課題演習 C1 最終レポート

0500-34-0042　大平 達也

2024 年 8 月 16 日

1　はじめに

本稿では, 課題演習 C1 のまとめとして, 磁場なしの衝撃波管問題を通して数値計算の手法について概観していく。

2　問題設定・目的

衝撃波管問題は, 物理量 $\boldsymbol{U}$ が初期に不連続を持つときの, 1 次元 Euler 方程式の解を求める問題である。より具体的には, 図 1 のように, 十分に長い, 仕切られた管の片側に高圧の, もう片側に低圧の流体をいれ, 仕切りを取り去ることで衝撃波等が発生することがある, というものである。

図 1　衝撃波管問題のイメージ図

3　物理的背景

3.1　設定について

簡単のため, ここで扱う流体は非粘性・圧縮流体を仮定する。また, 管に沿った方向以外の方向での流体の速度はゼロであると仮定する。扱う基礎方程式を記す:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(rhou^2 + p) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}\left[\frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2\right] + \frac{\partial}{\partial x}\left[\left(\frac{\gamma}{\gamma - 1}p + \frac{1}{2}\rho u^2\right)u\right] = 0 \tag{3}$$

上から順に, 連続の式, 運動量保存則, エネルギー保存則を表す。

ただし, 以下では, これと同値である, 1 つにまとめた次の形を用いることがある:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} = 0 \tag{4}$$

ただし,

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho v \\ \rho E \end{pmatrix}, \tag{5}$$

$$\boldsymbol{F} = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ \rho\rho H v \end{pmatrix}. \tag{6}$$

である。ここで, $H = E + \frac{p}{\rho}$ は単位質量あたりのエンタルピーとした。

また, 初期条件について, 初期での不連続点は原点 $x = 0$ にあるとし, 初期条件を次のように書くこととする:

$$\boldsymbol{U}(x, 0) = \begin{cases} \boldsymbol{U}_{\mathrm{L}} & (x < 0), \\ \boldsymbol{U}_{\mathrm{R}} & (x \geq 0). \end{cases}$$

3.2 現象の説明

数値解の一例を図 2 に示す。この通り, 3 つの急激な値の変化がみられる。これは左から, 膨張波, 接触不連続面, 衝撃波である。接触不連続面は, 密度は不連続であるものの, 圧力と速度が連続的に接している面であり, この面を境として温度が異なるガスが圧力平衡で接している。

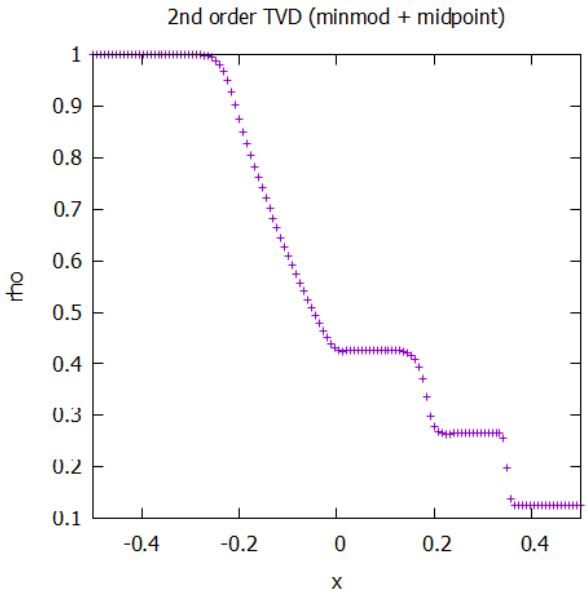

図 2　衝撃波管問題の数値解の一例

4　実装の方針

4.1　Riemann fan について

衝撃波管問題の解は, 衝撃波, 接触不連続面, 膨張波の組み合わせで構成され, 衝撃波, 接触不連続面, 膨張波はそれぞれ一定の速度で空間を伝播する。そのため, ある時刻の衝撃波管問題の解は, 別の時刻の解と相似的になっている。すなわち, $x-t$ 空間を考えると, 衝撃波, 接触不連続面, 膨張波の軌跡は原点から放射状に広がる。これを Riemann fan という (図 3 参照)。

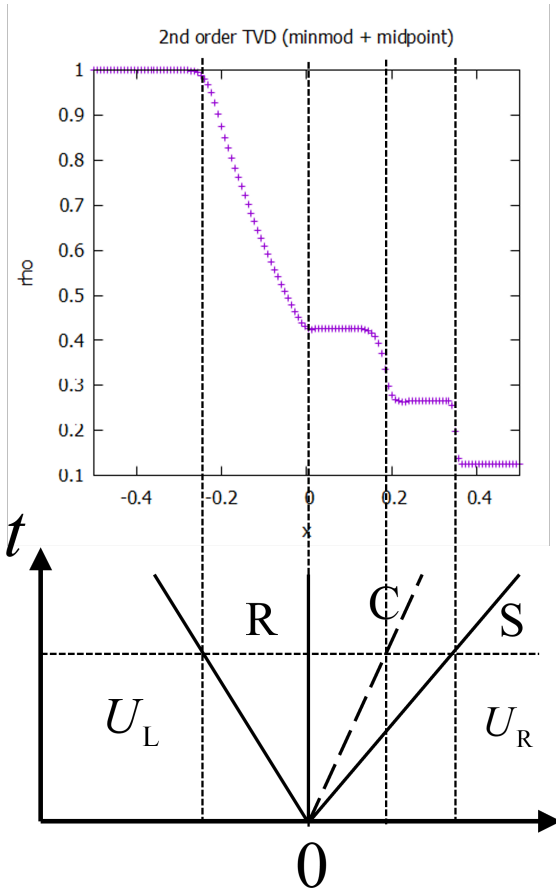

図 3　Riemann fan

4.2　有限体積法

まず, 今回数値計算を行う上で基本となる有限体積法を導出する。

Euler 方程式 (4) をセルの区間 $x_{i-1/2} \leq x \leq x_{i+1/2}$ において x で積分し, 時間刻みの区間 $t_n \leq t \leq t_{n+1}$ において積分する。その結果,

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{\Delta x}\left(\overline{\boldsymbol{F}}_{i+1/2} - \overline{\boldsymbol{F}}_{i-1/2}\right) \tag{7}$$

となる。ここで, 時刻 $t = t_n$ におけるセル i 内部の $\boldsymbol{U}$ の平均値を

$$\boldsymbol{U}_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{U}(x, t_n)dx \tag{8}$$

と定義した。また, セル境界 $x = x_{i+1/2}$ における時間区間 $t_n \leq t \leq t_{n+1}$ の流束 $\boldsymbol{F}$ の平均値を,

$$\overline{\boldsymbol{F}}_{i+1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \boldsymbol{F}(\boldsymbol{U}(x_{i+1/2}, t))dt \tag{9}$$

と定義する。

4.3 Godnov 法

前項で出てきた数値流束をどのように求めるかについて考えていく。

今回衝撃波管問題を解くのに用いたスキームは, Godnov スキームと呼ばれるものである。この節では, Godnov スキームについて概説する。

まず, 衝撃波管問題の解は, 左右の物理量の関数になっているため, 衝撃波管問題の解を

$$\boldsymbol{U}(x,t) = \mathcal{RP}\left(\frac{x}{t}, \boldsymbol{U}_{\mathrm{L}}, \boldsymbol{U}_{\mathrm{R}}\right) \tag{10}$$

と書くことができる。以下, $\mathcal{RP}$ を衝撃波管問題の解を返す関数として, この記法を用いる。

セル内の物理量 U が一定であるというのは, 有限体積法の考え方である。この考えのため, セル境界 $i + 1/2$ に不連続があると考えることができる。この不連続面において, 衝撃波管問題を考える。ここで, 時間刻み Δt の間に U が衝撃波管問題の解に従って変化し, 時間ステップの最後にセル内部の物理量 U を平均化することで, 区分的に一定の関数 $\boldsymbol{U}_i^{n+1}$ を求めるとすればよい。これが Godnov 法である。

このとき流束は,

$$\overline{\boldsymbol{F}}_{i+1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \boldsymbol{F}(\mathcal{RP}(0, \boldsymbol{U}_i^n, \boldsymbol{U}_i^{n+1}))dt \tag{11}$$

とかける[*1]。この関数 $\mathcal{RP}(0, \boldsymbol{U}_i^n, \boldsymbol{U}_i^{n+1})$ は時間的に一定であるため, 11 は,

$$\boldsymbol{F}_{I+1/2}^{*(\mathrm{Godnov})} = \boldsymbol{F}(\mathcal{RP}(0, \boldsymbol{U}_i^n, \boldsymbol{U}_i^{n+1})) \tag{12}$$

となる。これが Godnov 法の数値流束である。

4.4 Roe の方法

前節で, Godnov 法の数値流束について議論した。この説では, この数値流束の具体的な求め方について議論し, Roe の方法を説明する。

通常の Godnov 法においては, 数値流束を求める際に衝撃波管問題の解として厳密解を用いる。しかし, 衝撃波管問題の厳密解を求めるには, 反復法を用いる必要があり, 計算コストが高

[*1] ここで, セル境界 $x = x_{i+1/2}$ において衝撃波管問題を解くため, 第 1 変数はゼロになっている

い。そのため, 衝撃波管問題の近似解を用いることを考える。この近似解を求める方法として, 今回は Roe の方法を採用した。

まず, 流束は保存変数の関数として,

$$\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{U}) \tag{13}$$

とかけるため, 4 を次のように変形する:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x} = 0 \tag{14}$$

ここで, $\boldsymbol{A}$ は Jacobi 行列で,

$$\boldsymbol{A} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \begin{pmatrix} 0 & 1 & 0 \\ -(3-\gamma_{\mathrm{ad}})\frac{v^2}{2} & (3-\gamma_{\mathrm{ad}})v & \gamma_{\mathrm{ad}}-1 \\ -Hv+(\gamma_{\mathrm{ad}}-1)\frac{v^3}{2} & H-(\gamma_{\mathrm{ad}}-1)v^2 & \gamma_{\mathrm{ad}}v \end{pmatrix} \tag{15}$$

である。

$\boldsymbol{A}$ を対角化しておく。固有値 λ は,

$$\lambda_1 = v - c_s, \lambda_2 = v, \lambda_3 = v + c_s \tag{16}$$

右固有行列 $\boldsymbol{R}$ を,

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{r}^{(1)} & \boldsymbol{r}^{(2)} & \boldsymbol{r}^{(3)} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ v-c_s & v & v+c_s \\ H-vc_s & \frac{v^2}{2} & H+vc_s \end{pmatrix} \tag{17}$$

とする。これの逆行列を計算すると,

$$\boldsymbol{R}^{-1} = \begin{pmatrix} \frac{1}{2}\left(b_1+\frac{v}{c_s}\right) & -\frac{1}{2}\left(\frac{1}{c_s}+b_2 v\right) & \frac{1}{2}b_2 \\ 1-b_1 & b_2 v & -b_2 \\ \frac{1}{2}\left(b_1-\frac{v}{c_s}\right) & \frac{1}{2}\left(\frac{1}{c_s}-b_2 v\right) & \frac{1}{2}b_2 \end{pmatrix} \tag{18}$$

となり, これを $\boldsymbol{L}$ と定義する。ただし,

$$b_1 = \frac{\gamma_{\mathrm{ad}}-1}{c_s^2}\frac{v^2}{2}, b_2 = \frac{\gamma_{\mathrm{ad}}-1}{c_s^2} \tag{19}$$

とする。$\boldsymbol{R}, \boldsymbol{L}$ より, $\boldsymbol{A}$ が,

$$\boldsymbol{\Lambda} = \boldsymbol{L}\boldsymbol{A}\boldsymbol{R} \tag{20}$$

と対角化できた。ただし,

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{21}$$

である。

物理量は, セル境界を挟んで微小な変位 $\delta \boldsymbol{U} = \boldsymbol{U}_{i+1} - \boldsymbol{U}_i$ を持つが, 14 において行列 $\boldsymbol{A}$ は一定であると近似する[*2]。$\boldsymbol{A}$ が変わらないため, 固有値 λ_j や $\boldsymbol{L}, \boldsymbol{R}$ も一定である。

ここで, 変位 $\delta \boldsymbol{U}$ について, 新しい変数 $\partial \boldsymbol{W} = (\partial w_1, \partial w_2, \partial w_3)^{\mathrm{T}} = \boldsymbol{L}\delta \boldsymbol{U}$ を考える。各成分は,

$$\partial w_1 = \frac{\rho}{2c_s}\left(\frac{\delta p}{\rho c_s} - \delta v\right) \tag{22}$$

$$\partial w_2 = \delta \rho - \frac{\delta p}{c_s^2} \tag{23}$$

$$\partial w_3 = \frac{\rho}{2c_s}\left(\frac{\delta p}{\rho c_s} + \delta v\right) \tag{24}$$

次に, Euler 方程式 14 の第 2 項を離散化する。

$$\boldsymbol{A}\left(\frac{\partial \boldsymbol{U}}{\partial x}\right)^{(\mathrm{uw})} = \boldsymbol{R}(\boldsymbol{LAR})\boldsymbol{L}\left(\frac{\partial \boldsymbol{U}}{\partial x}\right)^{(\mathrm{uw})} \tag{25}$$

$$= \boldsymbol{R}\boldsymbol{\Lambda}\left(\frac{\partial \boldsymbol{W}}{\partial x}\right)^{(\mathrm{uw})} \tag{26}$$

$$= \begin{pmatrix} \boldsymbol{r}^{(1)} & \boldsymbol{r}^{(2)} & \boldsymbol{r}^{(3)} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \partial w_1/\partial x \\ \partial w_2/\partial x \\ \partial w_3/\partial x \end{pmatrix}^{(\mathrm{uw})} \tag{27}$$

$$= \sum_{j=1}^{3} \lambda_j \left(\frac{\partial w_j}{\partial x}\right)^{\mathrm{uw}} \boldsymbol{r}^{(j)} \tag{28}$$

となる。

すなわち, Euler 方程式を固有ベクトル $\boldsymbol{r}^{(\mathrm{j})}$ を用いて展開すると, ∂w_j に関する 3 つのスカラー線形波動方程式の重ね合わせとして書けることが分かった。

よって, それぞれの波について風上差分で評価すればよく, 数値流束は,

$$\boldsymbol{F}^*_{i+1/2} = \frac{1}{2}\left(\boldsymbol{F}_i + \boldsymbol{F}_{i+1}\right) - \frac{1}{2}\sum_{j=1}^{3} |\lambda_j| \partial w_j \boldsymbol{r}^{(j)} \tag{29}$$

となる。

ここから, 29 を非線形に拡張することを考える。すなわち, $\delta \boldsymbol{U}$ が有限の大きさのとき, セル境界における行列 $\boldsymbol{A}_{i+1/2}$ をどのように与えるかについて考えなければならない。

$\boldsymbol{A}$ は次の 3 つの性質を満たす行列 $\overline{\overline{\boldsymbol{A}}}$ であるべきである。

[*2] 線形近似である

- 物理量 $\boldsymbol{U}_i$, $\boldsymbol{U}_{i+1}$ について,

$$\boldsymbol{F}_{i+1} - \boldsymbol{F}_i = \overline{\overline{\boldsymbol{A}}}(\boldsymbol{U}_i, \boldsymbol{U}_{i+1})(\boldsymbol{U}_i - \boldsymbol{U}_{i+1}) \tag{30}$$

  が成立すること。
- 左右の物理量が等しいとき ($\boldsymbol{U}_i = \boldsymbol{U}_{i+1} = \boldsymbol{U}$),

$$\overline{\overline{\boldsymbol{A}}}(\boldsymbol{U}, \boldsymbol{U}) = \boldsymbol{A}(\boldsymbol{U}) \equiv \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} \tag{31}$$

  が成立すること。
- 行列 $\overline{\overline{\boldsymbol{A}}}$ が実数の固有値と線形独立な固有ベクトルを持つこと。

$\overline{\overline{\boldsymbol{A}}}$ がこの 3 条件を満たせば, 線形の場合の議論がそのまま非線形の場合に用いることができる。

$\boldsymbol{A}$ に現れる ρ, v, H を, 次で定義される $\overline{\rho}, \overline{v}, \overline{H}$ に置き換えて $\overline{\overline{A}}$ を計算すると, この 3 条件を満たす。

$$\overline{\rho}_{i+1/2} = \sqrt{\rho_i \rho_{i+1}} \tag{32}$$

$$\overline{v}_{1+1/2} = \frac{v_i\sqrt{\rho_i} + v_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{33}$$

$$\overline{\overline{H}}_{1+1/2} = \frac{H_i\sqrt{\rho_i} + H_{i+1}\sqrt{\rho_{i+1}}}{\sqrt{\rho_i} + \sqrt{\rho_{i+1}}} \tag{34}$$

これが Roe の方法である。

5 実装

コードを以下に記す。

Listing 1 衝撃波管問題 (Roe の方法)

```
1 #include <stdio.h>
2 #include <math.h>
3 #include <stdlib.h>
4 #define PI 3.14159265
5
6 #define gamma 1.4
7
8 double f(int i, double x){
9     double rhoL=1.;
```

```
10      double pL=1.;
11      double vL=0.9;
12      double rhoR=0.125;
13      double pR=0.1;
14      double vR=0.9;
15
16      if(i==0){ // rho
17          if(x<0) return rhoL;
18          else return rhoR;
19      }else if(i==1){ // rho*v
20          if(x<0) return vL*f(0, x);
21          else return vR*f(0, x);
22      }else if(i==2){ // rho*E
23          if(x<0) return pL/(gamma-1)+pow(f(1, x), 2.)/2./f(0,
                x);
24          else return pR/(gamma-1)+pow(f(1, x), 2.)/2./f(0, x);
25      }
26      else{
27          exit(1);
28      }
29  }
30
31  int main(void){
32      double a=-0.5;          // 始点
33      double b=0.5;           // 終点
34      double dx=1/128.;       // 空間刻みの幅
35      double nu=0.7;          // 数CFL
36      double dt=nu*dx/5;      // 時間刻みの幅
37      int I=(b-a)/dx+1;       // 空間ステップ数
38      int N=0.15/dt;          // 時間ステップ数
39
40      double u[I][3];         // 数値解を格納する配列
41      double tmp[I][3];       // 一時的に格納
42
43      // 初期条件の設定
```

```
44      for(int i=0; i<I; i++){
45          for(int index=0; index<3; index++){
46              u[i][index]=f(index, a+i*dx);
47              tmp[i][index]=u[i][index];
48          }
49      }
50
51      for(int step=0; step<N; step++){
52          // 解くパート
53          // 固定境界条件を課す
54          // を求めるflux
55          double flux[I-1][3];
56          for(int i=0; i<I-1; i++){
57              double rho0=u[i][0];
58              double rho1=u[i+1][0];
59              double v0=u[i][1]/rho0;
60              double v1=u[i+1][1]/rho1;
61              double E0=u[i][2]/rho0;
62              double E1=u[i+1][2]/rho1;
63              double p0=(gamma-1)*(rho0*E0-rho0*pow(v0, 2.)/2.)
                    ;
64              double p1=(gamma-1)*(rho1*E1-rho1*pow(v1, 2.)/2.)
                    ;
65              double H0=E0+p0/rho0;
66              double H1=E1+p1/rho1;
67
68              double brho=sqrt(rho0*rho1);
69              double bv=(v0*sqrt(rho0)+v1*sqrt(rho1))/(sqrt(
                    rho0)+sqrt(rho1));
70              double bH=(H0*sqrt(rho0)+H1*sqrt(rho1))/(sqrt(
                    rho0)+sqrt(rho1));
71
72              double bcs=(gamma-1)*(bH-pow(bv, 2.)/2.);
73
74              double lambda[3]={bv-bcs, bv, bv+bcs};
```

```
75
76              double r[3][3]={{1, bv-bcs, bH-bv*bcs},
77                              {1, bv, pow(bv, 2.)/2.},
78                              {1, bv+bcs, bH+bv*bcs}};
79
80              double drho=rho1-rho0;
81              double dv=v1-v0;
82              double dp=p1-p0;
83
84              double parw[3];
85              parw[0]=brho/2./bcs*(dp/brho/bcs-dv);
86              parw[1]=drho-dp/pow(bcs, 2.);
87              parw[2]=brho/2./bcs*(dp/brho/bcs+dv);
88
89              double f0[3];
90              f0[0]=rho0*v0;
91              f0[1]=rho0*pow(v0, 2.)+p0;
92              f0[2]=rho0*H0*v0;
93
94              double f1[3];
95              f1[0]=rho1*v1;
96              f1[1]=rho1*pow(v1, 2.)+p1;
97              f1[2]=rho1*H1*v1;
98
99              for(int j=0; j<3; j++){
100                 double tmpf=0;
101                 for(int k=0; k<3; k++){
102                     tmpf += fabs(lambda[k])*parw[k]*r[k][j];
103                 }
104                 flux[i][j]=(f0[j]+f1[j]-tmpf)/2.;
105             }
106         }
107
108         for(int i=0; i<I-2; i++){
109             for(int j=0; j<3; j++){
```

```
110                 tmp[i+1][j]=u[i+1][j]-dt/dx*(flux[i+1][j]-
                        flux[i][j]);
111             }
112         }
113
114         for(int i=0; i<I; i++){
115             for(int j=0; j<3; j++){
116                 u[i][j]=tmp[i][j];
117             }
118         }
119     }
120
121     // ファイルの用意
122     char filename[256];
123     sprintf(filename, "out.data");
124     FILE *file = fopen(filename, "w");
125     if(file == NULL){
126         perror("Failed to open file. ");
127         exit(1);
128     }
129
130     // ファイルへの書き込み
131     for(int i=0; i<I; i++){
132         double rho=u[i][0];
133         double v=u[i][1]/rho;
134         double E=u[i][2]/rho;
135         double p=(gamma-1)*(rho*E-rho*pow(v, 2.)/2.);
136         double H=E+p/rho;
137         double cs=(gamma-1)*(H-pow(v, 2.)/2.);
138         double S=log(p/pow(rho, gamma));
139
140         fprintf(file, "%e, %e, %e, %e, %e\n", a+i*dx, rho, v/
                cs, S, v-cs);
141     }
142
```

```
143     fclose(file);
144
145     return 0;
146 }
```

6　実行結果

上記コードを実行したのち, gnuplot で可視化した図を以下に示す。

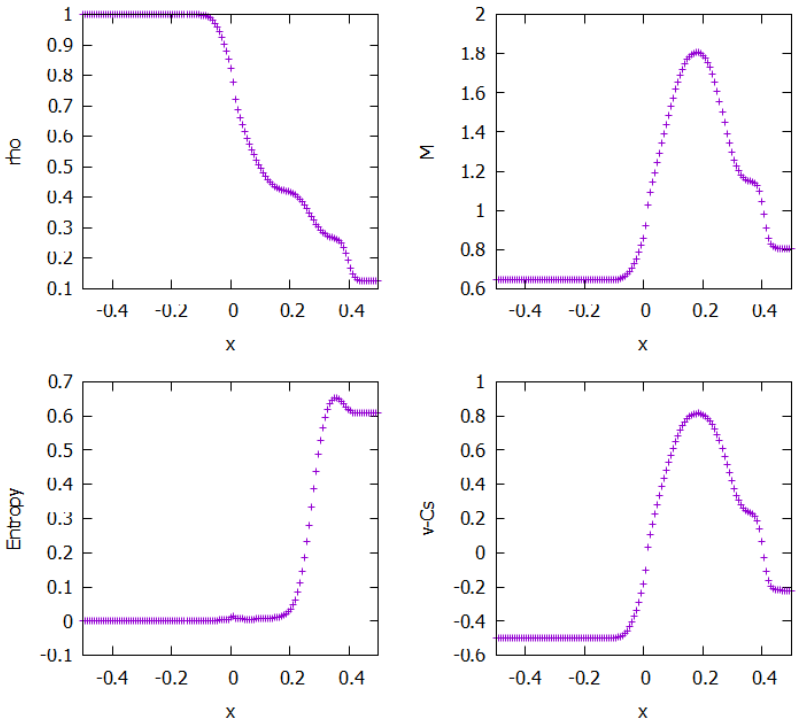

図 4　衝撃波管問題を Roe の方法で解いた plot

全体的になまった解ではあるが, 衝撃波と接触不連続面については安定に解けている。しかし, 膨張波について不連続が発生している。これは, 膨張衝撃波と呼ばれる, 現実には存在しない解である。これの対処について次の節で述べる。

7　エントロピー補正

通常の衝撃波では, 流体は衝撃波に対して超音速で流入し, 亜音速で流出し, その際に流体の運動エネルギーが熱エネルギーに変換される。この状況を時間反転した状態を考えると, エントロピーが減少してしまい, 物理的にありえない。これが膨張衝撃波である。

膨張衝撃波を抑制する方法として, エントロピー補正を考える。不連続面で波の速度がゼロにならないように補正する。そのため, Roe の数値流束に現れる $|\overline{\overline{\lambda}}|_{i+1/2}$ を次の $|\overline{\overline{\lambda}}|_{\mathrm{mod}}$ に置き換えて, 波の速度を補正する。この方法は膨張波において数値粘性を多めに加えることで膨張衝撃波をなまらせる方法であると考えることもできる。

$$|\overline{\overline{\lambda}}|_{\mathrm{mod}} = \begin{cases} |\overline{\overline{\lambda}}|_{i+1/2} & (|\overline{\overline{\lambda}}|_{i+1/2} \geq \epsilon), \\ \frac{1}{2}\left(\frac{\overline{\overline{\lambda}}^2_{i+1/2}}{\epsilon} + \epsilon\right) & (|\overline{\overline{\lambda}}|_{i+1/2} < \epsilon). \end{cases}$$

$$\epsilon = \max[0, \overline{\overline{\lambda}}_{i+1/2} - \lambda_i, \lambda_{i+1} - \overline{\overline{\lambda}}_{i+1/2}] \tag{35}$$

これを用いて plot した結果を次に示す。

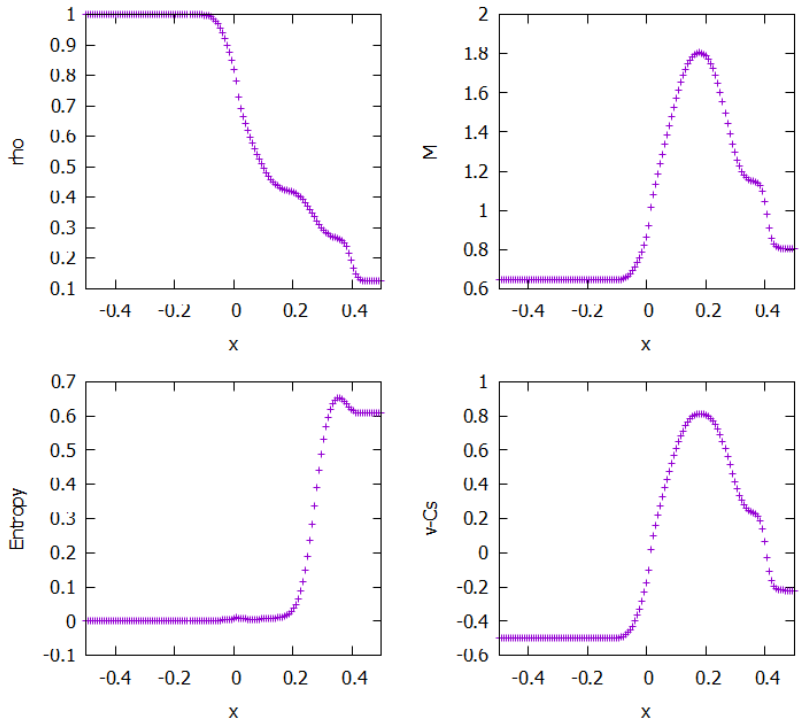

図 5　エントロピー補正したときの plot

8　高次元化

まず空間 2 次元の手法として MUSCL を用いた。そこに制限関数を後で導入する。次に時間についても 2 次元にしたい。本稿では中点法と Heun 法を用いる。

8.1　MUSCL

区間 $x_{i-1/2} < x < x_{i+1/2}$ 内で, 物理量を次のように展開する:

$$u(x) = u_i + (x - x_i)\left(\frac{\partial u}{\partial x}\right)_i + \frac{3\kappa}{2}\left[(x - x_i)^2 - \frac{(\Delta x)^2}{12}\right]\left(\frac{\partial^2 u}{\partial x^2}\right)_i \tag{36}$$

ここで, u_i はセル内の平均値とする。$\kappa = 1/3$ のとき, 36 は Taylor 展開に 2 次まで一致するため, 3 次精度になる。$\kappa \neq 1/3$ のときは 2 次精度となる。

数値流束を求める際に, セル境界の値が必要になる。これまではセル内の値を一定として数値流束を計算したが, MUSCL では 36 をにより, より近い値を補間し, それを用いることで高次元化を実現する。

8.2　制限関数

MUSCL を用いたときに解が振動してしまうことがある。たとえば図**??**は, $\kappa = -1$ としたときに, セル境界の値 $u^{\mathrm{L}}_{i+1/2}$ を求めようとしている状況である。

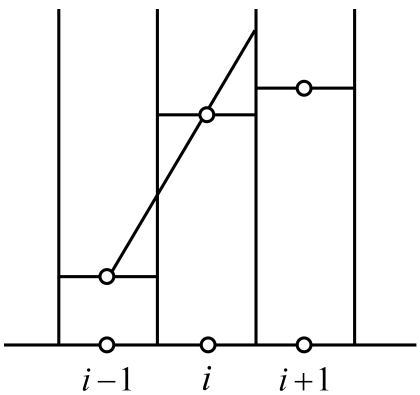

図 6　u_{i-1} と u_i を補間して $u^{\mathrm{L}}_{i+1/2}$ を求める様子

補間によって求められたセル境界の値は $u^{\mathrm{L}}_{i+1/2} > u_{i+1}$ となり, 元の値の大小関係が逆転してしまっている。そのため, 勾配に制限をもうけることが必要である。

今回は, 制限関数として最も強いもの (superbee 関数) と最も弱いもの (minmo d 関数) を用いて比較した。

ここで, minmod 制限関数は,

$$\Psi(r) = minmod(r, 1) \tag{37}$$

で, superbee 関数は

$$\Psi(r) = \max[0, \min(2r, 1), \min(r, 2)] \tag{38}$$

で与えられる。

8.3 実行結果と比較

それぞれの制限関数用いて衝撃波管問題を解いた結果を以下の図に示す。

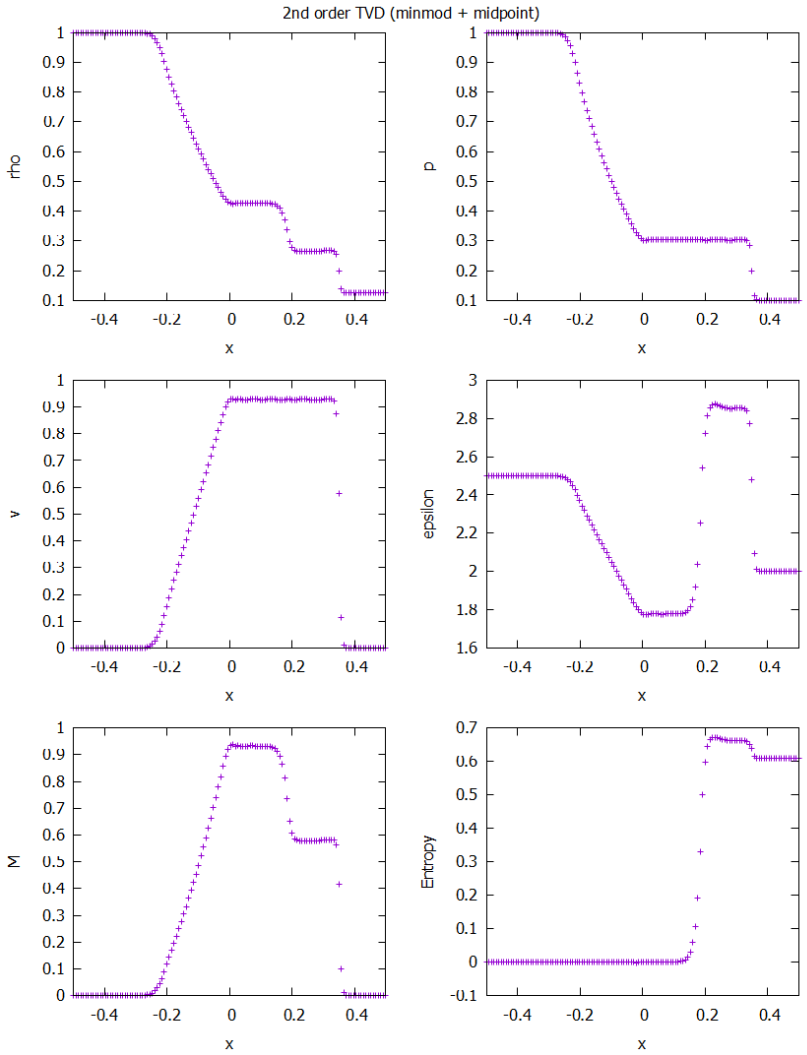

図 7　衝撃波管問題を minmod 制限関数を用いて解いた plot

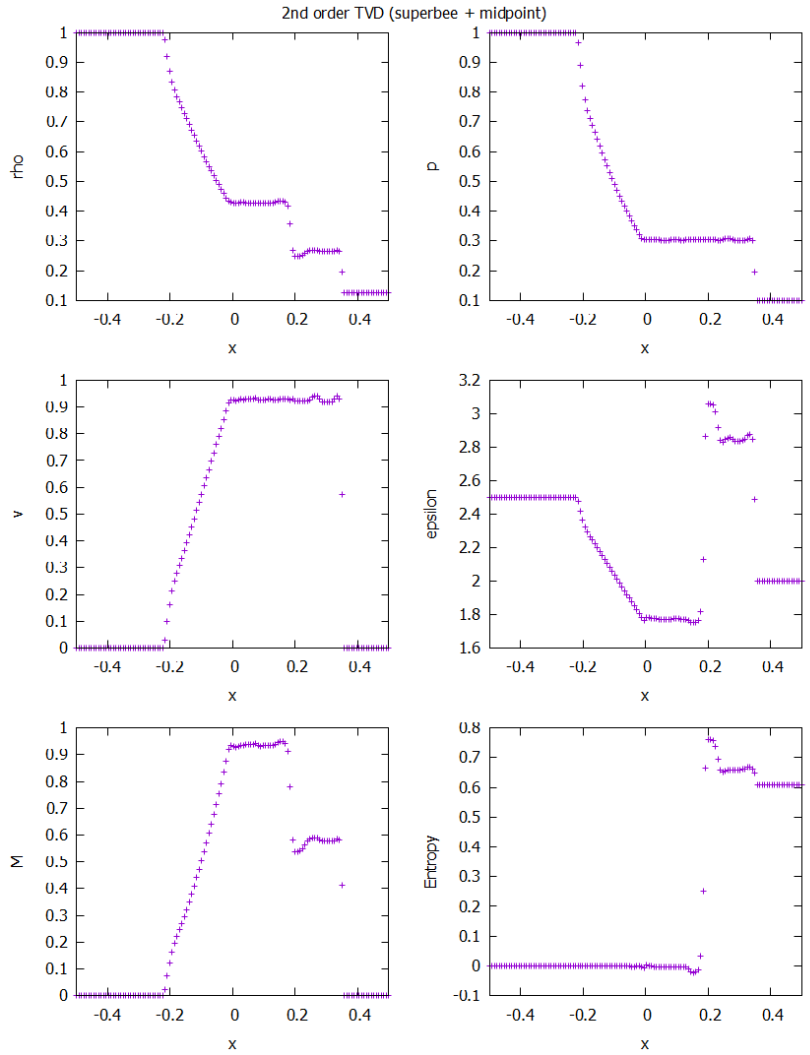

図 8　衝撃波管問題を superbee 制限関数を用いて解いた plot

この図の通り, (厳密解と比較しても) 2 次精度において解の精度が上がっていることが確認された。また, より強い superbee 制限関数を用いた場合の plot は, 全体的にシャープな印象を感じ, 特に接触不連続面と膨張波の部分において少ない点数で捉えれているように思った。しかし角が角張っている様子が見られた。この特徴は, 正弦波の線形移流方程式を解いた際にも見られたため, 矩形波を扱う分にはよいが, 多くの場合においては minmod 正弦関数が無難であるように考えられるが, 所々に物理量の振動が見られるため, 結局は精度自体をあげることがベストのように感じた。

9　最後に・感想

有名なテスト問題である衝撃波管問題を通して, 数値計算の様々な手法と考え方について学んだ。ただ, (HLL 法は実装できたものの,) 後半は HLLD 法のバグをずっととれずに時間を費やしてしまい, 磁場ありの問題についてレポートをかくことができず, 非常に残念である。今回悔しい結果で終わってしまったため, 今後とも数値計算の勉強自体は続けれていければと思っている。

また, 本レポートでは省略した制限関数を用いて衝撃波管問題を解いた際のコードなど, C1 で作成したほぼすべてのコードを, GitHub の Pubric Repositry に格納してあるので, 参照いただけますと幸いです。

`https://github.com/mito-nya/c1_2024`

参考文献

[1] 松本倫明ほか. 輻射電磁流体シミュレーションの基礎. 日本評論社, 2024.

[2] Eleuterio F. Toro. Riemann Solvers and Numerical Method for Fluid Dynamics. Springer, 2009.