

界面の大変形を伴う自由界面流れの計算手法*

北海道大学 電子科学研究所 横井 研介¹

(2000年5月8日受理)

日常世界において流体现象は、河川、キッチン (コンロ, 流し, やかんの水, など), 動植物内等いたる所に現れる. これらのような流体现象の多くは, 移動境界 (気-液, 固-液, 液-液, など) を持ち, それらの境界は一般に複雑であったり, 大きく変形したりする. このような現象の解析はチャレンジングかつ実用的にも重要であるが, 物理的対象としては複雑すぎ, 研究する手段も近年まで実験以外にはなかった. しかし, 近年の数値流体の計算法の発展により, これらの現象の数値的扱いが可能になりはじめ, 計算物理の対象になりつつある. 本原稿では, 自由界面流れを例に近年開発された移動境界流れの計算手法について紹介する.²

1 はじめに

移動境界を持つ流体现象に対する数値計算法は非常に多くあり [1], どの方法が良いかは現象に強く依存する. 本原稿では, 界面の大変形を伴う流体现象 (例えば波が崩れるような流体现象) に的をしぼり, 界面の大変形にも対応できる数値計算法について説明する.

移動境界を持つ流体现象のシミュレーションは, 境界の扱いの複雑さなどのため難しいと言われる. 私も実際にやってみてその難しさを痛感した. しかし, 流体の支配方程式には基本的に完成されたものがあり, 界面が大変形しても流体现象はこれらの支配方程式に従っている. 本原稿では, 以下の流体の支配方程式 (重力, 粘性, 表面張力の効果を含む) を採用する.

$$\frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla) \rho = -\rho \nabla \cdot \mathbf{u}, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{\nabla p}{\rho} + \mathbf{g} + \frac{\mu}{\rho} \Delta \mathbf{u} + \frac{\mathbf{F}_{sv}}{\rho}, \quad (2)$$

$$\frac{\partial e}{\partial t} + (\mathbf{u} \cdot \nabla) e = -\frac{p}{\rho} \nabla \cdot \mathbf{u}, \quad (3)$$

$$p = p(\rho, e), \quad (4)$$

ここで, ρ : 密度, \mathbf{u} : 速度, p : 圧力, \mathbf{g} : 重力, μ : 粘性係数, \mathbf{F}_{sv} : 表面張力, e : 内部エネルギー. また, (1), (2), (3), (4) は, 連続の式, 運動方程式, エネルギー式, 状態方程式である. 基本的には,

* 本稿は, 編集部から特にお願いして執筆していただいた記事である。

¹ E-mail: kensuke@aurora.es.hokudai.ac.jp

² ここで紹介する方法は私が開発したのではなく, 私はこれらの方法を現象の解析に用いているだけです. ユーザーの立場からこれらの方法を紹介させていただきます。

これらの式を差分化し計算機に計算させれば流体现象は再現されるはずである。しかし、以下に説明するように、それが容易ではない。

では、何がそれほど難しいのか？ 流体の支配方程式は、見てのとおり非線形の連立偏微分方程式であり、いかにも難しそうに見える。しかし、本当に難しいのは、連立偏微分方程式だからでも非線形だからでもなく左辺の移流項³、つまり移流方程式

$$\frac{\partial f}{\partial t} + (\mathbf{u} \cdot \nabla) f = 0, \quad (5)$$

の計算である。この方程式には、厳密解“ $f(\mathbf{r}, t) = f(\mathbf{r} - \mathbf{u}t, 0)$ ”まで存在し、その計算は簡単に見える。移流方程式の解の簡単な例として、一次元かつ全空間に一樣な速度 u が与えられている場合を図 1 に示す。この場合、 $t=0$ のプロファイルが dt 後に udt 移動するだけであり簡単であ

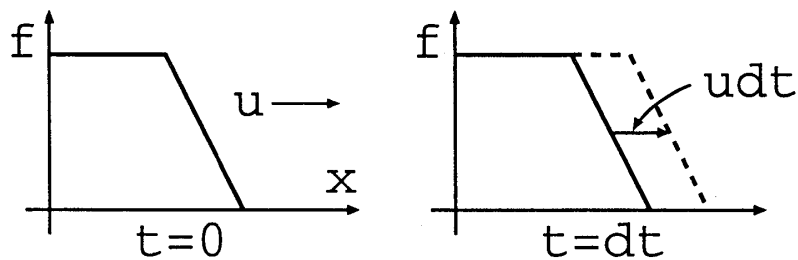


図 1: 一次元移流方程式の解。全領域に対して一定の速度 u が与えられている場合、初期プロファイル ($t=0$) が dt 後、右図のように udt だけ移動する。

る。しかし、これを具体的にメッシュ上で計算しようとするとき様々な困難に直面することがすぐに分かる。

まず、ラグランジュメッシュ(流体とともに動くメッシュ)を使う場合を考える。この場合、図 1 のような計算は容易に行なうことができる。しかし、2次元に拡張し、図 2 のような水波が壊れるような状況になるとメッシュの捻れや重なりのため計算が破綻してしまう。

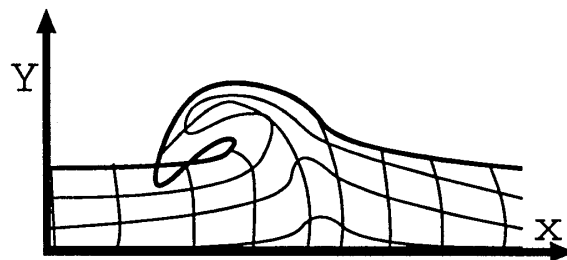


図 2: 太線は界面を表す。ラグランジュメッシュでは界面の大変形に対応できない。

次に、図 3 のようにオイラーメッシュ (固定メッシュ) 上で界面を動かすことを考える。液体の

³ この項は、対流項とも呼ばれ流体の移動を表す。

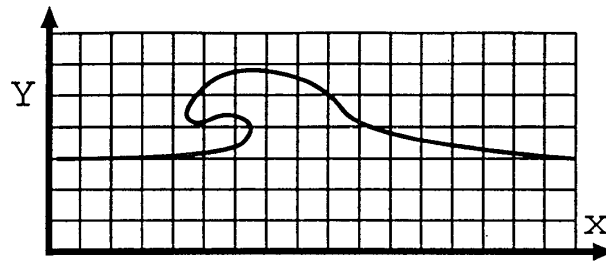


図 3: 太線は界面を表す。オイラーメッシュは界面の大変形には対応できるが、界面での数値拡散が問題となる。

部分を 1, 気体の部分を 0 とする密度関数を導入し, 密度関数の勾配が存在する所を界面とすれば, オイラーメッシュ上で界面を表現できる。この方法は一見良さそうに見えるが, 界面での数値拡散が大きな問題となる。簡単のため, 一次元の等間隔のグリッドを例に, そのことについて説明する。いま, 図 1 と同じ問題を固定グリッド上で計算することを考える。各々の格子点での

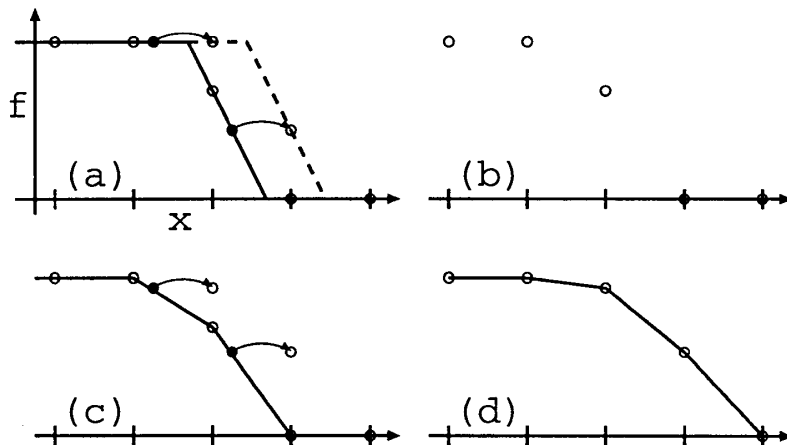


図 4: 一次元の移流方程式に線形補間を用いた計算。

dt 後の値を知るには udt 前の値 (図 4 (a) の黒丸の値) が分かれば良い。しかし, そこには一般にグリッドはなく (図 4 (b)), 黒丸の値を知るには, 何らかの工夫が必要である。もっとも簡単な方法は線形補間であるが, それをすると黒丸の位置が本来あるべき位置 (図 4 (a)) から図 4 (c) のようにずれてしまい, プロファイルが正しく伝わらない (図 4 (d))。このようにプロファイルが崩れていくことを数値拡散という。この方法でもメッシュを細かくとれば良いではないかと思われる方がいると思うが, この方法を使う限り時間発展のたびにプロファイルが崩れていくため, 本質的な解決にはならない。しかし, もし数値拡散もなく, 精度良く移流計算ができれば, 固定グリッドをベースにした方法は, メッシュが捻れて計算が破綻することがないため界面の大変形にも耐えられる。そこで, 本原稿では等間隔の固定グリッドを使って計算することを考える。

固定グリッドを使う場合は, 移流以外にも大きな問題が出てくる。例えば, 直交格子上で曲線

(例えば円形の水滴, 図 5) を表現しようとする場合である. 図 5 は, その一例であるが, 簡単に

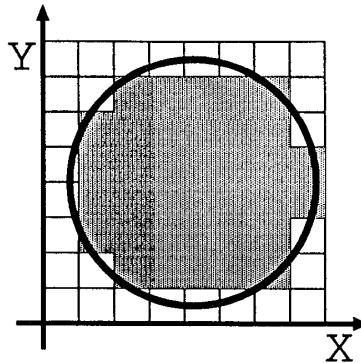


図 5: メッシュの中心が円の内側にある所を円と考えて円を表現した場合.

は表現できないことが分かって頂けると思う. この問題はメッシュを細かくとればある程度回避できるが, メッシュを多く取った分, 計算時間を必要とする. 表面張力の計算には, 曲率が必要のため, 曲線を精度良く表現することは重要である. しかし, 直交格子上で曲線 (特に移動する曲線) を精度良く記述することは容易ではない.

このように, 界面の大変形を伴う現象のシミュレーションが容易ではないことは分かって頂けると思う. しかし, これらのような問題は, 近年の数値計算法の進歩によりほぼ解決されている. 本原稿では, これらの数値計算法を紹介する. 具体的には, 以下の方法について紹介する.

- 高精度な移流方程式の数値解法の CIP 法 (第 2 章).
- 界面を扱うのに便利な level set 法 (第 3 章).
- 界面に境界条件を必要としない (計算の中で境界条件が自動的に満たされ, 実質境界条件を必要としない), 圧力ベースの Multi-Fluid, Multi-Phase Flow の数値解法の CUP 法 (第 4 章).

これらの計算法の計算例等が, [2, 3, 4] のホームページにあるので, まず見ることをお勧めする.

2 移流方程式の数値解法

本章では, 主に以下の 1 次元移流方程式:

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0, \quad (6)$$

を扱う. 移流方程式 (6) を固定グリッド上で計算することは, 図 4 から分かるように, 意外に難しい. この計算では, 有限の格子点 (図 6 の白丸) の情報から, いかに厳密解 (図 6 (a) の実線) に近いプロファイルを復元するかということが重要になる. 線形補間を使った場合, 図 6 (b) から分かるように著しくプロファイルが失われる. そこで, 線形補間ではなく, 2, 3, ... 次関数を使

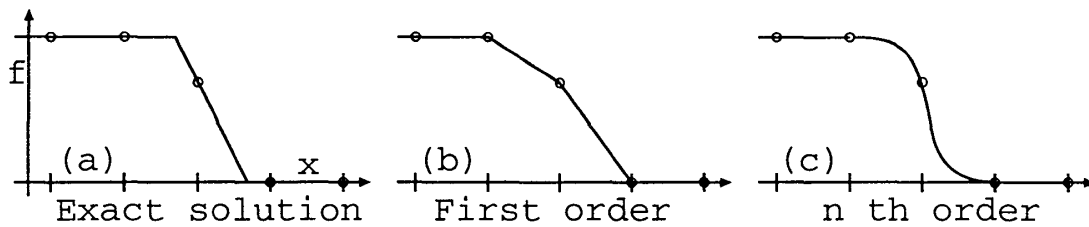


図 6: 格子間の補間の例. 線形補間 (b) を使った場合, プロファイルが大きく崩れる. n ($n > 1$) 次の補間を使う場合 (c), 線形補間 (b) より, 厳密解 (a) に近付くことが期待される.

例えば図 6 (c) の様に比較的厳密解に近いプロファイルが得られると考えられる. この様な, 2, 3, ... 次曲線を使う方法は非常に多くあるが, 大きく分けて 2 つのタイプに分けられる. 1 つ目は, ラグランジュ補間と呼ばれる方法で, $n + 1$ 個の格子点を用いて n 次曲線を構築する方法である. この方法で代表的なものは Lax-Wendroff 法で, 3 つの格子点の値を使って 2 次曲線を構築する. この方法の欠点は, 次数が上がるにしたがって計算に必要な格子点が多くなり境界条件を付けることが難しくなることである. 2 つ目は, スプライン補間と呼ばれる方法で, 格子の一区間 (例えば, i と $i + 1$ の間) を n 次曲線で補間する方法である. この方法は, 境界条件が付けやすく扱いやすいので, ここではこちらの補間方法を採用する. この方法で良く使われるのは, 3 次関数を使う方法である. 3 次関数の係数を求めるには多くの方法があるが, 今回はその中の一つの方法, 矢部等によって開発された Cubic interpolated propagation (CIP) 法 [5, 6] について紹介する.

格子の一区間を補間する 3 次関数に

$$F_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + f_i, \tag{7}$$

を採用する. ここで, $x_i = i \times \Delta x$, Δx : 格子間隔, $f_i \equiv f(x_i)$, a_i, b_i, c_i : 係数. 普通に考えれ

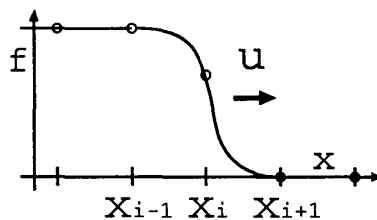


図 7: 格子間を 3 次関数で補間する. $u_i > 0$ の時は, x_{i-1} で連続の条件を課す.

ば, 3 つの係数 (a_i, b_i, c_i) を決めるために, 以下の条件, もし $u_i > 0$ ならば x_{i-1} ($u_i < 0$ ならば x_{i+1}) において (風上側において), $f, \frac{df}{dx}, \frac{d^2f}{dx^2}$ が連続という条件を課す. しかし, この条件を用いると行列計算をしなければならず多少面倒になる.

そこで, CIP 法では異なる方法を用いる. いま, 初期値として f だけでなく, f' も与えられて

いるとする。この時、補間関数は、

$$F_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + f'_i(x - x_i) + f_i, \quad (8)$$

となり、未知係数が1つ減る。従って、上で用いた条件も1つ減らすことができるため、格子間で $f, \frac{df}{dx}$ が連続という条件を与えれば係数が以下のように決まる。

$$a_i = \frac{(f'_i + f'_{i-1})}{\Delta x^2} - \frac{2(f_i - f_{i-1})}{\Delta x^3}, \quad (9)$$

$$b_i = \frac{3(f_{i-1} - f_i)}{\Delta x^2} + \frac{2f'_i + f'_{i-1}}{\Delta x}. \quad (10)$$

この様にすると行列計算をする必要がなくなる。そして、この3次関数を用いて、プロファイルを図1の様にシフトすれば、

$$f_i^{t+\Delta t} = F_i(x_i - u_i \Delta t), \quad (11)$$

のように f の時間発展が計算できる。この方法では、 f の時間発展を計算するためには、 f' の時間発展も計算する必要がある。いま、 u を定数と考え、(6)の空間微分をとると、

$$\frac{\partial f'}{\partial t} + u \frac{\partial f'}{\partial x} = 0, \quad (12)$$

になる。ここで、 $f' = \frac{\partial f}{\partial x}$ 。良く見ると(12)の形は、(6)と同じであり、この式には、 f 同様、厳密解 $f'(x, t) = f'(x - ut, 0)$ が存在する。 f' の補間関数は、(8)の空間微分

$$F'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + f'_i, \quad (13)$$

であるから、この補間関数を用い、

$$f_i^{t+\Delta t} = F'_i(x_i - u_i \Delta t), \quad (14)$$

により、 f' の時間発展を容易に計算することができる。CIP法の2, 3次元への拡張については[7, 8]を参照。

上では、 u を定数としたが、 u が定数でない問題にもこの方法を適用できる。 u が定数でない場合、(6)の空間微分は、

$$\frac{\partial f'}{\partial t} + u \frac{\partial f'}{\partial x} = -f' \frac{\partial u}{\partial x}, \quad (15)$$

となる。CIP法を使うため、(15)を以下のように時間的に分離し、

$$\frac{\partial f'}{\partial t} + u \frac{\partial f'}{\partial x} = 0, \quad (16)$$

$$\frac{\partial f'}{\partial t} = -f' \frac{\partial u}{\partial x}, \quad (17)$$

順に解く。まず、(16)に対してCIP法を用い、(14)により、 f' の時間発展を計算する。いま、その値を f'^* とする。次に、 f'^* を用い(17)を差分によって計算し、 $f_i^{t+\Delta t}$ を求める。

CIP法の意味を簡単に説明すると図8のようになる。 f' の情報(図8の矢印)も使い格子間を3次関数で補間するため、プロファイルを精度良く回復させることができる。

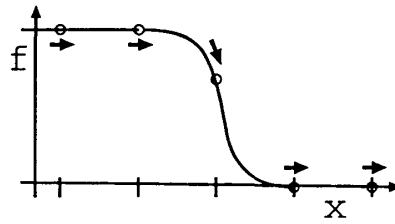


図 8: CIP 法の意味.

CIP 法と線形補間を使った方法 (1 次の風上差分) の違いを見るために矩形波伝搬の問題にこれらの方法を適用する. その結果が図 9 である. 線形補間ではプロファイルが大きく崩れ, また,

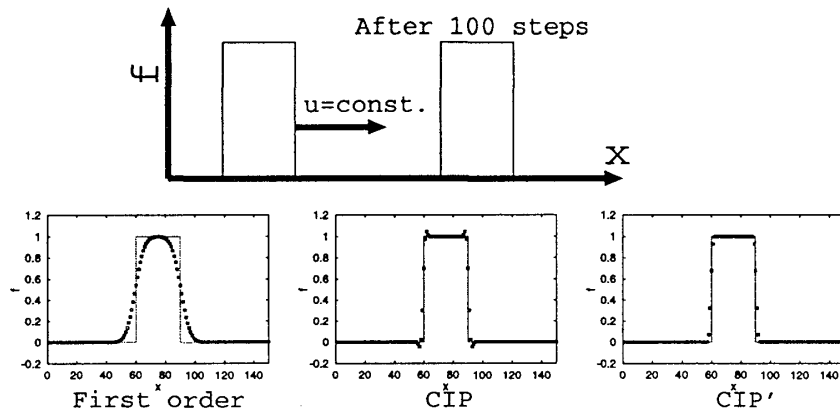


図 9: $\Delta x = 1, \Delta t = 0.5, u = 1$. 点 (四角) はシミュレーション, 点線は厳密解.

プロファイルは時間とともにさらに崩れていく. この方法では, 関数 f を密度関数として界面の識別に使うには, 数値拡散が大きすぎるため無理がある. 次に CIP 法であるが, 多少数値拡散があるもののプロファイルはほとんど崩れておらず, また, これが時間とともに崩れていくこともない. CIP 法では, 値が不連続な所で多少数値振動が起こるが, この問題は, Rational CIP 法 [9, 10] などを使えば図 9 の CIP' のように防ぐことができる. この方法であれば界面の識別に使える. 実際に, CIP 法は様々な問題に適用され多くの成果を上げている [11].

3 界面の取扱い

CIP 法は, 移流方程式の解法として, 数値拡散が少なく, 安定した, 優れた方法である. しかし, CIP 法は界面で数値拡散が起こらないとは保証しておらず, 界面が激しく変形するような場合, 数値拡散が問題となる場合があった. そこで, 界面での数値拡散を完全になくすために, Sethian 等によって開発された level set 法 [12, 13, 14] を導入する. 数値拡散を防ぐ方法は他にもあるが [15], この方法は数値拡散を防ぐだけでなく他にも多くのメリットを持つ. level set 法の特徴を以下に上げる.

- 界面のトポロジーの変化に対応できる。
- 格子点の情報だけで、格子の間にある界面を表現できる (図 5 の問題を避けられる)。
- 界面での法線ベクトルを正確に見積もることができる (法線ベクトルは、表面張力の計算に用いる)。
- 極めてシンプルな方法である。

level set 法では、界面を level set 関数 ψ によって追跡する。いま、ここでは気相、液相の 2 相流の問題を考え、level set 関数を以下の条件：

$$\begin{aligned} \psi > 0 & \quad \text{液相,} \\ \psi < 0 & \quad \text{気相,} \\ |\nabla\psi| = 1 & \quad \text{全領域,} \end{aligned} \tag{18}$$

を満たす関数とする。この時、界面は、zero level set $\psi = 0$ になる。1次元の場合を図 10 左に示す。図 10 から分かるように、界面は格子点上にある必要はない。物質の識別に用いる密度関数

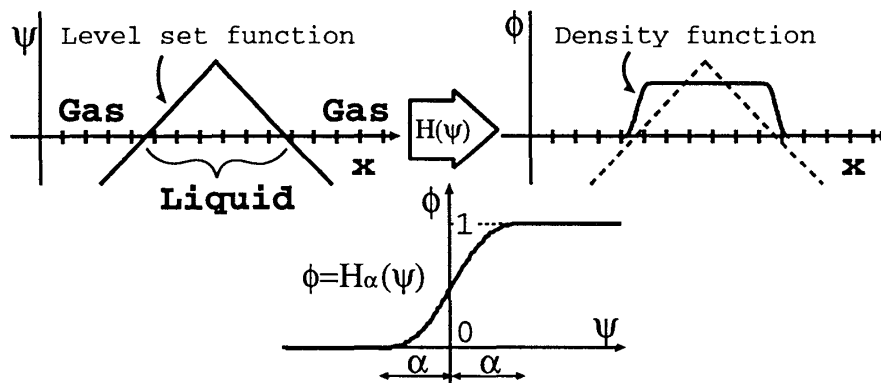


図 10: 左図は level set 関数。level set 関数を H_α (下図) により変換することにより、密度関数 (右図) を生成することができる。

ϕ (図 10 右) は、この level set 関数から H_α (図 10 下, Smoothed Heaviside 関数) を用い以下のように計算する。

$$\phi = H_\alpha(\psi), \tag{19}$$

ここでは、 H_α として

$$H_\alpha(\psi) = \begin{cases} 0 & \text{if } \psi < -\alpha \\ \frac{1}{2} \left[1 + \frac{\psi}{\alpha} + \frac{1}{\pi} \sin\left(\frac{\pi\psi}{\alpha}\right) \right] & \text{if } |\psi| \leq \alpha \\ 1 & \text{if } \psi > \alpha, \end{cases} \tag{20}$$

を採用する。ここで、 2α は仮想的な界面の厚さである。実際の界面に厚さはないが、界面に幅を持たせることにより計算を安定させることができ、また、表面張力を計算する際に必要になる。

そして、界面の時間発展は、移流方程式

$$\frac{\partial \psi}{\partial t} + (\mathbf{u} \cdot \nabla) \psi = 0, \quad (21)$$

によって計算される。ここでは、この移流方程式の計算にも CIP 法を用いる。

しかし、この移流の計算後の ψ は一般に level set 関数の性質がいくらか損なわれる (図 11 (a)). そこで、level set 関数の性質 (18) を回復させるために、zero level set は界面を正確に捉え

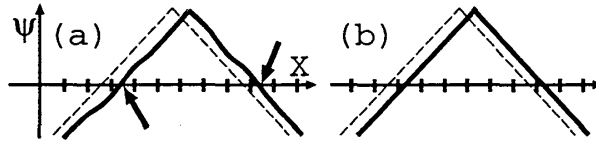


図 11: (a) の点線は初期の level set 関数, 実線は, 移流計算後の level set 関数, 矢印は zero level set を表す. (b) の実線は 再初期化 後の level set 関数である.

ていると考え、zero level set が動かないようにして、level set 関数を再初期化 (Reinitialization⁴) する (図 11 (b)). 再初期化の方法には色々あるが [13], ここでは、その中の一つ、界面を陽的に見つけることなく再初期化する方法 [14] を紹介する. 具体的には (22) の ψ が定常になるまで計算することにより再初期化する.

$$\frac{\partial \psi}{\partial \tau} = S(\psi)(1 - |\nabla \psi|), \quad (22)$$

ここで、 $S(\psi)$ は

$$S(\psi) = \frac{\psi}{\sqrt{\psi^2 + \delta^2}} \simeq \text{sign}(\psi). \quad (23)$$

$S(\psi)$ は、 $\psi = 0$ の所 (zero level set, 界面) では 0 であるため、(22) から分かるように界面は動かない. その他の所では、 $S(\psi)$ が 0 以外の値を持つため、 $|\nabla \psi| = 1$ になり、再初期化される⁵.

この方法では、界面を陽的に (直接) 追跡せず、陰的に (level set 関数を通して) 界面を追跡するため界面のトポロジーが変化するような状況 (例えば 2 つの水滴が衝突するような場合, 図 12) でも問題なく計算することができる.

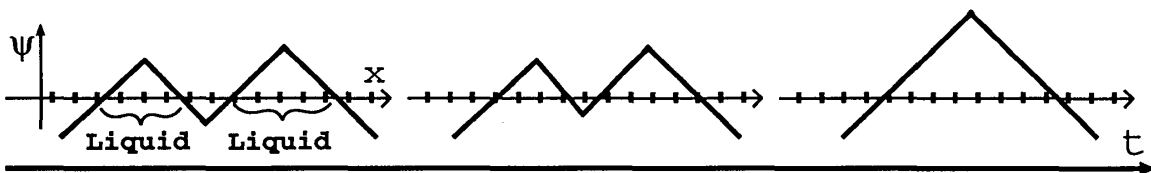


図 12: 水滴衝突時の level set 関数の時間発展. 界面のトポロジーが変化しても、level set 関数の計算では、そのことを気にする必要はない.

⁴ 直訳すると再初期化になるが、意味的には再構築の方が適当だと思われる.

⁵ 図 10 のように level set 関数がぶつかっている所 (とがっている所) は、 $|\nabla \psi| = 1$ にはもちろんならず計算が収束しないように見える. しかし、この問題は実際の計算に使う (22) の差分式を工夫することにより、うまく避けることができる [14].

また, level set 法は, 表面張力の計算にも威力を発揮する. 表面張力は, CSF (Continuum Surface Force) モデル [16] により密度関数の勾配を使い体積力 \mathbf{F}_{sv} として (24) により計算される.

$$\mathbf{F}_{sv} = \sigma \kappa \nabla \phi, \quad (24)$$

ここで, σ は表面張力係数, κ は曲率である. κ は

$$\kappa = -(\nabla \cdot \mathbf{n}), \quad (25)$$

によって計算され, \mathbf{n} は 界面に対する単位法線ベクトルである. \mathbf{n} は, level set 関数の勾配

$$\mathbf{n} = \nabla \psi, \quad (26)$$

により計算される. level set 関数の勾配は, 図 13 から分かるように, まさに単位法線ベクトル

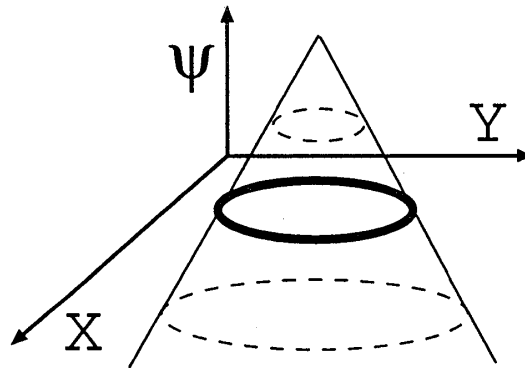


図 13: 2次元空間上に円形の水滴がある場合の level set 関数. 太線は, zero level set (界面).

である. そのため level set 法を用いると, 単位法線ベクトルを精度良く計算できる. 従って, 曲率が精度良く求まり, 表面張力を精度良く計算することができる.

4 Multi-Fluid, Multi-Phase Flow の数値解法

界面が大変形する流体现象のシミュレーションをする場合, 界面に境界条件を課すことは容易ではない. そこで, 界面に, 陽的に境界条件を課さない方法 (界面に特別に境界条件を付けなくても, 自動的に境界条件を満たすように計算できる方法) により計算することを考える. ここでは, そのような方法の一つである CIP-CUP (CIP-Combined Unified Procedure) 法 [17] について紹介する.

CIP-CUP 法では, CIP 法を利用するために, 支配方程式 (1), (2), (3) を Advection part と Non advection part に分離して計算する.

1. Advection part:

$$\frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla) \rho = 0, \quad (27)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = 0, \quad (28)$$

$$\frac{\partial e}{\partial t} + (\mathbf{u} \cdot \nabla) e = 0, \quad (29)$$

2. Non advection part:

$$\frac{\partial \rho}{\partial t} = -\rho \nabla \cdot \mathbf{u}, \quad (30)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{\nabla p}{\rho} + \mathbf{g} + \frac{\mu}{\rho} \Delta \mathbf{u} + \frac{\mathbf{F}_{sv}}{\rho}, \quad (31)$$

$$\frac{\partial e}{\partial t} = -\frac{p}{\rho} \nabla \cdot \mathbf{u}, \quad (32)$$

時間発展は，Advection part と Non-advection part を交互に計算することにより計算される．Advection part には，CIP 法を用い，Non-advection part は，差分法によって計算する．ただし，異なる音速を持つ混相流を扱うために陰的解法の CUP 法を用いる．異なる物質に対する音速は，密度関数により識別する．音速は $C_s^2 = \partial p / \partial \rho$ と定義でき，この関係は以下のように書き換えることができる．

$$\frac{\partial p}{\partial t} = C_s^2 \frac{\partial \rho}{\partial t} \quad (33)$$

(30) と (33) から

$$\frac{\partial p}{\partial t} = -\rho C_s^2 \nabla \cdot \mathbf{u}. \quad (34)$$

が得られる．そして，(30), (31), (34) を差分により計算する．

$$\frac{\rho^{t+\Delta t} - \rho^*}{\Delta t} = -\rho^* \nabla \cdot \mathbf{u}^{**}, \quad (35)$$

$$\frac{\mathbf{u}^{**} - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p^{t+\Delta t}}{\rho^*}, \quad (36)$$

$$\frac{\mathbf{u}^{t+\Delta t} - \mathbf{u}^{**}}{\Delta t} = \mathbf{g} + \frac{\mu}{\rho^*} \Delta \mathbf{u}^{**} + \frac{\mathbf{F}_{sv}}{\rho^*}, \quad (37)$$

$$\frac{p^{t+\Delta t} - p^*}{\Delta t} = -\rho^* C_s^2 \nabla \cdot \mathbf{u}^{**}, \quad (38)$$

* は Advection part を計算した後の値である．ここで，(36) の発散を (38) に代入すると圧力に対する以下の Poisson 方程式が得られる．

$$\nabla \cdot \left(\frac{\nabla p^{t+\Delta t}}{\rho^*} \right) = \frac{p^{t+\Delta t} - p^*}{\rho^* C_s^2 \Delta t^2} + \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}. \quad (39)$$

この Poisson 方程式 (39) を全領域 (液相部分だけでなく気相部分も含めて) に対して計算することにより，それぞれの音速に対応した圧力が計算される．そして，その $p^{t+\Delta t}$ を用い (35), (36), (37) からそれぞれの音速に対応した $\mathbf{u}^{t+\Delta t}$, $\rho^{t+\Delta t}$ が計算される．ここでは，異なる音速を扱う形式を扱ったが，より一般的な異なる状態方程式を扱うことができる形式 [18] も提案されている．

5 適用例

界面の大変形を伴う流体现象への適用例として円形跳水 [19, 20, 21] を取り上げる⁶。円形跳水は、蛇口から定常的に水を流し、その水流が流しに衝突した時に水流を中心にして形成される円形の水位の不連続のことである (図 14)。この現象は、層流かつ定常でありながら界面の大変形

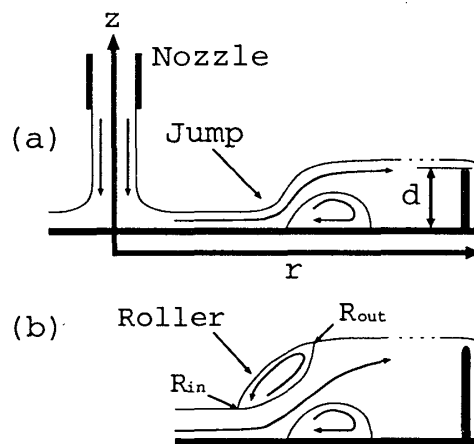


図 14: 円形跳水の概略図. d が 0 または低い時, (a) の様な流れの構造 Type I が形成される. Type I の状態から d を徐々に上げていくとある高さ d_c で, (b) の様な流れの構造 Type II に転移する.

を伴うという希な現象であり、界面の大変形を伴う流体现象の中では最も扱いが容易であると思われる。また、この実験では、常識的に考えれば円形であるはずの跳水が多角形に変化するという実験結果 [22, 23, 21] も報告されており非常に興味深い現象である。

この現象に対し、軸対象を仮定した 2 次元 (r - z) のシミュレーションを行なった。図 15 にシミュレーションと実験の界面プロファイルと比較した結果を示す。界面の変形、境界層、剥離渦、表面

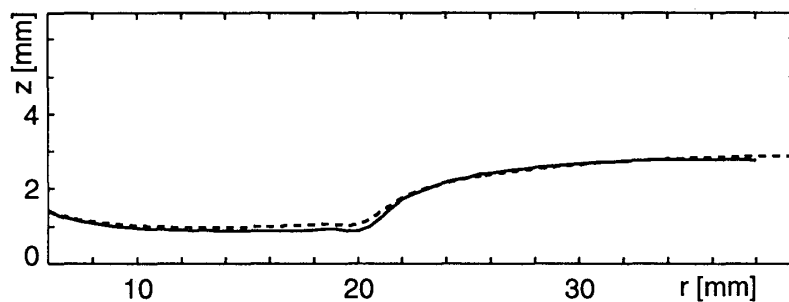


図 15: 実線がシミュレーション, 点線が実験の界面プロファイル. パラメータはすべて実験と同じものを使用. 液体の密度 $\rho_l = 1110 \text{ kg/m}^3$, 気体の密度 $\rho_a = 1.2 \text{ kg/m}^3$, 液体の動粘性係数 $\nu_l = 7.6 \times 10^{-6} \text{ m}^2/\text{s}$ 気体の動粘性係数 $\nu_a = 15.0 \times 10^{-6} \text{ m}^2/\text{s}$, 表面張力係数 $\sigma = 4.5 \times 10^{-2} \text{ N/m}$, 流量 $Q = 27 \text{ ml/s}$. 400×130 のメッシュを用いた.

⁶ 私の研究対象でもありますので、簡単に紹介させていただきます。

張力等の影響が複雑に絡み合った現象にしては、良く一致している。このシミュレーションにより、円形跳水における流れの構造形成について様々なことを明らかにすることができた [24, 25].

また、同様な計算法 (level set 法を使わず、CIP-CUP 法だけで計算) がミルククラウン [27, 2] や碎波 [26] などの自由界面流れの解析に使われ成果を上げている。

6 実際に数値計算する上での注意点

本原稿では、界面の大変形にも耐えられる数値計算法について紹介してきた。この様に書くと、この方法で自由界面流れは何でも計算できる、というような印象を与えてしまったかも知れないが、実際はそうではない。

現段階では、広く何にでも使える計算法はなく、現象により計算法を使い分ける必要がある (自由界面流れに限らず)。界面が大きく変形する現象に対しては、この原稿で紹介した方法は適していると思うが、そうでない自由界面の現象に対しては、ラグランジュメッシュを使う方が良いかもしれない。実際に計算をする場合は、現象により適切な計算法を選択することが重要である (何事もそうだと思いますが)。

ここで紹介した方法は、まだ多くの問題を持っている。一例として、保存則を完全には満たしていないことがあげられる。しかし、保存の破れはわずかであり、保存を保証しない代わりに得られるメリット (界面を陽的に扱うことができる等) が大きいことを忘れてはならない⁷。個人的な考えとして、保存の破れは良いことではないが、保存を保証する計算法では計算できず、多少の保存の破れがほとんど問題にならない現象に対しては、使っても良いのではないかと考える。実際に数値計算を行なう際は、数値計算法の問題点も考慮した上で使う必要がある。

7 おわりに

本原稿で紹介した計算法は、概念を理解するのは比較的容易である。しかし、実際にプログラムを組もうとすると敷居が高いと感じた (私の力不足かもしれませんが)。計算法の考え方はシンプルであっても実際の計算式は複雑であり、計算法とは直接関係ない境界条件に悩まされたり、高性能な行列計算法を使う必要があったりと苦勞が絶えなかった。また、このような計算には多くの経験を必要とするところもある。例えば、行列計算の方法は数多くあり、どの方法が良いかは計算対象に依存し、どの方法が適当かを見究めるのは、意外に難しい。流体の計算法の選択にしても同じである。しかし、それらを乗り越えれば、誰も取り組めなかった問題に取り組めるようになり新たな発見のチャンスが広がると思います。

最後に、共同研究を通じて数値計算法について議論して頂いた東工大 総合理工の肖 先生、原稿を読んで分かり難い箇所を指摘してくれた北大 電子研 情報数理の飯間さん、櫻井さん、牛島さん、野々村さん、上田君に感謝致します。

⁷ 保存の問題は、近年の研究 [28, 29, 13] により、上のメリットを残しながら解決されつつある。

参考文献

- [1] 保原, 大宮司 編, 数値流体力学 基礎と応用, 東京大学出版会, 1992, 数値流体力学編集委員会編, 数値流体力学シリーズ 4 移動境界流れ解析, 東京大学出版会, 1995, など.
- [2] T. Yabe, <http://www.mech.titech.ac.jp/>
- [3] F. Xiao, <http://www2.es.titech.ac.jp/~xiao/>
- [4] J. A. Sethian, http://www.math.berkeley.edu/~sethian/level_set.html
- [5] H. Takewaki, A. Nishiguchi and T. Yabe, *J. Comput. Phys.* **61**, 261 (1987).
- [6] T. Yabe and T. Aoki, *Comput. Phys. Commun.* **66**, 219 (1991).
- [7] T. Yabe et al., *Comput. Phys. Commun.* **66**, 233 (1991).
- [8] T. Aoki, *CFD Journal*, **4**, 279 (1995).
- [9] F. Xiao, T. Yabe and T. Ito, *Comput. Phys. Commun.* **93**, 1 (1996).
- [10] F. Xiao, et al., *Comput. Phys. Commun.* **94**, 103 (1996).
- [11] CIPUS Annual Report (1997).
- [12] S. Osher, and J. A. Sethian, *J. Comput. Phys.* **79**, 12 (1988).
- [13] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, 1999.
- [14] M. Sussman, P. Smereka, and S. Osher, *J. Comput. Phys.* **114**, 146 (1994).
- [15] T. Yabe, F. Xiao, *J. Phy. Soc. Jpn.* **62**, 2537 (1993).
- [16] J. U. Brackbill, D. B. Kothe, and C. Zemach, *J. Comput. Phys.* **100**, 335 (1992).
- [17] T. Yabe, and P. Y. Wang, *J. Phy. Soc. Jpn.* **60**, 2105 (1991).
- [18] F. Xiao, et al., *Comput. Phys. Commun.* **102**, 147 (1997).
- [19] T. Bohr et al, *Physica B* **228**, 1 (1996).
- [20] T. Bohr, V. Putkaradze, and S. Watanabe, *Phys. Rev. Lett.* **79**, 1038 (1997).
- [21] S. Watanabe, <http://mitukuni.sci.ibaraki.ac.jp/~shinya>
- [22] C. Ellegaard et al., *Nature* **392**, 767 (1998).

- [23] C. Ellegaard et al., *Nonlinearity* **12**, 1 (1999).
- [24] K. Yokoi and F. Xiao, *Phys. Lett. A* **257**, 153 (1999).
- [25] K. Yokoi and F. Xiao, *Phys. Rev. E* **61**, R1016 (2000).
- [26] 陸田 秀実, 大規模碎波による気液混相流体場における高精度数値計算手法の開発, 博士論文 (岐阜大学工学部), 2000.
- [27] Y. Zhang, T. Yabe, *CFD Journal*, in press.
- [28] R. Tanaka, T. Nakamura and T. Yabe, *Comput. Phys. Commun.***126**, 232 (2000).
- [29] M. Sussman and E. Fatemi, *SIAM J. Sci. Comput.* **20**, 1165 (1999).