

遺伝的プログラミングを利用したボルツマン方程式衝突項の 近似モデル探索

服部 篤樹（東大院），鈴木 宏二郎（東大）

Search for Approximate Model of Collision Term in Boltzmann Equation by Genetic Programming

HATTORI Atsuki, SUZUKI Kojiro (Univ. Tokyo)

ABSTRACT

The machine learning approach for exploring the approximate analytical description for the collision term in the Boltzmann equation is presented in this paper. Such approximate model is expected to significantly accelerate the numerical simulation of rarefied gas flows without severe degradation in the accuracy. The combination of a genetic programming (GP) method and a nonlinear least square method was adopted for the exploring. Training data was a solution of Direct Simulation Monte Carlo (DSMC) method, which is a stochastic method for rarefied gas dynamics. Considering the separation of the collision term from the translational term in the Boltzmann equation, we only solved the molecular collision by the DSMC method. The numerical solution of the obtained approximate model exhibited almost precise model. But there was still unphysical behavior in the profile of the velocity distribution function. It is suggested that GP may produce the equation including terms with numerical instability and that some restriction should be applied to assure the numerical stability as well as the physical accuracy at the exploring by GP.

1 序論

1.1 DSMC 法

流れの密度が十分に低下した，流れが希薄な状態では，Navier-Stokes 方程式は成立しなくなる．その場合には式 (1) の Boltzmann 方程式を基礎式として扱わなければならない．

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = \int_{\mathbf{v}_1} \int_{\Omega'} (f f_1' - f f_1) V \sigma d\Omega' d^3v_1 \quad (1)$$

ここで， f は速度分布関数， t は時間， \mathbf{r} は位置ベクトル， \mathbf{v} は分子速度ベクトル， Ω は衝突角， V は分子速度ベクトルの大きさ， σ は衝突断面積を表し， $'$ のあるものは衝突後，ないものは衝突前の速度を表す．Boltzmann 方程式では方程式の形が通常の流体力学の偏微分形とは異なった，気体分子運動論に基づいた形になっているため，差分法などの数値解析手法をそのまま適用することができない．そこで，一般的には確率解法として Direct Simulation Monte Carlo (DSMC) 法¹⁾²⁾ が用いられる．

DSMC 法は一般に粒子法と呼ばれる手法と同じく，計算領域内に大量のモデル粒子を配置し，その挙動を気体分子運動論に基づいて逐次時間ステップ毎に

各粒子の移動や衝突を計算する方法である．しかし，DSMC 法は複数回同様な計算を行い平均を計算することで精度を担保する確率解法となっており，常に統計的ノイズが伴うために精度を確保できない流れも存在する．例えば，マッハ数が 0.01 といった遅い流れでは巨視的な流れ場がノイズに埋没し，DSMC 法では解析できない．³⁾

一方で決定論的手法により Boltzmann 方程式を解くことを考えたとき物理空間×分子速度空間の超高次元空間で格子生成する必要があり，膨大なメモリと計算時間を必要とすることが予測される．また，特に決定論的手法で問題になるのは右辺の衝突積分の高精度な計算である．BGK 方程式のように模型方程式と呼ばれる衝突項を簡単に表すモデルも存在するが，DSMC 法の結果を再現できる精度を持たせることはできない．そこで衝突項を精度を損なわずに比較的簡単な式で記述できる方法が見つければ有用であるが，これまでの気体分子運動論から理論的、数学的に模型方程式を求めるアプローチで十分な精度を確保できるモデルは確立されていない．⁴⁾⁵⁾

そこで本稿では，DSMC 法の結果を入力データとして機械学習による式の探索を行い，十分な精度を確保できる模型方程式の作成を目的とした．

2 方程式探手法

2.1 方程式探手法アルゴリズム

方程式探手法アルゴリズムとは、大量のデータからそのデータ固有の式を探索する手法全体を指しており、手法、応用先も様々に存在する。例えば系に存在する保存則を求める手法⁶⁾や乱流モデルの改善⁷⁾、系の支配方程式を求める手法にもスパース回帰⁸⁾⁹⁾やガウス過程¹⁰⁾などがある。

本稿で注目したのはその中でも遺伝的プログラミングと呼ばれる手法である。スパース回帰やガウス過程では予め推定式にどのような項が含まれる可能性があるか予想し、その項で表現できる範囲内の解を求めている。しかし、本稿の対象のようにどのような項が含まれるか不明な場合には予想を立てることが困難であり、適用が難しい。一方で遺伝的プログラミングでは、その過程で計算機が新たな項を生み出し、より高い自由度で方程式探手法を行うことができる。よって、本稿の探手法では遺伝的プログラミングを採用した。

2.2 遺伝的プログラミング

遺伝的プログラミング¹¹⁾(Genetic Programming, 以降 GP と称する)は遺伝的アルゴリズムの一種であり、特徴として一個体が木構造として存在することである。各個体は例えば図 1 に示されるような構造で数式として存在している。図 1(a) を例としてその読み方を説明すると、

- 頂点の“-”が(左のノード) - (右のノード)を表す
- (左のノード)は“+”であり、さらに下の(左のノード) + (右の子ノード)を表す
- x, y は変数を表す

以下同様にする事で、図 1(a) は次式を表していることがわかる。

$$(x + (x + y)) - (1 \times (y + (x \times x)))$$

このようにして木構造を用いることで様々な式表現を行うことができる。

ここで、本稿で用いた GP の手法を簡潔にまとめると、以下のような手順で行われる。

1. 初期個体の生成
2. 木構造の数式化、簡略化
3. 評価関数に基づいて各個体の係数を最適化
4. 個体の選抜
5. 交叉, 突然変異による世代の更新
6. 2.へ戻る

ここで各手順に関して詳しく説明する。

表 1 ノードの種類

Group	children num.	Name of node (Symbol)
Value	0	Const (c_k), Variable (x_i, u_j)
Operator	2	Add (+), Subtract (-) Multiply (\times), Divide (/)

2.2.1 初期個体の生成

GP ではノードの組み合わせによって木構造を表現する。本研究では表 1 にあるノードを使用して木構造を構築した。Const は定数であり、各個体ごとに独立変数によらない実数が入る。Variable のうち、 x_i は独立変数、 u_j は従属変数を指す。初期個体は表 1 の一覧からランダムにノードを選択して繋げることで作成する。

2.2.2 木構造の数式化、簡略化

GP においては世代が更新されて行くにつれて木構造が複雑に、冗長なものになる傾向が見られる。この傾向は一般に Bloat と呼ばれ、Bloat により多大なメモリの消費となり、数式の多様性の妨げとなる。本研究では Python の記号計算ライブラリである SymPy を用いることでその問題を回避した。具体的には木構造で表現された各個体を一度数式へと変換し、SymPy により数式の簡略化を行う。その後逆ポーランド記法を用いて簡略化された数式を木構造へと変換し直す。この手法の導入により木構造の簡略化が可能になった。例えば、図 1(a)(b) はともに $2x - x^2$ という数式を表しているが、本手法の導入により (a) のような複雑な木構造も同図 (b) のような簡潔な木構造へと変換することが可能となった。

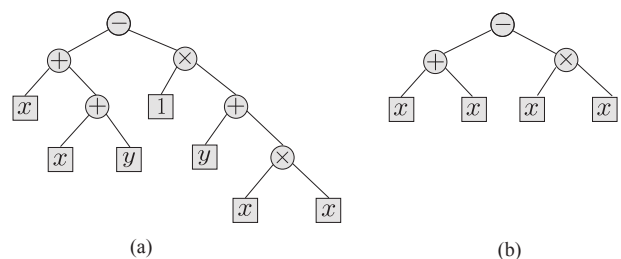


図 1 木構造による数式表現

2.2.3 評価関数に基づいて各個体の係数を最適化

本研究では推定対象となる式が微分方程式である。すなわち、独立変数 x_i 、従属変数 u_j 、定数 c_k を含む以下の性質を満たす方程式を探索することになる。

$$f(x_i, u_j, c_k) = 0 \quad (2)$$

しかし、実問題を考えた時に式 (2) を厳密に満たす方程式を見つけ出すことは不可能であり、アルゴリズム

中では以下の性質を満たす方程式を探すことになる。

$$\text{minimize } \|f(x_i, u_j, c_k)\|_2 \quad (3)$$

しかし、式 (3) では自明な解が得られてしまう。すなわち実際に求めたい方程式ではなく、恒等的に 0 になる式、例えば $x - x$ や 0 などの式が得られてしまう。故に本稿では

$$\frac{\partial u_0}{\partial t} = f(x_i, u_j, c_k) \quad (4)$$

という形式の方程式を推定対象とし、

$$\text{minimize } \left\| \frac{\partial u_0}{\partial t} - f(x_i, u_j, c_k) \right\|_2 \quad (5)$$

を評価関数として用いた。ここで、 $\partial u_0 / \partial t$ は学習データから得られた真値であり、 $f(x_i, u_j, c_k)$ は GP により得られた式から計算される推定値である。

GP では定数 $Const$ を最適化することで数式の各項に係数を与えることが可能であるが、一般にその方式による係数の最適化は効率が悪い¹²⁾。本研究では数式構造を GP により生成した後に各項の係数は後述する別の最適化アルゴリズムを用いた。

まず、数式構造を項ごとに分解する手法に関して、SymPy を用いて実装した。ここでの項とは、数式を線形和で表現した時にその数式を構成するそれ以上展開も分割もできない状態の単項式のことを指す。そして分解後に各項に係数 a_i を定めた。例えばある個体は以下のように分解、係数付けを行われる。

$$x \left(\frac{2x + c_1 + 2y}{c_1 + x + y} + x \right) \quad (6)$$

$$= \frac{x^2}{c_1 + x + y} + \frac{xy}{c_1 + x + y} + x + x^2 \quad (7)$$

$$\rightarrow a_0 \frac{x^2}{c_1 + x + y} + a_1 \frac{xy}{c_1 + x + y} + a_2 x + a_3 x^2 \quad (8)$$

次に係数の最適化手法に関して、評価関数として式 (5) を用いて、最小二乗法により係数を定める。また、係数に対して数式が線形とは限らないため非線形の最小二乗法を用いる必要がある。ここでは先行研究¹²⁾¹³⁾でも頻繁に用いられている Levenberg-Marquardt 法¹⁴⁾を用いた。

2.2.4 個体の選抜

各個体の係数を定めると評価関数式 (5) の値が決まる。この値を適応度と呼ぶことにする。適応度に応じてどの個体を次世代に残すか選抜する必要がある。その手法は様々に提案されているが、本稿ではトーナメント方式を採用した。

トーナメント方式とは、全個体からある一定数 N の個体をランダムに選び、その中で最も適応度が高いもののみを次世代に残す選抜方法である。 N が小さいほ

ど低い適応度を持つ個体も残りやすくなり、また、選ばれた個体群 N の中で最大になれば次世代に残れることから、比較的幅広い適応度の個体が次世代に残ることになる。これにより、GP による探索で局所解に陥ることを防いでいる。本稿では全て $N = 10$ として GP を行った。

2.2.5 交叉、突然変異による世代の更新

次世代個体の選抜後には個体間の交叉、突然変異を作用させることで次世代個体が全て用意し終わることになる。遺伝的アルゴリズムの遺伝子とは違い、GP では木構造という形になっているので、交叉の仕方も異なる。図 2 に交叉の方法を示す。まず 2 個体の親世代の木構造のうち、それぞれのあるノードを指定する。次に、そのノードより下層の部分木を 2 つの個体間で交換する。これにより子世代の 2 個体を生み出すことができる。また、突然変異の場合も交叉の時と同様に部分木の交換をすることで行うが、交叉と異なる点としては親世代は 1 個体しか選ばず、突然変異した部分木はランダムに生成し、選択されたノードと交換する点である。

交叉、突然変異はそれぞれある確率で発生する。また、交叉や突然変異後の子世代個体の木構造には深度の制限をかけて、過度に複雑な木構造となることを防いでいる。なお、深度とは木構造が第何階層まで持つかを指しており、例えば図 1(a) は深度 4 の木構造である。

本稿では、交叉確率 0.5、突然変異確率 0.4、最大深度 5 となるように GP を設定した。

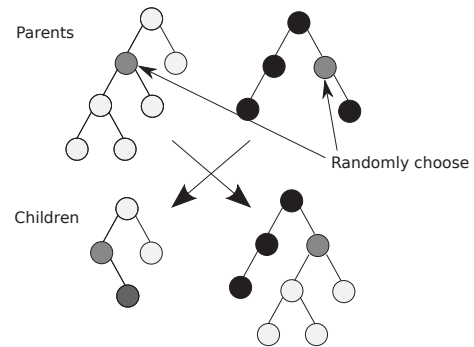


図 2 交叉手法

3 DSMC 法による衝突項の計算

本稿では簡略化のために式 (9) に示す移流項、外力項のない Boltzmann 方程式の衝突項を方程式探索した。

$$\frac{\partial f}{\partial t} = \int_{\mathbf{v}_1} \int_{\Omega'} (f f_1' - f f_1) V \sigma d\Omega' d^3 \mathbf{v}_1 \quad (9)$$

すなわち、通常速度分布関数は時刻、位置、分子速度に依存し $f(t, \mathbf{r}, \mathbf{v})$ となるが、本研究では速度分布関数は位置によらず一定であり、 $f(t, \mathbf{v})$ となると想定する。この想定下では速度分布関数 f は任意の初期条件から分子間衝突による速度交換を経て Maxwell-Boltzmann 分布に収束する。その収束に至る時定数は気体の希薄度を示すクヌーセン数 (Kn) に支配されており、クヌーセン数が違えば収束に至る時間も異なってくる。

GP により衝突項の近似モデル探索を行うために、入力するデータとして式 (9) に従う速度分布関数の時間履歴が必要となる。そこで最大衝突数法を用いた DSMC 法により式 (9) に従う、ある空間の一点における速度分布関数の履歴データを作成した。本稿では、特に $Kn = 0.4$ として計算を行い、次の二種類の初期速度分布からの時間履歴データを作成した。

- 二重正規分布型 (図 3)
- 異方正規分布型 (図 4)

図 3, 4 に速度分布関数の形をそれぞれ示す。二重正規分布型の場合、 x, y, z 方向には違いがないが、速度分布が正規分布から大きく離れた形になっている。一方、異方正規分布型の場合、速度分布の形は正規分布になっているが、 x, y, z 方向それぞれに対して分散が異なる。

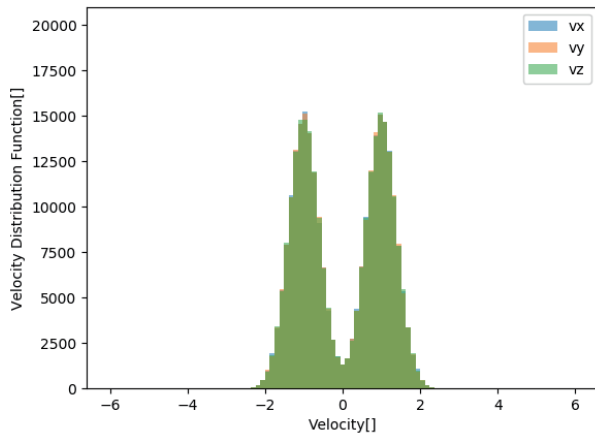


図 3 二重正規分布

4 非平衡緩和現象記述への GP の適用

4.1 条件設定

DSMC 法による衝突計算から得られたデータの GP への適用方法を述べる。評価関数に関して、 f_{DSMC} を DSMC 法から得られた速度分布関数の時間履歴データとし、 f_{GP} を GP から得られた推定式であるとする、評価関数は以下のように表現される。

$$\text{minimize} \left\| \frac{\partial f_{DSMC}}{\partial t} - f_{GP}(x_i, u_j, c_k) \right\|_2 \quad (10)$$

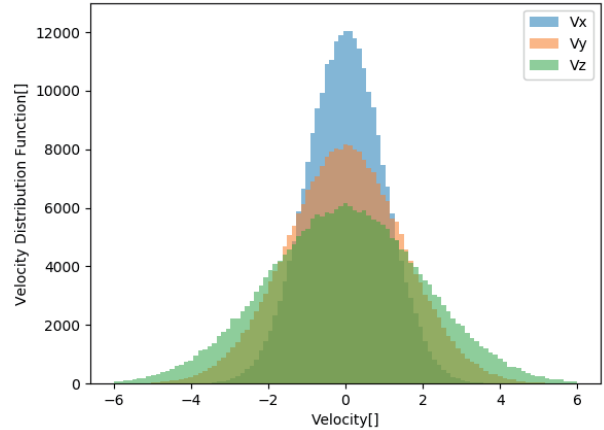


図 4 異方正規分布

GP に関して、 f_{GP} を表現するためにどのような変数を使用できるのか定める必要がある。表 2 に本稿で Boltzmann 方程式衝突項の探索に用いた変数の一覧を示す。 v_x, v_y, v_z は速度分布関数の軸となる分子速度、 f は速度分布関数の時間履歴データであり、 f_0 は収束状態の速度分布関数 f を示す。これらの変数と演算子の組み合わせにより式を構築し、 $\partial f_{DSMC} / \partial t$ を最もよく再現できるように係数を含めて最適化を行う。GP の各パラメータは表 3 を参照されたい。

4.2 適用結果

以上で述べた条件のもとで GP による方程式探索を実施し、Boltzmann 方程式衝突項の近似モデルを探索した。なお、推定の際に用いる $\partial f / \partial v_x$ のような偏微分項は全て 2 次の中心差分により評価を行った。

二重正規分布型の初期分布の場合

式 (11) のような方程式を得ることができた。

$$\frac{\partial f}{\partial t} = a_0 f_0 + a_1 f + a_2 f_0 f \quad (11)$$

$$\begin{aligned} a_0 &= 3.22 \times 10^0 & a_1 &= -3.41 \times 10^0 \\ a_2 &= -1.30 \times 10^{-5} \end{aligned}$$

異方正規分布型の初期分布の場合

式 (12) のような方程式を得ることができた。

$$\frac{\partial f}{\partial t} = a_0 f_0 + a_1 f + a_2 \frac{\partial f}{\partial v_x} + a_3 \frac{\partial^2 f}{\partial v_x^2} + a_4 \frac{\partial^2 f}{\partial v_y^2} + a_5 \frac{\partial^2 f}{\partial v_z^2} \quad (12)$$

$$\begin{aligned} a_0 &= 4.52 \times 10^0 & a_1 &= -4.57 \times 10^0 \\ a_2 &= -7.38 \times 10^{-3} & a_3 &= -7.91 \times 10^{-2} \\ a_4 &= -9.47 \times 10^{-2} & a_5 &= -1.10 \times 10^{-2} \end{aligned}$$

本稿では特に推定対象のデータが $Kn = 0.4$ と限定的にしているため、各項の係数は数値として得られて

表 2 GP で使用した変数一覧

Group	Name	Symbol
Independent variable	Velocity	v_x, v_y, v_z
Dependent variable	Distribution function	f, f_0
	1st derivative	$\frac{\partial f}{\partial v_x}, \frac{\partial f}{\partial v_y}, \frac{\partial f}{\partial v_z}$
	2nd derivative	$\frac{\partial^2 f}{\partial v_x^2}, \frac{\partial^2 f}{\partial v_y^2}, \frac{\partial^2 f}{\partial v_z^2}, \frac{\partial^2 f}{\partial v_x \partial v_y}, \frac{\partial^2 f}{\partial v_y \partial v_z}, \frac{\partial^2 f}{\partial v_z \partial v_x}$

表 3 GP のパラメータ設定

Parameters	Settings
Number of generations	5
Total populations size	100
Maximum tree depth	5
Tournament size	10

いる。得られた推定式 (11)(12) の注目すべき点は、右辺によく知られている緩和方程式と同様の $(f_0 - f)/\tau$ という式の一部が現れている点である。緩和方程式はしばしば Boltzmann 方程式の模型方程式として用いられるが、事前にその情報を与えることなく、本手法がその方程式を導出しているのは興味深い結果である。さらに、GP では世代を重ねるごとに複雑な式を表現することが可能になる一方で、得られた推定式 (11)(12) は簡単な線型結合のみで表現された式になっており、他の支配方程式と同等な簡潔さを保持していることも興味深い。

4.3 推定方程式の検証

今回推定した方程式がどの程度の精度の近似を果たしているか確認するために、逆に推定した方程式から DSMC 法で得られたデータを再現できるか確認する。そのために、推定式 (11) と推定式 (12) のそれぞれに対して、以下に示す差分法により解析を行った。

1. 初期条件設定
2. f_0 の計算
3. $\partial f / \partial v_x$ 等の偏微分項の計算
4. 推定式に基づいて f を更新
5. 2. へ戻る

推定式 (11)(12) に含まれる偏微分項は全て 2 次中心差分で評価を行った。また、収束状態 f_0 は以下のように推定した。

質量保存則から f, f_0 を全速度空間で積分すると同じ値になる。これを a_0 として算出する。

$$a_0 \equiv \int f dv = \int f_0 dv$$

運動量保存から各分子速度方向の平均は f, f_0 で同じ値になる。これを μ_i として算出する。

$$\mu_i \equiv \frac{1}{a_0} \int f v_i dv_i = \frac{1}{a_0} \int f_0 v_i dv_i$$

各方向の分子速度の分散を算出し、 σ_i^2 として算出する。

$$\sigma_i^2 \equiv \frac{1}{a_0} \int f (v_i - \mu_i)^2 dv_i \equiv \sigma_i^2$$

エネルギー等分配則から収束状態ではエネルギーは x, y, z 方向全てに等しく分配される。 f_0 におけるある方向への分子速度の分散 σ_0^2 は以下のように計算される。

$$\sigma_0^2 \equiv \frac{1}{3} \sigma^2 = \frac{1}{3} (\sigma_x^2 + \sigma_y^2 + \sigma_z^2)$$

以上の値を用いて f_0 は次の関数により与えられる。

$$f_0(\mathbf{v}) = \mathcal{N}(\mathbf{v} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (13)$$

$$= \frac{1}{(\sqrt{2\pi})^3 \sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{v} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{v} - \boldsymbol{\mu})\right)$$

なお、式中の変数は以下の式で与えられる。

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_0 & 0 & 0 \\ 0 & \sigma_0 & 0 \\ 0 & 0 & \sigma_0 \end{pmatrix} \quad (14)$$

f_0 は収束状態なのでマクスウェル分布であり、関数は多変量ガウス分布の式 (13) で与えられる。一方で f から f_0 に至る過程では質量保存則や運動量保存則、エネルギー保存則といった基本則は満たされるはずであり、それらから式 (13) に必要な $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ が定まる。

以上の手法で推定方程式から GP に入力したデータの再現を試みた。DSMC 法の結果と推定式を用いた差分法による解析結果の比較を図 5 に示す。

4.3.1 二重正規分布型の初期分布の場合

解析結果の時間履歴を図 5(a)(b)(c) に示す。薄青色に塗りつぶされている部分が DSMC 法から得られた結果であり、青色の実線が今回の推定式から差分法による解析で得られた結果である。二重正規分布型の場合には DSMC 法の場合も推定式の場合も x, y, z 方向

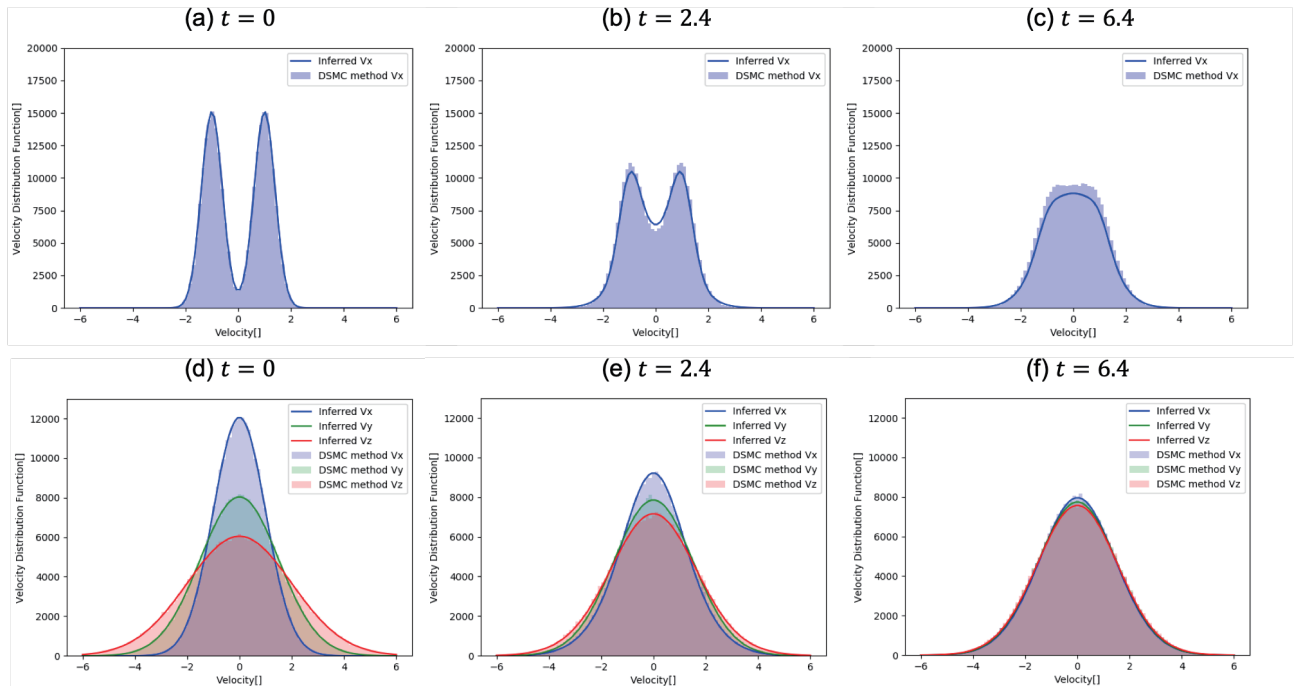


図5 推定方程式の差分法での解析結果

に違いがなかったため、今回は x 方向を代表してグラフにした。

グラフからわかるように、推定式に基づく差分法の結果は、学習データである DSMC 法の結果をよく再現していることがわかる。一方で、図 5(c) を見ると、推定された結果は DSMC 法の結果よりも全体的に値が低くなっていることがわかり、質量保存則が満たされていないことがわかる。

4.3.2 異方正規分布型の初期分布の場合

解析結果の時間履歴を図 5(d)(e)(f) に示す。半透明色に塗りつぶされている部分が DSMC 法から得られた結果であり、実線が得られた推定式を用いた差分法による解析結果である。また、青、緑、赤色はそれぞれ x, y, z 方向に対応する。

本ケースの場合も推定式に基づく解析は学習データである DSMC 法の結果をよく再現している。さらに、図 5(f) を見ると推定された結果と DSMC 法の結果に大きな差異は見られず、質量保存則が満たされていることがわかる。

5 今後の課題

まず、推定方程式の検証結果からわかるように、質量保存則を満たさない方程式が推定されてしまうことが挙げられる。本課題に対する解決案として、次式 (15) のように評価関数を変更することが考えられる。学習データである DSMC 法との差だけでなく、質量保存則からの差異の二乗も評価値に加えることで、DSMC

法を再現しながら質量保存則を満たす方程式が探索できるようにするはずである。

$$\text{minimize} \left\| \frac{\partial f_{DSMC}}{\partial t} - f_{GP} \right\|_2 + \alpha \|f_{GP}\|_2 \quad (15)$$

次に、初期分布の違いにより推定される方程式が異なることが課題として挙げられる。本課題に対しては、GP に 1 ケースずつ学習させるのではなく、多くのケースを同時に学習させることで解決できると考えている。1 ケースごとに探索しては、最適となる近似方程式にケースごとの偏りが生じてしまうと考えられる。

また、得られた推定式の形が x, y, z 方向に対して対称な形になっていないことも課題である。そこで解決策としては、GP に与える変数を全て x, y, z 方向に対して対称なものにすることが考えられる。

そして、係数をクヌーセン数等のパラメータにより評価することが今後必要になると見られる。これは、様々なクヌーセン数の場合に対して推定式の各項の係数を集めることで、定式化が可能になると考えている。

6 まとめ

本稿では、DSMC 法のデータから支配方程式である Boltzmann 方程式の右辺衝突項の近似モデル化を目指し、方程式探索アルゴリズムの構築を行った。また、独自の改良を施すことにより探索効率の向上を行った。同アルゴリズムは遺伝的プログラミングによる式構造の探索、非線形最小二乗法による係数の最適化か

らなる。式構造の探索では、個体の木構造が過度に複雑化する Bloat と呼ばれる問題に対して木構造の簡略化を施すことでメモリ負担の軽減を行った。係数の最適化に対しては、Levenberg-Marquardt 法を導入することで複雑な式に対しても最適な係数を発見できた。

同アルゴリズムを二種類の初期速度分布に対して適用し、Boltzmann 方程式衝突項の模型方程式を推定した。得られた二種類の推定式の評価を行うため、それぞれの式に対して差分法を適用して学習データである DSMC 法の結果を再現できるか確認した。どちらも DSMC 法の結果とよく一致していたが、一方で質量保存則が満たされないケースが存在した。

課題として、初期速度分布に依存して推定式が変化してしまうので、多くのケースを同時に学習させる必要がある。また、学習データへのフィッティングだけでは保存則が満たされなくなることが示唆されたので、今後は保存則を満たす方程式を探索するよう、評価関数へ保存則に関するペナルティを組み込む手法を開発する予定である。

参考文献

- [1] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Science Publications, 1994.
- [2] K. Nanbu, *Theory on the Monte Carlo Simulation of Rarefied Flows Based on the BGK, Holway, Kac, and Boltzmann Equations*, Mem. Inst. Fluid Sci. Tohoku Univ. Vol. 5, pp. 83-120, 1994.
- [3] S. Taguchi and K. Aoki, *Rarefied gas flow around a sharp edge induced by a temperature field*, J. Fluid Mech. Vol. 694, pp. 191-224, 2012.
- [4] R. Yano, K. Suzuki and H. Kuroda, *Analytical and numerical study on the nonequilibrium relaxation by the simplified Fokker-Planck equation*, Physics of Fluids, Vol. 21, No. 1, pp. 1-11, 2009.
- [5] L. Mieussens and H. Struchtrup, *Numerical comparison of Bhatnagar-Gross-Krook models with proper Prandtl number*, Physics of Fluids, Vol. 16, No. 8, pp. 2797-2813, 2004.
- [6] M. Schmidt and H. Lipson, *Distilling free-form natural laws from experimental data*. Science, Vol. 324, No. 5923, pp. 81-85, 2009.
- [7] J. Weatheritt and R. Sandberg, *A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship*. Journal of Computational Physics, Vol. 325, pp. 22-37, 2016.
- [8] S. H. Rudy, S. L. Brunton, J. L. Proctor and J. N. Kutz, *Data-driven discovery of partial differential equations*. Science Advances, 2017.
- [9] S. L. Brunton, J. L. Proctor and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*. Proc. Natl. Acad. Sci. U.S.A. Vol. 113, pp. 3932-3937, 2016.
- [10] M. Raissi and G. E. Karniadakis, *Hidden physics models: Machine learning of nonlinear partial differential equations*. Journal of Computational Physics, Vol. 357, pp. 125-141, 2018.
- [11] J. Koza. *On the Programming of Computers by means of Natural Selection*. 1998.
- [12] H. Cao, L. Kang, and Y. Chen. *Evolutionary modeling of systems of ordinary differential equations with genetic programming*. Genetic Programming and Evolvable Machines, Vol. 1, pp. 309-337, 2000.
- [13] 金森正史, 日高亜希子, 永井伸治. *方程式探索アルゴリズムを用いた再突入カプセルのピッチ運動に関するモデル方程式の導出*. 第 50 回流体力学講演会/第 36 回航空宇宙数値シミュレーション技術シンポジウム, 2018.
- [14] J. C. Nash. *Minimizing a non-linear sum of squares function on a small computer*. IMA Journal of Applied Mathematics, Vol. 19, No. 2, pp. 231-237, 1977.
- [15] 杉本直哉, 坂本栄里奈, 伊庭斉志. *遺伝的プログラミングによる微分方程式系の推定*. 人工知能学会論文誌, Vol. 19, No. 6, pp. 450-459, 2004.