] 株式会社ケイ・ジー・ティー 2002, AVS/Express モジュール・リファレンス
- [4] 株式会社ケイ・ジー・ティー 2003, AVS/Express チュートリアル・ガイド
- [5] 株式会社ケイ・ジー・ティー 2002, AVS/Express ユーザーズ・ガイド
- [6] 株式会社 富士通長野システムエンジニアリング 2001, AVS/Express Touch & Try 第3版
- [7] 株式会社 富士通長野システムエンジニアリング 2001, AVS/Express Touch & Try 添付資料 第2版
- [8] 株式会社 富士通長野システムエンジニアリング 2001, 2002, AVS/Express 可視化実習 (講習会テキスト)

第 10 章

観測的可視化の目的

立教大学 大須賀 健

近年、宇宙物理学における数値シミュレーション、特に、スーパーコンピュータを用いた大規模数値シミュレーションが、日本をはじめ世界中で盛んに行われている。数値シミュレーションの発展により、これまでになく現実的に、且つ定量的に天体現象を再現することが可能となりつつある。シミュレーションの結果と観測データとを直接つぎ合わせ、理論モデルを検証しつつ、天体現象の本質を探るのが今後の課題であるが、そこで重要となるのが「観測的可視化」である。

そもそも、多くの場合、数値シミュレーションの結果は観測可能量ではない。例えば磁気流体シミュレーションでは、密度場、速度場、温度場、そして磁場の時間発展が求まるが、実際に観測されるのは輻射スペクトルやイメージである。そこで、数値シミュレーションの結果を元に、輻射スペクトルやイメージ、更にはそれらの時間進化を理論的に作り出し、観測データと直接比較可能にすることが重要である。つまり、輻射輸送計算が必要なのである。

このように「観測的可視化」とはシミュレーションの結果と観測データとを直接比較するべく、理論的に輻射スペクトルやイメージを作り出すことである。数値データを人間が理解し易くする為に行う所謂「可視化」とは異なっている。「可視化」された密度分布や温度分布などは、物理現象の理解の助けにはなるがそれらは直接観測できる物理量ではないのである。例えばジェットやスパイラル等、観測されている構造と同形状の密度分布を数値シミュレーションで再現できたとする。しかしながら、観測されているのは輻射強度分布であり、シミュレーションで得られた密度分布をもって観測を再現できたというのは厳密に言うとは誤りである。「観測的可視化」を行い、観測データと比較する必要がある。

10.1 モンテカルロ輻射輸送計算法

ここではモンテカルロ法による輻射輸送計算法を説明する。発生した光子が吸収や散乱を受けながら伝播し、系外に放射される過程を確率的に取り扱い、観測されるであろう輻射スペクトルやイメージを作り出すのが目的である。この方法は、統計的に有意な結果を得るために多数の試行回数が必要であり、決して他の輻射輸送計算法と比べて計算量が少ないわけではない。むしろ計算量が多い方の部類に入る。しかし、数値計算コードの開発が容易で、且つ並列計算に向いているというメリットがあるので、計算機環境が整っている場合は比較的容易に観測的可視化が可能となる。

ここで本章で説明する計算法について幾つかの注意事項を述べておく。ここでは、(磁気)流体シミュレーションで系の構造を解き、その構造をもとに輻射輸送を解いて輻射スペクトルやイメージを

計算することを想定している。輻射輸送計算と流体計算とは完全に独立している。これは、輻射がダイナミクスに影響を与えず、電離や化学変化に与える影響も無視できる場合にのみ有効な方法である。輻射のダイナミクス等に与える効果が無視できない場合は、流体計算の時点で輻射の効果を取り入れる必要がある。輻射（磁気）流体シミュレーションが必要である。

10.1.1 計算手順

ここでは、効率良く計算を行うため Pozdnyakov, Sobol, & Sunyaev (1977) に従い、重み (w) を導入する。大まかな流れを図 1 に示す。

図中の P は escape probability で、光学的厚み (τ) を用いると $P = \exp(-\tau)$ となる。また、 α は scattering albedo、 N_p は試行回数、 δ は微量量である。まずデータを読み込み、各メッシュでの放射率を計算する。そして放射率に応じて光子を発生させる（この方法は次節参照）。発生した光子の一部はガスと相互作用することなく系外へ放射され、残りはガスによって散乱、もしくは吸収される。散乱を受けた光子の一部は透過率に従って系外へ放射され、残りの一部は 2 回目の散乱を受ける。これを重み (w) が小さくなるまで繰り返す。重み (w) が十分小さくなったら ($w < \delta$)、この光子に対する計算を終了し、次の光子を発生させる。これを N_p 回繰り返す。

10.1.2 光子発生

上記の手順中、モンテカルロ法に基づき、確率的な処理を行うのは光子発生とコンプトン散乱の部分である。乱数を発生させ、光子の発生位置、光子の方向、振動数の選択を行う。尚、計算領域の中心に点光源が存在するような問題の場合、発生位置は確定しており、確率的処理を行う必要はない。以下では例として、特定の光源が存在せず、系全体で放射が起こる場合の処理について説明する。

まず、手順に従い全メッシュの放射率を計算する。発生させた乱数を R_1 (0 から 1 の間の値をとる) とすると、

$$\sum_{j=1}^{k-1} \epsilon(j) < R_1 \sum_{j=1}^{N_m} \epsilon(j) < \sum_{j=1}^k \epsilon(j) \quad (10.1)$$

となる k 番目のメッシュで光子を発生させる。ここで、 N_p は全メッシュ数であり、 $\epsilon(j)$ は j 番目のメッシュでの放射率である。放射率の大きなメッシュで多数の光子が発生することになる。この方法は一例である。ここでは紹介しないが各メッシュで同数の光子を発生させる方法も行われている。

振動数の選択についても幾つかの方法がある。上記の方法と同様に放射率に応じて選択する場合、新たな乱数 R_2 を用いて

$$R_2 = \frac{\int^{\nu_0} \epsilon_\nu(j) d\nu}{\int \epsilon(j) d\nu} \quad (10.2)$$

となる ν_0 が発生光子の振動数となる。更に、方向ごとの放射率から光子の方向 ($\mathbf{n}$) を選択する。等方放射であれば簡単に

$$\mathbf{n} = (R_3, R_4, R_5) / (R_3^2 + R_4^2 + R_5^2)^{1/2} \quad (10.3)$$

とすることができる。¹

¹ 編者注: この方式では $\mathbf{n}$ は非等方になる。 $R_3 = (\cos \theta + 1)/2$ 、 $R_4 = \varphi/(2\pi)$ により θ と φ を求め、 $\mathbf{n} = (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$ とすると等方的になる。

10.1.3 コンプトン散乱

乱数を発生させ、散乱位置、散乱後の光子の方向、振動数の選択を行う。まず、散乱位置の選択の仕方について説明する。今、光線の方向を l 軸に取る。光子の位置を原点とし、 l 軸と計算境界との接点を L とする。光子が散乱を受ける位置 l_s は

$$R_6 = \frac{\int_0^{l=l_s} e^{-\tau_\nu} d\tau_\nu}{\int_0^{l=L} e^{-\tau_\nu} d\tau_\nu} \quad (10.4)$$

から得られる。ここで、 τ_ν は光子の位置 ($l = 0$) から l 軸に沿って計った振動数 ν での光学的厚みである。

$$\tau_\nu = \int \left\{ \frac{\rho(l)}{m_p} \sigma_\nu + \kappa_\nu^{\text{abs}}(l) \right\} dl \quad (10.5)$$

$\rho, m_p, \sigma, \kappa_{\text{abs}}$ はそれぞれ密度、陽子の質量、電子の散乱断面積 (Rybicki & Lightman 1979)、そして吸収係数である。ただし、現状の計算コードでは、これを

$$\tau_\nu = \left\{ \frac{\rho(0)}{m_p} \sigma_\nu + \kappa_\nu^{\text{abs}}(0) \right\} l \quad (10.6)$$

と近似し、計算量を抑えている。これは、計算領域内における密度等の物理量の変化が比較的小さい場合には良い近似となるが、そうでない問題を扱う場合には改善することが望ましい。

次に散乱後の方向、振動数について述べる。ここでは電子静止系でのコンプトン散乱の基礎方程式を用いて新たな方向と振動数を選択する (Pozdnyakov, Sobol & Sunyaev 1977)。実験室系で与えられている光子の方向、振動数を散乱相手となる電子の静止系に座標変換し、電子静止系での散乱後の方向と振動数を選択する。それを元の実験室系に変換するというのが基本的な方針である。手順は以下の通りである。

1. 散乱位置での流体の共同座標系に座標変換 (座標変換 1)
2. 散乱位置での電子温度をもとに散乱電子を選択
3. 散乱電子の静止系に座標変換 (座標変換 2)
4. 電子静止系での散乱後の方向と振動数を選択
5. 座標変換 2 の逆変換
6. 座標変換 1 の逆変換、実験室系での光子の方向、振動数を得る

10.1.4 スペクトル&イメージ

透過光子が我々が観測できる光子である。透過光子は位置、方向、振動数と観測データと比較すべき全ての情報を持っている。したがって、

1. 平均的な輻射スペクトル — 振動数別に透過光子を積算
2. 方向依存の輻射スペクトル — 方向、振動数別に透過光子を積算
3. 波長ごとのイメージ — 位置、方向、振動数別に透過光子を積算

とすることで理論的に輻射スペクトルやイメージを作り出すことができる。統計的に有意な結果を出す為、1 より 2、2 より 3 でより多くの光子数（試行回数）が必要となる。また、流体シミュレーションの各タイムステップで輻射輸送計算を行うことで、輻射スペクトルやイメージの時間発展を調べることができる。

10.2 3次元磁気降着円盤の輻射スペクトル

10.2.1 計算手順

ここでは実際の計算例として、3次元磁気降着円盤の輻射スペクトル計算の結果を紹介する。3次元磁気降着円盤のモデルとしては Kato, Mineshige, Shibata (2004) のシミュレーション結果を採用する。磁気流体シミュレーションではイオン温度 (T_i) と無次元した密度 (ρ_{MHD})、磁場 (B_{MHD}) を解いている。密度パラメータを ρ_0 を用いて実際の密度場、磁場に直す [$\rho = \rho_0 \rho_{\text{MHD}}$, $B = B_{\text{MHD}}(\rho_0 c^2)^{1/2}$]。また、輻射輸送計算には電子温度が必要である。磁気流体シミュレーションは一温度プラズマを仮定しており、イオンの温度は解いているが、電子温度は取り扱われていない。そこで、本研究では、輻射スペクトルと同時に、電子温度も自己矛盾無く解くという方法を取る。この為、計算手順は以下のようになる。

電子の加熱と冷却はシンクロトロン放射 / 吸収と制動放射 / 吸収を考慮して決める (Pacholczyk 1970; Stepney & Guilbert 1983)。輻射輸送計算と電子温度の補正を交互に行い、自己矛盾のない解が得られるまで繰り返す。

10.2.2 電子温度

電子温度は厳密には位置（メッシュ）によって異なる。しかし、ここでは電子の温度は鉛直方向に一定で、半径方向にのみ依存すると仮定する。そして輻射による冷却とイオンとのクーロン相互作用による加熱 [λ_{ie} (Stepney & Guilbert 1983)] の平衡温度を求めている。

$$\int_{-Z}^Z \int_{R_{\text{in}}}^{R_{\text{out}}} \lambda_{ie}(\rho, T_i, T_e) 2\pi r dr dz = \int L_\nu(R_{\text{in}} < r < R_{\text{out}}) d\nu. \quad (10.7)$$

ここで断っておくが、移流や膨張・圧縮による熱過程が電子の温度の影響を与えるような場合、この方法で正しい電子温度を見積もることができない。2温度磁気流体シミュレーションが必要となる。

10.2.3 結果

得られた輻射スペクトルを、Sagittarius A*(Sgr A*) の観測データ点とあわせて図3に示す。最も観測を再現するように密度パラメータを選んだもので、結果として得られた電子温度は $5 \times 10^9 \text{ K}$ であった。因みに試行回数はおよそ 10^9 であり、計算時間は Pentium4 の単体計算機で3時間程度である。

電波領域に現れているピーク ($\log \nu \sim 12$) はシンクロトロン放射 / 自己吸収で作られている。また、X線領域 ($\log \nu > 16$) は制動放射が卓越している。シンクロトロン放射の一部がコンプトン散乱を受けた為、 $\log \nu \sim 14$ 付近にピークが作られることが分かった。この結果では目立たないが、 $\log \nu \sim 16$ にセカンドコンプトンの成分が現れている。このスペクトルは磁気降着円盤の中心近傍からのスペクトルだけを計算した物である。図3から分かるように、この場合は磁気降着円盤がよく

観測データを再現できることが分かる。しかし、円盤全体からのスペクトルは観測と矛盾するという結果が得られている。なぜ、観測を再現できないのか？磁気流体シミュレーションの今後の課題である。

10.3 まとめと今後の展開

この章ではモンテカルロ法による観測的可視化の方法を紹介した。観測的可視化は数値シミュレーションと観測データを繋ぐ手段であり、理論モデルの検証、天体現象の解明に必要なものである。特に近年の計算機やシミュレーション技術の発展、観測装置の飛躍的な進化や大型プロジェクトの進行により、観測的可視化によるシミュレーションと観測の直接比較の重要性はますます高くなっていくと思われる。

モンテカルロ法による輻射輸送計算は、計算量は決して少なくないが、コード開発が容易で並列化に向いているという大きなメリットがある。PCクラスターやベクトル・並列計算機が使える環境では特に向いている計算法である。

今後は光学的厚みの見積もりをより正確なものに改善するとともに、コードの並列化、ベクトル化を行いたい。これは既に進行中であり、現在テスト中である。限られた時間でより多くの光子の伝播を計算することが可能となるので、これまで不可能であった角度ごとの輻射スペクトル、波長ごとのイメージを得られると期待できる。また、短時間でスペクトルを得られる為、細かい時間刻みでのスペクトルの変化を調べることが可能となる。円盤振動などの解析が期待できる。更には磁気流体シミュレーションコードとの結合し、例えば、これまでの磁気流体シミュレーションでは扱えなかった輻射冷却が効く状況での降着円盤の進化を調べたい。

参考文献

- [1] Kato, Y., Mineshige, S., & Shibata, K. 2004, ApJ, 605, 307
- [2] Pacholczyk, A. G. 1970, Radio Astrophysics (San Francisco: Freeman)
- [3] Pozdnyakov, L. A., Sobol, I. M., & Sunyaev, R. A. 1977, Soviet Astron., 21, 708
- [4] Rybicki, G. B. & Lightman, A. P. 1979, Radiative Processes in Astrophysics (New York: John Wiley & Sons, Inc.)
- [5] Stepney, S. & Guilbert, P. W. 1983, MNRAS, 204, 1269

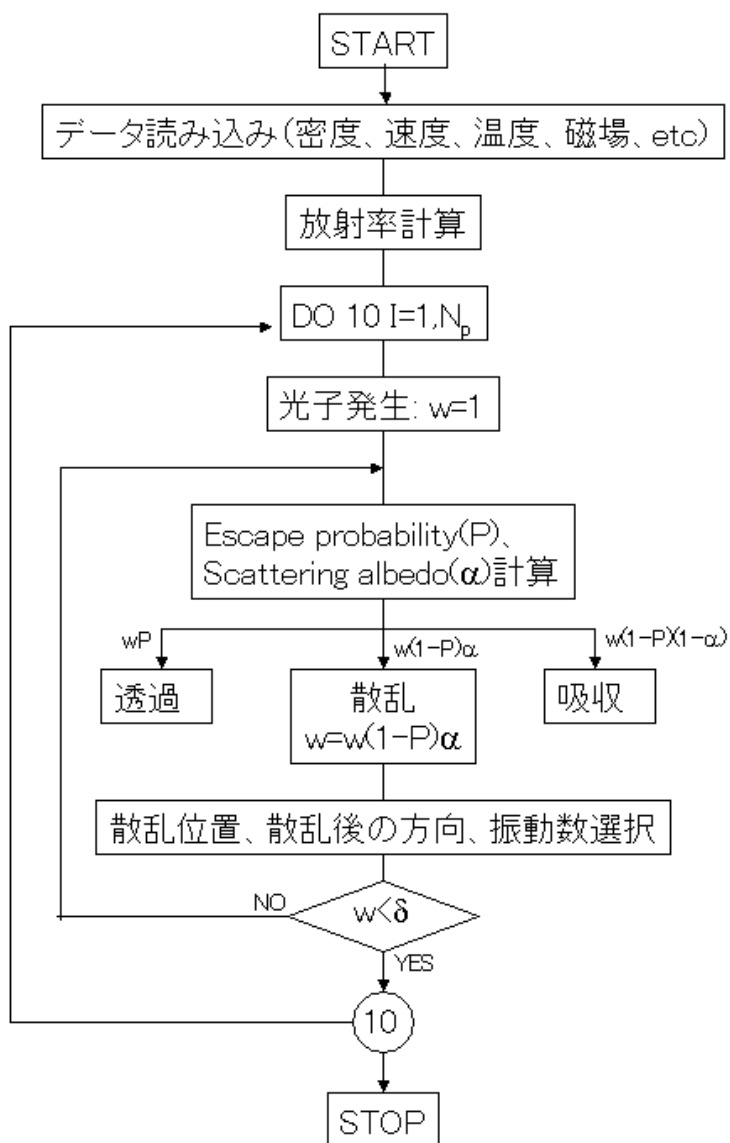


図 10.1: モンテカルロ輻射輸送計算法の手順

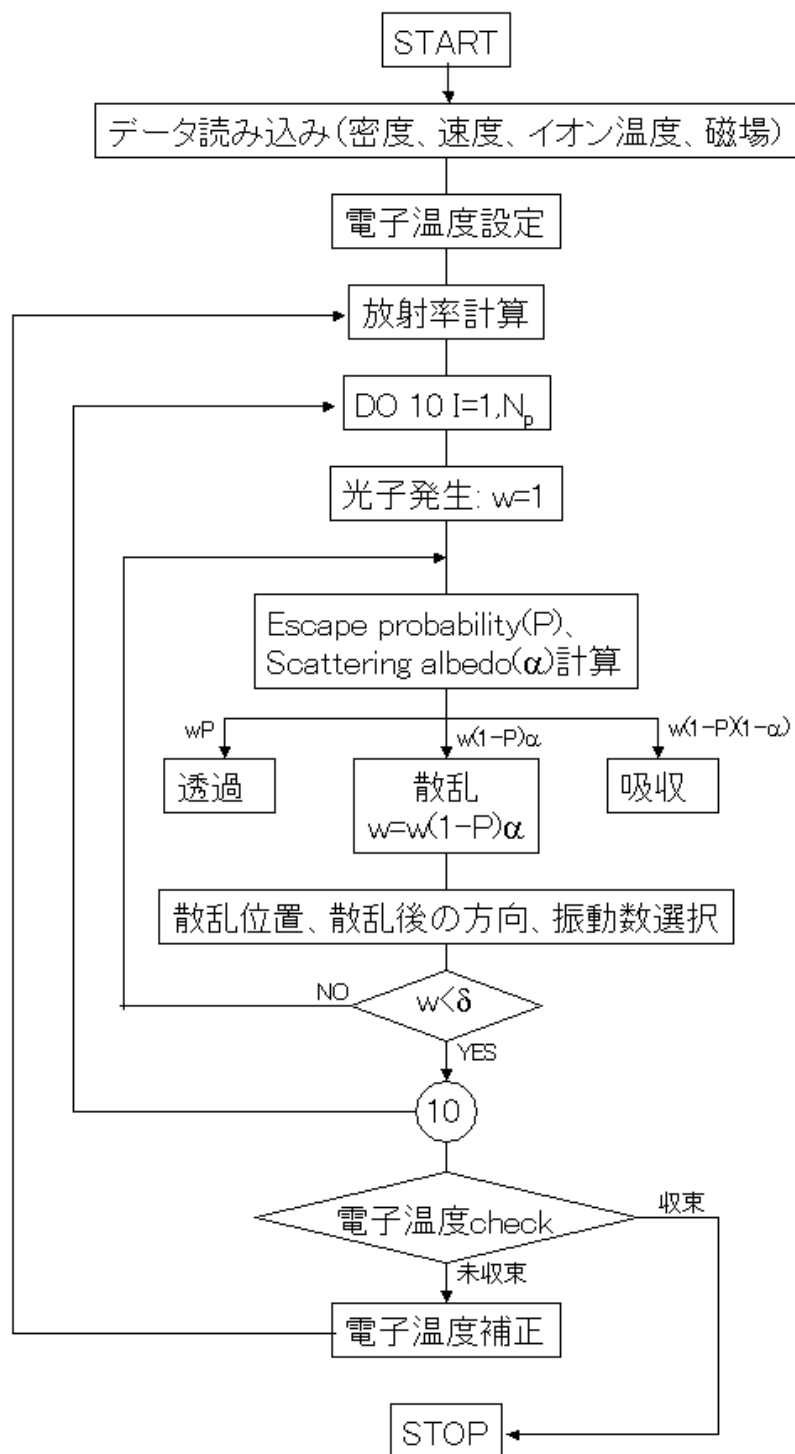


図 10.2: モンテカルロ輻射輸送計算法の手順 2

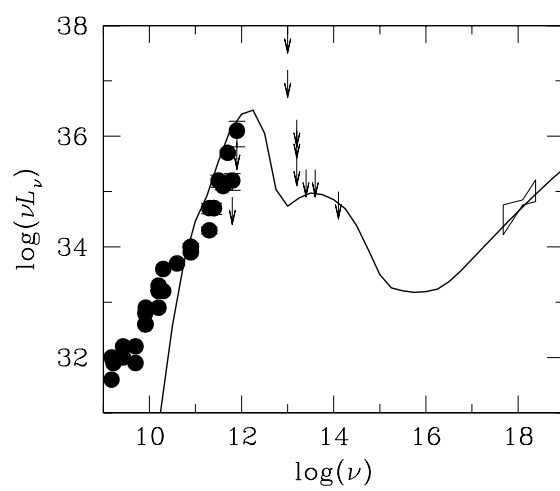


図 10.3: 3次元磁気降着円盤の预言する輻射スペクトル

第 III 部

数値シミュレーションのデータ解析

第 11 章

Fourier 解析

名古屋大学 杉本 香菜子

数値シミュレーションによって得られた密度分布や速度場、磁場がどのような波数のゆらぎから成るかを調べるには、フーリエ変換が有効である。本章では、富士通 SSL II (Scientific Subroutine Library II)¹を使用した 3 次元離散型の高速実フーリエ変換の手法と、スペクトルの計算法について解説する。解析の方法と手順に焦点を絞るため、フーリエ変換や高速フーリエ変換 (FFT) アルゴリズムに関する一般的な解説は各種テキストに譲り (例えば、章末の参考文献を参照)、SSL II の FFT サブルーチンが利用できることを前提に話をすすめる。

実空間のスカラー分布 $F[\mathbf{x}]$ と波数空間での複素振幅分布 $A[\mathbf{k}]$ との間の 3 次元離散 Fourier 変換と逆フーリエ変換は以下のように表せる。

$$\begin{aligned} A[\ell_1, \ell_2, \ell_3] &= \frac{1}{N_1 N_2 N_3} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} F[j_1, j_2, j_3] \\ &\quad \times \exp\left[-i \frac{2\pi j_1 \ell_1}{N_1}\right] \exp\left[-i \frac{2\pi j_2 \ell_2}{N_2}\right] \exp\left[-i \frac{2\pi j_3 \ell_3}{N_3}\right], \end{aligned} \quad (11.1)$$

$$\begin{aligned} F[j_1, j_2, j_3] &= \sum_{\ell_1=0}^{N_1-1} \sum_{\ell_2=0}^{N_2-1} \sum_{\ell_3=0}^{N_3-1} A[\ell_1, \ell_2, \ell_3] \\ &\quad \times \exp\left[i \frac{2\pi \ell_1 j_1}{N_1}\right] \exp\left[i \frac{2\pi \ell_2 j_2}{N_2}\right] \exp\left[i \frac{2\pi \ell_3 j_3}{N_3}\right], \end{aligned} \quad (11.2)$$

ここで、 N_1, N_2, N_3 はそれぞれ x_1, x_2, x_3 方向の格子数を表す。実際の数値計算では、 $F[\mathbf{x}]$ は密度、圧力、運動エネルギーなどの物理量の実数スカラー配列であり、 $A[\mathbf{k}]$ はそれらを波数成分に分解した振幅を表す複素数配列である。

さらに、実フーリエ変換 ($F[\mathbf{x}]$ が実数関数) では以下のような複素共役関係が成立する。

$$A[\ell_1, \ell_2, \ell_3] = A^*[N_1 - \ell_1, N_2 - \ell_2, N_3 - \ell_3]. \quad (11.3)$$

¹富士通 SSL II は科学用サブルーチンライブラリであり、国立天文台データ解析計算センターでは、Fortran および C 言語 (C-SSL II) で使用可能である (2004 年度現在)。また、パーソナルコンピュータなどの逐次計算用に加えて、ベクトル計算機用の SSL II/VP (Fortran 用) および C-SSL II/VP (C 言語用)、ベクトル並列計算機用の SSL II/VPP (VPP Fortran 用) および SSL II/HPF (HPF 用) も利用できる。詳細については SSL II 使用手引書 (国立天文台データ解析計算センターのホームページ等に掲載) を参照。

ただし、 $\ell_i = 0$ ($i = 1, 2, 3$) の時は、 $N_i - \ell_i = 0$ とする。

一般に、(11.1) または (11.2) 式の離散フーリエ変換は、 $N_1^2 \times N_2^2 \times N_3^2$ 回の乗算を要するが、離散点の数 N_1, N_2, N_3 の値によっては計算量を大幅に減らすことが出来る。例えば N_i ($i = 1, 2, 3$) が 2 の中乗で表せるときは、演算の回数は $N_1 N_2 N_3 \log_2(N_1 N_2 N_3)$ に減らせる。この他に計算効率がよくできるのは N_i が小さな素数の中乗で素因数分解できるときで、このときの素因数は基底 (または基数) と呼ばれる。

11.1 数値計算ライブラリ SSLII による高速フーリエ変換

本節では、SSL II/VPP のサブルーチン DP_V3DRCF を例に、FFT の手順を解説する。SSL II のその他の実 FFT サブルーチンとその機能については、§11.4 の一覧を参照して欲しい。サブルーチン DP_V3DRCF は VPP Fortran 用に開発された 3 次元離散型実フーリエ変換ルーチン (2, 3 及び 5 の混合基底: $N_i = 2^p 3^q 5^r$ の場合に使用可) であり、VPP Fortran で記述されたプログラム内で call 文によって呼び出すと、フーリエ変換または逆フーリエ変換として次のような結果を返す。3 次元フーリエ変換は

$$N_1 N_2 N_3 A[\ell_1, \ell_2, \ell_3] = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} F[j_1, j_2, j_3] \times \exp\left[-i \frac{2\pi j_1 \ell_1}{N_1} r\right] \exp\left[-i \frac{2\pi j_2 \ell_2}{N_2} r\right] \exp\left[-i \frac{2\pi j_3 \ell_3}{N_3} r\right], \quad (11.4)$$

3 次元逆フーリエ変換は

$$F[j_1, j_2, j_3] = \sum_{\ell_1=0}^{N_1-1} \sum_{\ell_2=0}^{N_2-1} \sum_{\ell_3=0}^{N_3-1} A[\ell_1, \ell_2, \ell_3] \times \exp\left[i \frac{2\pi \ell_1 j_1}{N_1} r\right] \exp\left[i \frac{2\pi \ell_2 j_2}{N_2} r\right] \exp\left[i \frac{2\pi \ell_3 j_3}{N_3} r\right], \quad (11.5)$$

ここで、 $r = 1$ または -1 。従って、 $A[\ell_1, \ell_2, \ell_3]$ を得るためには、FFT 実行後に規格化が必要である。

SSL II シリーズの他の 3 次元実フーリエ変換サブルーチンを使用する場合は、呼び出し時に必要な変数や作業領域等が異なるが、戻り値は (11.4) または (11.5) 式と等しい。個々のサブルーチンの用法については、「SSL II 使用手引書」を参照。

SSL II/VPP のサブルーチンを使用する時は、コンパイル時にオプション `-lssl2vpp` を指定する。

```
frt -Wx -lssl2vpp program_name.f
```

FFT のプログラムは以下の手順に分けられる。

1. 変数の宣言
2. 被変換関数 $F[j_1, j_2, j_3]$ を定義
3. 前処理: 作業用配列の初期化など

4. FFT の実行。DP_V3DRCF を呼び出す
5. 後処理: 結果の規格化、スペクトルの計算、データの出力など

以上の手順を VPP Fortran で記述したサンプル副プログラム (fft_FtoA.f) を本節の末尾に添付する。この副プログラムおよび次節のサンプル副プログラムは、付属の CD-ROM に含まれるメインプログラム (main_fft.f) を実行するときに呼び出されるように設定されている。このメインプログラムは、データファイル (init.data) から被変換配列 $F[j_1, j_2, j_3]$ を読み込んで、FFT を実行・出力する。メインプログラムの主要部分は以下の通りである。

```

      INTEGER NPE
      PARAMETER (NPE=2)
      !XOCL PROCESSOR PE(NPE)
      C
      !XOCL PARALLEL REGION                                ← 並列化の開始
      CCC READ INITIAL DATA FROM A FILE
      !XOCL SPREAD REGION /PE(1)
         FN1='init.data'
         OPEN(1,FILE=FN1,FORM='UNFORMATTED')
         READ(1) F_G                                       ← 初期配列 F をファイルから読み込む
         CLOSE(UNIT=1)
      !XOCL END SPREAD
      CCC FFT F[j1,j2,j3] -> A[l1,l2,l3]
         CALL FFT_FtoA                                    ← FFT を実行 (fft_FtoA.f の呼び出し)
      CCC OUTPUT F[j1,j2,j3] & A[l1,l2,l3]
         CALL OUT_FFT                                     ← FFT 結果をファイルに出力
      CCC CALCULATE & OUTPUT ENERGY SPECTRUM
         CALL SPECTRUM                                   ← スペクトルの計算・出力 (sec. 10.2)
      !XOCL END PARALLEL                                  ← 並列化の終了
      STOP
      END

```

サンプル副プログラム (fft_FtoA.f) はメインプログラム中の並列実行領域 (!XOCL PARALLEL REGION !XOCL END PARALLEL の間) から呼び出す。

このサンプル副プログラム (fft_FtoA.f) 中にも示した通り、DP_V3DRCF は

```
CALL DP_V3DRCF(X_G,KX,NX,N1,N2,N3,ISIN,ISN,WR_G,WI_G,KW,W,IW,ICON)
```

のようにして呼び出す。フーリエ変換 ([11.4] 式) のときは $ISN = 1$ 、逆フーリエ変換 ([11.5] 式) のときは $ISN = -1$ とする。被変換配列および変換結果は配列 $X_G[1:KX, 1:N2, 1:NX]$ に格納される。 KX および NX は配列 X_G の要素数を表す変数で、整数 $N1, N2, N3$ はそれぞれ x_1 -, x_2 -, x_3 - 方向 (プログラム中では、 i -, j -, k - 方向) の格子数である。整数 $ISIN = 1$ または -1 は変換の方向を表し、(11.4), (11.5) 式の r の値と一致させる。以降は作業領域 (WR_G, WI_G, W) と、その配列の要素数 (KW, IW)、および、コンディションコード ($ICON$) である。グローバル配列 X_G, WR_G, WI_G は全て、1 次元目を均等分割する。各引数の詳細に付いては表 11.1 または使用手引書を参照。

フーリエ変換が実行されると、配列 X_G の要素 $X_G[1:N1, 1:N2, 1:N3/2+1]$ には $A[0 : N_1 - 1, 0 : N_2 - 1, 0 : N_3/2]$ の実部が、 $X_G[1:N1, 1:N2, N3/2+2:2(N3/2+1)]$ には $A[0 : N_1 - 1, 0 : N_2 - 1, 0 : N_3/2]$ の虚部が格納される。残りのモードの振幅は (11.3) 式の複素共役関係から得られる。付属 CD-ROM のサンプルプログラムでは、配列 X_G のこれらの要素を配列 A に代入している。

サンプルメインプログラム (main_fft.f) を実行したときに読み込まれる初期配列 F と、そのフーリ

表 11.1: 3次元離散実フーリエ変換サブルーチン DP_V3DRCF の引数 (「富士通 SSL II/VPP 使用手引書」より編集)

引数	型	備考
X_G[KX, N2, NX]	(倍精度) 実数型グローバル配列	入力: 被変換配列、出力: 変換結果。1次元目を均等分割
KX	整数	配列 X_G の 1次元目の大きさ。N1 以上で 2*NPE の倍数 (NPE: 並列数) 最適値: $KX = 2*NPE*[(N1+2*NPE-1)/(2*NPE)]$
NX	整数	配列 X_G の 3次元目の大きさ。 $NX \geq 2*(N3/2 + 1)$
N1	整数	変換する実データの 1次元目の大きさ。 $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
N2	整数	変換する実データの 2次元目の大きさ $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
N3	整数	変換する実データの 3次元目の大きさ $N1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
ISIN	整変数	変換の方向。ISIN = -1 または 1。(11.4)、(11.5) 式の r と一致させる
ISN	整変数	1(変換)、-1(逆変換)
WR_G[KW, N2, KX]	倍精度実数型グローバル配列	作業領域。1次元目を均等分割
WI_G[KW, N2, KX]	倍精度実数型グローバル配列	作業領域。1次元目を均等分割
KW	整数	配列 WR_G と WI_G の 1次元目の大きさ。N3/2 + 1 以上で NPE の倍数 最適値: $KW = NPE*[(N3/2 + NPE)/NPE]$
W[IW]	倍精度実数型 1次元配列	作業領域
IW	整数	配列 W の大きさ。 $IW = 2*MAX(N1, N2, N3)$
ICON	整変数	コンディションコード

エ変換の結果 A をそれぞれある断面について示したのが図 11.1である。 $A(\mathbf{k})$ は 3次元の波数空間分布を持つため、このままではどのようなスケールのゆらぎが主な成分なのかが解りにくい。そこで、次節では $A(\mathbf{k})$ から 1次元的な波数空間でのスペクトル $P(k)dk$ をとる手順を解説する。

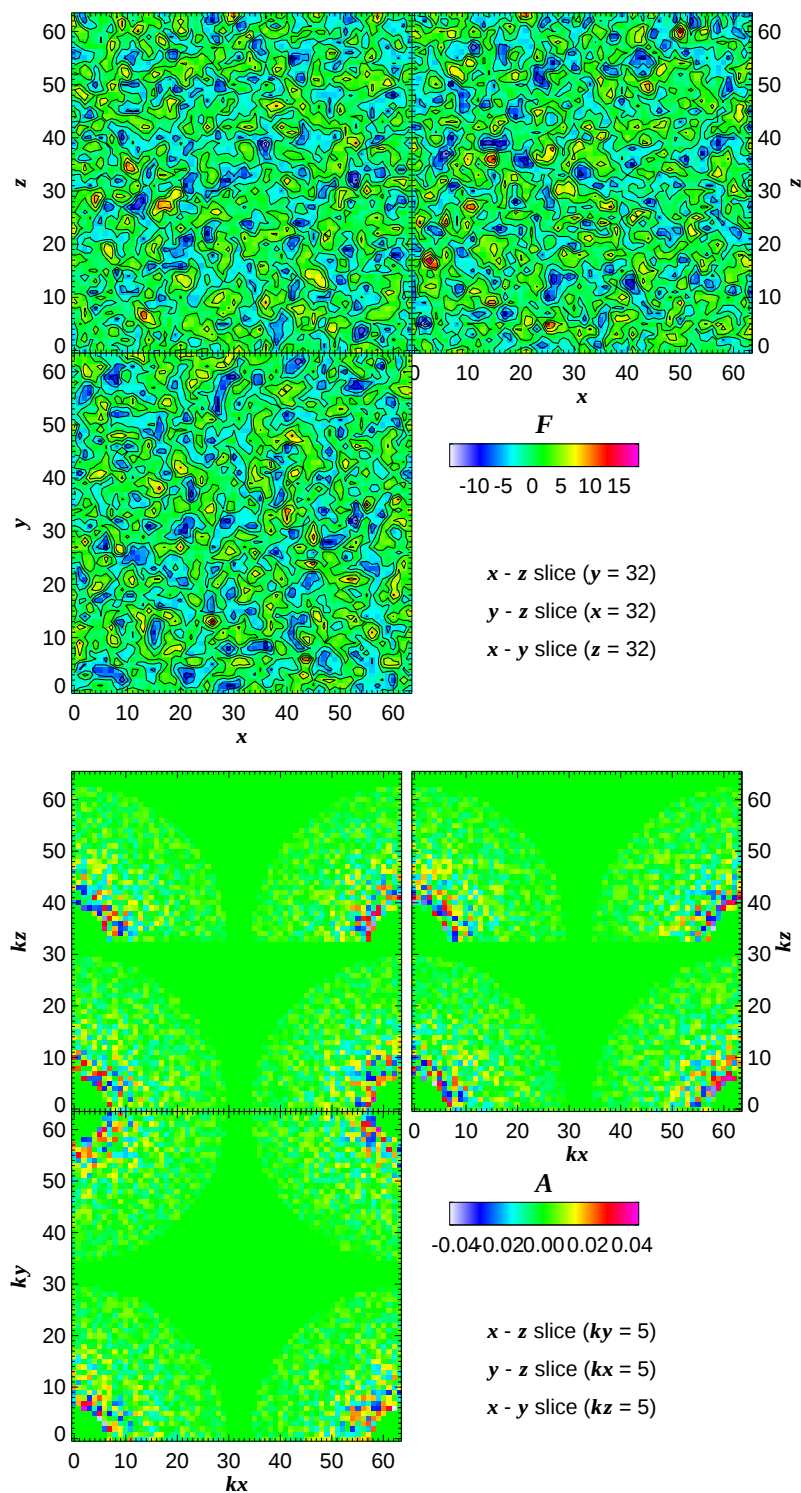


図 11.1: サンプルプログラム (main_fft.f) の被変換配列 $F[x, y, z]$ (上図) と FFT 結果 $A[kx, ky, kz]$ (下図) の断面図。

DP_V3DRCF による FFT のサンプルプログラム (fft_FtoA.f)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PROGRAM NAME: fft_FtoA.f
C WHAT CAN THIS DO?: Fast Fourier Transformation using DP_V3DRCF
C                      F[i,j,k] -> A[i,j,k].          ※このプログラムでは、非変換配列 F は既に
C                                                         与えられているとした。
C FILES TO OUTPUT: fft.data (binary)
C
C The original version by Kanako Sugimoto (2004/09/01)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
SUBROUTINE FFT_FtoA
CC --- DEFINE VARIABLES          ← (1) ここから変数、配列の宣言
      INTEGER NPE
      PARAMETER (NPE=2)
!XOCL PROCESSOR PE(NPE)
!XOCL SUBPROCESSOR PES(NPE)=PE(1:NPE)
      INTEGER nmax,N1,N2,N3
      PARAMETER (nmax=64,N1=nmax,N2=nmax,N3=nmax)
      INTEGER i,j,k
      REAL*8 normi
      REAL*8 F(N1,N2,N3),F_G(N1,N2,N3)
CC --- PARAMETERS FOR FFT
      INTEGER KX, NX, ISIN, ISN, KW, IW, ICON
      PARAMETER (IW=2*N1)
      PARAMETER (NX=2*(N3/2+1))
      PARAMETER (KX=(N1+2*NPE-1)/(2*NPE)*2*NPE)
      PARAMETER (KW=(N3/2+NPE)/NPE*NPE)
      REAL*8 A(N1,N2,NX),A_G(N1,N2,NX)
      REAL*8 X(KX,N2,NX),X_G(KX,N2,NX)
      REAL*8 W(IW), WR_G(KW,N2,KX), WI_G(KW,N2,KX)
      equivalence (F_G,F),(A_G,A),(X_G,X)
      common /func/ F_G, A_G
      common /ampx/ X_G
!XOCL INDEX PARTITION INF=(PES,INDEX=1:N3,PART=BAND)
!XOCL INDEX PARTITION INA=(PES,INDEX=1:N3+2,PART=BAND)
!XOCL INDEX PARTITION INFFT=(PES,INDEX=1:KX,PART=BAND)
!XOCL INDEX PARTITION INW=(PES,INDEX=1:KW,PART=BAND)
!XOCL LOCAL F(:, :, /INF(OVERLAP=(1,1))), A(:, :, /INA), X(/INFFT, :, :)
!XOCL GLOBAL F_G, A_G, X_G, WR_G(/INW, :, :), WI_G(/INW, :, :)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC PRE TRANSFORMATION PROCESSING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC INITIALIZE TEMPORAL ARRAYS          ← (3) 作業用配列の初期化 [推奨]
!XOCL SPREAD DO /INFFT
      DO i=1,KX
        DO k=1,NX
          DO j=1,N2
            X(i,j,k)=0.d0
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
!XOCL SPREAD DO /INW
      DO i=1,KW
        DO k=1,KX
          DO j=1,N2
            WI_G(i,j,k)=0.d0
            WR_G(i,j,k)=0.d0
          END DO
        END DO
      END DO

```

```

        END DO
    END DO
END DO
!XOCL END SPREAD DO
!XOCL SPREAD MOVE /INFFT
    DO i=1,N1
        DO k=1,N3
            DO j=1,N2
                X(i,j,k)=F_G(i,j,k)
            END DO
        END DO
    END DO
END DO
!XOCL END SPREAD(ID)
!XOCL MOVE WAIT(ID)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC CALL FFT FUNCTION
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC FOURIER TRANSFORMATION: F(i,j,k) -> A(i,j,k)
    ISN=1
CC DIRECTION FOR THE TRANSFORMATION:
CC    ISIN = 1(r_i=+1); 0(no-transformation); -1(r_i=-1)
    ISIN=1
C
    CALL
    & DP_V3DRCF(X_G,KX,NX,N1,N2,N3,ISIN,ISN,WR_G,WI_G,KW,W,IW,ICON)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC POST TRANSFORMATION PROCESSING
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC NORMALIZE AMPLITUDE
    NORMI=1.0d0/DFLOAT(N1*N2*N3)
!XOCL SPREAD DO /INFFT
    DO i=1,N1
        DO k=1,NX
            DO j=1,N2
                X(i,j,k)=X(i,j,k)*NORMI
            END DO
        END DO
    END DO
!XOCL END SPREAD
C
!XOCL SPREAD MOVE /INA
    DO k=1,NX
        DO j=1,N2
            DO i=1,N1
                A(i,j,k)=X_G(i,j,k)
            END DO
        END DO
    END DO
!XOCL END SPREAD MOVE(ID)
!XOCL MOVE WAIT(ID)
RETURN
END

```

← f[i,j,k] を入出力配列 X[i,j,k] に代入

← (4) FFT サブルーチンを CALL

← (5) ここから後処理

← 振幅の規格化

← X[i,j,k] を A[i,j,k] に代入

11.2 スペクトルの計算法

モード毎の振幅配列 $A(\mathbf{k})$ からスペクトル、

$$P(k) dk = \sum_{|\mathbf{k}|=k}^{k+\Delta k} A^2(\mathbf{k}), \quad (11.6)$$

を計算する手順は、

1. 変数の宣言と初期化: スペクトルを取る波数範囲と刻み幅などを設定する
2. スペクトルの波数目盛り $k(i)$ を定義する
3. 各モードの3次元波数の絶対値をとる: $\mathbf{k}[\ell_1, \ell_2, \ell_3] \rightarrow k[\ell_1, \ell_2, \ell_3] = |\mathbf{k}[\ell_1, \ell_2, \ell_3]|$
4. 各々の波数目盛り (ビン) ごとにモードの振幅 $A(k)$ を足し合わせる ([11.6] 式)
5. 結果の出力

である。

サンプル副プログラムを節末に添付してある (spectrum.f)。このサンプル副プログラムを基に、前節の図 11.1 の分布からスペクトルを計算した結果が図 11.2 である。サンプルプログラムで与えられた初期配列 F は、波数範囲 $8 \leq k \leq 30$ に、 $P(k)d \log k = \text{const}$ なるスペクトルをもつゆらぎ ($1/f$ ゆらぎ) であったことが解る。

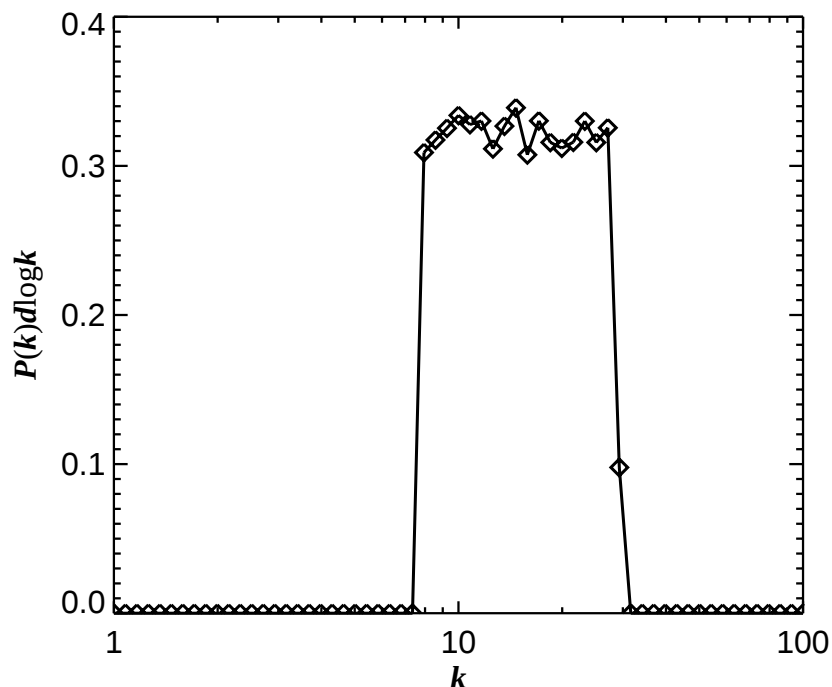


図 11.2: サンプルプログラムのペクトル。図 11.1の分布を元に計算した。

スペクトル計算のサンプルプログラム (spectrum.f)

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C PROGRAM NAME: spectrum.f
C WHAT CAN THIS DO?: CALCULATE SPECTRUM, A[i,j,k] -> A[1]^2.
C FILE TO OUTPUT: spcec.data (binary)
C
C The original version by Kanako Sugimoto (2004/09/02)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE SPECTRUM
      INTEGER NPE
      PARAMETER (NPE=2)
!XOCL PROCESSOR PE(NPE)
!XOCL SUBPROCESSOR PES(NPE)=PE(1:NPE)
      INTEGER nmax,N1,N2,N3,KX,NX
      PARAMETER (nmax=64,N1=nmax,N2=nmax,N3=nmax)
      PARAMETER (NX=2*(N3/2+1))
      PARAMETER (KX=(N1+2*NPE-1)/(2*NPE)*2*NPE)
      INTEGER i,j,k,normi
      INTEGER wnx(N1),wny(N2),wnz(NX)
      REAL*8 X(N1,N2,NX),X_G(N1,N2,NX)
      equivalence (X_G,X)
      common /nwave/ wnx,wny,wnz
      common /ampx/ X_G
!XOCL INDEX PARTITION INFFT=(PES,INDEX=1:KX,PART=BAND)
!XOCL LOCAL X(/INFFT,,:,:)
!XOCL GLOBAL X_G
CCC VARIABLES FOR SPECTRUM
      REAL*8 A2(KX,N2,NX)

```

← (1) ここから変数、配列の宣言

```

!XOCL LOCAL A2(/INFFT, :, :)
      INTEGER iskip,nbin,noffset,ntick,minp,maxp
      REAL*8 tickmin,tickmax,wnum
      PARAMETER (minp=0,maxp=2)
      PARAMETER (ntick=30,tickmin=1.0D1**minp,tickmax=1.0D1**maxp)
      PARAMETER (nbin=(maxp-minp)*ntick+1)
      PARAMETER (noffset=-minp*ntick+1)
      INTEGER L(kx,n2,n3),L_G(kx,n2,n3),nmode(nbin)
      DOUBLE PRECISION Pow(nbin),kwave(nbin)
      EQUIVALENCE (L_G,L)
      common /spec/ Pow,nmode,kwave
!XOCL GLOBAL L_G
!XOCL LOCAL L(/infft, :, :)
      CHARACTER*40 FN3
!XOCL SPREAD DO /INFFT                                ← (1) 'ここから配列の初期化
      DO I=1,KX
        DO K=1,NX
          DO J=1,N2
            A2(I,J,K)=0.0D0
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
      ISKIP=N3/2+1
!XOCL SPREAD DO /INFFT
      DO I=1,N1
        DO K=1,N3/2+1
          DO J=1,N2
            A2(I,J,K)=X(I,J,K)**2+X(I,J,K+ISKIP)**2
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
CCC INITIALIZE SPECTRUM ARRAY & DEFINE THE WAVENUMBER OF EACH BIN
      DO i=1,nbin
        nmode(i) = 0
        Pow(i) = 0.0D0
        kwave(i) = tickmin*1.0D1**(DFLOAT(i-1)/ntick) ← (2) スペクトルの波数目盛り(ビン)を定義
      END DO
CC DEFINE TICK LEVEL OF THE DENSITY FOR EACH MESH      ← (3) 各モードの波数がスペクトルの何番目の
!XOCL SPREAD DO /INFFT                                  ビンに相当するか
      DO i=1,N1
        DO k=1,N3/2+1
          DO j=1,N2
            WNUM=DSQRT(DFLOAT(wnx(i)**2+wny(j)**2+wnz(k)**2))
            L(i,j,k) =
&              IDINT(DLOG10(max(dfloating(wnum),1.D-6))*dfloating(ntick)
&              +dfloating(noffset))
            L(i,j,k) = max(L(i,j,k),1)
            L(i,j,k) = min(L(i,j,k),nbin)
          END DO
        END DO
      END DO
!XOCL END SPREAD DO
CC CALCULATE SPECTRUM
!XOCL SPREAD DO /INFFT
      DO i=1,N1
        DO k=1,N3/2+1

```

```

      DO j=1,N2
        Pow(L(i,j,k)) = Pow(L(i,j,k))+A2(i,j,k)    ← (4) ビン毎に振幅の2乗を合計
      END DO
    END DO
  END DO
!XOCL END SPREAD SUM(Pow)
!XOCL SPREAD DO /INFFT
  DO i=1,N1
    DO k=1,N3/2+1
      DO j=1,N2
        nmode(L(i,j,k)) = nmode(L(i,j,k))+1    ← (4)' ビン毎のモード数の合計
      END DO
    END DO
  END DO
!XOCL END SPREAD SUM(nmode)
CCC WRITE BINARY IN SINGLE PRECISION    ← (5) データ出力
  FN3='spec.data'
!XOCL SPREAD REGION /PES(1)
  OPEN(3,FILE=FN3,FORM='UNFORMATTED')
  write(3) nbin
  write(3) (real(kwave(i)),i=1,nbin)
  write(3) (real(Pow(i)),i=1,nbin)
  write(3) (nmode(i),i=1,nbin)
  CLOSE(UNIT=3)
!XOCL END SPREAD
  RETURN
END

```

11.3 おまけ：乱流の発生法、スペクトルの制御、非圧縮性ゆらぎの生成

§11.1に述べた通り、実 FFT サブルーチン DP_V3DRCF の呼び出し変数 $ISN = -1$ とすると、波数空間の複素振幅配列 A から実スカラー配列 F への逆フーリエ変換が出来る。(実は、サンプルプログラムの初期配列 F もこの手法で作成した。) 本節では、フーリエ逆変換によって、ランダムなスカラー場やベクトル場 (乱流場) を発生させるにあたって、筆者が必要だと感じた注意点とテクニックを述べる。

11.3.1 乱流の発生法 (逆実 FFT の注意点)

数値計算で密度揺らぎなどのランダムなスカラー場を発生させるためには、配列 $A[N_1, N_2, 2(N_3/2+1)]$ にランダムな振幅 (乱数) を与えて、逆フーリエ変換によって実空間の分布 $F[N_1, N_2, N_3]$ を計算すればよい。乱流速度場などのランダムなベクトル場は FFT サブルーチンを複数回呼んでベクトルの各方向成分を得る。このとき、 A と F の配列要素の数が異なることに注意が必要である。 $N_1 \times N_2 \times N_3$ 点の実スカラー分布 F を得るためには、振幅要素 $A[I, J, K]$ を $N_1 \times N_2 \times 2(N_3/2+1)$ 個用意する。FFT の変換結果と同様に、配列 A の要素 $A[1:N_1, 1:N_2, 1:N_3/2+1]$ には複素振幅の実部を、要素 $A[1:N_1, 1:N_2, N_3/2+2:2(N_3/2+1)]$ には複素振幅の虚部を与える。

また、変換後の配列が実数配列になるためには、複素振幅分布 A が (11.3) 式の複素共役関係を満たすように配列 A を与えなくてはならない²。SSL II による FFT サブルーチンでは、この複素共役関係を利用して配列 A の半分を実部、残りを虚部としているので、ほとんどの要素では A に任意の値を与えても複素共役関係を破らない。問題となるのは $k_3 = 1, N_3/2+1, N_3/2+2, 2(N_3/2+1)$ の面で、

$$\begin{aligned} A[I, J, 1] &= A[N_1+1-I, N_2+1-J, 1], \\ A[I, J, N_3/2+1] &= A[N_1+1-I, N_2+1-J, N_3/2+1], \\ A[I, J, N_3/2+2] &= -A[N_1+1-I, N_2+1-J, N_3/2+2], \\ A[I, J, 2(N_3/2+1)] &= -A[N_1+1-I, N_2+1-J, 2(N_3/2+1)], \end{aligned}$$

ここで、 $I, J = 1, 2, 3, \dots, N_1/2+1$ の要素である。

11.3.2 任意のパワースペクトルを持つゆらぎの与え方

一般に、振幅配列 A に定数や一様乱数を与えても、スペクトルは平坦にならない (例外は、1次元のスペクトル $P[k_1]dk_1$)。これは、図 11.3に示したように波数によってスペクトルのビン Δk あたりに含まれるモードの数が異なるためである。

図 11.3は、 $0 \leq k_1, k_2 \leq 4$ について波数幅 $\Delta k = 1.7$ でスペクトルを取ったときに、各ビンに含まれるモードを異なるマークで示している。波数域が $0 \leq k < 1.7$ のビンには4つのモードが、波数域が $1.7 \leq k < 3.4$ のビンには9つのモードが含まれることが解る。このように、波数 k が大きいほど $k \leq |k| < k + \Delta k$ の領域が波数空間に占める面積 (3D では体積) が増えるため、その内部に含まれるモードの数が大きくなる。また、数値計算では格子数が有限なので、扱える波数に上限がある。このため、波数の上限付近では境界の影響によるモード数の減少が起こる (図 11.3の 3, 4 番目の

²ただし、この複素共役関係を破るような A を与えても、数値計算上はエラーメッセージが出ない可能性がある。しかし、複素共役関係を満たさない配列 A を与えると、実行結果の実スカラー配列 F に不連続面が現われるなどの症状が出ることを筆者は確認している。また、これらの複素共役成分に与えた振幅はそれぞれフーリエ変換されて加算されるので注意が必要である。

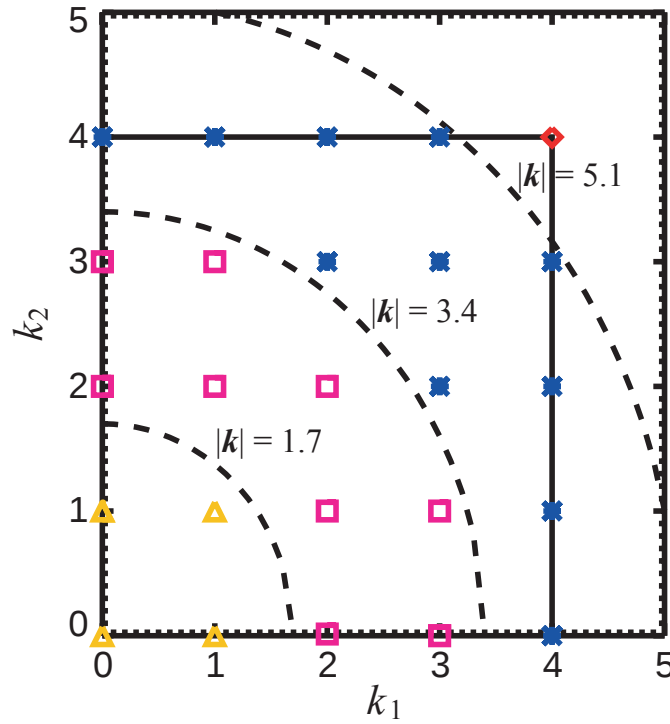


図 11.3: 波数空間分布とモードの数。波数空間 $0 \leq k_1, k_2 \leq 4$ において、 $\Delta k = 1.7$ でスペクトルを取るときの、各ビンに含まれるモードを異なる記号で記した。

ビン)。このようにビン毎のモード数が異なるとスペクトル、

$$P(k) dk = \sum_{|\mathbf{k}|=k}^{k+\Delta k} A^2(\mathbf{k}) = \langle A^2 \rangle n_{\text{mode}}[k \leq |\mathbf{k}| < k + \Delta k], \quad (11.7)$$

はモード数に比例した形になる。ここで、 n_{mode} は $k \leq |\mathbf{k}| < k + \Delta k$ に含まれるモードの数、 $\langle A^2 \rangle$ は一様乱数 (または定数) 配列 A の自乗平均である。

ビン毎のモード数の変化によるスペクトルの波数依存を回避するもっとも一般的な解決法は、波数が連続的に分布している極限での $k \leq |\mathbf{k}| < k + \Delta k$ に含まれるモードの数、即ち、波数空間の面積 (3D では体積) で振幅 A を予め割ることである。2次元波数空間では、

$$n_{\text{mode}} \propto 2\pi k dk \quad (11.8)$$

なので、(11.7) 式から、 $P(k) \propto k^p$ とするためには $A(\mathbf{k})$ を $k^{(p-1)/2}$ 倍すればよい。同様に、3次元の場合、

$$n_{\text{mode}} \propto 4\pi k^2 dk \quad (11.9)$$

なので、 $P(k) \propto k^p$ とするためには $A(\mathbf{k})$ を $k^{(p-2)/2}$ 倍すればよい。

また、サンプルプログラムのスペクトル (図 11.2) のように、 $\log k$ スケールで平坦な $1/f$ ゆらぎ ($kP[k] = \text{const}$) は $\Delta \log k$ 幅のモードの数を考えて、 $A(\mathbf{k})$ を $k^{-3/2}$ 倍すればよい (3次元の場合)。

11.3.3 非圧縮性ゆらぎの与え方

ランダムなベクトル場 $\mathbf{v}$ を初期条件として数値計算するときに、非圧縮性のゆらぎを与えたいことがある。非圧縮性のゆらぎとは、

$$\nabla \cdot \mathbf{v} = 0, \quad (11.10)$$

をみたすベクトル場のことである。圧縮性のゆらぎと非圧縮性のゆらぎを分けて考えるのは、これらが物理的に異なる性質を持つからである。また、磁場ゆらぎ $\delta \mathbf{B}$ を与えるときは、 $\nabla \cdot \delta \mathbf{B} = 0$ をみたすゆらぎを与える必要がある。

非圧縮性ゆらぎ $\mathbf{v}_s$ を与えるためには、ゆらぎのベクトルポテンシャル $\mathbf{V}$ を与えて、

$$\mathbf{v}_s = \nabla \times \mathbf{V}, \quad (11.11)$$

から速度ゆらぎを計算すればよい。これにより $\mathbf{v}_s$ は

$$\nabla \cdot \mathbf{v}_s = \nabla \cdot (\nabla \times \mathbf{V}) = 0, \quad (11.12)$$

をみたす。

プログラム手順は、

1. 振幅配列 $\mathbf{A}$ を逆フーリエ変換して、ランダムなベクトル場を生成する $\rightarrow \mathbf{F}$ 。3次元ベクトル場の場合はこの手順を3回繰り返す: $[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3] \rightarrow [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$ ($\mathbf{A}^*$ と $\mathbf{F}^*$ はそれぞれ $N_1 \times N_2 \times 2(N_3/2 + 1)$ と $N_1 \times N_2 \times N_3$ 要素の配列) 生成したベクトル場 $[\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$ をベクトルポテンシャル $\mathbf{V}$ とする, $\mathbf{V} = [\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3]$

2. $\mathbf{v}_s = \nabla \times \mathbf{V}$ によってランダムな非圧縮性ベクトル場を得る

ここで注意しなくてはならないのは、初期の $[\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3]$ が一様な乱数配列であっても、 $\mathbf{v}_s$ のフーリエ成分は k に比例する波数依存を持つてしまうことである。即ち、ベクトルポテンシャル

$$\mathbf{V} = \sum_{k_1} \sum_{k_2} \sum_{k_3} \mathbf{A} \exp[-i\mathbf{k} \cdot \mathbf{x}] \quad (11.13)$$

として、 $\mathbf{A}$ を一様乱数ベクトルとすると、 $\nabla \times \mathbf{V}$ によって与えられる非圧縮性乱流場は、

$$\mathbf{v}_s = -i \sum_{k_1} \sum_{k_2} \sum_{k_3} \mathbf{k} \times \mathbf{A} \exp[-i\mathbf{k} \cdot \mathbf{x}] \quad (11.14)$$

となり、振幅は典型的に $k = |\mathbf{k}|$ に比例する。これを防ぐためには、初期の振幅 $\mathbf{A}$ を各々の波数 k で割ってから逆FFTをして $\mathbf{V}$ を得ればよい。

同様の理屈で、逆フーリエ変換によってスカラーポテンシャル ϕ を与えて $\mathbf{v} = \nabla \phi$ とすれば、 $\nabla \times \mathbf{v} = 0$ (渦なし) のベクトル場が得られる。

11.4 富士通 SSL II の離散型実フーリエ変換サブルーチン

「富士通 SSL II 使用手引書」より抜粋、編集

表 11.2: SSL II (FORTRAN、逐次計算、ベクトル計算機用)

名前	機能	備考
RFT	1 次元、実 FFT (2 基底)	配列の要素数 $N_1 = 2^p$ (p は正整数)

表 11.3: SSL II/VP (FORTRAN、ベクトル機用)

名前	機能	備考
VRFT1	1 次元、実 FFT (性能優先型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
VRFT2	1 次元、実 FFT (メモリ節約型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
VMRF2	1 次元、多重、多次元実 FFT (混合基底)	1 次元目の要素数 N_1 は偶数。その他の要素数は任意
VMRFT	多次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 $N(1 : M)$ を持つ M 次元のデータを変換。 $N(1 : M-1)$ のうち少なくとも 1 つは偶数かつ $N_i = 2^p 3^q 5^r$ ($i = 1, 2, \dots, M; p, q, r$ は 0 又は正整数)
VRPF3	3 次元、素因子実 FFT	1 次元目の要素数 N_1 は偶数、かつ、 N_i ($i = 1, 2, 3$) は $\{2, 3, 4, 5, 7, 8, 9, 16\}$ のうち互いに素な数の積で表せる
VSRFT	1 次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 を持つ、 M 本のデータを変換。 N_1 または M の一方は偶数、かつ、 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)

表 11.4: SSL II/VPP (VPP FORTRAN 用)

名前	機能	備考
DP_V1DRFT	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 n は偶数、かつ、 $m, n/2 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DP_V1DRCF	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DP_V2DRCF	2 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2; p, q, r$ は 0 または正整数)
DP_V3DRCF	3 次元、実 FFT (2, 3 及び 5 の混合基底)	本文参照。各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2, 3; p, q, r$ は 0 または正整数)

表 11.5: SSL II/HPF (HPF, VPP 用)

名前	機能	備考
DH_V1DRFT	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 n は偶数、かつ、 $m, n/2 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DH_V1DRCF	1 次元、実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 の 1 次元配列を $N_1 = m \times n$ の 2 次元配列にして変換。 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
DH_V2DRCF	2 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2; p, q, r$ は 0 または正整数)
DH_V3DRCF	3 次元、実 FFT (2, 3 及び 5 の混合基底)	各次元方向の要素数 $N_i = 2^p 3^q 5^r$ ($i = 1, 2, 3; p, q, r$ は 0 または正整数)

表 11.6: C-SSL II (C 言語、逐次計算、ベクトル計算機用)

名前	機能	備考
c_dvmrft	多次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 $N(1 : M)$ を持つ M 次元のデータを変換。 $N(1 : M-1)$ のうち少なくとも 1 つは偶数かつ $N_i = 2^p 3^q 5^r$ ($i = 1, 2, \dots, M; p, q, r$ は 0 又は正整数)
c_dvsrft	1 次元、多重実 FFT (2, 3 及び 5 の混合基底)	要素数 N_1 を持つ、 M 本のデータを変換。 N_1 または M の一方は偶数、かつ、 $N_1 = 2^p 3^q 5^r$ (p, q, r は 0 または正整数)
c_dvrft1	1 次元、実 FFT (2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
c_dvrft2	1 次元、実 FFT (メモリ節約型、2 基底)	配列の要素数 $N_1 = 2^p$ (p は 0 または正整数)
c_dvrpf3	3 次元、素因子実フーリエ変換	1 次元目の要素数 N_1 は偶数、かつ、 N_i ($i = 1, 2, 3$) は $\{2, 3, 4, 5, 7, 8, 9, 16\}$ のうち互いに素な数の積で表せる

参考文献

- [1] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., 1992, Numerical Recipes in C [日本語版] (技術評論社), 379 (12 章: フーリエ変換)
- [2] 今村 勤, 「物理とフーリエ変換」 (岩波書店)
- [3] 小形 正男, 裳華房テキストシリーズ - 物理学「振動・波動」 (裳華房), 133 (7 章: 波束とフーリエ変換)
- [4] 富士通株式会社, 2001, 「FUJITSU C-SSL II 使用手引書」, 第 4 版 (富士通株式会社)
- [5] 富士通株式会社, 1987, 「FUJITSU SSL II 使用手引書」, 初版 (富士通株式会社)
- [6] 富士通株式会社, 1991, 「FUJITSU SSL II 拡張機能使用手引書」, 第 2 版 (富士通株式会社)
- [7] 富士通株式会社, 2001, 「FUJITSU SSL II 拡張機能使用手引書 II」, 第 5 版 (富士通株式会社)
- [8] 富士通株式会社, 2001, 「FUJITSU SSL II/VPP 使用手引書」, 第 2 版 (富士通株式会社)
- [9] 富士通株式会社, 2000, 「FUJITSU SSL II/VPP 使用手引書」, 初版 (富士通株式会社)

第 12 章

Minkowski functionals を用いた構造解析

名古屋大学 杉本 香菜子

本章では、Minkowski functionals を用いた構造解析の手法を解説する¹。Minkowski functionals は、任意のスカラー分布において、閾値を越えた点のトポロジカル (位相幾何学) な性質を表す統計量である。Minkowski functionals は宇宙の大規模構造の解析 (天文学での適用例は、Mecke et al. 1994; Schmalzing & Górski 1998、など) や、医療用画像の解析などに使われている。

筆者は、自己重力を考慮した MHD 乱流の進化を数値計算しており、形成されたガス雲の構造解析に Minkowski functionals を利用している。これによって、ある閾値 ρ_{th} 以上の密度を持つガス塊の個数や辺の長さ、表面積、体積の総和に相当する量を得られる。また、Minkowski functionals はこれらの量を座標軸の方向に分けて求められるので、形成されたガス雲がある座標軸方向に細長く伸びていて、かつ、それらの向きが揃っていると、異方性が検出できる。

Minkowski functionals の利点は、

- 任意の密度分布 (正確には任意のスカラー場) に対して適用できる
- 離散データの場合の計算が簡単
- 1 次元、2 次元、3 次元のいずれにも使える
- 数学的な性質が調べられている
 - 座標の取り方に依らない
 - 位相 (トポロジー) を定量化できる
 - 閾値を変えることにより、構造の特徴を知ることができる

ことである。

12.1 定義と計算法

本節では、Minkowski functionals の定義とその計算法をそれぞれ 1, 2, 3 次元のスカラー分布 ρ について解説する。3 次元の Minkowski functionals は genus, $\mathcal{G}$, curvature, $\mathcal{C}$, surface, $\mathcal{S}$, volume,

¹本章の解説は本サマースクールの実行委員の一人であり、筆者の共同研究者でもある花輪 知幸氏 (千葉大学 先進科学教育センター) が 2004 年 4 月に行ったセミナーのレジメを非常に参考にさせていただきました。

$\mathcal{V}$ の 4 つの統計量からなる。より低い次元の場合は、これらの内の一部の統計量からなる。スカラー分布 ρ の次元と Minkowski functionals の種類を表 12.1 にまとめた。

表 12.1: スカラー分布 ρ の次元と Minkowski functionals

	1 次元	2 次元	3 次元
	ρ_i	$\rho_{i,j}$	$\rho_{i,j,k}$
個数	$\mathcal{G}$	$\mathcal{G}$	$\mathcal{G}$
長さ	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{C}$
面積	—	$\mathcal{S}$	$\mathcal{S}$
体積	—	—	$\mathcal{V}$

以下に計算法の説明をするにあたり、スカラー分布の閾値を ρ_{th} とする。また、閾値で二値化した分布 n を使用する。 n は各点で、

$$n = \begin{cases} 1 & (\rho \geq \rho_{\text{th}}) \\ 0 & (\rho < \rho_{\text{th}}) \end{cases}, \quad (12.1)$$

である。

12.1.1 1 次元の Minkowski functionals ($\mathcal{G}$, $\mathcal{C}$)

スカラーデータが 1 次元の配列 ρ_i で与えられている場合、

$\mathcal{C}(\rho_{\text{th}})$: $\rho_i \geq \rho_{\text{th}}$ であるセルの個数

$\mathcal{G}(\rho_{\text{th}})$: $\rho_i \geq \rho_{\text{th}}$ であるセルが連続した領域の個数。

例 1: $\square$ ($=\rho_i < \rho_{\text{th}}$ のセル) と $\blacksquare$ ($=\rho_i \geq \rho_{\text{th}}$ のセル) が次のように並んでいる場合、

$\square \blacksquare \blacksquare \square \square \blacksquare \blacksquare \blacksquare \square \blacksquare \square \square \square \blacksquare \blacksquare \square$

$$\mathcal{C}(\rho_{\text{th}}) = 9, \quad (12.2)$$

$$\mathcal{G}(\rho_{\text{th}}) = 4. \quad (12.3)$$

計算法

$$\mathcal{C}(\rho_{\text{th}}) = \mathcal{N}_0, \quad (12.4)$$

$$\mathcal{G}(\rho_{\text{th}}) = \mathcal{N}_0 - \mathcal{N}_1, \quad (12.5)$$

記号	図解	数式
$\mathcal{N}_0$	$\blacksquare$ の数	$\sum_i n_i$
$\mathcal{N}_1$	$\blacksquare \blacksquare$ の数	$\sum_i n_i n_{i+1}$
または、 $\min(n_i, n_{i+1})$.		

ここで、

$$n_i = \begin{cases} 1 & (\rho_i \geq \rho_{\text{th}}) \\ 0 & (\rho_i < \rho_{\text{th}}) \end{cases} . \quad (12.6)$$

例1の場合は、 $\mathcal{N}_0 = 9, \mathcal{N}_1 = 5$ になる。

12.1.2 2次元の Minkowski functionals ($\mathcal{G}, \mathcal{C}, \mathcal{S}$)

スカラーデータが2次元の配列 $\rho_{i,j}$ で与えられている場合、

$\mathcal{S}(\rho_{\text{th}})$: $\rho_{i,j} \geq \rho_{\text{th}}$ であるセルの個数 (面積)

$\mathcal{C}(\rho_{\text{th}})$: $\rho_{i,j} \geq \rho_{\text{th}}$ であるセルの領域を囲む辺の長さの半分

$\mathcal{G}(\rho_{\text{th}})$: $\rho_i \geq \rho_{\text{th}}$ であるセルが連続した領域の個数から、■に囲まれた□のセルが連続して存在する領域 (穴) の個数を引いた数。

例2: □ ($=\rho_{i,j} < \rho_{\text{th}}$ のセル) と ■ ($=\rho_{i,j} \geq \rho_{\text{th}}$ のセル) が次のように並んでいる場合。

```

□□□□□□□□□□□□□□
□■□□■□□□□■□□□□
□■□□■□□□□■□□□□
□□□□■□□□□■□□□□
□□□□□□□□□□□□□□

```

$$\mathcal{S}(\rho_{\text{th}}) = 22, \quad (12.7)$$

$$\mathcal{C}(\rho_{\text{th}}) = 17, \quad (12.8)$$

$$\mathcal{G}(\rho_{\text{th}}) = 3. \quad (12.9)$$

例2: □ ($=\rho_{i,j} < \rho_{\text{th}}$ のセル) と ■ ($=\rho_{i,j} \geq \rho_{\text{th}}$ のセル) が次のように並んでいる場合。

```

□□□□□□□□□□□□□□
□■□□■□■□□□■□□□
□■□□■□■□□□■□□□
□□□□■□□□□■□□□□
□□□□□□□□□□□□□□

```

この例では右端の連結成分に穴が空いている。

$$\mathcal{S}(\rho_{\text{th}}) = 19, \quad (12.10)$$

$$\mathcal{C}(\rho_{\text{th}}) = 21, \quad (12.11)$$

$$\mathcal{G}(\rho_{\text{th}}) = 2. \quad (12.12)$$

計算法

$$\mathcal{S}(\rho_{\text{th}}) = \mathcal{N}_0, \quad (12.13)$$

$$\mathcal{C}(\rho_{\text{th}}) = 2\mathcal{N}_0 - \mathcal{N}_1, \quad (12.14)$$

$$\mathcal{G}(\rho_{\text{th}}) = \mathcal{N}_0 - \mathcal{N}_1 + \mathcal{N}_2, \quad (12.15)$$

記号	図解	数式
$\mathcal{N}_0$	■ の数	$\sum_i \sum_j n_{i,j}$
$\mathcal{N}_1$	■■ と ■ の数	$\sum_i \sum_j (n_{i,j} n_{i+1,j} + n_{i,j} n_{i,j+1})$
$\mathcal{N}_2$	■■■ の数	$\sum_i \sum_j n_{i,j} n_{i+1,j} n_{i,j+1} n_{i+1,j+1}$

ここで、

$$n_{i,j} = \begin{cases} 1 & (\rho_{i,j} \geq \rho_{\text{th}}) \\ 0 & (\rho_{i,j} < \rho_{\text{th}}) \end{cases} . \quad (12.16)$$

12.1.3 3 次元の Minkowski functionals ($\mathcal{G}$, $\mathcal{C}$, $\mathcal{S}$, $\mathcal{V}$)

データが 3 次元の配列 $\rho_{i,j,k}$ で与えられている場合。

$\mathcal{V}(\rho_{\text{th}})$: $\rho_{i,j,k} \geq \rho_{\text{th}}$ であるセルの個数 (体積)

$\mathcal{S}(\rho_{\text{th}})$: $\rho_{i,j,k} \geq \rho_{\text{th}}$ であるセルを覆う表面積の半分

$\mathcal{C}(\rho_{\text{th}})$: $\rho_{i,j,k} \geq \rho_{\text{th}}$ であるセルが連続した領域の凸辺の長さの和から凹辺の長さの和を引いた数
 $\times \frac{1}{4}$ (図 12.1 参照)

$\mathcal{G}(\rho_{\text{th}})$: $\rho_{i,j,k} \geq \rho_{\text{th}}$ であるセルが連続した領域の数と $\rho_{i,j,k} \geq \rho_{\text{th}}$ のセルに囲まれた空洞 (cavity) の数の和から、貫通した穴 (ハンドル) の数を引いたもの

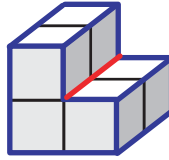


図 12.1: 3 次元の $\mathcal{C}$ を増加させる凸辺(青) と減少させる凹辺(赤)

計算法

$$\mathcal{V}(\rho_{\text{th}}) = \mathcal{N}_0, \quad (12.17)$$

$$\mathcal{S}(\rho_{\text{th}}) = 3\mathcal{N}_0 - \mathcal{N}_1, \quad (12.18)$$

$$\mathcal{C}(\rho_{\text{th}}) = 3\mathcal{N}_0 - 2\mathcal{N}_1 + \mathcal{N}_2, \quad (12.19)$$

$$\mathcal{G}(\rho_{\text{th}}) = \mathcal{N}_0 - \mathcal{N}_1 + \mathcal{N}_2 - \mathcal{N}_3. \quad (12.20)$$

$$\mathcal{N}_0(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} \quad (12.21)$$

$$\mathcal{N}_1(\rho_{\text{th}}) = \sum_i \sum_j \sum_k (n_{i,j,k} n_{i+1,j,k} + n_{i,j,k} n_{i,j+1,k} + n_{i,j,k} n_{i,j,k+1})$$

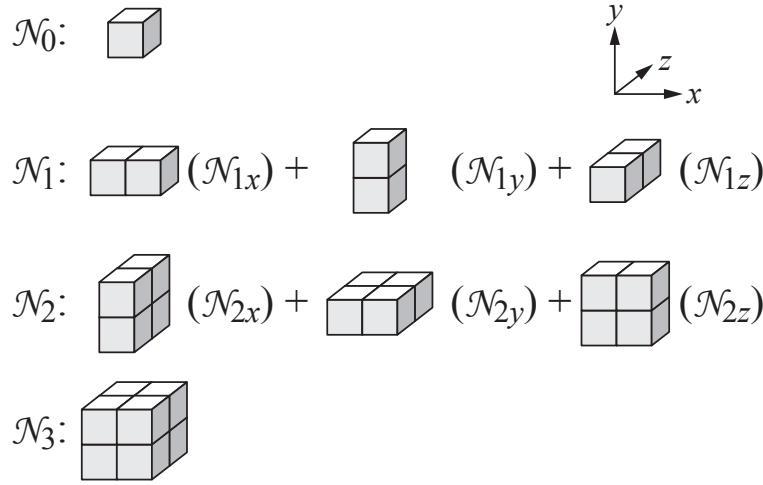


図 12.2: 3 次元の Minkowski functionals における、 $\mathcal{N}_i$ の定義 ($i = 0, 1, 2, 3$)。立方体は $\rho \geq \rho_{\text{th}}$ のセルを表す。

$$\mathcal{N}_2(\rho_{\text{th}}) = \sum_i \sum_j \sum_k (n_{i,j,k} n_{i,j,k+1} + n_{i,j,k} n_{i,j+1,k} n_{i,j+1,k+1} + n_{i,j,k} n_{i,j,k+1} n_{i,j+1,k+1} + n_{i,j,k} n_{i,j,k+1} n_{i+1,j,k} n_{i+1,j,k+1}) , \quad (12.22)$$

$$\mathcal{N}_3(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i+1,j,k} n_{i,j+1,k} n_{i+1,j+1,k} \times n_{i,j,k+1} n_{i+1,j,k+1} n_{i,j+1,k+1} n_{i+1,j+1,k+1} . \quad (12.23)$$

$$\mathcal{N}_3(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i+1,j,k} n_{i,j+1,k} n_{i+1,j+1,k} \times n_{i,j,k+1} n_{i+1,j,k+1} n_{i,j+1,k+1} n_{i+1,j+1,k+1} . \quad (12.24)$$

ここで、

$$n_{i,j,k} = \begin{cases} 1 & (\rho_{i,j,k} \geq \rho_{\text{th}}) \\ 0 & (\rho_{i,j,k} < \rho_{\text{th}}) \end{cases} . \quad (12.25)$$

12.2 閾値 ρ_{th} による Minkowski functionals の変化

閾値 ρ_{th} を変化させたときの Minkowski functionals の振舞いを図 12.3 に示した。図 12.3 の下図は、磁気乱流の減衰のシミュレーションによって得られた密度分布 (図 12.3 の上図) を基に、周期境界条件で計算した Minkowski functionals である。

考えている領域が $N_x \times N_y \times N_z$ 個のセルに分割されているとき。

- 閾値が密度の最大値より大きい場合 ($\rho_{\text{max}} < \rho_{\text{th}}$)、

$$1 \text{ 次元 } \mathcal{C} = \mathcal{G} = 0$$

$$2 \text{ 次元 } \mathcal{S} = \mathcal{G} = 0$$

$$3 \text{ 次元 } \mathcal{V} = \mathcal{S} = \mathcal{C} = \mathcal{G} = 0$$

- 閾値が密度の最小値より小さい場合 ($\rho_{\text{min}} > \rho_{\text{th}}$)

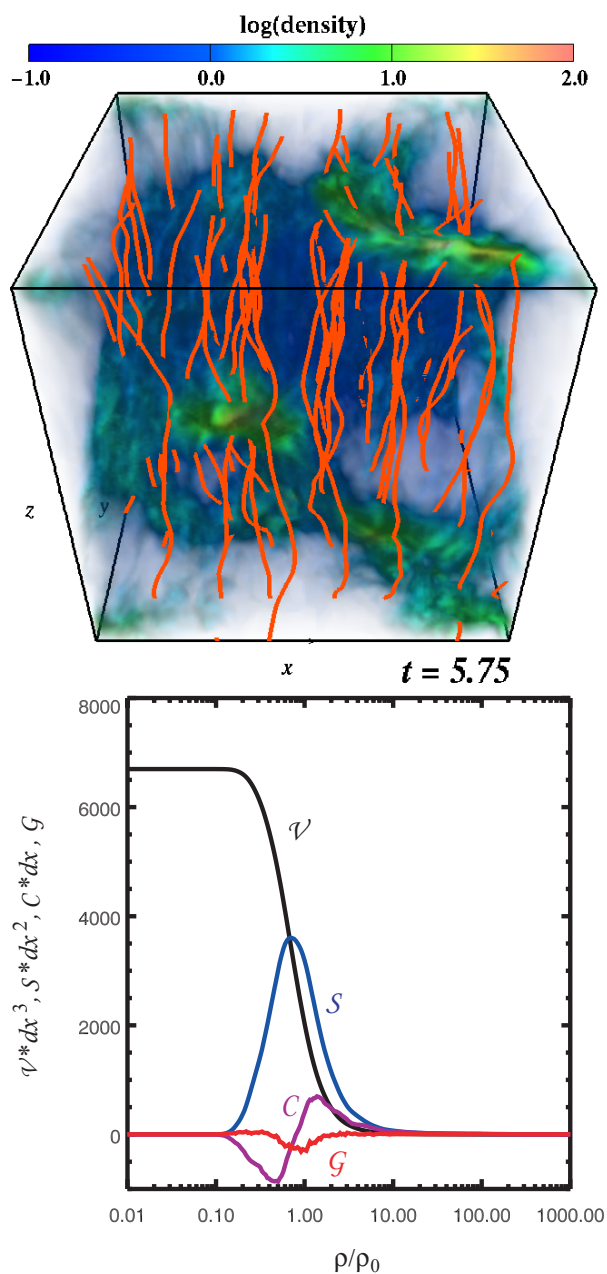


図 12.3: 磁気乱流の減衰の 3 次元シミュレーションの密度分布 (上図) と、これを基に計算した Minkowski functionals (下図)。上図: 密度分布 (対数スケール) を鳥瞰図として 3 次元的に表している。橙色の曲線は磁力線の例である。下図: 数値計算のメッシュ幅 dx の巾乗で規格化した Minkowski functionals を密度の閾値 (ρ) の関数としてプロットしている。周期境界条件によって Minkowski functionals を計算している。

1 次元 $\mathcal{C} = N_x$, $\mathcal{G} = 0$ (周期境界)、1 (孤立境界)

2 次元 $\mathcal{S} = N_x N_y$,

$\mathcal{C} = \mathcal{G} = 0$ (周期境界); $\mathcal{C} = N_x + N_y$, $\mathcal{G} = 1$ (孤立境界)

3 次元 $\mathcal{V} = N_x N_y N_z$,

$\mathcal{S} = \mathcal{C} = \mathcal{G} = 0$ (周期境界);

$\mathcal{S} = N_x N_y + N_y N_z + N_z N_x$, $\mathcal{C} = N_x + N_y + N_z$, $\mathcal{G} = 1$ (孤立境界)

- $\rho_{\min} \leq \rho_{\text{th}} \leq \rho_{\max}$ の場合

閾値 ρ_{th} を $\rho_{\min}$ から $\rho_{\max}$ まで増加させたときの Minkowski functionals の振る舞い

次元	1 次元	2 次元	3 次元
$\mathcal{G}$	増大して減少 (常に正)	極小 (−) と極大 (+)	極大 (+)- 極小 (−)- 極大 (+)
$\mathcal{C}$	単調減少 (常に正)	増大して減少 (常に正)	極小 (−) と極大 (+)
$\mathcal{S}$	—	単調減少 (常に正)	増大して減少 (常に正)
$\mathcal{V}$	—	—	単調減少 (常に正)

12.3 異方性の検出

スカラー分布 ρ が 2 次元および 3 次元の場合、 $\mathcal{N}_1$ と $\mathcal{N}_2$ を座標軸方向に分けて計算することにより、異方性を測ることが出来る。

2 次元の場合

$$\mathcal{N}_1(\rho_{\text{th}}) = \mathcal{N}_{1x}(\rho_{\text{th}}) + \mathcal{N}_{1y}(\rho_{\text{th}}), \quad (12.26)$$

$$\mathcal{N}_{1x}(\rho_{\text{th}}) = \sum_i \sum_j n_{i,j} n_{i+1,j}, \quad (12.27)$$

$$\mathcal{N}_{1y}(\rho_{\text{th}}) = \sum_i \sum_j n_{i,j} n_{i,j+1}. \quad (12.28)$$

x -, y - 方向に分けた $\mathcal{C}$ は、

$$\mathcal{C}(\rho_{\text{th}}) = \mathcal{C}_x(\rho_{\text{th}}) + \mathcal{C}_y(\rho_{\text{th}}), \quad (12.29)$$

$$\mathcal{C}_x(\rho_{\text{th}}) = \mathcal{N}_0(\rho_{\text{th}}) - \mathcal{N}_{1y}(\rho_{\text{th}}), \quad (12.30)$$

$$\mathcal{C}_y(\rho_{\text{th}}) = \mathcal{N}_0(\rho_{\text{th}}) - \mathcal{N}_{1x}(\rho_{\text{th}}). \quad (12.31)$$

3 次元の場合

$$\mathcal{N}_1(\rho_{\text{th}}) = \mathcal{N}_{1x}(\rho_{\text{th}}) + \mathcal{N}_{1y}(\rho_{\text{th}}) + \mathcal{N}_{1z}(\rho_{\text{th}}), \quad (12.32)$$

$$\mathcal{N}_{1x}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i+1,j,k}, \quad (12.33)$$

$$\mathcal{N}_{1y}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i,j+1,k}, \quad (12.34)$$

$$\mathcal{N}_{1z}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i,j,k+1}, \quad (12.35)$$

$$\mathcal{N}_2(\rho_{\text{th}}) = \mathcal{N}_{2x}(\rho_{\text{th}}) + \mathcal{N}_{2y}(\rho_{\text{th}}) + \mathcal{N}_{2z}(\rho_{\text{th}}), \quad (12.36)$$

$$\mathcal{N}_{2x}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i,j+1,k} n_{i,j,k+1} n_{i,j+1,k+1}, \quad (12.37)$$

$$\mathcal{N}_{2y}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i+1,j,k} n_{i,j,k+1} n_{i+1,j,k+1}, \quad (12.38)$$

$$\mathcal{N}_{2z}(\rho_{\text{th}}) = \sum_i \sum_j \sum_k n_{i,j,k} n_{i+1,j,k} n_{i,j+1,k} n_{i+1,j+1,k}. \quad (12.39)$$

x -, y -, z - 方向に分けた $\mathcal{S}$ は、

$$\mathcal{S} = \mathcal{S}_x + \mathcal{S}_y + \mathcal{S}_z, \quad (12.40)$$

$$\mathcal{S}_x = \mathcal{N}_0 - \mathcal{N}_{1x}, \quad (12.41)$$

$$\mathcal{S}_y = \mathcal{N}_0 - \mathcal{N}_{1y}, \quad (12.42)$$

$$\mathcal{S}_z = \mathcal{N}_0 - \mathcal{N}_{1z}. \quad (12.43)$$

x -, y -, z - 方向に分けた $\mathcal{C}$ は、

$$\mathcal{C} = \mathcal{C}_x + \mathcal{C}_y + \mathcal{C}_z, \quad (12.44)$$

$$\mathcal{C}_x = \mathcal{N}_0 - (\mathcal{N}_{1y} + \mathcal{N}_{1z}) + \mathcal{N}_{2x}, \quad (12.45)$$

$$\mathcal{C}_y = \mathcal{N}_0 - (\mathcal{N}_{1x} + \mathcal{N}_{1z}) + \mathcal{N}_{2y}, \quad (12.46)$$

$$\mathcal{C}_z = \mathcal{N}_0 - (\mathcal{N}_{1x} + \mathcal{N}_{1y}) + \mathcal{N}_{2z}. \quad (12.47)$$

図 12.4は図 12.3の上図の密度分布を基に、(12.41) - (12.47) 式によって計算した、 $\mathcal{C}$ と $\mathcal{S}$ の座標軸方向に分けた成分をプロットしたものである。高密度領域 ($10 \leq \rho \leq 300$) では $\mathcal{C}_z < \mathcal{C}_x, \mathcal{C}_y$ および $\mathcal{S}_z > \mathcal{S}_x, \mathcal{S}_y$ となっている。これは、平均的な磁場の向き (z - 軸方向) に対して垂直な方向に伸びた構造であることを示している。

12.4 閾値 ρ_{th} を変化させた計算

閾値 ρ_{th} を固定したとき、任意の $N_x \times N_y \times N_z$ 個のセルの Minkowski functionals は $\mathcal{G}, \mathcal{C}, \mathcal{S}, \mathcal{V}$ の 4 つの整数である。このように、Minkowski functionals はデータを非常に効率的に圧縮するので、閾値 ρ_{th} を変えながら計算してもデータの容量はあまり問題にならない。閾値を変化させながら計算する際に、以下のような手法を取ると、 ρ_{th} を変化させる毎に各点の密度 ρ と ρ_{th} の大小を比較して Minkowski functionals を計算するよりも計算時間が大幅に短縮される。

1 次元密度分布 ρ_i ($i = 1, 2, \dots, N$) を例にとると、閾値を

$$a_m = a_0 + b m \quad 0 \leq M, \quad (12.48)$$

と変化させる場合、計算手続きは以下の様になる。プログラム例は下に示す。

1. 閾値目盛り (ビン) の設定
2. 各点の密度 ρ_i がどのビンに当たるかを計算 (プログラム中の `bin[0:N]`)
3. ビン毎に分けて $\mathcal{N}_0, \mathcal{N}_1$ を計算

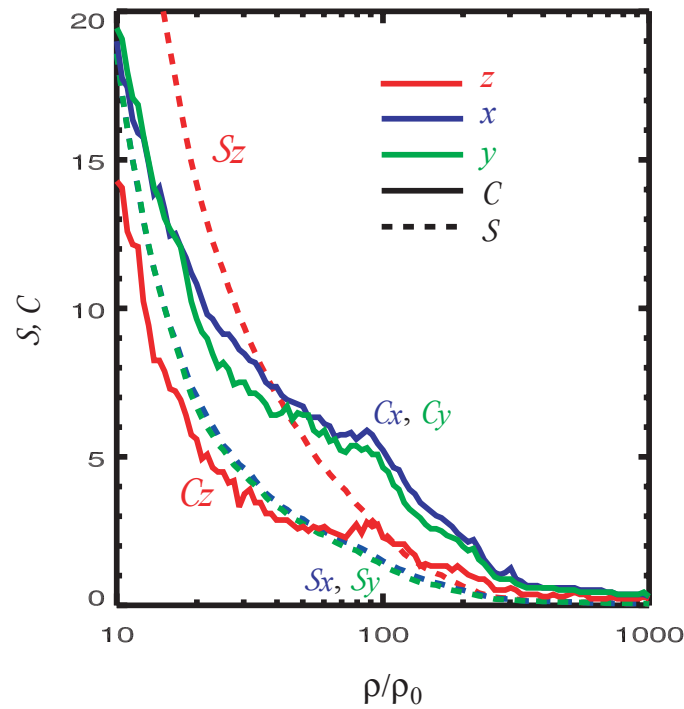


図 12.4: 座標軸の方向に分けた Minkowski functionals, $\mathcal{C}$, $\mathcal{S}$, の高密度部分。図 12.3 の密度分布を基に Minkowski functionals を計算した。 z - 方向が平均磁場の向きを表す。

$\mathcal{N}_0$: ビン毎の $\mathcal{N}_0$ を意味する $\text{dN0}(1:M)$ の、 ρ_i が対応するビンに 1 加える

$\mathcal{N}_1$: ρ_i と ρ_{i+1} のうち小さい方の密度が対応するビンの dN1 に 1 を加える

4. 各ビンの $\mathcal{N}_0$, $\mathcal{N}_1$ は自分自身のビンより高い閾値を持つビンを全て足し合わせた値になる

この手法は、より高い閾値を越える点はより低い閾値のビンでも必ず加算されるという特徴を利用して $\mathcal{N}_0$, $\mathcal{N}_1$ を高速に計算している。 $\mathcal{N}_2$, $\mathcal{N}_3$ も同様に高速化できる。

閾値を変化させた Minkowski functionals の高速計算プログラム例 (1 次元の場合)

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      M, N, rho, a0, b は既に与えられていると仮定する。
C
C      Modification history:
C 2004/04/      Tomoyuki Hanawa (original version: N0)
C 2004/09/09   Kanako Sugimoto (extended for N1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      dimension rho(N),bin(0:N),N0(0:M),dN0(0:M),N1(0:M),dN1(0:M)
C
      do j=0,M                                ← 値の初期化
        N0(j)=0
        dN0(j)=0
        N1(j)=0
        dN1(j)=0
      end do
C
      do i=1,N                                ← (2) 各点がどのビンに入るかを計算
        em=(rho(i)-a0)/b
        bin(i)=em
        bin(i)=min(bin(i),M)
        bin(i)=max(0,bin(i))
        dN0(bin(i))=dN0(bin(i))+1            ← (3) ビン毎のN0を計算
      enddo
      do i=1,N-1
        j=min(bin(i),bin(i+1))                ← (3) ビン毎のN1を計算
        dN1(j)=dN1(j)+1
      enddo
      N0(M)=dN0(M)
      N1(M)=dN1(M)
      do j=M-1,1,-1
        N0(j)=N0(j+1)+dN0(j)                  ← (4) N0を計算
        N1(j)=N1(j+1)+dN1(j)                  ← (4) N1を計算
      enddo
```

参考文献

- [1] Mecke, K. R., Buchert, T., & Wagner, H., 1994, A&A, 288, 697
- [2] Schmalzing, J., & Górski, K. M., 1998, MNRAS, 297, 355
- [3] 鳥脇 純一郎, 1988, 「画像理解のためのデジタル画像処理 [II]」 (昭晃堂)
- [4] 鳥脇 純一郎, 2002, 「3次元デジタル画像処理」 (昭晃堂)

索引

- #AVS, 202
- 1-D Newton's method, 119
- 1-D Newton's method, 123, 128
- 2-D Newton's method, 119
- 2-D Newton's method, 121

- AGN, 111
- Alfven Wave, 130
- ARROW, 172
- Arrow1, 214, 217
- AVS/Express, 196, 201
- Axis3D, 204

- BEGINP, 179
- Blast Wave, 135
- bounds, 204

- C-SSL II, 270
- CANS, 12, 130
- Cardano's method, 125, 126
- combine vect, 213
- Comoving frame, 104
- coord, 203
- crop, 204, 206, 213, 235
- curvature, $\mathcal{C}$, 272–275

- data, 202
- data math, 206, 209
- DECOMPOSED, 161
- DEVICE, 161
- Device font system, 173
- dim1, dim2, dim3, 202
- directional splitting method, 98, 110
- downsize, 214
- DP_V3DRCF (SSL II), 256–258
- energy-at-infinity, 113

- EOT, 231

- fft_FtoA.f, 257, 260
- FIDO, 103
- Fiducial observer, 103
- field, 202
- file, 202
- FINDFILE, 187
- FPlane, 213
- frame-dragging effect, 108

- genius, $\mathcal{G}$, 272–275
- glyph, 214
- GRB, 101
- GRMHD, 101, 111, 115

- HLL Flux, 129
- HLL scheme, 128

- IDL_DIR, 161, 191
- IDL_PATH, 160, 191
- IDL_STARTUP, 161
- init.data, 257
- isosurface, 204
- isovolume, 206
- iteration, 121, 137

- label, 202
- Laboratory frame, 103
- lapse function, 104, 108
- Lax-wendroff scheme, 110, 128
- LegendHoriz, 217
- LegendVert, 217
- Loop, 232–235

- main_fft.f, 257, 259
- MHD shock-tube, 133

- Minkowski functionals, 272
- ndim, 202
- netCDF, 228
- Newton's method, 124
- Newton-Raphson iteration method, 110, 115
- nspace, 202
- Orthoslice, 209, 214, 235, 237
- OutputField, 224, 232, 234, 237
- PS1, 183
- Rd netCDF Fld, 228
- Read Field, 203, 230, 232
- RETAIN, 162
- RHD, 101
- RMHD, 101
- Schwarzschild radius, 108
- set minmax, 206
- shift velocity, 104, 108, 109
- simplified TVD method, 109
- SIZE, 187
- skip, 202
- spectrum.f, 262, 263
- SRMHD, 101, 105, 110
- SSL II, 255, 268
- SSL II/VP, 269
- SSL II/VPP, 256, 269, 270
- streamlines, 213
- stride, 203
- STRING, 172
- String, 232, 233, 235
- STRMID, 187
- surface, $\mathcal{S}$, 272, 274, 275
- Symplectic Integrator, 148
- TextString3D, 220
- TextTitle, 220
- timestep, 231
- True Type font system, 173
- TRUE_COLOR, 162
- TVD, 109
- TVLCT, 188
- TVRD, 188
- Uviewer3D, 203, 204, 220
- variable, 202
- veclen, 202
- volume, $\mathcal{V}$, 273, 275
- VPP Fortran, 256, 257
- Write Image, 224, 234, 237
- WRITE_PNG, 182, 188
- XYOUTS, 172
- IDL, 18
- EPS ファイルの出力, 183
- EPS ファイルの大きさ, 183
- EPS ファイルの名前, 183
- 一般相対論的磁気流体力学, 101
- 一般相対論的流体力学, 116
- 異方性, 272, 278
- インスタンス, 198, 201
- 上付添字, 173
- 渦なし, 268
- 宇宙ジェット, 101, 111, 115
- 運動量エネルギーテンソル, 120
- 運動量エネルギー保存の法則, 119
- X 線連星, 101
- エネルギー運動量保存, 102
- $1/f$ ゆらぎ, 262, 267
- MHD ペンローズ過程, 113
- MOC 法, 71, 80
- エルゴ領域, 108, 111, 113
- オームの法則, 102
- 凹辺, 275
- オブジェクトエディター, 233
- Kerr-Schild 座標, 115
- カーブラックホール, 101, 108, 111
- 回転パラメータ, 108
- 活動銀河核, 101
- カラーバーの位置と向き, 179
- カラーコントラスト, 206, 209
- カラーテーブル, 189

- カラーテーブル, 158
- カラーバー, 217, 220
- 環境設定プログラム, 158, 161
- ガンマ線バースト, 101, 115
- 画像の出力, 224
- 基数, 256
- 基礎方程式, 102
- 基底, 256
- 基本量, 121, 128
- 逆フーリエ変換, 255–257, 266, 268
- 境界, 204
- 共動座標系, 104
- 近似 Riemann 解法, 46
- クーラン条件, 130
- 計量, 102, 103, 108
- ケプラー速度, 111
- 高速フーリエ変換, 255
- 降着円盤, 101, 111
- 固有値, 128
- 固有ベクトル, 128
- コロナ, 111
- コントラストの色, 169
- コントラストレベル, 168
- コントロールパネル, 198
- コンパイルオプション (SSL II/VPP), 256
- 最大値、最小値, 206
- 差分方程式, 30
- 三次方程式, 124, 125
- 3+1 形式, 103, 106
- サンプルデータ, 157, 192
- 座標軸, 203, 204
- GRMHD 方程式, 102
- CIP-CSL2 法, 91
- CIP 法, 71, 72
- 磁気流体力学, 46
- 時空の引きずり効果, 108, 113
- 事象の地平面, 108
- 下付添字, 172
- 実験室系, 103
- 実フーリエ変換, 255
- CT 法, 71, 79
- 修正 Lax-Wendroff, 22
- 出力ポート, 199
- シュバルツシルト時空, 109
- シュバルツシルト半径, 108
- 衝撃波, 82
- シングルウィンドウ, 199
- 磁気流体, 123
- 磁気流体力学, 29
- 人工粘性, 36, 82, 130
- 数式演算処理, 209
- スカラー分布, 255, 272
- スカラーポテンシャル, 268
- スカラーデータの可視化, 204
- スタガード格子, 71
- スペクトル, 255, 258, 262
- 線の太さ, 173
- 相対論的磁気流体力学, 101
- 相対論的磁気流体, 119
- 相対論的磁気流体力学方程式, 119
- 相対論的流体力学, 101
- タイトル, 220
- タイトルの指定, 172
- 中性子星形成, 46
- 超新星爆発, 46
- ツールバー, 198
- デバイスの設定, 161
- 等値ボリュームの表示, 206
- 等値面, 204, 206
- 特殊相対論, 120
- 特殊相対論的磁気流体力学, 101
- 特性曲線, 80
- 特性方程式, 80
- 凸辺, 275
- 動画, 232, 234, 235
- 入力ポート, 199
- ネットワークエディタ, 198
- パスの設定, 160
- バルサー, 46
- パワースペクトル, 266
- 反復解法, 121
- 反復法, 123, 137

- 非圧縮性ゆらぎ, 268
- PNG ファイルの出力, 181, 185
- PNG ファイルの大きさ, 183
- PNG ファイルの名前, 182
- 引数, 257, 258
- ビューワーウインドウ, 198
- フーリエ解析, 255
- フーリエ変換, 255-257
- field データファイル, 201, 203
- 複素共役関係, 255, 257, 266
- 複素振幅分布, 255
- ブラックホール, 101, 111, 115
- プロットシステム変数, 174
- 平均自由行程, 82
- ベクトル形式, 107
- ベクトルポテンシャル, 268
- ペンローズ過程, 113
- ベクトルデータの可視化, 213
- ベクトルデータの作成, 213
- Boyer-Lindquist 座標, 103, 108, 111
- 保存量, 120, 124
- マイクロクエーサー, 101
- Maxwell 方程式, 119
- マックスウェル方程式, 102
- 密度コントラ, 209
- ミンコフスキー時空, 104, 108
- ムービー用の作画, 185
- ムービーの作成, 188
- 文字の大きさ, 173
- 文字の種類, 172
- モジュール, 196, 198, 199
- モジュールごとの設定, 220
- モジュールの選択, 220
- モジュールの場所, 203
- 矢印間隔, 169
- 矢印の色, 171
- 矢印の長さ, 170
- 矢印, 214, 217
- 4次元形式, 102
- 乱数, 266
- ランダムなスカラー場, 266
- ランダムなベクトル場, 266, 268
- 乱流, 266
- 離散型実フーリエ変換サブルーチン, 268
- 理想 MHD 条件, 102
- リーマン不変量, 80
- 粒子数保存の法則, 119
- 粒子数保存, 102
- 流線, 213, 214
- 流線の表示, 179
- 流線の始点, 213
- Roe 平均, 48, 49
- Roe 法, 47
- ローレンツ係数, 101, 120, 128
- Lorentz 収縮, 137
- Roe 法, 128
- Wald 解, 111

数値天文学 テクニカルマニュアル

発行日	2004 年 9 月 10 日
初版発行	千葉大学 宇宙物理学研究室
編者	花輪 知幸 町田 正博
著者	花山 秀和 佐藤 裕司 上原 一浩 木暮 宏光 天野 孝伸 水野 陽介 富高 真 越智 康浩 錦織 弘充 大須賀 健 杉本 香菜子
