

Formura

サンプルコード解説

Version 1.1.0

2019年3月29日

変更履歴

Ver.	更新日時	説明
1.0.0	2019/3/15	初版発行
1.1.0	2019/3/29	第 2 版発行

目次

1	概要	1
2	サンプルプログラムの実行方法	1
3	1次元オイラー方程式	2
3.1	基礎方程式	2
3.2	HLLC スキーム	3
3.3	時間発展	5
3.4	MUSCL 補間・TVD 条件	5
3.5	初期条件	7
3.6	ドライバコード	7
3.7	実行結果	8
4	3次元オイラー方程式	9
4.1	基礎方程式	9
4.2	3次元 HLLC スキーム	9
4.3	時間発展	10
4.4	MUSCL 補間・TVD 条件	10
4.5	初期条件	13
4.6	ドライバコード	13
4.7	実行結果	13
5	1次元 MHD 方程式	14
5.1	基礎方程式	14
5.2	HLLD スキーム	14
5.3	時間発展	17
5.4	MUSCL 補間・TVD 条件	17
5.5	初期条件	17
5.6	ドライバコード	18
5.7	実行結果	18

6	3次元 MHD 方程式	18
6.1	基礎方程式	18
6.2	3次元 HLLD スキーム	20
6.3	9-wave 法	20
6.4	時間発展	22
6.5	MUSCL 補間・TVD 条件	22
6.6	初期条件	23
6.7	ドライバコード	23
6.8	実行結果	23

1 概要

このドキュメントでは，チュートリアルに付属しているサンプルプログラムについて解説する．サンプルプログラムは 1 および 3 次元のオイラー方程式および MHD の方程式のシミュレーションコードである．

サンプルプログラムはディレクトリ `sample` 以下にあり，それぞれ

- `sample/1d_h11c/`
1 次元オイラー方程式の計算コード
- `sample/3d_h11c/`
3 次元オイラー方程式の計算コード
- `sample/mhd_1d_h11d/`
1 次元 MHD 方程式の計算コード
- `sample/mhd_3d_h11d/`
3 次元 MHD 方程式の計算コード

である．

2 サンプルプログラムの実行方法

サンプルプログラムが入っているディレクトリ上で，

```
$ make
```

を実行すると Formura コンパイラが起動し，C 言語の計算コードおよびヘッダを出力する．

```
$ make main
```

を実行すると，必要なら Formura コンパイラが起動しさらに C コンパイラがプログラムの実行可能ファイルを出力する．

```
$ make run
```

を実行すると，必要なら Formura コンパイラおよび C コンパイラが起動し，さらにプロ

グラムを実行する.

```
$ make plot
```

を実行すると, プログラムの実行結果のファイルを可視化したグラフが画面上および PNG ファイルに出力される.

```
$ make clean
```

を実行すると, 中間生成ファイル, 計算結果および画像が削除される.

```
$ make run plot
```

を実行すると, プログラムのコンパイル, 実行, プロットまですべて行われる.

3 1次元オイラー方程式

Formura サンプルプログラム `sample/1d_hllc/hllc.fmr` が解いている問題とソルバーを解説する.

3.1 基礎方程式

1次元のオイラー方程式は保存量ベクトル \mathbf{U} , 流束ベクトル \mathbf{F} を用いて,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (1)$$

と書ける. \mathbf{U} と \mathbf{F} はそれぞれ

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ (e + p)u \end{pmatrix} \quad (2)$$

である. ここで, ρ は密度, u, v, w はそれぞれ流速 \mathbf{V} の x, y, z 軸成分, p は圧力, e はエネルギー密度である.

流体は理想気体であると仮定する. したがって, 適当な単位系の元で

$$p = \rho T \quad (3)$$

の関係が成り立つ。また、単位質量あたりの内部エネルギー U に対して、

$$U = \frac{T}{\gamma - 1} \quad (4)$$

が成り立つ。ここで γ は比熱比である。単位体積あたりの総エネルギー e との関係は

$$e = \rho U + \frac{1}{2} \rho |\mathbf{V}|^2 \quad (5)$$

である。

3.2 HLLC スキーム

サンプルプログラムでは、リーマンソルバーの一種である HLLC 法を用いてオイラー方程式の数値解を計算する。ここでは、HLLC 法の概略だけを説明する。HLLC 法の詳細は例えば Toro [1] の教科書等を参照のこと。

HLLC 法は近似リーマンソルバーの一種である。近似リーマンソルバーは、各セルの境界で近似的にリーマン問題を解き、その解から境界を流れる流束ベクトルを計算する方法である。伝統的な近似リーマンソルバーである HLL 法がリーマンファン内部の状態を単一の状態で近似してしまうのに対して、HLLC 法はリーマンファンの内部を接触不連続面を挟んで2つの領域に分割して、それぞれの領域が均一の状態であると仮定する。このため、HLLC 法は接触不連続面を解像することができる。

リーマン問題の初期状態では、セル内の物理量は位置によらず一定値を取り、セル境界において不連続な飛びがあると仮定する。リーマン問題を厳密に解くと、セル境界面からは衝撃波、膨張波、接触不連続面等の構造が現れ、それぞれの特性速度に従ってセル内部を進行する。その結果、ある一定時間経過した後のセル内部の物理状態は、初期状態のままの領域と、境界面から発生した衝撃波等によって乱された領域に分かれる。境界面から発生した波の影響を受けた領域をリーマンファンと呼ぶ。HLLC 法では、リーマンファンの内部を2つの領域に分けてそれぞれの領域で物理量が一定値をとると仮定する。

あるセル境界に着目して、境界の左側の物理量に $_L$ 、右側の物理量に対して $_R$ の添字をつけて区別することにする。Formura コード中では変数名に $_L$, $_R$ をつける。また、リーマンファン内部の左側の物理量には *_L 、右側の物理量には *_R の添字をつける。Formura コード中では変数名に $_stL$, $_stR$ をつける。

リーマンファンの左端の進行速度を S_L 、右端の進行速度を S_R とする。 S_L, S_R の評価

方法には様々な方法があるが、ここでは、Miyoshi & Kusano [2] を参考にして、

$$S_L = \min(u_L, u_R) - \max(a_L, a_R) \quad (6)$$

$$S_R = \max(u_L, u_R) + \max(a_L, a_R) \quad (7)$$

とする（関数 `fastest_signal_speed`）. ここで、 a_L, a_R はそれぞれの領域の物理量を使って評価した音速であり、

$$a_\alpha = \sqrt{\frac{\gamma p_\alpha}{\rho_\alpha}} \quad (8)$$

と定義される（ α には L または R の文字が入る）（関数 `sound_speed`）.

リーマンファン内部の2つの領域の境界の進行速度は、

$$S_M = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)} \quad (9)$$

で与えられる（関数 `middle_speed`）.

リーマンファンの内部では、

$$p_L^* = p_R^* = p^*, \quad (10)$$

$$u_L^* = u_R^* = u^* \quad (11)$$

が成り立つと仮定する. p^*, u^* はそれぞれ、

$$u^* = S_M \quad (12)$$

$$p^* = \frac{\rho_L p_R (S_L - u_L) - \rho_R p_L (S_R - u_R) + \rho_L \rho_R (S_R - u_R) (S_L - u_L) (u_L - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)} \quad (13)$$

である（関数 `pressure_star`）. 他の物理量は

$$\rho_\alpha^* = \rho_\alpha \frac{S_\alpha - u_\alpha}{S_\alpha - S_M} \quad (14)$$

$$v_\alpha^* = v_\alpha \quad (15)$$

$$w_\alpha^* = w_\alpha \quad (16)$$

$$e_\alpha^* = \frac{(S_\alpha - u_\alpha) e_\alpha - p_\alpha u_\alpha + p^* S_M}{S_\alpha - S_M} \quad (17)$$

とで計算できる（ α には L または R の文字が入る）（関数 `state_star`）.

以上の計算でリーマンファン内部の物理量 $\mathbf{U}_L^*, \mathbf{U}_R^*$ が求まった．これらの量と $\mathbf{U}_L, \mathbf{U}_R$ を合わせて，境界面における流束ベクトルが求まる．

$$\mathbf{F}_{\text{HLLC}} = \begin{cases} \mathbf{F}_L, & (0 \leq S_L), \\ \mathbf{F}_L^*, & (S_L \leq 0 \leq S_M), \\ \mathbf{F}_R^*, & (S_M \leq 0 \leq S_R), \\ \mathbf{F}_R, & (S_R \leq 0), \end{cases} \quad (18)$$

(関数 `mean_flux`) ここで，

$$\mathbf{F}_\alpha = \mathbf{F}(\mathbf{U}_\alpha) \quad (19)$$

$$\mathbf{F}_\alpha^* = \mathbf{F}_\alpha + S_\alpha(\mathbf{U}_\alpha^* - \mathbf{U}_\alpha) \quad (20)$$

(α には L または R の文字が入る) である (関数 `Flux_x, mean_flux`).

3.3 時間発展

HLLC 法を使って求めたフラックス \mathbf{F}_{HLLC} から，ある時間 Δt の間にセルに流入する保存量の計算し，次ステップにおける保存量の値 \mathbf{U}_{next} を決定する．

$$\mathbf{U}_{\text{next},i} = \mathbf{U}_i - \frac{\Delta t}{\Delta x} (\mathbf{F}_{\text{HLLC},i+\frac{1}{2}} - \mathbf{F}_{\text{HLLC},i-\frac{1}{2}}) \quad (21)$$

(`time_evol` 関数) ここで， i は格子点のインデックスである．半整数インデックスはセル境界上の点を表す．

3.4 MUSCL 補間・TVD 条件

MUSCL 補間はセル内の物理量の分布を多項式で近似することで計算の精度を向上させる方法である．サンプルプログラムでは，時間・空間ともに 2 次精度の MUSCL-Hancock 法を使用している．一方で，高次精度のスキームは不連続面の周辺で非物理的な振動を生じるため，計算結果が破綻しやすいという問題がある．この問題を回避するため TVD 条件を満たすような流束制限を施す必要がある．本サンプルコードでは MINMOD 流束制限関数を使用して数値的な安定性を維持している．

MUSCL-Hancock 法およびその TVD 版の詳しい説明は Toro[1] の教科書等載っている．

隣接する格子点間の差分 $\Delta_{i+\frac{1}{2}}$ を

$$\Delta_{i+\frac{1}{2}} = \mathbf{U}_{i+1} - \mathbf{U}_i \quad (22)$$

と定義する (関数 TVD_MUSCL). 変数が持つ半整数の格子点番号は, その変数が格子点と格子点の間 (セル境界) で定義されていることを表す. $\Delta_{i+\frac{1}{2}}, \Delta_{i-\frac{1}{2}}$ から, TVD 条件を満たすような制限されたスロープを

$$\overline{\Delta}_i = \begin{cases} \max[0, \min(\beta\Delta_{i-\frac{1}{2}}, \Delta_{i+\frac{1}{2}}), \min(\Delta_{i-\frac{1}{2}}, \beta\Delta_{i+\frac{1}{2}})], & (\Delta_{i+\frac{1}{2}} > 0), \\ \min[0, \max(\beta\Delta_{i-\frac{1}{2}}, \Delta_{i+\frac{1}{2}}), \max(\Delta_{i-\frac{1}{2}}, \beta\Delta_{i+\frac{1}{2}})], & (\Delta_{i+\frac{1}{2}} < 0), \end{cases} \quad (23)$$

と定義する (関数 TVD_MUSCL). $\beta = 1$ は MINMOD 制限関数, $\beta = 2$ は SUPER BEE 制限関数に対応する.

$\overline{\Delta}_i$ を用いて, セルの両端における保存量の値を

$$U_i^L = U_i - \frac{1}{2}\overline{\Delta}_i \quad (24)$$

$$U_i^R = U_i + \frac{1}{2}\overline{\Delta}_i \quad (25)$$

と定義する (関数 TVD_MUSCL) (ここでの左右はセルの中心から見たときの左右であり, 境界から見た左右とは異なるので注意).

次に, U_i^L, U_i^R の値から境界における流束ベクトルを計算し $\frac{1}{2}\Delta t$ だけタイムステップを進める.

$$\overline{U}_i^L = U_i^L - \frac{\Delta t}{2} \frac{\mathbf{F}(U_i^R) - \mathbf{F}(U_i^L)}{\Delta x} \quad (26)$$

$$\overline{U}_i^R = U_i^R - \frac{\Delta t}{2} \frac{\mathbf{F}(U_i^R) - \mathbf{F}(U_i^L)}{\Delta x} \quad (27)$$

(関数 left_right_state)

このようにして求めた $\overline{U}_i^L, \overline{U}_i^R$ をリーマンソルバーの境界の左右の値として用いる.

$$U_{L,i+\frac{1}{2}} = \overline{U}_i^R \quad (28)$$

$$U_{R,i+\frac{1}{2}} = \overline{U}_{i+1}^L \quad (29)$$

(関数 left_right_state, 関数 MUSCL_Hancock)

3.5 初期条件

サンプルプログラムで用いられている初期条件は Toro [1] の 10.8 節の Test1 に, v, w を追加して,

$$\rho_L = 1.0, \quad \rho_R = 0.125, \quad (30)$$

$$u_L = 0.75, \quad u_R = 0.0, \quad (31)$$

$$v_L = 0.0, \quad v_R = 0.0, \quad (32)$$

$$w_L = 0.0, \quad w_R = 0.0, \quad (33)$$

$$p_L = 1.0, \quad p_R = 0.1, \quad (34)$$

で与えられる (関数 `init`). また, $\gamma = \frac{7}{5}$ である.

Formura では CFL 条件のためのクーラン数 C_{CFL} を陽に設定することができないので, あらかじめ計算中に現れる最大の信号速度 S_{max} を仮定した上で, その速度に対してクーラン数を決めて時間刻み幅を

$$\Delta t = \frac{C_{\text{CFL}} \Delta x}{S_{\text{max}}} \quad (35)$$

と設定する.

今回の初期条件のもとでは, 音速の概算値は $a \sim \sqrt{\frac{\gamma p}{\rho}} \sim 1$, 流速の概算値は $u \sim 1$ なので, 最速の信号伝搬速度は $S \sim 2$ 程度であり, 余裕を持って $S_{\text{max}} = 4$ と仮定する. この速度に対してクーラン数 $C_{\text{CFL}} = 0.8$ とする. このときの時間刻み幅は $\Delta t = 0.2 \Delta x$ である.

今回は計算領域の長さを $L_x = 4.0$, 格子点数を $N_x = 400$ とした. したがって, $\Delta x = \frac{L_x}{N_x} = 0.01$ である. また, 計算の終了時刻は, $t_{\text{fin}} = 0.2$ である. したがって, 計算するタイムステップ数は $N_t = \frac{t_{\text{fin}}}{\Delta t} = 100$ となる.

3.6 ドライバコード

3.6.1 ドライバコードの動作

ドライバコード (`sample/1d_hllc/main.c`) はタイムステップが 100 になるまで, タイムステップを進める関数 `Formura_Forward()` を呼び出す. ドライバコードは, 初期時刻における保存量ベクトル場の状態と終了時刻における保存量ベクトル場の状態をファイルに書き出す.

3.6.2 出力ファイル形式

ドライバコードが出力するファイルは、空白区切りのテキストデータである。ファイル名の書式は、{計算ステップ番号}_{MPI ランク番号}.dat である。

一つの格子点に含まれる情報がデータファイルの1行に対応する。各行には、一つの格子点についての{x 軸座標} {y 軸座標} {z 軸座標} {密度} {圧力} {温度} {x 軸方向流速} {y 軸方向流速} {z 軸方向流速} がこの順番で並んでいる。

Python モジュール `formura_data_load.py` の `load_data` 関数はこの形式のデータを読み込み、結合した上で単一の `ndarray` データとして返す。

3.7 実行結果

ディレクトリ `sample/1d_hllc/` 上で `make run plot` を実行すると、Formura のコンパイル・実行が行われ、図1のようなグラフが出力される。

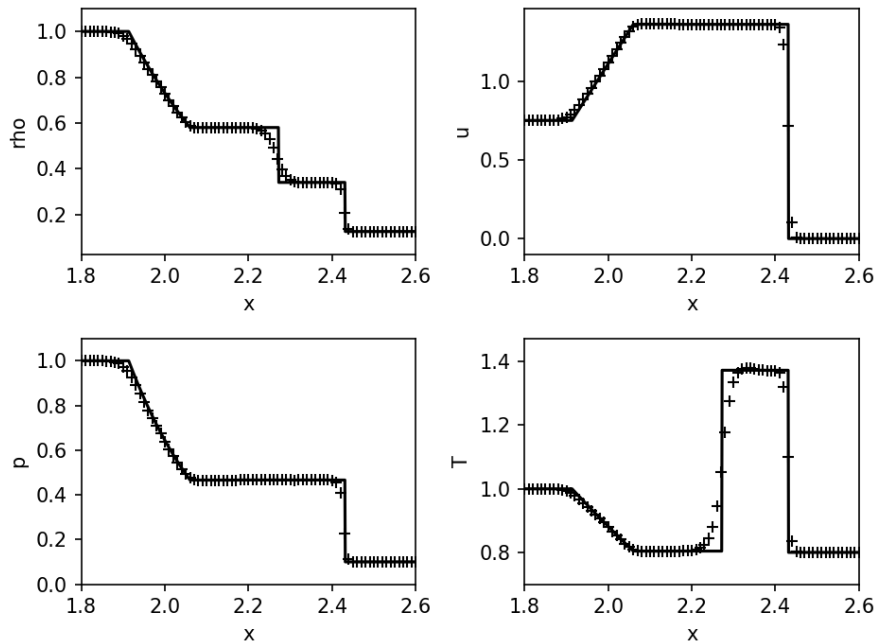


図1 オイラー方程式の衝撃波管問題の解. $t = 0.2$, $\Delta t = 0.002$, $\Delta x = 0.01$. 左上のパネルは密度, 右上のパネルは x 方向流速, 左下のパネルは圧力, 右下のパネルは温度である. 十字点は Formura による計算結果, 実線はリーマン問題の厳密解 (Toro[1]) を表す.

4 3次元オイラー方程式

サンプルコード `sample/3d_hllc/3d_hllc.fmr` の解説をする。

4.1 基礎方程式

このサンプルコードでは、3次元オイラー方程式を解く。使用するスキームは1次元のときと同じで、HLLC法である。MUSCL-Hnacock法による2次精度化も行う。

3次元のオイラー方程式は、

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0 \quad (36)$$

と書かれる。ここで、 \mathbf{U} は保存量の密度、 $\mathbf{F}, \mathbf{G}, \mathbf{H}$ はそれぞれ x 軸、 y 軸、 z 軸方向の流束ベクトルである。 $\mathbf{U}, \mathbf{F}, \mathbf{G}, \mathbf{H}$ は、

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ (e + p)u \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho w v \\ (e + p)v \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ (e + p)w \end{pmatrix} \quad (37)$$

である。気体の状態方程式とエネルギー密度は1次元のときと同様に式(3), (4), (5)で与えられる。

4.2 3次元 HLLC スキーム

このサンプルでは3次元の時間発展に有限体積法を用いる。3次元有限体積法については、Toro[1]の16.4節を参考にした。

3次元のフラックス計算は、1次元のときと同様にセルの境界に対してリーマン問題の近似解を求めて、その解から流束ベクトルを与える。

x 軸方向、 y 軸方向、 z 軸方向はそれぞれ独立に流束ベクトルを決定することができるため、 x 軸1次元のときの流束ベクトルの計算方法をそのまま y 軸、 z 軸にも適応できる。

Formuraコード上では、 $\mathbf{F}, \mathbf{G}, \mathbf{H}$ は関数 `HLLC_Flux_x`, `HLLC_Flux_y`, `HLLC_Flux_z` で計算している。これらは、内部で1次元版の流束ベクトル計算関数 `HLLC_Flux` を呼び出している。 y 軸方向を計算するときは、入力の状態ベクトルの x, y, z 成分をサイク

リックに並べ替えて, $\mathbf{U}_{\text{cyc},y} = (\rho, \rho v, \rho w, \rho u, e)^T$ としてから x 軸方向の流束ベクトルを求め, 座標軸を逆周りに入れ替えて \mathbf{G} を計算する (関数 `HLLC_Flux_1D` および関数 `YZXcycle`). z 軸方向も同様に, 状態ベクトルを $\mathbf{U}_{\text{cyc},z} = (\rho, \rho w, \rho u, \rho v, e)^T$ と入れ替えてから x 軸方向流束ベクトルを計算し, 座標軸を逆周りに回転させて \mathbf{H} を求めている (関数 `HLLC_Flux_1D` および関数 `YZXcycle`).

4.3 時間発展

HLLC 法を使って求めたフラックス $\mathbf{F}_{\text{HLLC}}, \mathbf{G}_{\text{HLLC}}, \mathbf{H}_{\text{HLLC}}$ から, ある時間 Δt の間にセルに流入する保存量の計算し, 次ステップにおける保存量の値 \mathbf{U}_{next} を決定する.

$$\begin{aligned} \mathbf{U}_{\text{next},i,j,k} = \mathbf{U}_{i,j,k} - \Delta t & \left(\frac{\mathbf{F}_{\text{HLLC},i+\frac{1}{2},j,k} - \mathbf{F}_{\text{HLLC},i-\frac{1}{2},j,k}}{\Delta x} \right. \\ & + \frac{\mathbf{G}_{\text{HLLC},i,j+\frac{1}{2},k} - \mathbf{G}_{\text{HLLC},i,j-\frac{1}{2},k}}{\Delta y} \\ & \left. + \frac{\mathbf{H}_{\text{HLLC},i,j,k+\frac{1}{2}} - \mathbf{H}_{\text{HLLC},i,j,k-\frac{1}{2}}}{\Delta z} \right) \end{aligned} \quad (38)$$

(関数 `time_evol`) ここで, i, j, k は格子点のインデックスである. 半整数インデックスはセル境界上の点を表す.

4.4 MUSCL 補間・TVD 条件

3次元版の MUSCL-Hancock 法の実装は Toro[1] の 16.5 節を参考にした.

格子点間の x, y, z それぞれの方向についての差分を

$$\Delta_{i+\frac{1}{2},j,k} = \mathbf{U}_{i+1,j,k} - \mathbf{U}_{i,j,k} \quad (39)$$

$$\Delta_{i,j+\frac{1}{2},k} = \mathbf{U}_{i,j+1,k} - \mathbf{U}_{i,j,k} \quad (40)$$

$$\Delta_{i,j,k+\frac{1}{2}} = \mathbf{U}_{i,j,k+1} - \mathbf{U}_{i,j,k} \quad (41)$$

と定義する (関数 `TVD_MUSCL`). これを用いて, TVD 条件を考慮した制限されたスロー

プは,

$$\overline{\Delta}_x = \begin{cases} \max[0, \min(\beta\Delta_{i-\frac{1}{2},j,k}, \Delta_{i+\frac{1}{2},j,k}), \min(\Delta_{i-\frac{1}{2},j,k}, \beta\Delta_{i+\frac{1}{2},j,k})], & (\Delta_{i+\frac{1}{2},j,k} > 0), \\ \min[0, \max(\beta\Delta_{i-\frac{1}{2},j,k}, \Delta_{i+\frac{1}{2},j,k}), \max(\Delta_{i-\frac{1}{2},j,k}, \beta\Delta_{i+\frac{1}{2},j,k})], & (\Delta_{i+\frac{1}{2},j,k} < 0), \end{cases} \quad (42)$$

$$\overline{\Delta}_y = \begin{cases} \max[0, \min(\beta\Delta_{i,j-\frac{1}{2},k}, \Delta_{i,j+\frac{1}{2},k}), \min(\Delta_{i,j-\frac{1}{2},k}, \beta\Delta_{i,j+\frac{1}{2},k})], & (\Delta_{i,j+\frac{1}{2},k} > 0), \\ \min[0, \max(\beta\Delta_{i,j-\frac{1}{2},k}, \Delta_{i,j+\frac{1}{2},k}), \max(\Delta_{i,j-\frac{1}{2},k}, \beta\Delta_{i,j+\frac{1}{2},k})], & (\Delta_{i,j+\frac{1}{2},k} < 0), \end{cases} \quad (43)$$

$$\overline{\Delta}_z = \begin{cases} \max[0, \min(\beta\Delta_{i,j,k-\frac{1}{2}}, \Delta_{i,j,k+\frac{1}{2}}), \min(\Delta_{i,j,k-\frac{1}{2}}, \beta\Delta_{i,j,k+\frac{1}{2}})], & (\Delta_{i,j,k+\frac{1}{2}} > 0), \\ \min[0, \max(\beta\Delta_{i,j,k-\frac{1}{2}}, \Delta_{i,j,k+\frac{1}{2}}), \max(\Delta_{i,j,k-\frac{1}{2}}, \beta\Delta_{i,j,k+\frac{1}{2}})], & (\Delta_{i,j,k+\frac{1}{2}} < 0), \end{cases} \quad (44)$$

となる (関数 TVD_MUSCL). さらに, これを使ってセル内の物理量を補間すると,

$$U^{-x} = U - \frac{1}{2}\overline{\Delta}_x, \quad U^{+x} = U + \frac{1}{2}\overline{\Delta}_x \quad (45)$$

$$U^{-y} = U - \frac{1}{2}\overline{\Delta}_y, \quad U^{+y} = U + \frac{1}{2}\overline{\Delta}_y \quad (46)$$

$$U^{-z} = U - \frac{1}{2}\overline{\Delta}_z, \quad U^{+z} = U + \frac{1}{2}\overline{\Delta}_z \quad (47)$$

となる (関数 TVD_MUSCL). 補間された保存量ベクトルから境界における流束ベクトルを

計算し, $\frac{1}{2}\Delta t$ だけタイムステップを進める.

$$\bar{U}^{-x} = U^{-x} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (48)$$

$$\bar{U}^{+x} = U^{+x} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (49)$$

$$\bar{U}^{-y} = U^{-y} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (50)$$

$$\bar{U}^{+y} = U^{+y} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (51)$$

$$\bar{U}^{-z} = U^{-z} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (52)$$

$$\bar{U}^{+z} = U^{+z} - \frac{\Delta t}{2} \left(\frac{F(U^{+x}) - F(U^{-x})}{\Delta x} + \frac{G(U^{+y}) - G(U^{-y})}{\Delta y} + \frac{H(U^{+z}) - H(U^{-z})}{\Delta z} \right) \quad (53)$$

$$(54)$$

(関数 `left_right_state`) この状態量を改めて境界の左右におけるリーマン問題の初期値であるとして, HLLC 法による流束ベクトルを評価する.

$$U_{L,i+\frac{1}{2},j,k} = \bar{U}_{i,j,k}^{+x} \quad (55)$$

$$U_{R,i+\frac{1}{2},j,k} = \bar{U}_{i+1,j,k}^{-x} \quad (56)$$

$$U_{L,i,j+\frac{1}{2},k} = \bar{U}_{i,j,k}^{+y} \quad (57)$$

$$U_{R,i,j+\frac{1}{2},k} = \bar{U}_{i,j+1,k}^{-y} \quad (58)$$

$$U_{L,i,j,k+\frac{1}{2}} = \bar{U}_{i,j,k}^{+z} \quad (59)$$

$$U_{R,i,j,k+\frac{1}{2}} = \bar{U}_{i,j,k+1}^{-z} \quad (60)$$

$$(61)$$

(関数 `left_right_state`)

4.5 初期条件

初期条件（関数 `init`）は1次元のときと同じである。ただし、 x 軸の代わりに y 軸方向の衝撃波管問題になっている。

4.6 ドライバコード

ドライバコードは1次元のときとほぼ同じである。ただし、出力するグリッドが3次元になっているため、ファイル出力時には3次元の格子点すべての情報を書き込む。ファイル出力の際の内ループは x 軸方向で最外ループは z 軸方向である。出力ファイルの列は1次元オイラー方程式のサンプルコードと同様である。

4.7 実行結果

ディレクトリ `sample/3d_hllc/` 上で `make run plot` を実行すると、Formura のコンパイル・実行が行われ、図2のようなグラフが出力される。

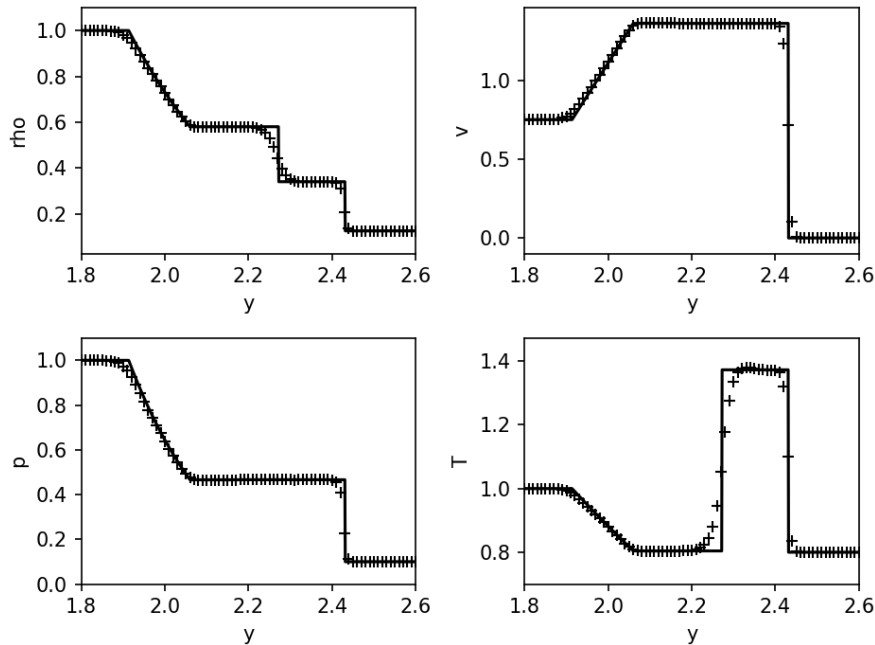


図2 3次元 HLLC コードを用いた、 y 軸方向 1次元衝撃波管問題の解。十字点は Formura の計算結果。実線はリーマン問題の厳密解 (Toro[1]) である。左上のパネルは密度、右上のパネルは x 方向流速、左下のパネルは圧力、右下のパネルは温度である。

5 1次元 MHD 方程式

1次元 MHD 方程式のサンプルプログラム (`sample/mhd_1d_hlld/`) の解説をする. このサンプルでは, HLLC 法を発展させた HLLD 法 [2] を用いて, 1次元の MHD 方程式を解く.

5.1 基礎方程式

1次元 MHD 方程式は式 (1) と同じく双曲型方程式である. 保存量ベクトル \mathbf{U} および流束ベクトル \mathbf{F} は,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ B_y \\ B_z \\ e \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p_T - B_x^2 \\ \rho v u - B_x B_y \\ \rho w u - B_x B_z \\ B_y u - B_x v \\ B_z u - B_x w \\ (e + p_T)u - B_x(\mathbf{V} \cdot \mathbf{B}) \end{pmatrix} \quad (62)$$

と書ける. ここで, B_x, B_y, B_z は磁束密度の x 成分, y 成分, z 成分である. また, p_T および e は

$$p_T = p + \frac{1}{2}|\mathbf{B}|^2 \quad (63)$$

$$e = \rho U + \frac{1}{2}\rho|\mathbf{V}|^2 + \frac{1}{2}|\mathbf{B}|^2 \quad (64)$$

である. 気体の状態方程式および内部エネルギーは式 (3), (4) とする.

5.2 HLLD スキーム

HLLD 法 [2] は HLLC 法を発展させた方法である. HLLC 法がリーマンファンを 2 状態に分割するのに対して, HLLD 法はリーマンファンを 4 状態に分割する. これによって HLLD 法は接触不連続面だけでなく, 回転不連続面も解像することができる.

リーマンファン内部の状態は左から順番に $\mathbf{U}_L^*, \mathbf{U}_L^{**}, \mathbf{U}_R^{**}, \mathbf{U}_R^*$ と書くことにする. \mathbf{U}_L^{**} と \mathbf{U}_R^{**} の境界の速度を S_M とする. HLLD 法の S_M は, 式 (9) の p を p_T に置き換えることで得られる.

$$S_M = \frac{p_{T_R} - p_{T_L} + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)} \quad (65)$$

(関数 `middle_speed`) リーマンファン内部の x 方向速度はどの領域でも等しく $u_L^* = u_L^{**} = u_R^{**} = u_R^* = u^*$ である. 圧力 p_T も同様に, $p_{TL}^* = p_{TL}^{**} = p_{TR}^{**} = p_{TR}^* = p_T^*$ である. p_T^*, u^* はそれぞれ,

$$u^* = S_M \quad (66)$$

$$p_T^* = \frac{\rho_L p_{TR}(S_L - u_L) - \rho_R p_{TL}(S_R - u_R) + \rho_L \rho_R (S_R - u_R)(S_L - u_L)(u_L - u_R)}{\rho_L(S_L - u_L) - \rho_R(S_R - u_R)} \quad (67)$$

である (関数 `pressure_star`).

スター領域における物理量は,

$$\rho_\alpha^* = \rho_\alpha \frac{S_\alpha - u_\alpha}{S_\alpha - S_M} \quad (68)$$

$$v_\alpha^* = v_\alpha - B_x B_{y\alpha} \frac{S_M - u_\alpha}{\rho_\alpha(S_\alpha - u_\alpha)(S_\alpha - S_M) - B_x^2} \quad (69)$$

$$w_\alpha^* = w_\alpha - B_x B_{z\alpha} \frac{S_M - u_\alpha}{\rho_\alpha(S_\alpha - u_\alpha)(S_\alpha - S_M) - B_x^2} \quad (70)$$

$$B_{y\alpha}^* = B_{y\alpha} \frac{\rho_\alpha(S_\alpha - u_\alpha)^2 - B_x^2}{\rho_\alpha(S_\alpha - u_\alpha)(S_\alpha - S_M) - B_x^2} \quad (71)$$

$$B_{z\alpha}^* = B_{z\alpha} \frac{\rho_\alpha(S_\alpha - u_\alpha)^2 - B_x^2}{\rho_\alpha(S_\alpha - u_\alpha)(S_\alpha - S_M) - B_x^2} \quad (72)$$

$$e_\alpha^* = \frac{(S_\alpha - u_\alpha)e_\alpha - p_{T\alpha} u_\alpha + p_T^* S_M + B_x(\mathbf{V}_\alpha \cdot \mathbf{B}_\alpha - \mathbf{V}_\alpha^* \cdot \mathbf{B}_\alpha^*)}{S_\alpha - S_M} \quad (73)$$

で与えられる (関数 `state_star`) (α には L または R が入る).

ダブルスター領域における密度は, 隣のスター領域の値に等しい.

$$\rho_\alpha^{**} = \rho_\alpha^* \quad (74)$$

ダブルスター領域における垂直方向の速度と磁束密度は左右で等しい.

$$v_L^{**} = v_R^{**} = v^{**} \quad (75)$$

$$w_L^{**} = w_R^{**} = w^{**} \quad (76)$$

$$B_{yL}^{**} = B_{yR}^{**} = B_y^{**} \quad (77)$$

$$B_{zL}^{**} = B_{zR}^{**} = B_z^{**} \quad (78)$$

ここで, $v^{**}, w^{**}, B_y^{**}, B_z^{**}$ はそれぞれ,

$$v^{**} = \frac{\sqrt{\rho_L^*} v_L^* + \sqrt{\rho_R^*} v_R^* + (B_{y_R}^* - B_{y_L}^*) \text{sign}(B_x)}{\sqrt{\rho_L^*} + \sqrt{\rho_R^*}} \quad (79)$$

$$w^{**} = \frac{\sqrt{\rho_L^*} w_L^* + \sqrt{\rho_R^*} w_R^* + (B_{z_R}^* - B_{z_L}^*) \text{sign}(B_x)}{\sqrt{\rho_L^*} + \sqrt{\rho_R^*}} \quad (80)$$

$$B_y^{**} = \frac{\sqrt{\rho_L^*} B_{y_R}^* + \sqrt{\rho_R^*} B_{y_L}^* + \sqrt{\rho_L^* \rho_R^*} (v_R^* - v_L^*) \text{sign}(B_x)}{\sqrt{\rho_L^*} + \sqrt{\rho_R^*}} \quad (81)$$

$$B_z^{**} = \frac{\sqrt{\rho_L^*} B_{z_R}^* + \sqrt{\rho_R^*} B_{z_L}^* + \sqrt{\rho_L^* \rho_R^*} (w_R^* - w_L^*) \text{sign}(B_x)}{\sqrt{\rho_L^*} + \sqrt{\rho_R^*}} \quad (82)$$

で与えられる (関数 `state_star_star`). ダブルスター領域のエネルギー密度は,

$$e_\alpha^{**} = e_\alpha^* \mp \sqrt{\rho_\alpha^*} (\mathbf{V}_\alpha^{**} \cdot \mathbf{V}^{**} \cdot \mathbf{B}^{**}) \text{sign}(B_x) \quad (83)$$

で与えられる (関数 `state_star_star`) (α には L または R が入る).

U_L^* と U_L^{**} の境界の速度 S_L^* および, U_R^* と U_R^{**} の境界の速度 S_R^* は,

$$S_L^* = S_M - \frac{|B_x|}{\sqrt{\rho_L^*}} \quad (84)$$

$$S_R^* = S_M + \frac{|B_x|}{\sqrt{\rho_R^*}} \quad (85)$$

で与えられる (関数 `star_speed`).

以上でリーマンファンの内部状態が求まった. ここから, 境界面の流束ベクトルは,

$$\mathbf{F}_{\text{HLLD}} = \begin{cases} \mathbf{F}_L, & (0 \leq S_L), \\ \mathbf{F}_L^*, & (S_L \leq 0 \leq S_L^*), \\ \mathbf{F}_L^{**}, & (S_L^* \leq 0 \leq S_M), \\ \mathbf{F}_R^{**}, & (S_M \leq 0 \leq S_R^*), \\ \mathbf{F}_R^*, & (S_R^* \leq 0 \leq S_R), \\ \mathbf{F}_R, & (S_R \leq 0), \end{cases} \quad (86)$$

で与えられる (関数 `mean_flux`). ここで, $\mathbf{F}_L, \mathbf{F}_R, \mathbf{F}_L^*, \mathbf{F}_L^{**}, \mathbf{F}_R^*, \mathbf{F}_R^{**}$ は

$$\mathbf{F}_\alpha = \mathbf{F}(\mathbf{U}_\alpha) \quad (87)$$

$$\mathbf{F}_\alpha^* = \mathbf{F}_\alpha + S_\alpha (\mathbf{U}_\alpha^* - \mathbf{U}_\alpha) \quad (88)$$

$$\mathbf{F}_\alpha^{**} = \mathbf{F}_\alpha + S_\alpha (\mathbf{U}_\alpha^* - \mathbf{U}_\alpha) + S_\alpha^* (\mathbf{U}_\alpha^{**} - \mathbf{U}_\alpha^*) \quad (89)$$

(α には L または R の文字が入る) である (関数 `MHD_Flux_x`, `mean_flux`).

5.3 時間発展

HLLD 法を使って求めたフラックス \mathbf{F}_{HLLD} から, ある時間 Δt の間にセルに流入する保存量の計算し, 次ステップにおける保存量の値 \mathbf{U}_{next} を決定する.

$$\mathbf{U}_{\text{next},i} = \mathbf{U}_i - \frac{\Delta t}{\Delta x} (\mathbf{F}_{\text{HLLD},i+\frac{1}{2}} - \mathbf{F}_{\text{HLLD},i-\frac{1}{2}}) \quad (90)$$

(関数 `time_evol`) ここで, i は格子点のインデックスである. 半整数インデックスはセル境界上の点を表す.

5.4 MUSCL 補間・TVD 条件

MUSCL 補間および流束制限に関する計算は 1 次元オイラー方程式のときと同様である.

5.5 初期条件

1 次元 MHD サンプルコードの初期条件は,

$$\rho_L = 1.08, \quad \rho_R = 1.0, \quad (91)$$

$$u_L = 1.2 \quad u_R = 0.0, \quad (92)$$

$$v_L = 0.01 \quad v_R = 0.0, \quad (93)$$

$$w_L = 0.5 \quad w_R = 0.0, \quad (94)$$

$$p_L = 0.95, \quad p_R = 1.0, \quad (95)$$

$$B_x = \frac{2.0}{\sqrt{4\pi}}, \quad (96)$$

$$B_{yL} = \frac{3.6}{\sqrt{4\pi}}, \quad B_{yR} = \frac{4.0}{\sqrt{4\pi}} \quad (97)$$

$$B_{zL} = \frac{2.0}{\sqrt{4\pi}}, \quad B_{zR} = \frac{2.0}{\sqrt{4\pi}} \quad (98)$$

で与えられる. また, 比熱比は $\gamma = \frac{5}{3}$ である.

この初期条件における最大速度は $S \sim |\mathbf{V}| + \sqrt{\frac{\gamma p + |\mathbf{B}|^2}{\rho}}$ で評価すると $S \sim 3$ である. また, クーラン数を $C_{\text{CFL}} = 0.3$ とする. よって, $\Delta t = 0.1 \Delta x$ となる.

設定ファイル `config.yaml` の設定は, グリッド数を 3200, 空間方向長さを 4.0 としている. よって, $\Delta x = \frac{4.0}{3200}$ である.

5.6 ドライバコード

5.6.1 ドライバコードの動作

ドライバコード `main.c` は, `FormuraForward` を 1600 ステップ ($t = 0.2$) 実行して停止する. 初期状態と終了状態をファイルに出力する.

5.6.2 出力ファイル形式

出力ファイルの形式は, 1次元オイラー方程式のサンプルコードと同様である. ただし, 出力ファイルに磁束密度に関するカラムが追加されている.

ファイルの1行は, {x 軸座標} {y 軸座標} {z 軸座標} {密度} {圧力} {温度} {x 軸方向流速} {y 軸方向流速} {z 軸方向流速} {磁束密度 x 成分} {磁束密度 y 成分} {磁束密度 z 成分} である. 1行が一つの格子点に対応し, すべての格子点に対して書き出される.

5.7 実行結果

ディレクトリ `sample/mhd_1d_hl1d/` 上で `make run plot` を実行すると, `Formura` のコンパイル・実行が行われ, 図3のようなグラフが出力される.

6 3次元 MHD 方程式

3次元 MHD 方程式のサンプルプログラム (`sample/mhd_3d_hl1d/mhd_3d_hl1d.fmr`) の解説をする.

6.1 基礎方程式

3次元 MHD 方程式の基礎方程式は, 式 (36) と同じく双曲型偏微分方程式である.

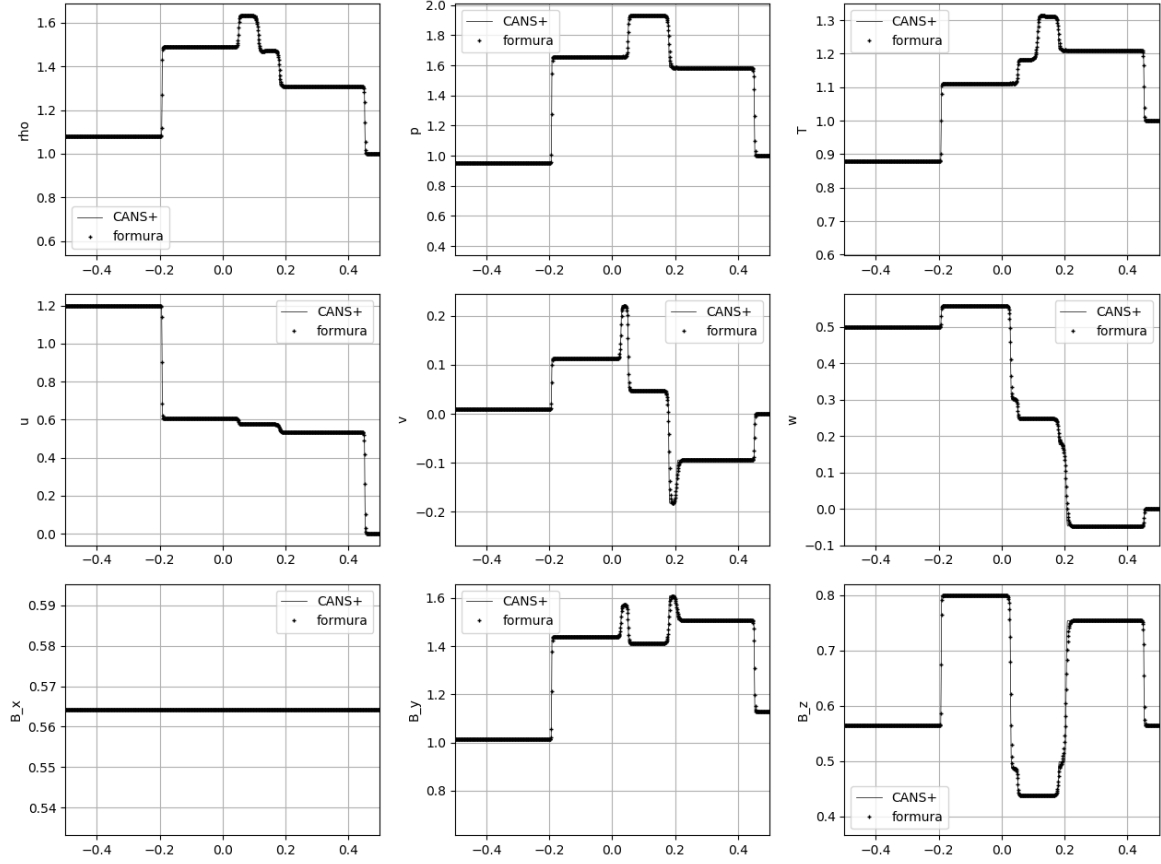


図3 MHD 方程式の衝撃波管問題の解. $t = 0.2$, $\Delta t = \frac{1}{8000}$, $\Delta x = \frac{1}{800}$. 最上段のパネルは左から密度, 圧力, 温度である. 中段のパネルは左から, 流速の x 成分, y 成分, z 成分である. 最下段のパネルは, 左から磁束密度の x 成分, y 成分, z 成分である. 十字点は Formura による計算結果, 実線は CANS+[3] による同一問題の計算結果である.

U, F, G, H は,

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ B_x \\ B_y \\ B_z \\ e \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p_T - B_x^2 \\ \rho v u - B_x B_y \\ \rho w u - B_x B_z \\ 0 \\ B_y u - B_x v \\ B_z u - B_x w \\ (e + p_T)u - B_x(\mathbf{V} \cdot \mathbf{B}) \end{pmatrix}, \quad (99)$$

$$G = \begin{pmatrix} \rho v \\ \rho u v - B_y B_x \\ \rho v^2 + p_T - B_y^2 \\ \rho w v - B_y B_z \\ B_x v - B_y u \\ 0 \\ B_z v - B_y w \\ (e + p_T)v - B_y(\mathbf{V} \cdot \mathbf{B}) \end{pmatrix}, H = \begin{pmatrix} \rho w \\ \rho u w - B_z B_x \\ \rho v w - B_z B_y \\ \rho w^2 + p_T - B_z^2 \\ B_x w - B_z u \\ B_y w - B_z v \\ 0 \\ (e + p_T)w - B_z(\mathbf{V} \cdot \mathbf{B}) \end{pmatrix} \quad (100)$$

である.

6.2 3次元 HLLD スキーム

原理的には, 3次元 HLLC の実装と同じく各軸方向に 1次元のフラックスを求めて, 足し合わせることで物理量の時間発展を計算することが可能である. しかし, 3次元 MHD の場合オイラー方程式のときとは異なり, 愚直な三次元拡張版スキームでは正しい解を得ることができない.

愚直な実装では, 3次元 MHD では磁束密度のガウスの法則

$$\operatorname{div} \mathbf{B} = 0 \quad (101)$$

を満たすことができない. したがって, $\operatorname{div} \mathbf{B}$ が小さい値を保つような工夫が必要になる.

6.3 9-wave 法

9-wave 法は Dedner et al.[4] が開発した $\operatorname{div} \mathbf{B}$ 解消法である. 9-wave 法は $\operatorname{div} \mathbf{B}$ の値が非ゼロになった場合, $\operatorname{div} \mathbf{B}$ が自動的に拡散・消滅するように MHD 方程式を修正する手法である. 非物理的な状態変数 ψ を導入し, ψ に関する方程式と MHD 方程式と連立させることで, $\operatorname{div} \mathbf{B}$ の値を拡散させる.

9-wave 法を適応した MHD 方程式は源泉項 \mathbf{S} を持ち,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{S} \quad (102)$$

となる．状態ベクトル \mathbf{U} および流束ベクトル $\mathbf{F}, \mathbf{G}, \mathbf{H}$ は,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ B_x \\ B_y \\ B_z \\ e \\ \psi \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p_T - B_x^2 \\ \rho v u - B_x B_y \\ \rho w u - B_x B_z \\ \psi \\ B_y u - B_x v \\ B_z u - B_x w \\ (e + p_T)u - B_x(\mathbf{V} \cdot \mathbf{B}) \\ c_h^2 B_x \end{pmatrix}, \quad (103)$$

$$\mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v - B_y B_x \\ \rho v^2 + p_T - B_y^2 \\ \rho w v - B_y B_z \\ B_x v - B_y u \\ \psi \\ B_z v - B_y w \\ (e + p_T)v - B_y(\mathbf{V} \cdot \mathbf{B}) \\ c_h^2 B_y \end{pmatrix}, \mathbf{H} = \begin{pmatrix} \rho w \\ \rho u w - B_z B_x \\ \rho v w - B_z B_y \\ \rho w^2 + p_T - B_z^2 \\ B_x w - B_z u \\ B_y w - B_z v \\ \psi \\ (e + p_T)w - B_z(\mathbf{V} \cdot \mathbf{B}) \\ c_h^2 B_z \end{pmatrix} \quad (104)$$

である．また，源泉項 \mathbf{S} は,

$$\mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{c_h}{c_r} \psi \end{pmatrix} \quad (105)$$

である．

1次元 MHD 方程式に対する HLLD 法を 3次元の場合にも流用するため，セル境界面における ψ の値と \mathbf{B} のセル境界面に対して垂直な成分を求める必要がある．これは，磁束密度の垂直成分 B_n と ψ に関するリーマン問題を解くことで，

$$B_n = \frac{B_{n_L} + B_{n_R}}{2} - \frac{1}{2c_h}(\psi_R - \psi_L) \quad (106)$$

$$\psi = \frac{\psi_L + \psi_R}{2} - \frac{c_h}{2}(B_{x_R} - B_{x_L}) \quad (107)$$

(関数 `normal_B`) と与えられる [3]. ここで求めた B_n の値を用いて x, y, z 各軸方向に対して 1 次元 MHD 方程式に対する HLLD 法の計算を行い, $\mathbf{F}, \mathbf{G}, \mathbf{H}$ を得る. そして, ψ に対して有限体積法の時間発展を評価した後の値 ψ_{temp} に対して, 源泉項の評価を行い,

$$\psi_{\text{next}} = \psi_{\text{temp}} \exp\left(-\Delta t \frac{c_h}{c_r}\right) \quad (108)$$

(関数 `time_evol`) として, 次ステップにおける ψ の値を決定する [3].

ここで, c_h と c_r はフリーパラメータである. c_h の値には CFL 条件から許される最大速度, c_r の値は 0.18 を採用する [4],[3].

6.4 時間発展

HLLD 法を使って求めたフラックス $\mathbf{F}_{\text{HLLD}}, \mathbf{G}_{\text{HLLD}}, \mathbf{H}_{\text{HLLD}}$ から, ある時間 Δt の間にセルに流入する保存量の計算し, 次ステップにおける保存量の値 \mathbf{U}_{next} を決定する.

$$\begin{aligned} \mathbf{U}_{\text{next},i,j,k} = \mathbf{U}_{i,j,k} - \Delta t & \left(\frac{\mathbf{F}_{\text{HLLD},i+\frac{1}{2},j,k} - \mathbf{F}_{\text{HLLD},i-\frac{1}{2},j,k}}{\Delta x} \right. \\ & + \frac{\mathbf{G}_{\text{HLLD},i,j+\frac{1}{2},k} - \mathbf{G}_{\text{HLLD},i,j-\frac{1}{2},k}}{\Delta y} \\ & \left. + \frac{\mathbf{H}_{\text{HLLD},i,j,k+\frac{1}{2}} - \mathbf{H}_{\text{HLLD},i,j,k-\frac{1}{2}}}{\Delta z} \right) \end{aligned} \quad (109)$$

(関数 `time_evol`) ここで, i, j, k は格子点のインデックスである. 半整数インデックスはセル境界上の点を表す. ただし, ψ に関しては, 源泉項を評価するため, 式 (109) を評価したあと, 式 (108) に従って ψ_{next} を決定する.

6.5 MUSCL 補間・TVD 条件

MUSCL 補間および TVD 条件を満たすための流束制限は, 3 次元オイラー方程式の場合と同様である.

6.6 初期条件

3次元 MHD 方程式のサンプルコードでは, Orszag-Tang の渦問題 [5] を解く. 初期条件は,

$$\rho = \gamma^2 \quad (110)$$

$$u = -\sin(y) \quad (111)$$

$$v = \sin(x) \quad (112)$$

$$w = 0 \quad (113)$$

$$p = \gamma \quad (114)$$

$$B_x = -\sin(y) \quad (115)$$

$$B_y = \sin(2x) \quad (116)$$

$$B_z = 0 \quad (117)$$

$$\psi = 0 \quad (118)$$

である. また $\gamma = \frac{5}{3}$ である.

最大信号速度は $S_{\max} = 3$ と仮定し, クーラン数は $C_{\text{CFL}} = 0.3$ とする. $\Delta t = 0.1\Delta x$ である. また, 9-wave 法の c_h は, $c_h = S_{\max}$ とする.

設定ファイルでは, x 軸および y 軸方向の計算領域の幅を 2π としている. グリッド数は x,y 軸方向に 200, z 軸方向に 5 とする. 格子幅は $\Delta x = \Delta y = \Delta z = \frac{\pi}{100}$ である.

6.7 ドライバコード

6.7.1 ドライバコードの動作

ドライバコードはタイムステップ 1000($t = \pi$) まで計算して終了する.

6.7.2 出力ファイル形式

出力ファイル形式は 1次元 MHD のときと同様である. ただし, ψ のカラムが追加されている. 3次元オイラー方程式のサンプルコードと同様に, 内側から, x 軸,y 軸,z 軸の順番でループを回し, 格子を走査しながら各格子点のデータをファイルに出力する.

6.8 実行結果

ディレクトリ `sample/mhd_3d_hl1d/` 上で `make run plot` を実行すると, Formura のコンパイル・実行が行われ, 図 4, 5, 6 のようなグラフが出力される.

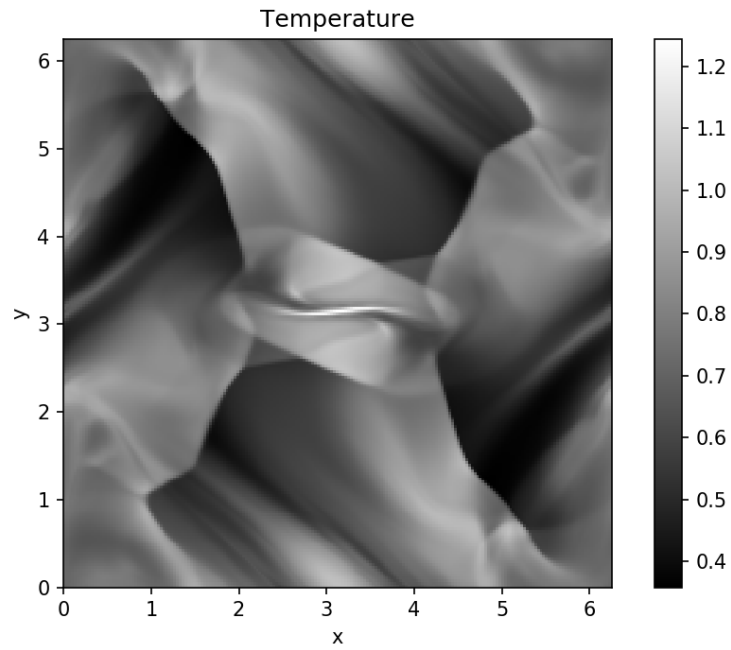


図4 HLLD法を用いた Orszag-Tang 渦問題の解（温度分布）.

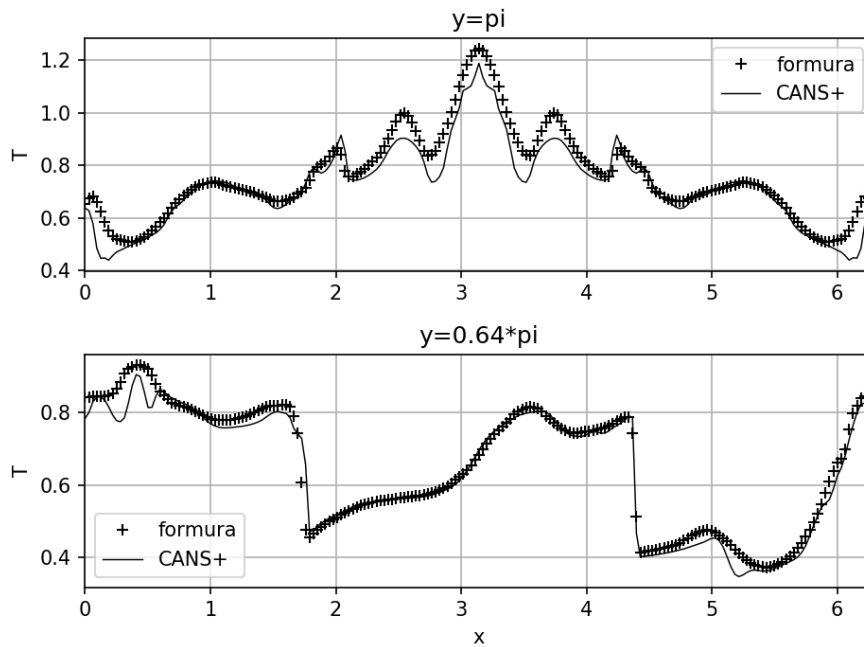


図5 HLLD法を用いた Orszag-Tang 渦問題の解（図4の断面）. 実線はCANS+の計算結果，十字点はFormuraの計算結果. 上のパネルが $y = \pi$ の断面，下のパネルが $y = 0.64\pi$ の断面に対応する.

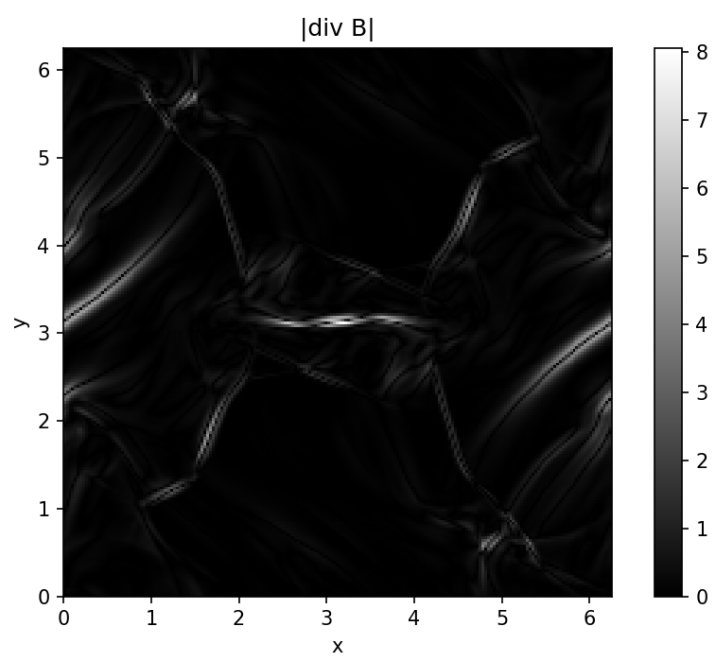


図 6 $|\text{div } \mathbf{B}|$ のグレースケールマップ.

参考文献

- [1] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, 2009.
- [2] Takahiro Miyoshi and Kanya Kusano. A multi-state hll approximate riemann solver for ideal magnetohydrodynamics. *Journal of Computational Physics*, Vol. 208, No. 1, pp. 315 – 344, 2005.
- [3] Yosuke Matsumoto, Yuta Asahina, Yuki Kudoh, Tomohisa Kawashima, Jin Matsumoto, Hiroyuki R. Takahashi, Takashi Minoshima, Seiji Zenitani, Takahiro Miyoshi, and Ryoji Matsumoto. Magnetohydrodynamic Simulation Code CANS+: Assessments and Applications. *arXiv e-prints*, p. arXiv:1611.01775, Nov 2016.
- [4] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic Divergence Cleaning for the MHD Equations. *Journal of Computational Physics*, Vol. 175, pp. 645–673, January 2002.
- [5] S. A. Orszag and C.-M. Tang. Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *Journal of Fluid Mechanics*, Vol. 90, pp. 129–143, January 1979.