

国立天文台 天文シミュレーションプロジェクト

## 流体学校の手引

# Contents

<b>1</b>	<b>流体方程式の数値解法</b>	<b>1</b>
1.1	一次元数値流体計算コード . . . . .	2
1.1.1	空間1次精度スキーム . . . . .	2
1.1.2	プログラミングするための準備 . . . . .	3
1.1.3	ゴーストセルを使った境界条件の処理 . . . . .	5
1.1.4	時間空間一次精度サンプルプログラムでの計算手順 . . . . .	6
1.2	実習課題 . . . . .	9
1.2.1	衝撃波管問題 . . . . .	9
1.2.2	音波の伝播 . . . . .	12
<b>2</b>	<b>磁気流体力学の数値解法</b>	<b>13</b>
2.1	1次元磁気流体方程式 . . . . .	13
2.2	双曲磁場発散除去法 . . . . .	14

## Chapter 1

# 流体方程式の数値解法

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0 \quad (1.1)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P \\ \rho v_x v_y \\ \rho v_x v_z \\ (E + P)v_x \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v_y \\ \rho v_y v_x \\ \rho v_y^2 + P \\ \rho v_y v_z \\ (E + P)v_y \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho v_z \\ \rho v_z v_x \\ \rho v_z v_y \\ \rho v_z^2 + P \\ (E + P)v_z \end{pmatrix}, \quad (1.2)$$

## 1.1 一次元数値流体計算コード

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (1.3)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P \\ (E + P)v_x \end{pmatrix} \quad (1.4)$$

天文学の多くのシミュレーションでは、有限体積法が用いられている。空間をセルに分割し、セル内の物理量の時間発展を、セルの両側の境界で流れる流束の差し引きで求める。セル中心の座標を  $x_i$  とし、セル境界の座標を  $x_{i+1/2}$  とする。 $i$  番目のセルの保存量を  $U_i$  とし、その左と右の境界を流れる流束を  $F_{i-1/2}$  と  $F_{i+1/2}$  とすると、有限体積法による保存量の時間進化は以下の式で表現される。

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} (F_{i+1/2} - F_{i-1/2}) \quad (1.5)$$

詳しくは講義で話されるが、 $x_{i+1/2}$  での数値流束を、 $x_i$  と  $x_{i+1}$  の保存量を使って出した流束の単純な平均 (例えば、質量保存式の場合  $F_{i+1/2} = \{(\rho v_x)_i + (\rho v_x)_{i+1}\}/2$ ) として表して計算すると、移流方程式での中心差分に相当し、数值的に不安定となる。

天文シミュレーションで用いられる多くのコードでは、セル境界の左と右の状態を使って Riemann 問題を解いた結果からセル境界での数値流束を評価する Godunov 法と呼ばれる手法が用いられる。形式的に書くと、

$$F_{i+1/2} = \text{RiemannProblem}(U_{L,i+1/2}, U_{R,i+1/2}) \quad (1.6)$$

となる。

### 1.1.1 空間 1 次精度スキーム

空間 1 次精度のスキームでは、右と左の物理量にセル中心の値をつかう。

$$F_{i+1/2} = \text{RiemannProblem}(U_i, U_{i+1}) \quad (1.7)$$

MUSCL 法を使った空間 2 次精度のスキームでは、セル中心からセル境界に線形補間した

### 1.1.2 プログラミングするための準備

Fortran を使ってプログラミングをおこなうための準備をする。 $x_i$  や  $\rho_i$ 、 $F_{i+1/2}$  など、空間に依存する量はセルの番号をインデックスとする配列を使って表現する。注意するのは、有限体積法ではセル中心で定義されている物理量 ( $\rho_i$ ,  $P_i$  など) とセル境界で定義されている物理量 ( $F_{i+1/2}$ ) があることである。Fortran では、配列のインデックスは整数でなければならないので、数値流束  $F_{i+1/2}$  の半整数  $i+1/2$  をインデックスにできない。たとえば、数値流束  $F_{i+1/2}$  を  $F(i)$  と表現することにして、頭の中で  $1/2$  だけずらす必要がある。

#### 座標に関する変数と配列

図 1.1 に示す通り、計算領域を  $x_{\min} \leq x \leq x_{\max}$  とし (変数名 `x_min`、`x_max`)、空間を  $N_{\text{grid}}$  個 (変数名 `ngrid`) のセルに分割する。

??節で述べた通り、境界条件を課すために使うゴーストセルを左右に用意する (図 1.1 の赤破線)。ゴーストセルの数は、Riemann 問題の左右の状態を求める手法に依存する。例えば、空間 2 次精度 MUSCL 法では、ゴーストセルは計算領域を挟み左右にそれぞれ 2 つ必要である。ゴーストセルの数を、自由に変更できるようにするため、`mgn` という変数にする。

図 1.1 に示すように、セル中心境界  $x_{i-1/2}$  を表す配列を `x1f(i)` とし、セル中心座標を表す配列  $x_i$  を `x1v(i)` とする。実は fortran では配列の開始インデックスは任意の整数から始められるが、ここではデフォルトの 1 とする (C 言語の場合 0)。ゴーストセルが加わるため、計算領域左端のセル中心座標のインデックスは 1 ではない値となり、これを `is` とする (これは `mgn+1` と等しい)。同様に右端のセル中心座標のインデックスを `ie` とする (これは `ngrid+mgn` と等しい)。セル境界座標の `x1f(is)` は  $x_{\min}$  と等しく、`x1f(ie+1)` は  $x_{\max}$  に等しい。

セル中心座標 `x1v` の総要素数は `ngrid+2*mgn` で、セル境界座標 `x1f` の総要素数は `ngrid+2*mgn+1` である。サンプルプログラムでは大きい方 `ngrid+2*mgn+1` を `in` としている。

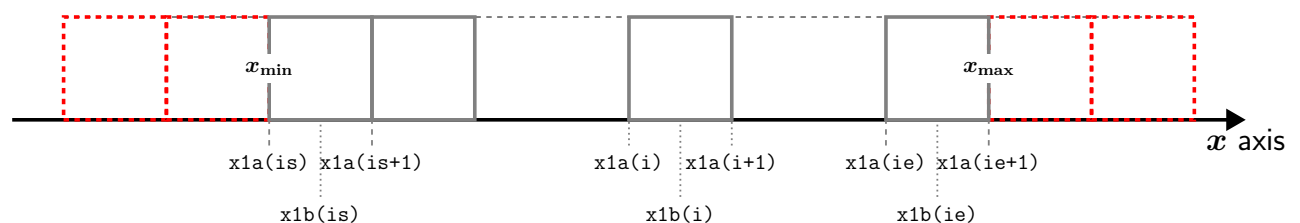


Figure 1.1:

Table 1.1: 座標に関する変数

変数名/配列名	説明
<code>nx</code>	計算領域内のセル総数
<code>ngh</code>	ghost cell 数
<code>nxtot</code>	ghost cell を含めたセル総数
<code>is</code>	計算領域左端のセル番号
<code>ie</code>	計算領域右端のセル番号
<code>x1min</code>	計算領域左端の座標
<code>x1max</code>	計算領域右端の座標
<code>x1c(i)</code>	セル中心の座標 $x_i$ (要素数 <code>nxtot</code> )
<code>x1b(i)</code>	セル境界の座標 $x_{i+1/2}$ (要素数 <code>nxtot+1</code> )

### 流体変数に関する変数と配列

物理量は、 $x$  と  $t$  の 2 つの従属変数をもつので、空間方向のセル番号と時間ステップの番号を引数とする 2 次元配列、例えば `d(i,n)` で定義するのが自然と思うかもしれない。しかし、例えばオイラー時間積分法で解く場合、 $t = t^{n+1}$  での  $U$  を求めるとき、前の時刻  $t = t^n$  の全空間の  $U$  のデータを保存しておけば十分で、それ以前  $t \leq t^{n-1}$  のデータは不要である。したがって、通常、時間ステップ番号を引数とすることはなく、空間 1 次元のシミュレーションをおこなう場合は、セル番号を引数とする配列 (例えば `U(i)`) を宣言する。多段の時間積分法を使う場合は、サブステップの物理量を保存しておく必要があるため、必要な数の配列を宣言する。

サンプルコードでは、保存量と基本量をそれぞれ 2 次元配列 `U(i,n)` と `W(i,n)` で定義している。一番目の引数 `i` はセル番号を表し、二番目の引数 `n` は流体変数のインデックスを表す。1 次元流体計算では式 (1.4) で示すように、3 つの要素をもつ。たとえば、`U(i,0)` は、`i` 番目のセルの密度を表し、`W(i,2)` は `i` 番目のセルの圧力を表す。プログラム内で、番号で流体変数の種類を指定するのは間違いの元なので、変数名を流体変数と対応付けた整数変数を定義する。詳しくは表 1.1.2 を参照。

表 1.1.2 にサンプルプログラムで使われている配列を示す。

Table 1.2: 流体変数に関する変数

変数名/配列名	説明
<b>基本量</b> (primitive variables) (IDEN=0, IVX=1, IPRE=2)	
W(i, IDN)	$x = x_i$ における密度 (要素数 <code>nxtot</code> ),
W(i, IV1)	$x = x_i$ における速度 (要素数 <code>nxtot</code> )
W(i, IPR)	$x = x_i$ における圧力 (要素数 <code>nxtot</code> )
<b>保存量</b> (primitive variables) (IDEN=0, IMX=1, IENE=2)	
U(i, IDN)	$x = x_i$ における密度 (要素数 <code>nxtot</code> )
U(i, IM1)	$x = x_i$ における運動量 (要素数 <code>nxtot</code> )
U(i, IEN)	$x = x_i$ における全エネルギー (要素数 <code>nxtot</code> )
<b>セル境界の流束</b> (numerical flux) (IDEN=0, IMX=1, IENE=2)	
F(i, IDN)	$x = x_{i+1/2}$ における質量流束 (要素数 <code>nxtot+1</code> )
F(i, IM1)	$x = x_{i+1/2}$ における運動量流束 (要素数 <code>nxtot+1</code> )
F(i, IEN)	$x = x_{i+1/2}$ におけるエネルギー流束 (要素数 <code>nxtot+1</code> )

### 1.1.3 ゴーストセルを使った境界条件の処理

流体方程式は、偏微分方程式なので解くためには境界条件が必要となる。時間方向の境界条件は初期条件と呼ばれ、シミュレーション上では、 $t = 0$  において、全てのセルの  $U$  に値を代入することに対応する。

現実の空間は無限に広がっているが、シミュレーションをおこなう場合は、有限の計算領域を設定せざるを得ず、必ず空間方向の境界条件 (以下では単に境界条件と呼ぶ) を与える必要がある。

代表的な境界条件としては、

- 周期境界条件  $U(x + L) = U(x)$
- ディリクレ境界条件  $\partial U / \partial x = U'_0$ .
- ノイマン境界条件  $U = U_0$

がある。その他、問題設定に応じて適切な条件を設定する。

左端セルの密度の時間発展は、

$$U(is, IDEN) = U(is, IDEN) - dt / (x1f(is+1) - x1f(is)) * (F(is+1, IDEN) - F(is, IDEN))$$

となる。ここで  $F(is, IDEN)$  は  $U(is-1, IDEN)$  と  $U(is, IDEN)$  から計算されるはずだが、 $U(is-1, IDEN)$

は計算領域外にあるので、このままでは  $F(is, IDEN)$  が計算できない。計算領域の右端のセルも同様に時間発展できない。

境界条件を設定する方法は色々ある。たとえば、計算領域の境界における数値流束  $F(IDEN, is)$  を、境界条件を満たすように直接与えるのも一つの手である。流束が 0 の境界条件であれば、 $dflux(is)=0$  にすればよい。ただその場合は、境界のセルを特別扱いする必要がある。

昨今の公開コードで多く用いられているのは、計算領域の外に「ゴースト」セルを用意する方法である。設定したい境界条件が満たされるようにゴーストセルに値を事前に代入すれば、計算領域の境界と接しているセルも、そうでないセルと全く同じ手順で計算できるため、コードが簡潔になる。

- 周期境界条件の場合  $U(x+L) = U(x)$

$x1v(is-1)$  と  $x1v(ie)$  が同一視される。また、 $x1v(ie+1)$  と  $x1v(is)$  が同一視される。

したがって、ゴーストセルに代入すべき値は、

$$U(is-1, IDN) = U(ie, IDN)$$

$$U(ie+1, IDN) = U(is, IDN)$$

- ディレクレ条件の場合  $\partial U / \partial x = U'_0$

左端に着目すると、計算領域の右端の境界  $xf(is)$  での  $U$  の勾配を差分化すると、

$$\frac{U_{is} - U_{is-1}}{\Delta x} = U'_0 \quad (1.8)$$

となる。したがって、 $U_{is-1} = U_{is} - U'_0 \Delta x$  となる。同様に、左端の境界  $xf(ie+1)$  では、 $U_{ie+1} = U_{ie} + U'_0 \Delta x$  が得られる。

- ノイマン条件の場合  $U = U_0$

物理量がセル中心でしか定義されていないので、ゴーストセルの値を使って、セル境界の値を最も簡単に表すと、 $U(x_{is}) = (U_{is} + U_{is-1})/2$  となる。これが  $U_0$  に等しいので、 $U_{is-1} = 2U_0 - U_{is}$  が得られる ( $U_0$  と  $U_{is-1}$  を使った外挿になっている)。右端の境界でも同様にして、 $U_{ie+1} = 2U_0 - U_{ie}$  を得る。

#### 1.1.4 時間空間一次精度サンプルプログラムでの計算手順

以下にサンプルプログラムの計算手順を示す (図 1.2 のフローチャットも参照)。対応する関数およびサブルーチンを載せている。

引数に色を付けている。赤がインプットに対応する変数/配列 (サブルーチン内で変更しない) で、青がサブルーチン内で値を代入する変数/配列を表す。



## 1. シミュレーションを始める前の準備

- まず、セル境界とセル中心の座標の設定をする。
  - `GenerateGrid( xf, xv )`
- 基本量の初期分布を設定する。
  - `GenerateProblem( x1c, x1b, W )`
- 保存量に変換
  - `ConsvVariable( W, U )`

## 2. シミュレーションのメインループ (ここで物理量の時間を更新する)

- CFL 条件を満たすように  $\Delta t$  を設定する。
  - `Real(8) Function TimeControl( x1a, W )`
- 境界条件を設定するためにゴーストセルに適切な値を代入する。
  - `BoundaryCondition(W)`
- 数値流束を求める。
  - `NumericalFlux( x1a, xv, W, F )`
- 保存量を更新する。
  - `UpdateConsv( dt, xf, Uold, F, U )`
- 保存量から基本量を求める。
  - `PrimVariable(U, W)`
- 必要なら結果を出力する。
  - `Output( x1a, x1b, U, W )`
- 時間を更新し (`time = time + dt`)、シミュレーションの終了条件を満たしているか確認し、満たしていたらメインループを抜ける。満たしていなかったら、メインループの始めに戻る。

## サンプルコードで使われている変数の説明

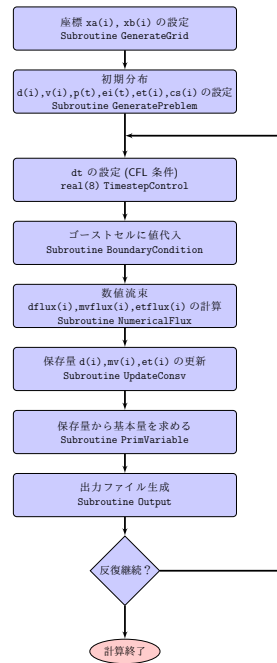


Figure 1.2: サンプルプログラムの計算手順のフローチャート。

Table 1.3: 時間発展に関数する変数

変数名/配列名	説明
nhy	時間ステップ数
nhymax	最大時間ステップ数
time	時刻
timemax	計算終了時刻
dt	時間幅

Table 1.4:  $nhy > nhymax$  または  $time > timemax$  を満たすと計算が終了する。

## 1.2 実習課題

### 1.2.1 衝撃波管問題

まずは問題設定が単純な衝撃波管問題でコードが正しく動いているかを確認する。

#### 理論

詳しくは講義資料を参照のこと。不連続面で仕切られた2つの一様なガスの時間進化には、厳密解が知られており、数値流体計算コードが正しく動いているかをテストするためによく使われる。

Sod 解は有名な衝撃波管問題の解で、計算コードの論文によく登場する。左右の物理量を添え字”L”と”R”を使ってそれぞれ表すと、Sod 解の初期条件は、

$$\begin{pmatrix} \rho_L \\ v_L \\ P_L \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \rho_R \\ v_R \\ P_R \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0 \\ 0.1 \end{pmatrix} \quad (1.9)$$

である。伝統的に Sod 解での比熱比は  $\gamma = 1.4$  が使われる。

計算結果と比較するための解析解を生成する python スクリプト (`RiemannSolver.py`) を用意した。すでに生成したファイル `sod_ana.dat` (時刻  $t = 0.2$  での解析解。1 列目は  $x$ 、2 列目は  $\rho$ 、3 列目は  $v$ 、4 列目は  $P$ 。) をディレクトリの中に入れている。

図 1.3 は  $t = 0.2$  における厳密解を表す。左のガスの方が高圧なので、右のガスから左のガスが押されて、右のガスに衝撃波が伝わり、逆に右のガスには希薄波が生じる。

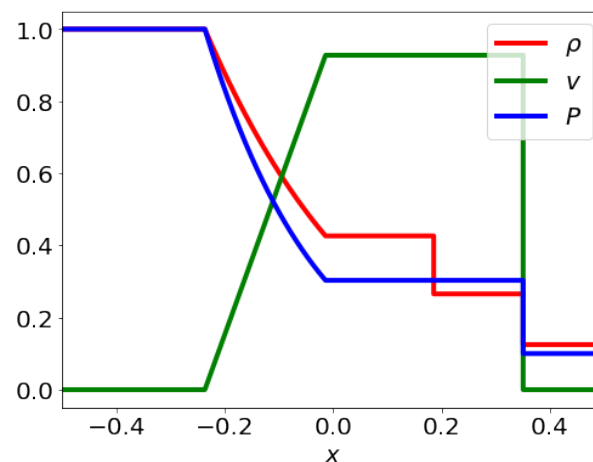


Figure 1.3:  $t = 0.2$  における Sod 解

## 出力

サンプルコードでは、dtout 毎にスナップショットファイルの出力をおこなっている。対応するサブルーチンは Output である。スナップショットファイルは、ディレクトリ snap/の下に作られる。

## 可視化

可視化はシミュレーション結果を確認する強力な方法である。この実習では計算コードを実装しつつ、衝撃波管問題の解析解と比較する。参考のために gnuplot と python による可視化のためのサンプルコードを用意した。

- gnuplot によるインタラクティブな可視化

コマンドラインで gnuplot と入力すると起動する。

```
gnuplot> plot "snap/t00001.dat" using 1:2
```

と入力すると横軸を 1 列目、縦軸を 2 列目にとったグラフが出力される。

解析解と同時に出力したい場合は、

```
gnuplot> plot "snap/t00001.dat" using 1:2 w lp, "sod_ana.dat" using 1:2 w l
```

using の後の列を指定する箇所は、演算した結果を使うこともできる。

たとえば、using (\$1\*2):(\$2\*\$3) にすると、1 列目を 2 倍した値を横軸とし、2 列目と 3 列目の積の値を縦軸にとったプロットを作れる。

- gnuplot によるパラパラ漫画出力

gnuplot の出力画面上で、パラパラ漫画のように各時刻のスナップショットを連続的に表示できる。コマンドラインで、

```
gnuplot RealtimeAnime.plt
```

を実行する。gnuplot を起動した上で実行する場合は、load "RealtimeAnime.plt"を実行する。

- gnuplot による動画の作成

MakeAnime.sh は、snap/に出力されたファイルをつかって、自動で動画を作るスクリプトである。

コマンドラインで、

```
./MakeAnime.sh
```

を実行すると、動画ファイル `animation.mp4` が作成される。再生する場合は、コマンドラインで

```
mplayer animation.mp4
```

を実行する。

- python による動画の作成

コマンドラインで

```
python MakeAnime.py
```

を実行する。

## 課題

サンプルプログラムに以下を実装し

- 数値流束 HLL 法 (HLL 法を実装したあと余力があれば HLLC 法も)
- outflow 境界条件の設定 (すべての基本量の勾配を境界で 0 にする) `BoundaryCondition`

シミュレーション結果が厳密解と整合的か確認する。特に接触不連続面の両側の物理量が厳密解と一致しているか。

## 1.2.2 音波の伝播

## 理論

基礎方程式は以下になる。ここで便利のために全エネルギー保存の式を断熱の式に置き換えている。

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial x} = 0 \quad (1.10)$$

$$\rho \left( \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial x} \right) = - \frac{\partial P}{\partial x} \quad (1.11)$$

$$\left( \frac{\partial}{\partial t} + v \frac{\partial}{\partial x} \right) \ln \left( \frac{P}{\rho^\gamma} \right) = 0 \quad (1.12)$$

非摂動状態として一様な静止したガスを考える (密度  $\rho = \rho_0$ 、速度  $v = 0$ 、圧力  $P = P_0$ )。摂動量を  $\delta \rho$ 、 $\delta v$ 、 $\delta P$  とし、流体方程式を線形化すると、

$$\frac{\partial \delta \rho}{\partial t} + \rho_0 \frac{\partial \delta v}{\partial x} = 0 \quad (1.13)$$

$$\rho_0 \frac{\partial \delta v}{\partial t} = - \frac{\partial \delta P}{\partial x} \quad (1.14)$$

$$\frac{\delta P}{P_0} - \gamma \frac{\delta \rho}{\rho_0} = 0 \quad \Rightarrow \quad \delta P = c_s^2 \delta \rho, \quad (1.15)$$

ここで、 $c_s = \sqrt{\gamma P_0 / \rho_0}$  は音速である。

式 (1.15) を式 (1.14) に代入し、式 (1.13) を使って、 $\delta v$  を消去すると、以下の  $\delta \rho$  についての波動方程式を得る。

$$\frac{\partial^2 \delta \rho}{\partial t^2} - c_s^2 \frac{\partial^2 \delta \rho}{\partial x^2} = 0 \quad (1.16)$$

ちなみに  $\delta v$  と  $\delta P$  も全く同じ波動方程式に従う。波動方程式の厳密解は、よく知られており、波形を保ったまま左右に伝播する波を表す ( $\delta \rho(x, t) = g(x - ct) + f(x + ct)$ )。

## 課題 1

密度摂動が、

$$\delta \rho(x, t) = A \sin(k(x - c_s t)) \quad (1.17)$$

に従って、右に伝わる様子をシミュレーションしよう。ここで  $k$  は波数である。

密度摂動が上の式で与えられたとき、速度摂動と圧力摂動がどう与えられるかは、摂動方程式からわかるので、適切な初期条件を与えること。

## Chapter 2

# 磁気流体力学の数値解法

### 2.1 1次元磁気流体方程式

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

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \\ B_y \\ B_z \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P_T - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ (E + P_T)v_x - B_x(\mathbf{v} \cdot \mathbf{B}) \\ B_y v_x - B_x v_y \\ B_z v_x - B_x v_z \end{pmatrix} \quad (2.2)$$

ここで、 $E = P/(\gamma - 1) + \rho \mathbf{v}^2/2 + \mathbf{B}^2/2$  は全エネルギー、 $P_T$  は全圧力 ( $P + \mathbf{B}^2/2$ ) を表す。 $B_x$  は  $\nabla \cdot \mathbf{B} = 0$  から一定である必要がある。

## 2.2 双曲磁場発散除去法

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S} \quad (2.3)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ E \\ B_x \\ B_y \\ B_z \\ \psi \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + P_T - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ (E + P_T)v_x - B_x(\mathbf{v} \cdot \mathbf{B}) \\ \psi \\ B_y v_x - B_x v_y \\ B_z v_x - B_x v_z \\ c_h^2 B_x^2 \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \psi/\tau \end{pmatrix} \quad (2.4)$$