

保存保証型多次元セミラグランジュスキームの開発 Development of Multi Dimensional Conservative Semi-Lagrangian Scheme

滝沢研二, 東京工業大学機械宇宙学科, 〒152-8552 東京都目黒区大岡山 2-12-1, ktaki@es.titech.ac.jp
矢部孝, 東京工業大学機械物理学専攻, 〒152-8552 東京都目黒区大岡山 2-12-1, yabe@mech.titech.ac.jp
Kenji TAKIZAWA, Dept. Mechano-Aerospace Engineering Tokyo Inst. of Tech. 2-12-1 O-okayama, Tokyo 152-8552
Takashi YABE, Dept. Mechanical Sciences and Engineering Tokyo Inst. of Tech. 2-12-1 O-okayama, Tokyo 152-8552

Conservative Semi-Lagrangian scheme that is one of the new CIP methods developed lately guarantees total conservative quantity. We have developed multi-dimensional semi-Lagrangian scheme by applying non-splitting method on the basis of one-dimensional CIP-CSL2 method. Two-dimensional scheme needs both line and surface densities. In addition to the two-dimensional case, three-dimensional scheme needs line, surface and volume densities altogether.

1. 概要

この方法は基礎方程式

$$\frac{\partial f}{\partial t} + \nabla \cdot (vf) = 0 \quad (1)$$

の差分解法となっているが, 実際には保存方程式で解かれているのは保存量と定義する値のみで, 他の物理量は CIP と同じ, 非保存系に対して, 非移流相計算で求める方法である。この非移流相計算は, CIP と同様に Fractional Step 法を用いる。

1) Fig.1 に示すような移流相計算を行う。

2) 上の式に**含まれない**発散を計算する。

また, この一次元保存セミラグランジュスキーム CIP-CSL2 を多次元に拡張する際に線密度と面密度という物理量を導入している。

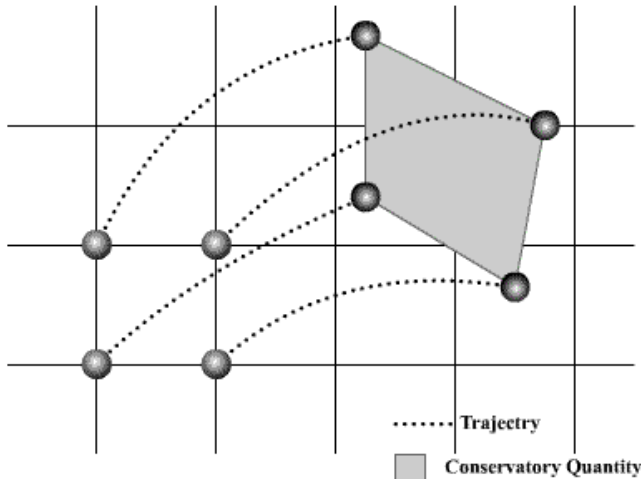


Fig. 1

2. 独立変数

1) 各点の値

2) 点の間に定義される線上を積分した値

a. x 方向 x

b. y 方向 y

3) 保存量に相当する

の3種類4つの独立変数を定義する。ただし, それぞれの定義位置は, 同じ位置ではない。従って, 独立変数はおよそ4と呼ぶのが正しい。

3. 補間関数の定義

補間関数は, 2 次関数補間で下記のようなものである。

$$F_{i,j}(X,Y) = \begin{pmatrix} X^0 & X^1 & X^2 \end{pmatrix} \cdot \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \cdot \begin{bmatrix} Y^0 \\ Y^1 \\ Y^2 \end{bmatrix} \quad (2)$$

ただし, 関数の定義区間を

$$\begin{cases} 0 \leq \frac{x - x_i}{x_{i+1} - x_i} \equiv X \leq 1 \\ 0 \leq \frac{y - y_i}{y_{i+1} - y_i} \equiv Y \leq 1 \end{cases} \quad (3)$$

と規格化している。

4. 関数の補間係数 Matrix の計算方法

2. の定義に従った値をそれぞれ計算していくと4点に囲まれる領域(関数の定義域)には9つの制約条件が存在する。

1) 各点の値の一致 (× 4)

2) 各辺の積分値の一致 (× 4)

3) 保存量の一致 (× 1)

である。また上記4つは線上に1次元 CIP-CSL2 を定義することを意味しており, **関数値の連続**を意味している。

ところで, 補間関数(2)の積分を考えてみると, 右辺の3つの項はお互いに独立である。従って x 方向の積分や, y 方向の積分はそれぞれの項を積分するだけでよい。

ここで, 補間関数の定義をもう一度書き直すと

$$f(x) \equiv \begin{pmatrix} 1 & x & x^2 \end{pmatrix}$$

$$g(x) \equiv \int f(x) dx = \begin{pmatrix} x & \frac{x^2}{2} & \frac{x^3}{3} \end{pmatrix}$$

$$C \equiv \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}$$

(4)

$$F(X,Y) = f(X) \cdot C \cdot f(Y)$$

(5)

*t は転置行列

このように定義した場合,9 つの制約条件は

$$\begin{bmatrix} f_{i,j} & \sigma_{i,j+1/2} & f_{i,j+1} \\ \sigma_{i+1/2,j} & \rho_{i+1/2,j+1/2} & \sigma_{i+1/2,j+1} \\ f_{i+1,j} & \sigma_{i+1,j+1/2} & f_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(0) \\ \mathbf{g}(1) - \mathbf{g}(0) \\ \mathbf{f}(1) \end{bmatrix} \cdot \mathbf{C} \cdot \begin{bmatrix} {}^t\mathbf{f}(0) & {}^t\mathbf{g}(1) - {}^t\mathbf{g}(0) & {}^t\mathbf{f}(1) \end{bmatrix} \quad (6)$$

ここで,(6)の右辺の両端の項を \mathbf{P} とおくと,それぞれ転置の関係になっており,またこの逆行列を求めることができ,これは

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 6 & -2 \\ 3 & -6 & 3 \end{bmatrix} \quad (7)$$

従って,

$$\mathbf{C} = \mathbf{P}^{-1} \cdot \begin{bmatrix} f_{i,j} & \sigma_{i,j+1/2} & f_{i,j+1} \\ \sigma_{i+1/2,j} & \rho_{i+1/2,j+1/2} & \sigma_{i+1/2,j+1} \\ f_{i+1,j} & \sigma_{i+1,j+1/2} & f_{i+1,j+1} \end{bmatrix} \cdot {}^t\mathbf{P}^{-1} \quad (8)$$

なお,この係数行列は 1 次元や 3 次元にも適応できる。

1 次元

$$\mathbf{C} = \mathbf{P}^{-1} \cdot \begin{bmatrix} f_i \\ \rho_{i+1/2} \\ f_{i+1} \end{bmatrix} \quad (9)$$

3 次元

$$\mathbf{C}^*_k = \mathbf{P}^{-1} \cdot \begin{bmatrix} f_{i,j,k} & \sigma_{i,j+1/2,k} & f_{i,j+1,k} \\ \sigma_{i+1/2,j,k} & \rho_{i+1/2,j+1/2,k} & \sigma_{i+1/2,j+1,k} \\ f_{i+1,j,k} & \sigma_{i+1,j+1/2,k} & f_{i+1,j+1,k} \end{bmatrix} \cdot {}^t\mathbf{P}^{-1} \quad (10)$$

$$\begin{bmatrix} \mathbf{C}_0 \\ \mathbf{C}_{1/2} \\ \mathbf{C}_1 \end{bmatrix} \equiv \mathbf{P}^{-1} \begin{bmatrix} \mathbf{C}^*_0 \\ \mathbf{C}^*_{1/2} \\ \mathbf{C}^*_1 \end{bmatrix} \quad (11)$$

ただし,補間関数は,

$$F(X,Y) = \mathbf{f}(X) \cdot (\mathbf{C}_0 + \mathbf{C}_{1/2}Z + \mathbf{C}_1Z^2) \cdot {}^t\mathbf{f}(Y) \quad (12)$$

また, ρ や σ を便宜的に表示に用いているが,本質的には差はなく,

$\rho_{i,j,k}$ において i,j,k のうち 1/2 が 1 つのものは

線密度,2 つのものは面密度,3 つのものは体積密度と呼ばれる値である。

5. 時間更新

3 種類の物理量はそれぞれ違った時間更新が必要である。

具体的には,

1) f の時間更新

$$\frac{Df}{Dt} = -f \text{div} \mathbf{v} \quad (12)$$

2) σ の時間更新

$$\frac{D\sigma}{Dt} = -\sigma \frac{\partial v_\eta}{\partial \eta} \quad (13)$$

(η : normal line direction)

3) ρ の時間更新

$$\frac{D\rho}{Dt} = 0 \quad (14)$$

となっている。

1) は CIP と同様な完全に発散を計算するもの,

2) は線密度方向には保存して解いているため残りの法線方向のみの発散を計算しないといけな。

3) はこのスキームの核心であり保存方程式を完全に解いた形になっている。

具体的には,この手法はセミラグランジュスキームであるため高精度で移流軌跡を求めていく必要がある。従ってこの作業と同時にそこにおける発散を計算していく手法を提案する。また,確実に接線方向の発散を計算することは困難であるのでその近似法も次のセクションにおいて提案させていただく。

6. 計算アルゴリズム

全体は時間の Fractional Step によって構成される。

$$f^* - f^n = \text{Advection Phase}$$

$$f^{n+1} - f^* = \text{Non - Advection Phase} \quad (15)$$

1) 移流相計算

セミラグランジュスキームとして移流相計算をするにあたって,非常に難しい領域の積分が要求される。この点について軽く説明します。

2) 非移流相計算

非移流相としては 5. の発散の計算方法を示します。なお,これは 3) での軌跡を求める際に計算することになるので,計算理論を示します。

3) 移流軌跡の求め方

大きな CFL で計算するには速度を線形補間したものでは不十分である。従って \mathbf{v} をさらに細かい時間ステップ (virtual step) を設け,これで精度よく遺留原点を求めます。また,同時に非移流相の計算も行います。

以下,具体的に見ていきます。

1) 移流相計算

(移流原点を次のようにあらわすとする。)

$$\mathbf{x}_{i,j} = \mathbf{x}_{i,j}^{n+1} - \int_0^{\Delta t} \mathbf{v} dt \quad (16)$$

a)値の更新

CIP 同様その場所すなわち(16)を用いて

$$f_{i,j}^{n+1} = F(\mathbf{x}_{i,j}) \quad (17)$$

ただし,F は \mathbf{x} のある領域の補間関数で計算を行います。

b)線密度 の計算

$$\sigma_{i+1/2,j}^{n+1} = \int_{\mathbf{x}_{i,j}}^{\mathbf{x}_{i+1,j}} F(\mathbf{x}) d\mathbf{x} \quad (18)$$

ただし,x は線形補間するすなわち

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ ax+b \end{bmatrix} \quad (19)$$

$$\sigma_{i+1/2,j}^{n+1} = \int_{x_{i,j}}^{x_{i+1,j}} F(x, ax+b) \sqrt{a^2+1} dx \quad (20)$$

これも,また補間関数はその領域ごとの関数を用いる。

y 方向の場合は j を 1/2 足したものを考える。

また,セミラグランジュにおいては a が大きくなることもありえ計算上発散してしまう場合もある。

従って, a の大きさによって違う計算方法をとることが望ましい。

具体的には,

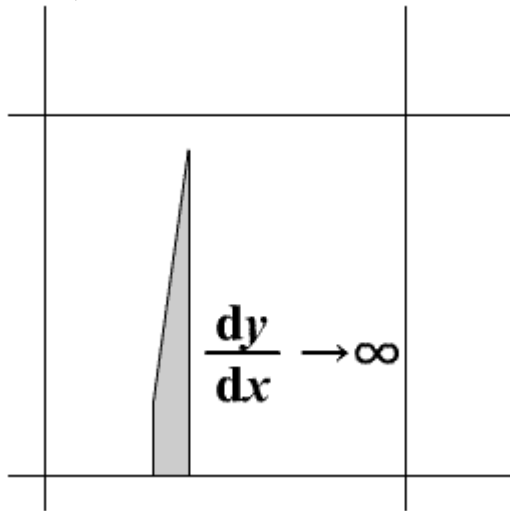


Fig.2

もし,(19)において, $|a| \geq 1$ となるような場合は

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a^* y + b^* \\ y \end{bmatrix} \quad (21)$$

$$\sigma_{i+1/2,j}^{n+1} = \int_{y_{i,j}}^{y_{i+1,j}} F(a^* y + b^*, y) \sqrt{a^{*2}+1} dy \quad (22)$$

と計算するとよい。(この場合 $|a^*| \leq 1$)

c)面密度 の計算方法

概略は Fig.3 に示すような 4 つの移流原点に囲まれた四角形領域の積分を行う。

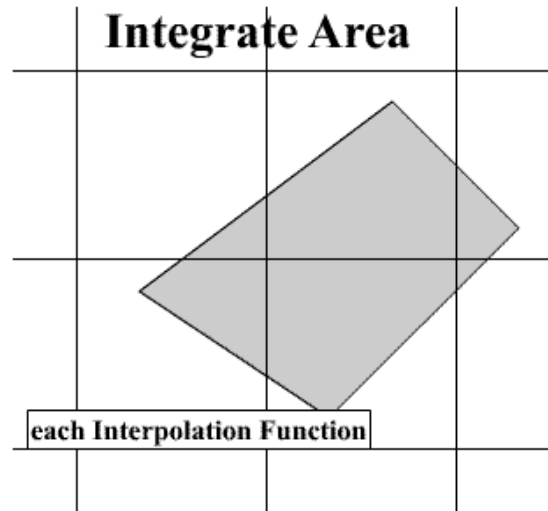


Fig.3

まず,このような領域をどのように選択していくかを考える。媒介変数を設け,周回積分することでこの領域の積分とする。

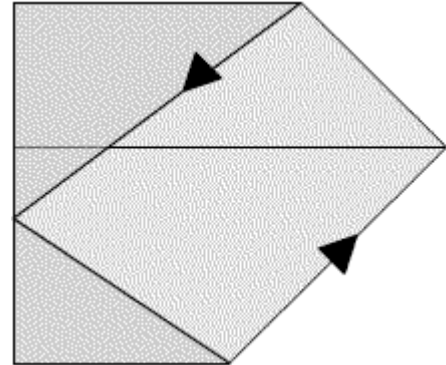


Fig.4-1

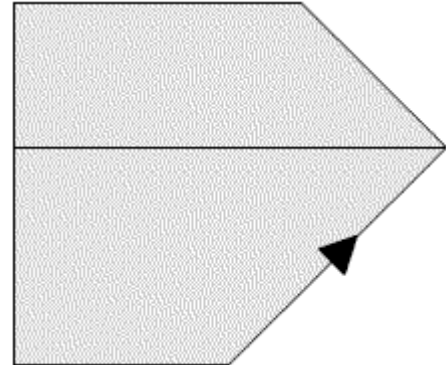


Fig.4-2

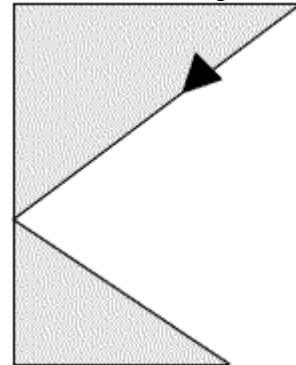


Fig.4-3

Fig4のように例えば x の最小値を基準として半時計回りに4つの直線を順次計算していく手法が考えられる。このことにより、各領域は独立して計算することが可能になる。次に台形の領域をどのように積分するかということを考える。

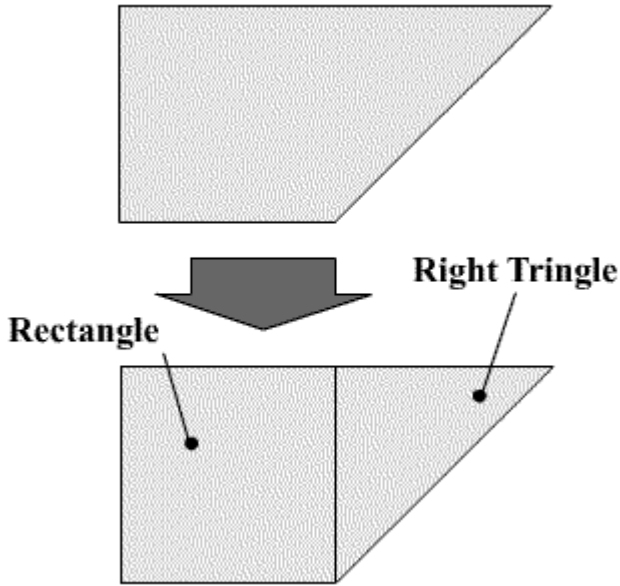


Fig.5

Fig.5のように長方形領域と直角三角形部に分けて計算する。この2つの解析解を用意することですべての場合に対応することが可能である。また、この場合もFig.2に示すように三角形が鋭くなる場合は傾きの緩やかな方向から見た式を用いることが計算精度を上げるために必要である。また、実際にはFig.3に示すようにこの領域はさらに多くの補間関数領域をまたいでいる場合が考えられるので、それぞれの領域でこれを計算することになる。

2) 非移流相計算

a) 値の更新
(12)より

$$\frac{df}{f} = -\text{div} \mathbf{v} \cdot f dt$$

$$f^{n+1} = f^* \exp\left(-\int \text{div} \mathbf{v} dt\right)$$

(23)

(*は、移流前の値)

$\exp\left(-\int \text{div} \mathbf{v} dt\right)$ は、この特性線に特有の値として3)で述べる。

b) 線密度の更新

$$\frac{D}{Dt} \int_0^l f d\xi = \int_0^l \left(\frac{Df}{Dt} \right) d\xi + \int_0^l f \frac{D}{Dt} (d\xi)$$

$l : \text{length}(t)$

(24)

length は線密度の長さ、 $\frac{D}{Dt}$ は接線すなわち

$$\frac{D\sigma}{Dt} = -\int_0^l f \text{div} \cdot \mathbf{v} d\xi + \int_0^l f \frac{dv_\xi}{d\xi} d\xi$$

$\therefore (12)$

(25)

というのが本来求めるべき計算であるが、それぞれの点における f を正確に計算するのは不可能であり、あくまでこの積分値が計算されているに過ぎない。また、この積分値は(25)の項により、時々刻々変わってゆく。従ってこれを厳密に解くことは妥当ではなく、次のように近似して解くことにする。

$$\frac{D\sigma}{Dt} = -\sigma \left(\overline{\text{div} \cdot \mathbf{v}} - \frac{\partial l}{\partial t} \right)$$

(26)

$$\sigma_{i+1/2,j}^{n+1} = \sigma_{i+1/2,j}^* \exp \left(-\frac{\int \text{div} \mathbf{v} dt_{i,j} + \int \text{div} \mathbf{v} dt_{i+1,j}}{2} + l(\Delta t) - l(0) \right)$$

(27)

c) 面密度 は非移流相なし

3) 移流軌跡の求め方

同時に、発散も計算するため Runge-Kutta 法を用いて順次計算する。いま時間ステップを Δt とすると、Virtual Step 数 N でわった $\Delta t/N$ 時間おきに補間された速度場を移動する。例えば、点に値しかない速度場が与えられていたなら4点を用いた1次関数補間で次のように求める。

$$\mathbf{V} = (u_{00} + u_{10}x + u_{01}y + u_{11}xy, v_{00} + v_{10}x + v_{01}y + v_{11}xy)$$

(28)

また、速度も CIP や CSL で持っているのであれば2次関数補間や3次関数補間も可能である。

この時、補間した関数には、発散の解析解が存在するのでこれを求め、足し合わせて(23)などの値として用いる。

例えば(28)のときは

$$\text{div} \mathbf{V} = u_{10} + u_{11}y + v_{01} + v_{11}x$$

(29)

となる。

7. 計算例

1) Zalesak's Experiments

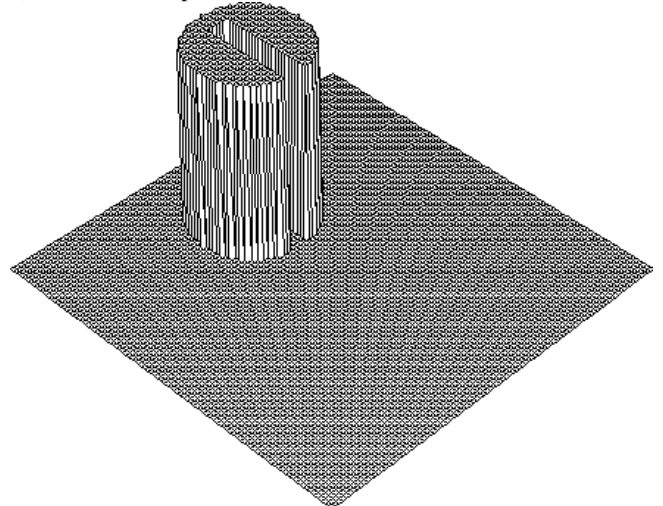


Fig.7 (Initial Condition and True Solution)
回転角速度 $2\pi/100\text{step}$ の剛体回転の場合

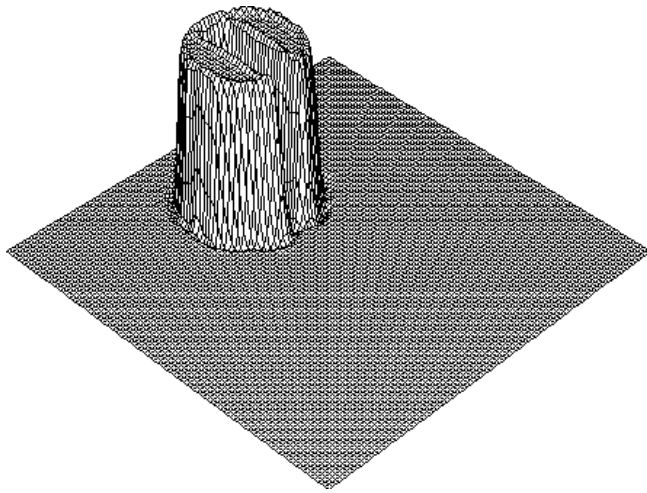


Fig.8

計算格子 100×100
100step 後の計算結果

ここで, Mass Error として次の定義をする。

$$MassError = \left| \frac{Mass_0 - Mass}{Mass_0} \right|$$

$$Mass = \sum_i \sum_j \iint F_{i,j}(X,Y) dXdY$$

(30)

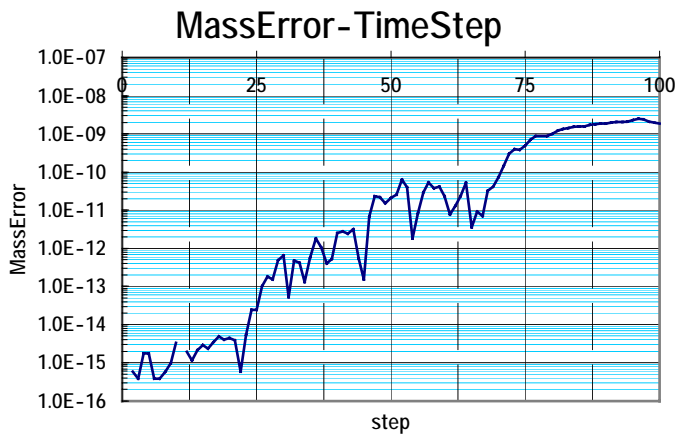


Fig.9

25step おきに MassError が大きくなっているのは, 拡散により広がった裾が計算領域の外に出てしまっている分である。境界の影響を無視すればほぼ計算機の精度で保存が成立している。

2) Doswell's Frontogenesis Experiment

$$v_T = \text{sech}^2(r) \tanh(r)$$

$$r = \sqrt{x^2 + y^2}$$

$$\phi(x, y) = -\tanh\left(\frac{y}{2}\right)$$

(30)

速度は r の垂直方向右回り, 無次元時間 16 後の計算結果

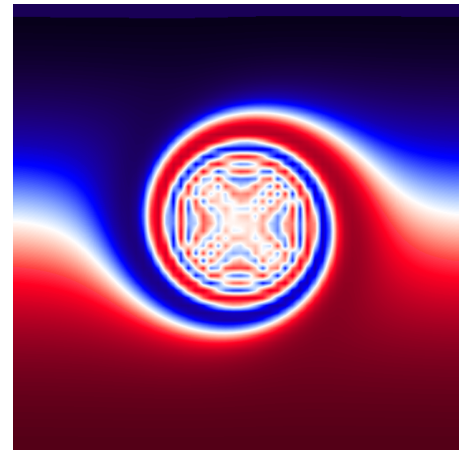


Fig.10(True Solution)
解析解

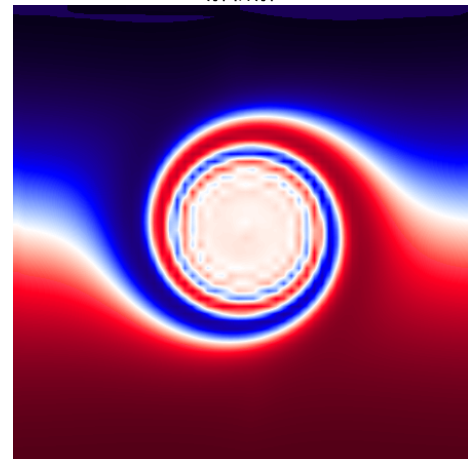


Fig.11(CFL=4.243)
CSL2 による計算結果

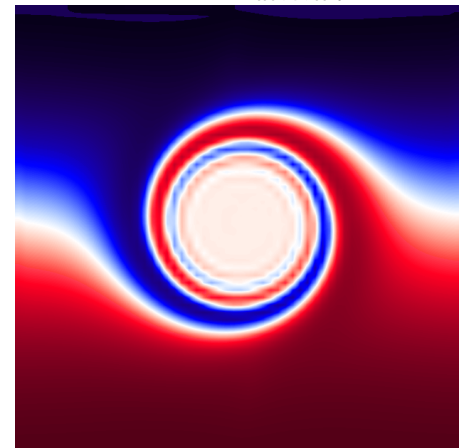


Fig.12(CFL=1.061)

CSL2 による計算結果

この計算は, ほとんど圧縮の項がはいっていないこともあり, かなりよい精度の計算結果が得られている。実際, 二乗平均誤差 (RMS 値) は次の Table のようになっている。このことからわかるように, CFL を大きくとった方が, 数値拡散が小さくなり, 精度が上がるようである。

CFL	RMS
4.243	5.6020487995647343e-02
2.121	6.6485016416233098e-02
1.061	7.3757156348096875e-02

Table

3) 圧縮性速度場における計算結果

計算条件

$$\mathbf{v} = (u, v)$$

$$\begin{cases} u = \frac{1}{1 + 0.5 \sin(2\pi x / L)} \\ v = \frac{1}{1 + 0.5 \sin(2\pi y / L)} \end{cases}$$

(31)

格子: 100 × 100

初期条件

$$\begin{cases} f = 1.0 & (10 \leq x, y \leq 30) \\ f = 0.0 & (\text{otherwise}) \end{cases}$$

(32)

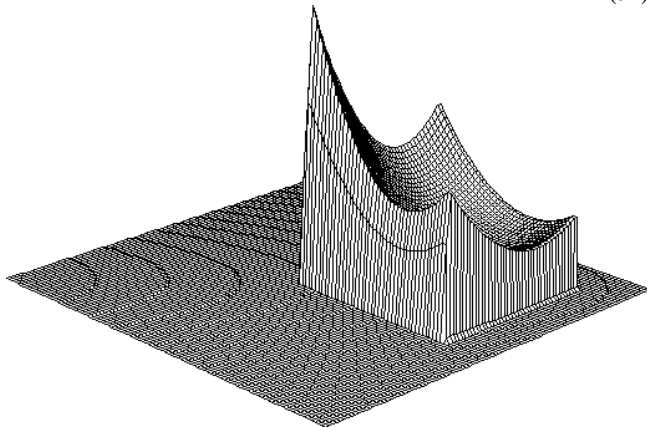


Fig.13 (True Solution)

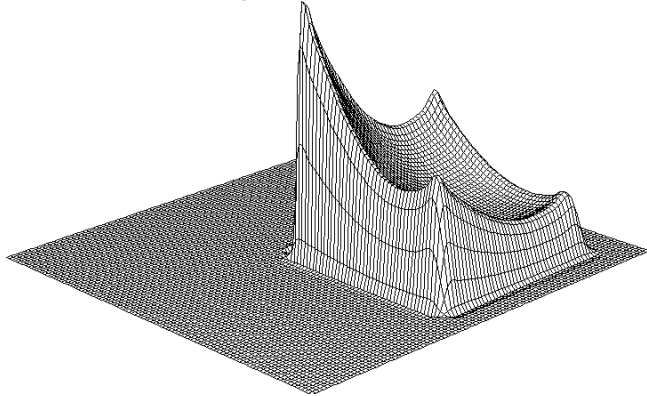


Fig.14

CFL=2.0 (50step 後の計算結果)

圧縮,膨張のある場においてもよい結果を得ている。

8.考察と今後の改良に向けて

1次元 CIP-CSL2 法の Non-Splitting Method による多次元拡張としてその方法を示すことができた。ただし,3次元においてはその領域分割の方法が問題となり,また独立変数が 8 値,線密度 × 3,面密度 × 3,体積密度) となってしまう点が問題となっている。

この対策として 線密度を用いずに回りの値などから求める手, Fig.15 参照, や 線密度の変わりに微分値を用いて計算する方法, Fig.16, も開発中である。(Fig.8 に示したものと同条件の計算結果)

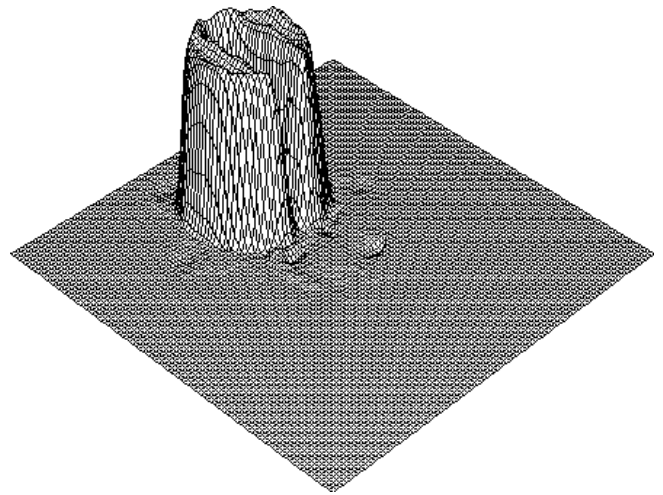


Fig.15

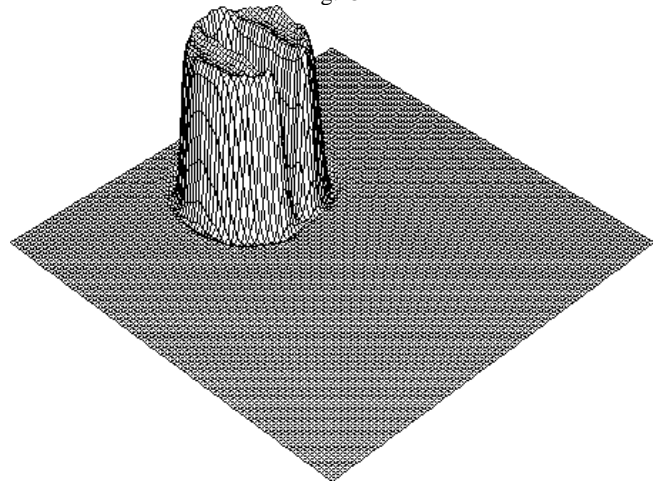


Fig.16

また同じ多次元拡張である,空間を方向分割して Fractional Step を用いた手法では CFL が大きくなった場合に精度が落ちるという報告があるが,この手法は CFL の大きさには拠らない。また変形 CFL < 1 において計算が破綻することなく確実に求めることができる点は優れていると思われる。ただし,今回示したように領域別の計算方法が難しく,また計算に時間がかかることも予想される。この積分方法の最適化や,また別の手法で計算の効率化を考える必要があるかもしれない。