

# 応用数学の概念を用いた物理モデルをいかに Julia でプログラミングを行うか

---

降旗 大介 (大阪大学サイバーメディアセンター)

`daisuke.furihata.cmc@osaka-u.ac.jp`

数学と物理における Julia の活用@九州大学 IMI, 2023.07.11

## Tips: 数値計算プログラムを組む上で Julia だと嬉しいこと

- 粗い近似解をある程度良い近似解へ近づける (繰り返し) 手法・アルゴリズムはたくさんあるし、原理もしっかりした、明快なものが多い。しかしその後、  
**ある程度良い近似解から、相当に良い近似解  $\cong$  ほぼ正しい解を得る**  
 **$\iff$  非線形方程式の根を求める**  
方法は結構実装が面倒。これに対し、Julia には “NLsolve” package があり、とても助かる…
- 科学技術計算プログラムでは連立一次方程式の解を求めるシーンが多いのに、通常のコンピュータ言語にはそういったライブラリは標準装備ではない。ライブラリを探して、関数をマニュアルで調べて、使い方を確認して… 面倒だ！  
 $\implies$  Julia だと “LinearAlgebra” package が標準装備なので、とても楽。  
例: 行列  $A$  とベクトル  $x, b$  に対して  $Ax = b$  を解いて  $x$  を求めたい場合、 $x = A \setminus b$  と書くだけで良い！
- 既に多くの科学技術系のライブラリが存在する！ そのうえ、dump コマンドをうまく使うと、ライブラリの計算結果を自由に抽出できたりする。  
 $\implies$  ライブラリのソースを読まなくても、内部結果などを知ることができる。

- Cahn–Hilliard 方程式系統の問題 (相分離現象などのモデル方程式) の数値解析をしたい

▶ movie(method of line による数値解)

▶ movie(DVDM による数値解)

- これらの問題には質量保存, エネルギー減少性などの大域的な性質がある. そもそも Cahn–Hilliard にかぎらずこうした問題は多い.
- これらを保存する数値解析法 構造保存数値解法 structure-preserving method の研究はそれなりに内外で発展している (例: SciCADE という国際研究集会ではこの内容のセッションがいつもある).
- structure-preserving method は数値解が優れていることが多い.
- ODE だとハミルトン系  $+\alpha$  で考えることが多いが, PDE だと変分構造を介して考えることが多い (PDE のそれは離散変分導関数法などと称していて日本発).
- これらの手法は基本的に微積を始めとした「連続極限で定義される operator」を離散的に定義してその一貫性を担保する方法. ある意味数学的に厳密. **ただし計算量は大きめで, そこが難点だが, 近年では高速化手法も充実してきた.**
- 上の枠組みはそれなりに充実してきているので, そろそろ根本的に違う切り口も考えたい

相分離現象のモデル方程式の一つで, 点  $(x, t)$  において状態が A 相, B 相のいずれに近いかを示す物理量  $u(x, t)$  に対し以下のような PDE で記述されるもの (多くの場合, A 相状態を  $u = 0$ , B 相状態を  $u = 1$  などとする).

$$\frac{\partial u}{\partial t} = \Delta \left[ -\epsilon \Delta u + \frac{d}{du} P(u) \right] \quad \text{for } t \in [0, \infty), x \in \Omega \quad (1)$$

ただし,  $\Delta$  は  $x$  に関するラプラシアン,  $0 < \epsilon \ll 1$ ,  $P(u)$  は A 相状態, B 相状態を底にもつ二重井戸型関数, 境界条件は説明略.

このとき, (適切な境界条件のもとで) 次の 2 つの大域的性質がある.

## 質量保存性

$$\frac{d}{dt} \int u \, dx = 0 \quad (2)$$

## エネルギー減少性

$$\frac{d}{dt} \int G(u, \nabla u) \, dx \leq 0 \quad (3)$$

ただし,  $G$  はエネルギー関数と呼ばれる下記の量.

$$G(u, \nabla u) = \frac{\epsilon}{2} |\nabla u|^2 + P(u) \quad (4)$$

## Cahn–Hilliard 方程式 概要 (2)

とくにエネルギー減少性は重要で、解の存在性を証明する数学的なキーでもある。その方程式との関連性は明瞭で以下の通り。

エネルギー関数  $G$  に対して、その変分導関数が

$$\frac{\delta G(u, \nabla u)}{\delta u} = -\epsilon \Delta u + \frac{d}{du} P(u) \quad (5)$$

と導出できるので、これを用いて Cahn–Hilliard 方程式は下記の形に書き換えられる (**この形式がポイント**)。

$$\frac{\partial u}{\partial t} = \Delta \frac{\delta G}{\delta u} \quad (6)$$

よってエネルギー減少性は

$$\begin{aligned} \frac{d}{dt} \int G(u, \nabla u) dx &= \int \frac{\delta G}{\delta u} \frac{\partial u}{\partial t} dx + \text{境界項}^* \\ &= \int \frac{\delta G}{\delta u} \Delta \frac{\delta G}{\delta u} dx = - \int \left\| \nabla \frac{\delta G}{\delta u} \right\|^2 dx + \text{境界項}^* \leq 0 \end{aligned} \quad (7)$$

\* : 想定されている境界条件によって 0 となる。

注: 離散変分導関数法はこのページを丸々離散化する

PDE に対する普通の解法，特殊な解法等が使える。

- **method of line**(空間方向を先に離散化して連立 ODE で近似し，ODE 数値解法で解く)。

▶ [この解法の解説 pdf](#)

時間方向の離散化幅  $\Delta t$  を小さくすれば一応使えるが，質量保存性，エネルギー減少性はときおり破れ，そのせいで物理的にみておかしい数値解になりがち (とくに長時間発展させた場合)。

- 構造保存数値解法としての**離散変分導関数法** (差分法，有限要素法等いろいろあり．線形スキームもある)。

長時間発展も考えると現在の本命はこちらか．基本的にきちんと動くし，数値解の様子が物理的におかしいということもあまり無い．数値解の一意存在性や安定性が証明できるケースもある．ただし，多くの場合数値スキームが時間方向に陰的 (未知の数値解に対して連立非線形方程式を解いて数値解を求めるという形のこと) で，**計算量は大きめ**。

線形スキームを設計する方法論 (多段化) もあるが，問題の非線形性が多項式の形状でないと使えないこと，4 次を越えると数値的な不安定性が強くなることなどの制限がある。

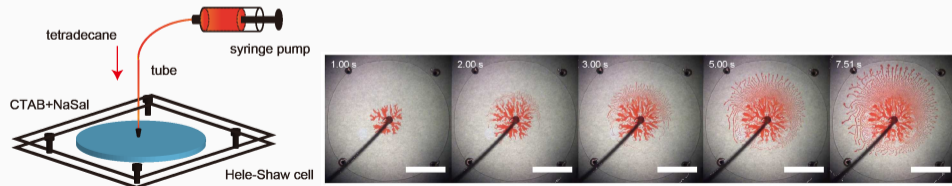
「妥当な数値解を高速に得る」という方法論については，充実しているとはまだ言い難い。

## Cahn–Hilliard 方程式の order reduction: 現状

妥当な数値解を得るための方法の一つに, order reduction (問題をより低次元の問題に置き換えて解く)がある.

- Alikakos らが  $\epsilon \rightarrow 0$  のときこの粗視化過程の相の境界面の挙動が Hele–Shaw 問題 (領域境界面の挙動問題) の解に収束することを証明<sup>†</sup>.
- order reduction としては上の結果は数学的にはほぼ満足 (問題の次元が文字通り 1 次元下がる).
- ただし, Hele–Shaw 問題の数値解析は厄介. むしろ「Hele–Shaw 問題の数値解法として Cahn–Hilliard 方程式を解く方法」が提案される状態.

<sup>†</sup> N. D. Alikakos, P. W. Bates and X. Chen, Convergence of the Cahn–Hilliard equation to the Hele–Shaw model, Arch. Rat. Mech. Anal. 128 (1994), 165–205.



Hele–Shawflow の実験模式図と実験画像 (K.Yoshii and Y.Sumino, arXiv 1904.10673, 2019. より引用)

背景・アイデアのスタート地点:

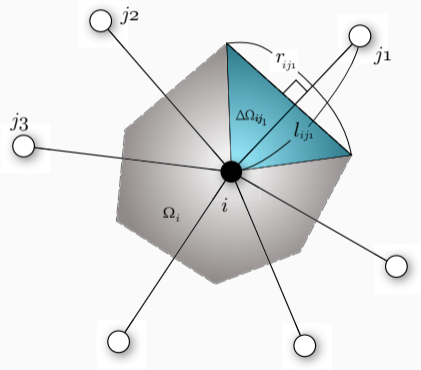
- 微分幾何をベースに, 粒子法的な particle dynamics model 方程式を作る. つまり, 各地点での  $u(x, t)$  が時間変化するという問題ではなく, (たとえば)  $u \cong 1$  な粒子が移動する, という問題に書き換える.
- このとき解決すべき問題は大きくわけて以下の2つ.
  - 1 「解  $u$  の値が大きい  $\iff$  粒子がそこで密集している」という関係性をどうやって定量化するか.
  - 2 関数  $u(x, t)$  に対するラプラシアン  $\Delta$  などの微分作用素を, 粒子の位置関係によって発生する力などにどうやって解釈し直すか.
- 実は上の2つの問題とも, 「粒子の位置を母点, 生成点とした空間分割」をもとにした**離散微分**を用いることで近似として解決可能  $\implies$  Voronoi 分割 (次スライド) などが代表的.



## ボロノイ分割とは: 一筋の良い空間分割

- 任意の点はもっとも近いボロノイ領域に含まれるとする. つまり, 母点  $x_i$  によるボロノイ領域  $\Omega_i$  を以下のように定義する.

$$\Omega_i \stackrel{\text{def}}{=} \{x \mid \|x - x_i\| \leq \|x - x_j\| \text{ for any } j \neq i\}$$



- 母点, 生成元 = 空間をこの点をもとにボロノイ分割する
- ボロノイ点 = ボロノイ領域の頂点
- $\Omega_i$  と  $\Omega_j$  が隣接しているとき

$$\begin{cases} r_{ij} & \stackrel{\text{def}}{=} |\Omega_i \cap \Omega_j|, \\ l_{ij} & \stackrel{\text{def}}{=} \|x_i - x_j\|. \end{cases}$$

▶ 母点から Voronoi 分割が進む様子の movie

▶ Voronoi 分割等の離散化に伴う微積分の離散化

## 方程式を粒子法相当に真面目に離散化した方程式






モデル方程式の全容は以下ようになる.

$$\begin{aligned}\frac{dx_i}{dt} &= -\left(\frac{1}{4m}\right) \sum_{(M_{ad})_{ij} \text{ is true}} \left. \frac{\delta G}{\delta u} \right|_i n_{ij} r_{ij}, \\ \left. \frac{\delta G}{\delta u} \right|_i &= p(2u_i - 1) + r(u_i - 1)^3 + \frac{2q}{|\Omega_i|} \sum_{(M_{ad})_{ij} \text{ is true}} \left(\frac{u_j - u_i}{l_{ij}}\right) r_{ij}, \\ u_i &= \frac{m}{|\Omega_i|}.\end{aligned}$$

ただし,  $x_i(t)$  は  $i$ -番 particle の位置,  $m$  は particle の大きさ (面積),  $p, q, r$  は Cahn–Hilliard 方程式の定数.  $|\Omega_i|$  は母点  $x_i$  により生成されたボロノイ領域  $\Omega_i$  の大きさで, 行列  $M_{ad}$  はボロノイ領域の隣接関係を列挙した隣接関係行列.  $n_{ij}, l_{ij}, r_{ij}$  は隣接している場合のみ非ゼロで, それぞれ particle  $i$  と particle  $j$  間境界の ( $i$  からみた) 外向き単位法線ベクトル, particle 間の距離, 同境界の長さである.

このモデル方程式は数学的には根拠がある合理的なものだが, 粒子の位置が近づきすぎたり, 領域の外へ行ってしまったり (密度  $\rightarrow 0$  を実現しようとして) すると Voronoi 分割が不安定になるため, 実際の計算にはそうとうの繊細さが必要となる.

## 数値計算結果より見えてくること (method 1)

-  数学的にはきちんと近似しているので、(計算が可能なら) 計算結果はそれなりに信頼できそう.
-  密度の高いところに粒子がたくさん集まり空間分割の解像度が高くなるので、メッシュ調整の一種になっている.
-  離散近似の際に空間次元に依存する箇所がないので、3次元、4次元でも問題ない.
-  点の距離が近くなると Voronoi 分割の数値計算は不安定になりがち.
-  空間の「外へ」出ていこうとする粒子が存在しうるので、この対処が必要.

というわけで、もう少し「簡潔なモデル」を考えたい.

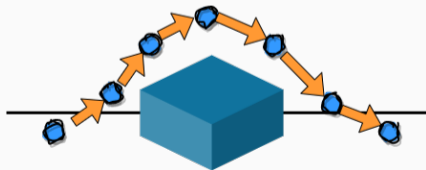
## Cahn–Hilliard 方程式の粗い order reduction を作るには (method 2)

背景・アイデアのスタート地点:

- Cahn–Hilliard 方程式を数学的な手法で order reduction するのはいったん止めて…
- もともとの相分離現象における「粗視化過程 (coarsening process)」のみを再現する素朴なモデルを考える
- その際、少なくとも質量保存性を再現する (そうしないと物理的にかなり奇妙に見える)

粗視化過程を観察すると:

- 全体に「流体」現象のような挙動
- 少領域同士に主に距離に依存した引力的な相互作用がありそう
- 流体的挙動と考えると、近くの領域を「直接越えて」力が働くモデルには違和感



流体では、物質同士の相互作用により力が障害物を迂回して伝わる ⇒ 迂回しての隣接関係が成り立つようにしたい…

というわけで、

## particledynamics model of Cahn–Hilliard 方程式 (1/2)

- $u \cong 1$  で半径が一定の球 (particle) の位置移動のみを記述するモデルを考える.
- particle は増減しないし, 大きさも変わらないし, 重ならない. これによって質量保存性を確保.
- particle 同士は, 互いの距離に応じて引き合う (とりあえず逆二乗則)
- ただし, particle が重ならないように非常に近い場合は体積排除効果的な力が働く.
- ただし, 近くの particle を何重にも越えての力はあまり働かない (緩和隣接関係)

これは例えば下記のようにしてモデル化する.

- 1 空間上の particle をそれぞれ母点として空間を **Voronoi 分割**し, Voronoi 領域の隣接関係から作られる隣接行列を  $M_{ad}$  とする.
  - 2 モデルパラメータ  $p$  (正整数) に対し, 緩和隣接行列を  $M_p \stackrel{\text{def}}{=} (M_{ad})^p$  とし,  $M_p$  によって隣接していると判断される particle 間でのみ力が発生するとする.
- 化学物質の易動度の概念のように, particle に働く力  $\propto$  particle の速度
  - 流体抵抗のように, particle の速度には上限あり

…という, いわば **re-modeling** を行う

モデル方程式の全容は、具体的には以下の通り。

$$\begin{aligned}\frac{dx_i(t)}{dt} &= C_1 \left( \frac{\tanh(C_2 \|f_i(t)\|)}{C_2 \|f_i(t)\|} \right) f_i(t), \\ f_i(t) &= \sum_{(M_p)_{ij} \text{ is true}} f_{ij}(t), \\ f_{ij}(t) &= \begin{cases} C_3 \frac{x_j - x_i}{\|x_j - x_i\|^3} & : \|x_j - x_i\| \geq r_{\text{cri}}, \\ -2C_3 \frac{x_j - x_i}{\|x_j - x_i\|^3} & : \|x_j - x_i\| < r_{\text{cri}}. \end{cases}\end{aligned}$$

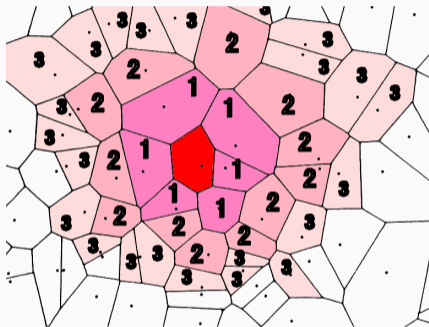
ただし、 $x_i(t)$  は  $i$ -番 particle の位置、 $C_1, C_2, C_3$  は正定数。

行列  $M_p$  は  $M_p \stackrel{\text{def}}{=} (M_{\text{ad}})^p$  ( $p$  は正整数, 3 程度), その計算過程で必要な真偽値計算 (ブール値の計算) は  $a \cdot b = a$  and  $b$ ,  $a + b = a$  or  $b$  となる。

行列  $M_{\text{ad}}$  はボロノイ領域の隣接関係を列挙した隣接関係行列で、母点  $x_i$  により生成されたボロノイ領域  $V_i$  と母点  $x_j$  により生成されたボロノイ領域  $V_j$  とが隣接しているときは  $(M_{\text{ad}})_{ij}$  が真, そうでなければ偽である。

## ポロノイ領域の隣接関係

図の中心の赤いポロノイ領域  $V_i$  を対しての隣接関係行列  $(M_{\text{ad}})^p$  ( $p$  は正整数) の意味は…



空間のポロノイ分割の例.

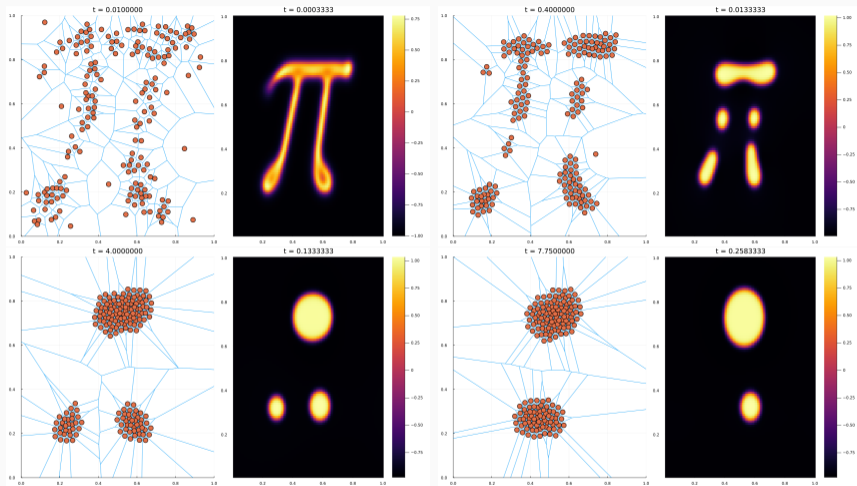
- 1 が書かれた領域 …  $(M_{\text{ad}})^1$  の  $(i, j)$  成分が「真」となるポロノイ領域  $V_j$ .
- 2 が書かれた領域 …  $(M_{\text{ad}})^2$  の  $(i, j)$  成分が「真」となるポロノイ領域  $V_j$ .
- 3 が書かれた領域 …  $(M_{\text{ad}})^3$  の  $(i, j)$  成分が「真」となるポロノイ領域  $V_j$ .

… 注目！ この隣接関係は「迂回路」を持ちうる  
→ 流体の「回り込み」挙動をモデリングできる.

▶ この解法の julia code

▶ >> movie (初期配置が random)

# 実験結果: 粗視化過程をそれなりに re-modeling できたのか?



左: particle モデルによる数値解, 右: Cahn-Hilliard 方程式による数値解

>> movie (初期配置が pi 形状)



- 👍 **非常に遅い挙動が発現している**. この挙動は領域の移動・変形過程で見られる特徴の一つである.
- 👍 **領域の移動速度がそのサイズに依存する** (大きいほど遅い). こうした効果をモデルに直接取り入れているわけではないので興味深い.
- 👍 **質量保存則が成り立つ**. これはモデルの設計からして当然.
- ⚠️ **決定方法が不明な未知パラメータが含まれる**モデルである. 具体的には,  $C_1, C_2, C_3, p$  などが未知パラメータ. これらをどう決めると Cahn–Hilliard 方程式の解挙動を近似できるのか, は future work.
  - 領域の生成・消滅メカニズムが陽的にはこのモデルには組み込まれていない. ただし, 擬似的にこうしたメカニズムが再現はされている様子.  
なお, この欠如は今回のモデルとしては対象現象から切り捨てたから当然であるが, 組み込めるならばそれに越したことはない. (→ future work)

- 👍 Cahn–Hilliard 方程式の解の挙動をシミュレートするためのシンプルな粗視化モデルとして **particle motion model** を設計した.
- 👍 数値計算例をみるとこのモデルで領域の移動・変形に**遅い挙動が発現**しており, Cahn–Hilliard 方程式の粗視化モデルとして期待できる.
- ⚠️ モデルに含まれる未知パラメータの決定方法は現状不明.
  - Cahn–Hilliard 方程式の大域エネルギーに相当するこのモデルでの特徴量を定義・計算し, その挙動を調べる必要がある. なぜなら Cahn–Hilliard 方程式の大域エネルギーは時間発展に伴い減少する散逸量であることが判明しているため.
  - 3次元領域でのこのモデルでの数値計算の実験等を行いたい.  
なお, 下記のように, 空間次元を上げて, 計算コストは指数的に増大したりしない.
    - 2次元問題では空間のボロノイ分割の計算コストは多くのアルゴリズムで  $O(n \log n)$ .  
ただし  $n = \# \text{ points}$ .
    - Delaunay 三角分割を行ってボロノイ分割を計算する Bowyer–Watson アルゴリズムは, 問題の次元によらず (おお!), 普通は計算コストが  $O(n \log n)$ .  
ただし最悪のケース (点配置が退化している場合) では  $O(n^2)$  になりうる.

- 👍 複雑な問題に対しても, Julia + jupyter 環境だと, 少しずつ試行錯誤することで意外に気楽にプログラムの完成にこぎつけられる.
- 👍 想像するより多くのライブラリが有り, かなり「頼る」ことができる.
- 👍 他の言語にはもう戻れない!