

卒業論文

題目

流体解析におけるベルレリストの  
機械学習を用いたパラメータ最適  
値推定

指導教員

大野和彦 講師

2018年

三重大学 工学部 情報工学科  
コンピュータアーキテクチャ研究室

馬瀬 義陽 (413852)

## 内容梗概

SPH 粒子法とは宇宙物理学の問題を解くために提案された手法であるが、計算モデルの単純さから幅広く応用が利くため、流体解析にも使用される。SPH 粒子法では粒子同士が相互作用を及ぼす範囲から近傍粒子を探索しており、ベルレリストを用いるとその探索を高速化できる。ベルレリストは SPH 粒子法におけるデータ構造であり、各粒子に自身の近傍粒子を記録する近傍リストを持たせる手法である。しかしそのリストサイズを決める変数パラメータは動的に求めることが難しく、ユーザーがあらかじめ設定しなければならない問題がある。そこで、本研究では流体解析プログラムである DualSPHysics の GPU 版におけるベルレリストの変数パラメータの最適値を推定モデル式によって導出した。初めに、ベルレリストのパラメータを変化させながらシミュレーションに要する実時間を計測する。次に、十分な実行時間のデータを元にシミュレーション毎に最適なパラメータを求める。そして、その結果を目的変数とし機械学習による回帰分析を行い、最適なパラメータの推定モデル式を導出した。

# Abstract

SPH method is a method proposed to solve the problem of astrophysics, but because it is widely applicable from the simplicity of the calculation model, it is also used for fluid analysis. In the SPH method, neighboring particles are searched from the range where particles interact with each other, and if Verletlist is used, the search can be speeded up. Verletlist is a data structure in the SPH method, and it is a method to make each particle have a neighborhood list for recording its neighboring particles. However, it is difficult to dynamically obtain variable parameters that determine the list size, and there are problems that the user has to set in advance. Therefore, in this study, the optimum value of the Verletlist's variable parameter in the GPU version of DualSPHysics which is the fluid analysis program is derived by the estimation model equation. First, measure the real time required for simulation while changing the parameters of Verletlist. Next, optimum parameters are obtained for each simulation based on data with sufficient execution time. Then, regression analysis by machine learning is performed with the result as the objective variable, and an estimation model equation of the optimum parameter is derived.

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>背景</b>	<b>2</b>
2.1	GPU . . . . .	2
2.2	SPH 粒子法 . . . . .	2
2.2.1	セルリンクリスト . . . . .	3
2.2.2	ベルレリスト . . . . .	4
2.2.3	ベルレリストの問題点 . . . . .	6
2.3	機械学習における回帰 . . . . .	6
<b>3</b>	<b>提案手法</b>	<b>8</b>
3.1	手法の概要 . . . . .	8
3.2	実装 . . . . .	8
<b>4</b>	<b>性能評価</b>	<b>11</b>
4.1	評価方法 . . . . .	11
4.2	評価結果 . . . . .	13
<b>5</b>	<b>おわりに</b>	<b>18</b>
	謝辞	18
	参考文献	19

## 目 次

2.1	SPH 粒子法 . . . . .	3
2.2	セルリンクリスト法 . . . . .	4
2.3	ベルレリスト法 . . . . .	5
3.4	実験の流れ . . . . .	9
4.5	セルリンクに対する実行速度の伸び率 . . . . .	14
4.6	学習させた $c$ の実測値とモデル式から計算した $c$ の値 . . .	15
4.7	学習させていない $c$ の実測値とその予測値 . . . . .	16

## 表 目 次

4.1	本実験に用いたテストケース . . . . .	12
4.2	実行時間 [s] . . . . .	13
4.3	説明変数とその係数 . . . . .	17

# 1 はじめに

SPH 粒子法とは宇宙物理学の問題を解くために提案された手法であるが、計算モデルの単純さから幅広く応用が利くため、流体解析にも使用される。SPH 粒子法では粒子同士が相互作用を及ぼす範囲から近傍粒子を探索しており、ベルレリストを用いるとその探索を高速化できる。ベルレリストは SPH 粒子法におけるデータ構造であり、各粒子に自身の近傍粒子を記録する近傍リストを持たせる手法である。しかしそのリストサイズを決める変数パラメータは動的に求めることが難しく、ユーザーがあらかじめ設定しなければならない問題がある。そこで、本研究では流体解析プログラムである DualSPHysics[1] の GPU 版におけるベルレリストの変数パラメータの最適値を推定モデル式によって導出した。初めに、ベルレリストのパラメータを変化させながらシミュレーションに要する実時間を計測する。次に、十分な実行時間のデータを元にシミュレーション毎に最適なパラメータを求める。そして、その結果を目的変数とし機械学習による回帰分析を行い、最適なパラメータの推定モデル式を導出した。

## 2 背景

### 2.1 GPU

GPUは元々画像処理に特化した演算装置であるが、演算を行うコアを多数搭載しており、多数の処理を同時に実行することができる。このため、単純な演算の繰り返し等をCPUの代わりにGPUに行わせることで処理速度が飛躍的に向上する。このことから、計算が膨大となるSPH粒子法においてGPUを用いた高速化の研究が行われてきた。

### 2.2 SPH粒子法

粒子法は計算対象を粒子の集まりとして、連続体に関する方程式を数値的に解析する手法である。SPH粒子法では1個の粒子に対して位置、速度、密度、圧力などの属性を持つ構造体の配列に各粒子の属性値を格納する。この配列の要素番号と、各粒子のIDを対応させることで、粒子の属性値を参照できるようにする。図2.1のように、粒子同士が相互に作用し合う影響範囲があり、近傍粒子を探索してこれらのデータを用いて計算を行う。そのため、タイムステップ毎に各粒子に対して相互作用する近傍粒子を探索する必要がある。



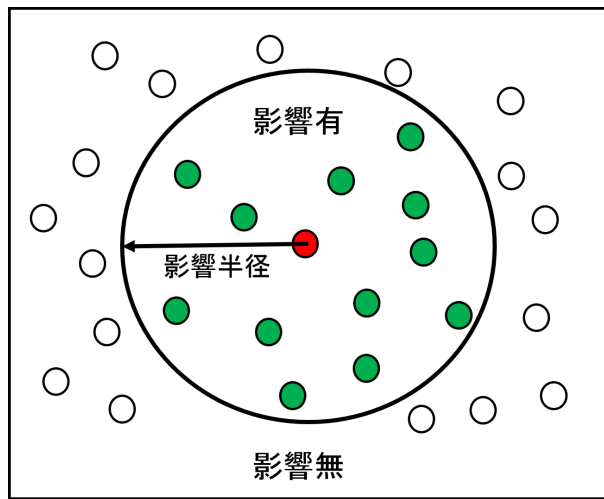


図 2.1: SPH 粒子法

### 2.2.1 セルリンクリスト

近傍粒子探索の高速化手法としてセルリンクリスト法がある。セルリンクリスト法ではシミュレーション空間を同じサイズのセルに分割しておき、各粒子がどのセル内に所属(存在)するかあらかじめ登録する。図 2.2 はセルリンクリスト法を用いた 2 次元での近傍粒子探索の例である。セルのサイズをカットオフ半径  $r_C$  にすると、計算対象の粒子  $p_i$  に対する近傍粒子の候補は、 $p_i$  が所属するセルおよびそれに隣接する 8 セル内の粒子に限定される。これらの候補の全粒子について  $p_i$  との距離計算を行い、距離が  $r_C$  以下かどうかを判定する。このように探索範囲を限定することで、計算コストを削減できる。

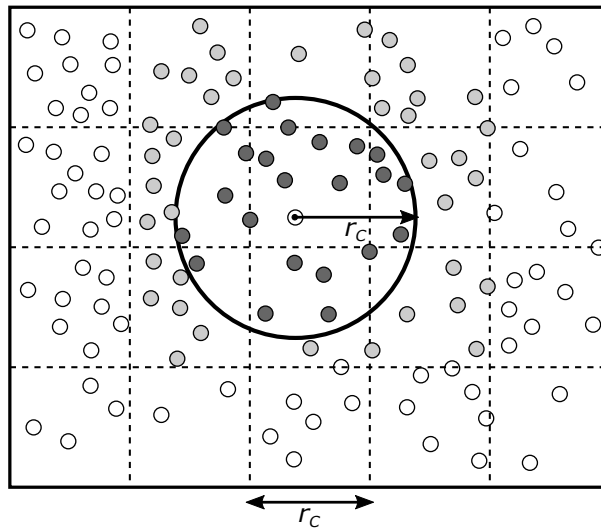


図 2.2: セルリンクリスト法

### 2.2.2 ベルレリスト

ベルレリスト法では、各粒子に近傍粒子を記録する近傍リストを持たせる。あらかじめこの近傍リストを構築しておき、相互作用計算での近傍粒子探索時に近傍リストを参照することで、探索範囲を限定する。セルリンクリスト法に比べて探索時の近傍粒子の候補が少ないので、データアクセスおよび分岐を削減できる [2].

図 2.3 は 2 次元でのベルレリスト法を用いた近傍リスト構築の例である。リスト構築時にセルリンクリスト法を用いることで、高速化することができる。また、近傍リストには粒子間の距離がカットオフ半径  $r_C$  以下の粒子ではなく、 $R_C (> r_C)$  以下の粒子を登録する。これにより、リスト

の再構築を数ステップに1度に削減できる。SPH法でベルレリスト法を用いる場合、近傍リストに登録する半径  $R_C$  は以下の式で求められる [2]。

$$R_C = r_C + \Delta h \quad (1)$$

$$\Delta h = 2 \cdot V_{max} \cdot C \cdot \Delta t \quad (2)$$

ここで  $r_C$  はカットオフ半径,  $V_{max}$  は全粒子中での最大速度,  $C$  はリストを維持するステップ数である。

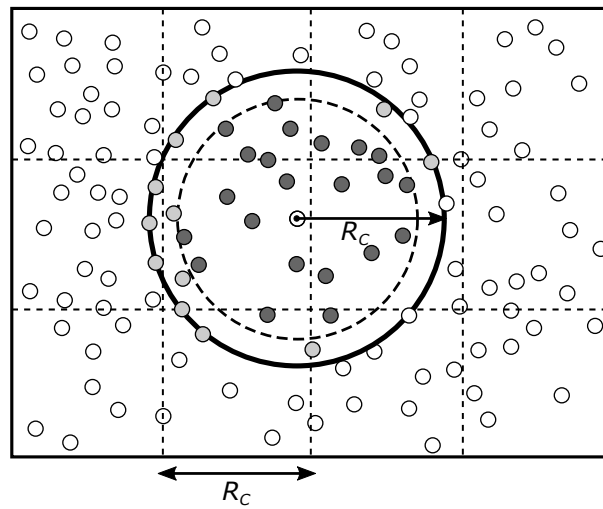


図 2.3: ベルレリスト法

### 2.2.3 ベルレリストの問題点

ベルレリスト法では近傍リストの構築コストがかかるため、更新頻度は低いことが望まれる。しかしリストの維持ステップ数である  $C$  の値を大きくするとリストに登録する粒子数が増えるため、粒子の相互作用計算時間が大きくなる。このようにリストの構築コストと粒子の計算コストはトレードオフの関係があり、 $C$  の最適値は粒子の運動状態に依存する。この  $C$  の値は予め予測することが難しく、従来では  $C = 10$  等で固定され用いられることが多い。また大規模シミュレーションにおいて実装コストや計算コストの面から、計算によって動的に求めることも難しいと考えられる。

## 2.3 機械学習における回帰

機械学習とは計算によって解を求めることが難しい問題を、ある程度の入力データを解析してパターンを自動的に抽出することで解く手法である。機械学習は教師あり学習と教師なし学習に分かれる。教師あり学習では事前に入力に対する出力が与えられ、そのデータを元に分類問題や回帰問題を解く。回帰とは目的変数  $Y$  と説明変数  $X$  の間に  $Y = f(X_1, X_2, \dots, X_n)$  というモデル式を当てはめることである。説明変数  $X$  の係数と切片によっ

て目的変数  $Y$  の値を予測する。回帰で使われるモデルとして線形回帰がある。線形回帰では最小二乗法によって係数と切片を決定する。近年では、回帰や分類における計算は、機械学習ライブラリによって簡易に実装することが可能になっている。

## 3 提案手法

### 3.1 手法の概要

本研究では、予め実行によって求めた  $C$  の最適値と、解析中の粒子状態の情報を用いて機械学習による回帰を行い、 $C$  の予測式を設計する。まず、 $C$  の値を変化させてシミュレーション時間を実測する。そして、各テストケースにおける実行時間のデータを元に最適な  $C$  の値を求め、これを目的変数とする。粒子の動きが密な場所では近傍リストの更新頻度が高くなり、疎な場所では逆にリストを長時間維持できると考えられる。このことからリスト維持ステップ数  $C$  は、粒子の速度や密度、圧力などと関係があると予測できるため、それらをパラメータ化する。パラメータ化された粒子情報を説明変数とし、Python の機械学習ライブラリである scikit-learn[3] の線形回帰の機能を用いて、 $C$  のモデル式のパラメータを導出する。

### 3.2 実装

流体解析シミュレーションの実行時間計測には、流体解析プログラムである DualSPHysics に実装されているテストケースを使用した。DualSPHysics ではダム崩壊や津波シミュレーションなどの問題を SPH 法を用

いた流体解析により検証できる. 実験の流れを図 3.4 に示す.

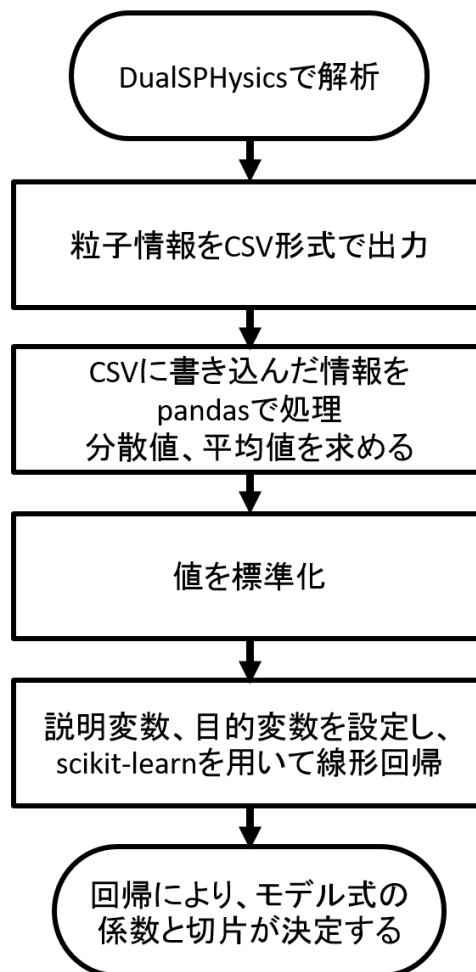


図 3.4: 実験の流れ

まず, DualSPHysics で解析を行い, 各テストケースにおける  $C$  の値の変化によるシミュレーション時間を実測する. 実行時間から求めた  $C$  の

最適値を目的変数とする。DualSPHysics は解析実行後に各実行ステップでの数十万個程度の各粒子の状態を csv 形式で出力できる。そのデータに対して python のデータ解析ライブラリである pandas を用いて、各実行ステップでの各粒子の速度と密度、圧力の、それぞれ平均値、分散値を求める。各実行ステップにおけるこれらの値の平均を取ることで、シミュレーションでの粒子の運動状態の激しさを数値化する。この値の他に、粒子の最大速度、テストケース毎の粒子数、シミュレーション空間の大きさを説明変数とする。これらの値は規格が異なるため、標準化を行う。最後に、設定した目的変数と説明変数より、python の機械学習ライブラリである scikit-learn の線形回帰の機能を用いて、 $C$  のモデル式のパラメータを導出する。



## 4 性能評価

### 4.1 評価方法

精度評価は，DualSPHysics のテストケース 16 個のうち 13 個を学習に用い，残りの 3 個を評価に用いた．テストケースの詳細を表 4.1 に示す．評価環境は Intel Core i7-930，メモリ 5.71GB，CPU GeForce GTX 1070 を搭載した計算機を使用した．

$C$  の値を逐次変更し，各テストケースにおける実行時間を計測した結果，テストケース DambreakVal2D において  $C = 17$  での実行時間が最も高速となった．それ以上の値では他のテストケースにおいても速度が低下したため，これを上限とする．

最後に精度評価として，実際に実行することで求めた  $C$  の最適値と，モデル式から導いた  $C$  の予測値を比較する．

テストケース	空間	粒子数	概要
DamBreak	3D	171496	ダム崩壊シミュレーション
DamBreak2D	2D	21001	ダム崩壊シミュレーション
Periodicity	2D	23808	右の壁を通った粒子が左の壁から流れる
MoveSquare	2D	20302	正方形の剛体が一定速度で水の中を進む
MoveSquare_Noshifting	2D	20302	正方形の剛体が一定速度で水の中を進む 内部の流れにシフトが使用されない
Forces	3D	44063	外力を加え、タンク内の水をかき回す
SloshingMotion	2D	22064	水の入ったタンクを回転させる
SloshingAcc	2D	22064	水の入ったタンクを回転させる 流体粒子を加速させる
WaveMaker2D	2D	53087	一定周期の波を生成する
WaveMaker	3D	518744	一定周期の波を生成する
WavesREG	2D	51234	規則的な波を生成する
WavesIRREG	2D	51234	不規則な波を生成する
FloatingSphereVal2D	2D	56844	箱を浮かべた水に波を生成する
Floating	3D	469508	箱を浮かべた水に波を生成する
FloatingWavesVal	3D	602920	箱を浮かべた水に波を生成する 非直線波が作用する
Solids	3D	115016	空間内に剛体粒子を配置したダム崩壊

表 4.1: 本実験に用いたテストケース

## 4.2 評価結果

テストケース DambreakVal2D と Forces における  $C$  の変化による実行時間を表 4.2 に示す。また、セルリンク版における実行時間と比較した、 $C$  の変化による実行時間の伸び率を図 4.5 に示す。DambreakVal2D では  $C$  の最適値が 17 に対して、Forces では 5 が最適値となっている。これにより、テストケース毎に  $C$  の最適値が変わり、実行時間に大きく影響することが分かる。

	Wavemaker	DambreakVal_2D
c=1	18211.71292	53.685878
c=2	13701.08008	36.194283
c=5	11380.27373	23.022678
c=7	11954.86971	22.190403
c=10	13469.89308	19.806642
c=15	17569.35405	18.881513
c=17	-	18.761505
c=20	-	19.300433

表 4.2: 実行時間 [s]

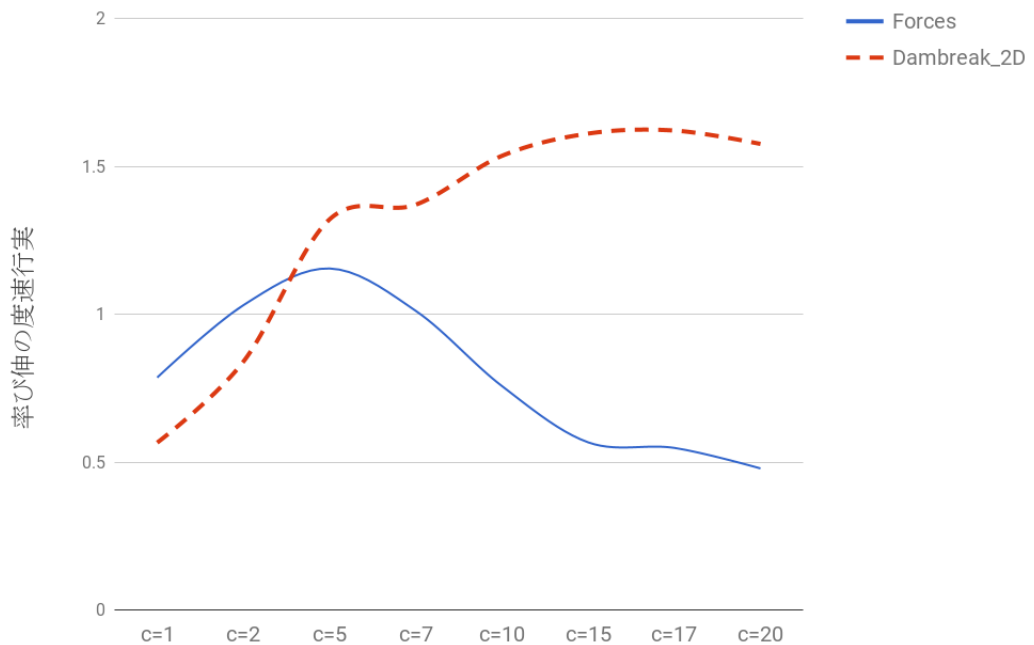


図 4.5: セルリンクに対する実行速度の伸び率

図 4.6 は、学習データに対するモデル式の適合率を測るため、学習に用いた  $c$  の実測値とモデル式から導いた  $c$  の値を比較したものである。次に、学習させていない  $c$  の実測値とその予測値を図 4.7 に示す。平均二乗誤差による精度評価を行った結果、平均二乗誤差は 2.378 となり、ベルレリストの変数パラメータの値を最適値から  $\pm 2$  以下の誤差で予測することができた。図 4.6 でモデル式の当てはまりの度合を確認できていることから、正しく予測されていると考えられる。表 4.2, 図 4.5 が示すように、

この誤差では最適な  $C$  の場合の実行時間から大きく変わらないため、この値を用いることは十分有用であると考えられる。

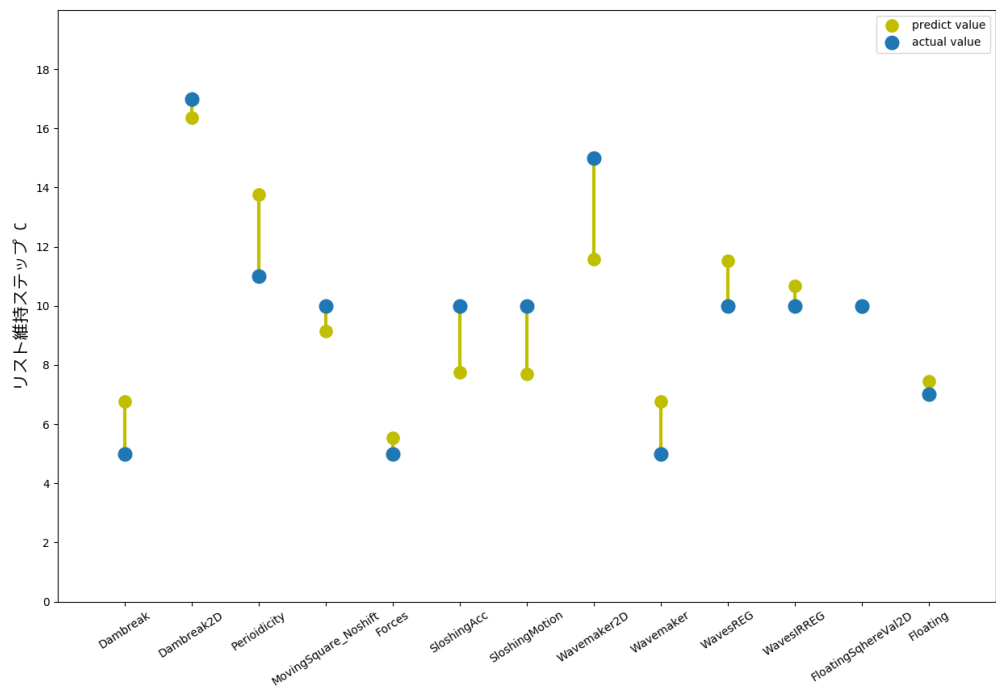


図 4.6: 学習させた  $c$  の実測値とモデル式から計算した  $c$  の値

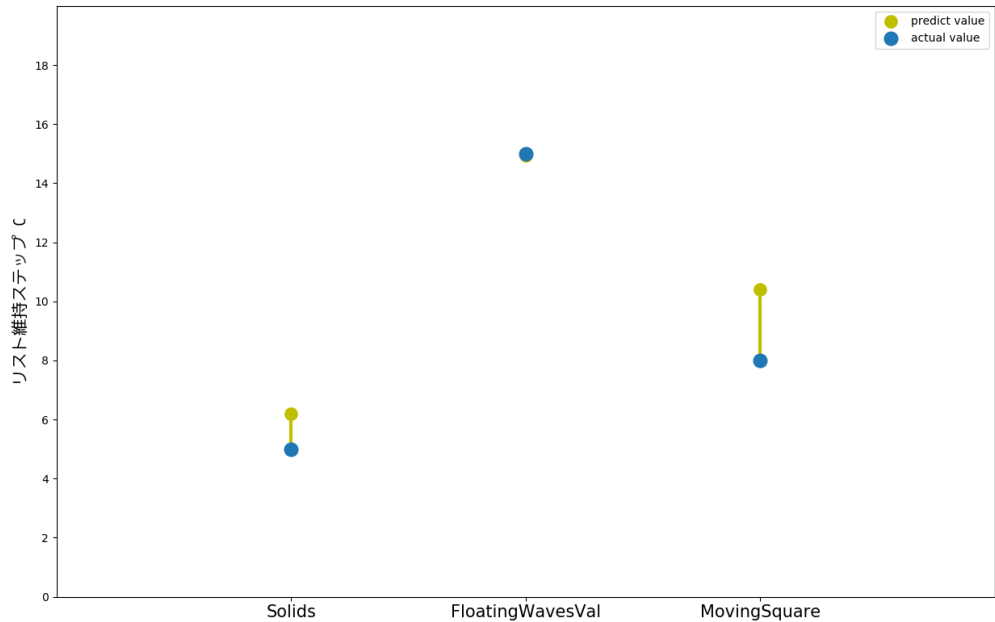


図 4.7: 学習させていない c の実測値とその予測値

設計したモデル式の確からしさを検証する。設定した説明変数と、線形回帰により求めた係数を表 4.3 に示す。これを見ると速度の影響が大きく、速度によって c が一意に決まるように考えられる。しかし検証として、速度以外の要素を消し回帰を行ったところ、平均二乗誤差は 5.074 となり、元の平均二乗誤差 2.378 に比べ精度が下がった。このことから、速度以外の影響度の低い要素も必要であると考えられる。また、予備実験として密度の平均値を説明変数に含めた場合の評価を取ったところ、MovingSquare

の予測において  $C$  の最適値が 8 であるところを, 48.9 と大きく外れて予測してしまった. このとき平均二乗誤差は 558.6 となった. 線形回帰を行う際, サンプル数に対し説明変数は少なければならない. 本実験では学習データが少なく, 学習データ 13 個に対し, 速度以外の影響度の低い説明変数の割合が増えたため, 精度が落ちたと考えられる.

粒子数/空間サイズ	0.665664
密度の分散値	0.180163
圧力の平均値	-6.565695
圧力の分散値	6.972883
速度の平均値	8.495207
速度の最大値	-19.970967
速度の分散値	13.115613
バイアス項	9.61538461538

表 4.3: 説明変数とその係数

## 5 おわりに

本研究では、SPH 粒子法の高速化のため、ベルレリストのリスト維持ステップ数の最適値を予測するモデル式を設計した。この予測式から得られる  $C$  の値を用いての実行時間は、最適値を用いた場合と比較し十分に速いと言える。しかし精度評価に用いたテストケースが少なすぎるため、更に検証が必要である。また、学習に用いるテストケースを変えた結果、予測値が大きくズレる場合があった。学習データを増やす、又はモデルの設定を工夫することで精度の向上が期待できる。

## 謝辞

本研究に際して、様々なご指導を頂いた指導教員の大野和彦講師に感謝致します。また、多くのご指摘を下さいましたコンピュータソフトウェア研究室の皆様に感謝します。



## 参考文献

- [1] DualSPHysics ,<http://dual.sphysics.org/>
- [2] 高田 貴正: SPH 法による流体解析の GPU 上での高速化, 情報処理学会論文誌, (p.3-4), (2017/02/18)
- [3] scikit-learn,<http://scikit-learn.org/stable/>