

Fintech Data Championship

流体を用いた情報処理装置による株価予測

平成 31 年 2 月 22 日 提出

大阪大学大学院基礎工学研究科
機能創成専攻 1 年

小橋 敬太

目次

第 1 章 諸言	1
第 2 章 定式化	2
§ 2.1 運動方程式	2
§ 2.1.1 境界条件	3
§ 2.1.2 埋め込み境界法 (IBM)	3
§ 2.2 流体情報処理装置の定式化	4
§ 2.2.1 RC の定式化	4
§ 2.2.2 流体実装	5
§ 2.3 誤差の指標	5
第 3 章 対象タスクのデータ選定・前処理	6
§ 3.1 データの選定	6
§ 3.2 データの前処理	6
第 4 章 計算結果	7
§ 4.1 プログラムの概要	7
§ 4.2 Task1: ローレンツモデル	9
§ 4.2.1 入力 $x(t)$, 出力 $y(t), z(t)$	10
§ 4.2.2 入力 $x(t)$, 出力 $x(t+10)$	11
§ 4.3 Task2: 株価予測	15
第 5 章 結言	17

第1章 諸言

今回, FDC に参加させていただいた最大の理由は「現在行っている RC の物理実装の研究への刺激になれば」との思いからである. 研究の一部として行っているカオス時系列予測の観点から FDC のタスクを見る. すなわち利益最大化と株価予測を同一視して以降この論文を書き進める.

近年の機械学習技術における発展の影響は多岐にわたる. ビッグデータの解析や音声認識, 物体認識, 市場予測などの分野では既に機械学習なしには語れない. 時系列データを分析する際, 古くから用いられてきた方法として RNN (リカレントニューラルネットワーク) が挙げられる [3]. この方法は学習の計算コストが高く計算が煩雑な点が欠点とも言える. そこで近年 RC (リザーバーコンピューティング) という方法が注目され始め研究が活発に行われている [4]. RC は RNN に比べ計算コストが比較的低いとともに数理的原理はとても簡潔である. またこの簡潔さから RC 自身の自由度も上げることも容易であり時系列データの分析にとっても適している. しかし, 「RC がなぜ予測することができるのか」という根本的な問題はまだ未解決のままである. 現状はまだ「理由は判明していないがうまく予測できるため RC を使用する」という時点に過ぎない [2].

現在, RC の物理実装についても盛んに研究されている [5]. この理由として高速且つ省エネルギーな機械学習を実現できる可能性が示唆されているからである. 現在すでに研究なされている実装法としてレーザのダイナミクスを利用した RC やタコ足を利用した RC など, 実装方法は様々である. しかし時間遅延ダイナミクスを利用した実装法には高速性や情報処理性能に限界があることが容易に予想される. この限界を突破すべく革命的な実装法「連続時空間ダイナミクスを用いた RC を流体のダイナミクスを用い実装, そして株価を予測する.」ことを目的する. 今回は数値計算シミュレーションを用い実際に流体の RC を構築した. 以降, この RC を「流体情報処理装置」と呼ぶ.

本論文の構成を以下の通りである. 第2章では本コンペティションで使用する RC を定式化. 第3章では, 対象タスクのデータ選定・前処理. 第4章では, 前章までに得られた式を実際の現象に用い数値解析結果を解析する. そして第5章では結言としてこれまでの結果をまとめ, これからの研究における課題を記す.

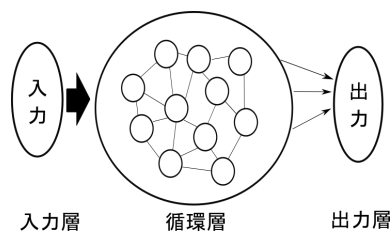


図 1.1: RC のイメージ図

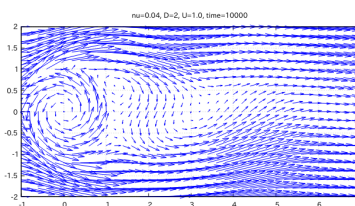


図 1.2: 流体情報処理装置のイメージ図

第2章 定式化

今回はデータの選定及び前処理に Python を用い、それ以外の流体のシミュレーションと株価予測に Fortran を使用した。前者の理由として Python はデータの加工を得意としており、多数のファイルの種類に対応していることが挙げられる。後者の理由は流体のシミュレーションには数値計算に向いている Fortran がより適するということが挙げられる。また原理のわかっていないライブラリを一切使用しないことを心がけた。

§ 2.1 運動方程式

2次元の円柱後流についての数値計算シミュレーションを定式化する。ここで図 1.2 のような二次元流を考える。二次元のナビエ・ストークスの方程式は以下のようになる。

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \frac{1}{Re} \nabla^2 \mathbf{u} \quad (2.1)$$

ここで連続の式も満たしていることが要請される。非圧縮条件下における連続の式は

$$\nabla \cdot \mathbf{u} = 0 \quad (2.2)$$

と表される。また、式 (2.1) の両辺の $\nabla \cdot$ をとると、

$$\Delta P = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) \quad (2.3)$$

これらの式 (2.1), (2.2), (2.3) を適切な境界条件を考慮し解く。

式 (2.1)

$$\begin{cases} \frac{\partial v_x}{\partial t} = -v_x \frac{\partial v_x}{\partial x} - v_y \frac{\partial v_x}{\partial y} - \frac{\partial P}{\partial x} + \nu \left(\frac{\partial^2 v_x}{\partial t^2} + \frac{\partial^2 v_x}{\partial y^2} \right) \\ \frac{\partial v_y}{\partial t} = -v_x \frac{\partial v_y}{\partial x} - v_y \frac{\partial v_y}{\partial y} - \frac{\partial P}{\partial y} + \nu \left(\frac{\partial^2 v_y}{\partial t^2} + \frac{\partial^2 v_y}{\partial y^2} \right) \end{cases}$$

ここで $\partial v_x, \partial v_y$ の時間微分について前進差分より

$$\begin{cases} \frac{\partial v_x(t+\delta t) - \partial v_x(t)}{\delta t} \\ \frac{\partial v_y(t+\delta t) - \partial v_y(t)}{\delta t} \end{cases}$$

と表せる。よって次ステップの速度 $\partial v_x, \partial v_y$ は右辺を R_x, R_y とすると

$$\begin{cases} v_x(t + \delta t) = v_x(t) + R_x \delta t \\ v_y(t + \delta t) = v_y(t) + R_y \delta t \end{cases}$$

となる。ここで R_x, R_y は単位量あたりの流体に加わる x, y 方向の力である。

式 (2.3)

次ステップの圧力のポアソン方程式について解く. この式には次ステップの速度が必要となる. よって速度を進めた後このプログラムを持ってくる. 式 (2.3) より,

$$\Delta P = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u})$$

と表せるが連続の式をみたすようにプログラム中では手を加える. 連続の式 (2.2) を満たすため右辺の

$$-\frac{\partial}{\partial t} \nabla \cdot \mathbf{u}$$

が消えているがここでこの項を単純に消すのではなく以下のように残しておく.

$$\Delta P = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) - \frac{\partial}{\partial t} \nabla \cdot \mathbf{u} \quad (2.4)$$

ここで時間微分を前進差分で考え

$$\begin{aligned} -\frac{\partial}{\partial t} \nabla \cdot \mathbf{u} &= -\frac{\nabla \cdot \mathbf{u}(t + \delta t) - \nabla \cdot \mathbf{u}(t)}{\delta t} \\ &= -\frac{\nabla \cdot \mathbf{u}(t + \delta t)}{\delta t} + \frac{\nabla \cdot \mathbf{u}(t)}{\delta t} \end{aligned} \quad (2.5)$$

次ステップで連続の式 (2.2) を満たせば, 時間を進めても常に連続の式 (2.2) を満たすと考えられるので式 (2.5) の第一項を 0 とする.

$$-\frac{\partial}{\partial t} \nabla \cdot \mathbf{u} \approx \frac{\nabla \cdot \mathbf{u}}{\delta t}$$

よって式 (2.4) は

$$\Delta P = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) + \frac{\nabla \cdot \mathbf{u}}{\delta t} \quad (2.6)$$

§ 2.1.1 境界条件

流入速度, 流出速度, そして上下端の速度を主流の速度とする. ここで主流の速度は時間依存しない定数 U とする. よって図 2.1 を見て, 4 辺の境界の速度は x 方向のみに速度 U して解くことになる.

§ 2.1.2 埋め込み境界法 (IBM)

運動方程式と連続の式より速度, そしてポアソン方程式より圧力が求まる. ここで流体の場に存在する円柱を埋め込み境界法 (IBM) を用いてとく. ここで円柱の表面では速度が 0 と考える. 言い換えると円柱表面では速度が 0 になるように流体に力が加わっていると考えることもできる. よって式において円柱表面先程求めた R_x, R_y を

$$\begin{cases} R_x = R_x + \frac{V_x(t+\delta t) - V_x(t)}{\delta t} = R_x + \frac{0 - V_x(t)}{\delta t} \\ R_y = R_y + \frac{V_y(t+\delta t) - V_y(t)}{\delta t} = R_y + \frac{0 - V_y(t)}{\delta t} \end{cases}$$

とすれば良い.

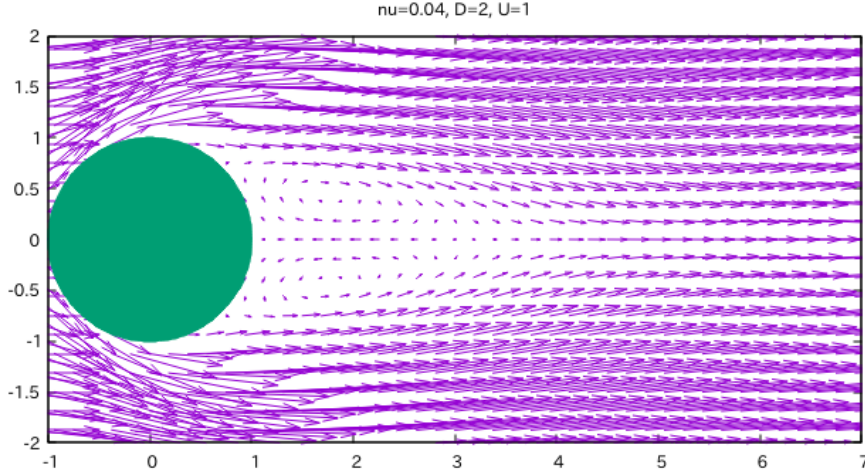


図 2.1: IBM を用い解いた図

§ 2.2 流体情報処理装置の定式化

§ 2.2.1 RC の定式化

本論文で使用する RC のイメージは図 1.1, 1.2 のように描ける. RC は入力層, 循環層, 出力層から構成される. RC の特徴として循環層の信号から出力信号を生成する場合の重みのみを学習する. この際, 線形回帰である最小二乗法を使用する. 以下, 入力の次元数を $m1$, 出力の次元数を $m2$, RC の次元数を n , 教師データの長さ k とする. 入力 $\mathbf{x}(t)$, 出力 $\mathbf{s}(t)$, リザーバーの信号 $\mathbf{r}(t)$ とすると,

$$\mathbf{s}(t) = \mathbf{W}^{out} \mathbf{r}(t) \quad (2.7)$$

$$\mathbf{r}(t) = \phi[\mathbf{r}(t-1), \mathbf{x}(t)] \quad (2.8)$$

注意するのは重みとして 3 つの行列 $\mathbf{A}(n \times n \text{ 行列})$, $\mathbf{W}^{in}(n \times m1 \text{ 行列})$, $\mathbf{W}^{out}(n \times m2 \text{ 行列})$ があるが前者の 2 つは固定し, 後者の一つのみ教師データから学習する.

\mathbf{W}^{out} の学習の方法については以下の通りである. 以降, $[\cdot](i\Delta t) = [\cdot]_i$ まず教師データの組 $(\mathbf{x}_i, \mathbf{s}_i)_{i=1}^k$ を考える. 式 (2.13) に従って対応する \mathbf{r}_i を生成すると行列 $\mathbf{R}(k \times n \text{ 行列})$ ができる.

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_k \end{bmatrix} \quad (2.9)$$

RC の出力を $\tilde{\mathbf{S}} = [\tilde{s}(1), \dots, \tilde{s}(n)]^T$ とすると式 (2.7) より

$$\tilde{\mathbf{S}}_{i,j} = \mathbf{R}_{i,k} \mathbf{W}^{out}_{k,j} \quad (2.10)$$

ここで RC による出力 $\tilde{\mathbf{S}}$ と教師データ \mathbf{S} の二乗を E とすると

$$E = \sum_{i=1}^n \sum_{j=1}^m \left(\tilde{\mathbf{S}}_{i,j} - \mathbf{S}_{i,j} \right)^2 \quad (2.11)$$

最小二乗法より $\frac{\partial E}{\partial \mathbf{W}_{ij}^{out}} = 0$ を満たす \mathbf{W}^{out} 求めると

$$\mathbf{W}^{out} = \left(\mathbf{R}^T \mathbf{R} \right)^{-1} \mathbf{R}^T \mathbf{S} \quad (2.12)$$

が導かれる.

§ 2.2.2 流体実装

循環層を流体に置き換えた場合を考える. 流体の任意の領域の速度 \mathbf{u} を循環層の値として使用する. この循環層として使用する速度は先程の循環層内部の信号 \mathbf{R}_{ij} に対応する. つまり ϕ が流体自身となる. そして入力信号 $\mathbf{x}(t)$ は図 2.1 の円柱を回すことによって, 信号を入力するとした. 以上を踏まえて流体情報処理装置は以下の式で表せる.

$$\begin{aligned} \mathbf{s}(t) &= \mathbf{W}^{out} \mathbf{r}(t) \\ \mathbf{r}(t) &= \phi[\mathbf{r}(t-1), \mathbf{x}(t)] \\ &= \mathbf{u}(x, y, t) \end{aligned} \quad (2.13)$$

\mathbf{W}^{out} の学習の方法については 2.12 と同様に行う.

§ 2.3 誤差の指標

誤差の定量的評価として RMSE を以下のように定義する.

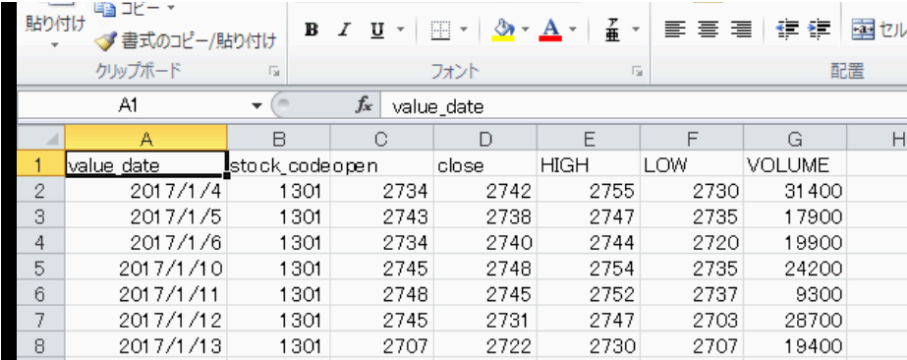
$$\text{err} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left([\tilde{\cdot}]_i - [\cdot]_i \right)^2} \quad (2.14)$$

ここで $[\tilde{\cdot}]_i$ は RC による出力値, $[\cdot]_i$ は真値を表す. 最小二乗法により最適化しているので二乗誤差を使用するのは自然である.

第3章 対象タスクのデータ選定・前処理

株価予測について使用データはMILIZE様より与えられた4つのCSVファイル (stock_2016.csv, stock_2017.csv, stock_2018.csv, stock_2019_2_7.csv) のみを使用した。この章ではその選定と前処理の詳細を記す。

§ 3.1 データの選定



	A	B	C	D	E	F	G	H
1	value_date	stock_code	open	close	HIGH	LOW	VOLUME	
2	2017/1/4	1301	2734	2742	2755	2730	31400	
3	2017/1/5	1301	2743	2738	2747	2735	17900	
4	2017/1/6	1301	2734	2740	2744	2720	19900	
5	2017/1/10	1301	2745	2748	2754	2735	24200	
6	2017/1/11	1301	2748	2745	2752	2737	9300	
7	2017/1/12	1301	2745	2731	2747	2703	28700	
8	2017/1/13	1301	2707	2722	2730	2707	19400	

図 3.1: 与えられたデータ

与えられたデータを表示すると図 3.1 ようになる。今回は「データの中で流動性が高い 30 銘柄」をまず選んだ。この理由として流動性リスクを回避することがあげられる。[6] によって与えられた 4 つのファイルから 3 年超えの出来高の合計から上位 30 銘柄を選び株数をまず絞った。

§ 3.2 データの前処理

まずデータの前処理として正規化を行った。流体情報処理装置の特徴として 0 を中心に回すことが私の研究より判明している。入力信号は円柱を回すことで与えるが、時計回りと反時計回りの両方があるとより精度が高いということである。よって正規化処理を行うことと 0 を中心に入力信号を与えるということは同義であると思っていよい。

また流体情報処理装置の重要なパラメータの一つに「最大表面角速度」がある。入力信号の振動数は定数として扱うとし振幅はとても重要なパラメータである。データ加工の際に正規化処理を行ったあと最大表面角速度が適切な値になるように選定した。その選定過程については次章のローレンツモデルを例にし考察する。

第4章 計算結果

§ 4.1 プログラムの概要

プログラムの概要をフローチャート形式で図 4.1 に示した。また Python と Fortran コードは以下ようになる。

1. Python コード

- (a) データ選定.
- (b) データの正規化.
- (c) Fortran 向けにデータを整形.

2. Fortran コード

- (a) 初期化（ファイル，パラメータ）.
- (b) 数値シミュレーションのグリッド設定.
- (c) 初期状態の速度，圧力を解く.
- (d) 定常状態になるまでスキップステップ.
- (e) 学習ステップ.
- (f) 予測ステップ.

埋め込み境界法を用いナビエストークス方程式を解くと以下の図のようになる。

Re=100 としたときの図である。カルマン渦列のような双子渦が確認できる。

今回のメインタスクに関する結果の前に一つローレンツモデルの結果を示す。この理由は「流体情報処理装置」の有効性を示すためである。そのあと今回のタスク「株価予測」に移る。基本パラメータは以下のように設定する。ここで図 4.3 を見てほしい。横軸がエ

表 4.1: 基本パラメータ

パラメータ	
Re	10
Δt	0.01
$k(= T/\Delta t)$	20000
F	$\omega_{max} * 2$

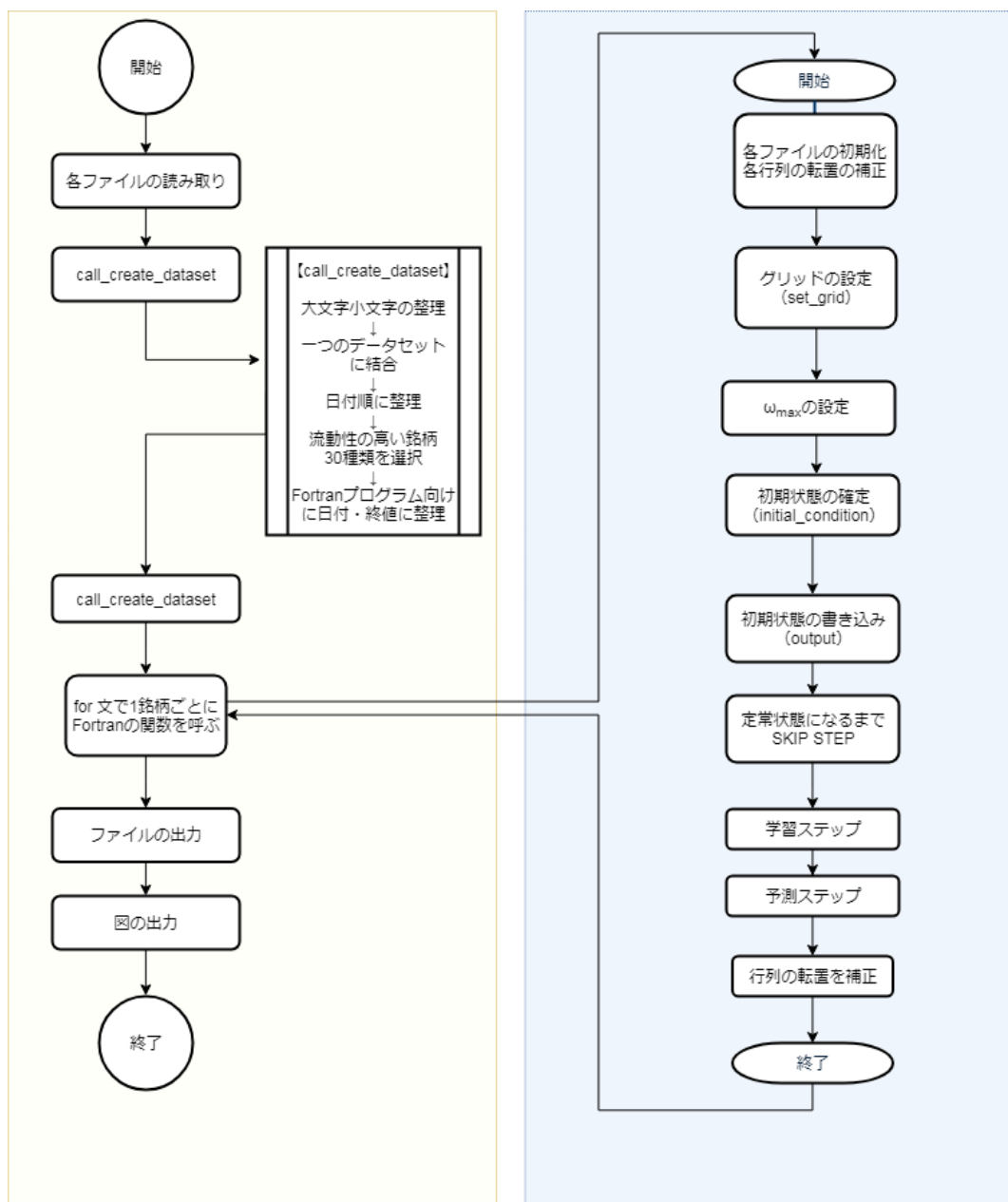


図 4.1: 全体のプログラムの流れ (黄色 : Python コード, 青 : Fortran コード)

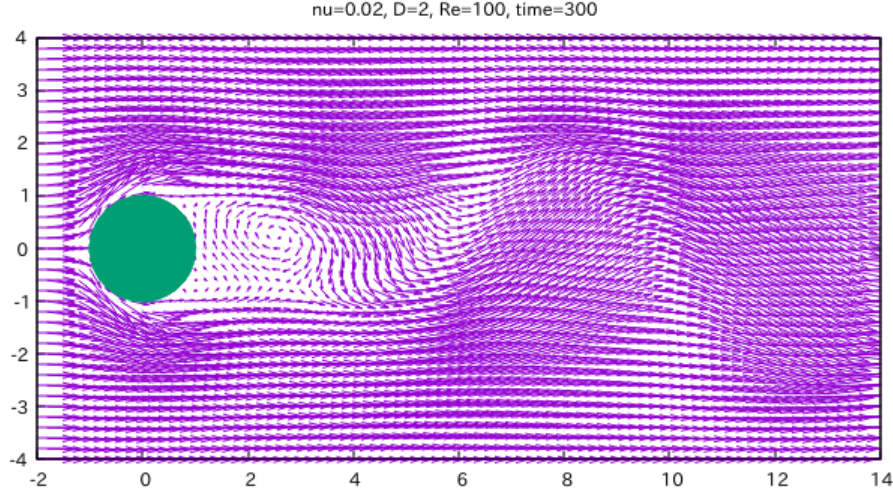


図 4.2: 流体情報処理装置 ($Re=100$)

ラーで縦軸が最大角速度，色がエラーの大きさを示している．リザーバーコンピュータの重要な数理としてリャプノフ指数が限りなく 0 に近い場所で最も性能が良くなる傾向がある [1]. この先行研究を踏まえて流体を考えると，円柱後流のながれが定常会から周期解になるあたりの Re 数が適切であると思われる．しかし私の先行研究より，「定常解であればよく周期解から定常解の分岐点である必要はない」ということが判明した（図 4.3）．また流れがない場合 ($Re = 0$) に比べ少しでもある ($Re > 0$) 方が性能が良くなることがわかる．これらを踏まえ，さらに計算時間の問題から今回は基本パラメータに $Re = 10$ に決定した．

※ ここで F は流体に信号を入力する際の最大回転角速度の 2 倍である．よって例えば $F = 10$ であれば円柱の最大回転速度 ω_{max} は 5 となる．今回，使用する円柱の半径は大きさ 1 に設定してあるので滑りなし条件から円柱表面速度も 5 である．

§ 4.2 Task1: ローレンツモデル

以下のローレンツモデルを使用した．この式はカオス的振る舞いをする代表的な非線形方程式である．

$$\begin{aligned}\frac{dx}{dt} &= -ax + ay \\ \frac{dy}{dt} &= bx - y - xz \\ \frac{dz}{dt} &= cz + xy\end{aligned}\tag{4.1}$$

$a = 10, b = 8/3, c = 8/3$ とする．教師データとして式 (4.1) をルンゲクッタで四次の精度で数値的に解き $\Delta t = 10^{-2}$ で出力したものを使用する．

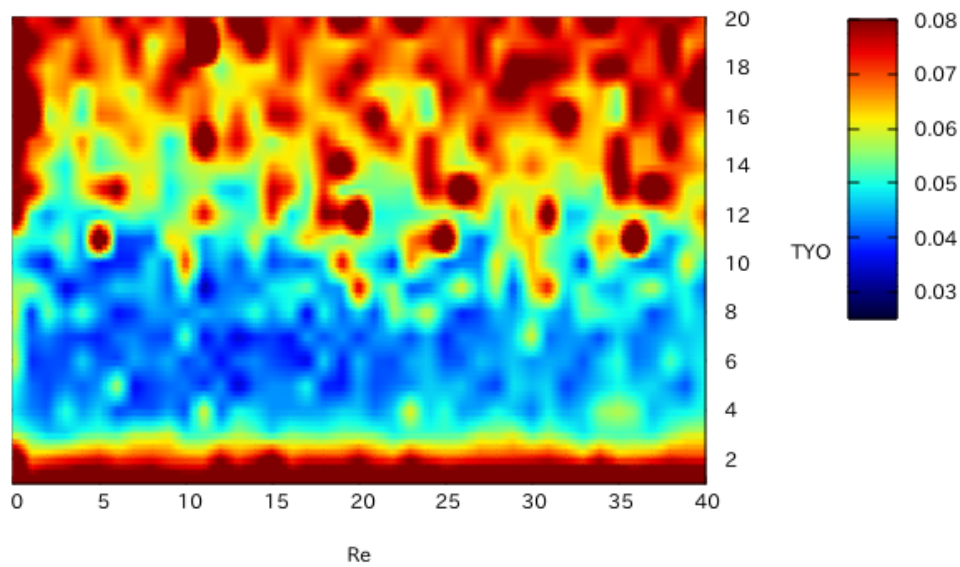


図 4.3: Re を変化させたことによるエラーの推移

§ 4.2.1 入力 $x(t)$, 出力 $y(t), z(t)$

ローレンツモデルの x 座標を入力値, y, z 座標を出力値として学習した. また循環層の役割をなす流体のスパン方向の速度成分を図 4.4 の部分の 100 点から読み取り使用した.

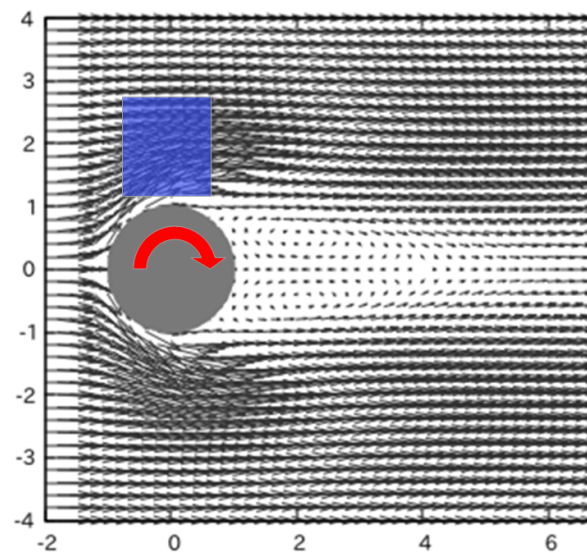


図 4.4: 流体情報処理装置の信号として読み取るおおよその場所

学習方法は上記の通りである. 結果を図 4.5, 4.6 に示す. 今回は真値と推定値との誤差に関して数値的な言及はしないが良い精度で流体情報処理装置が推定値を出力していることがわかる.

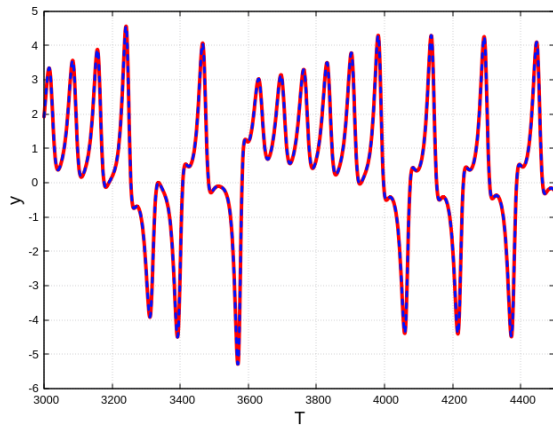


図 4.5: y 座標の推定値と真値

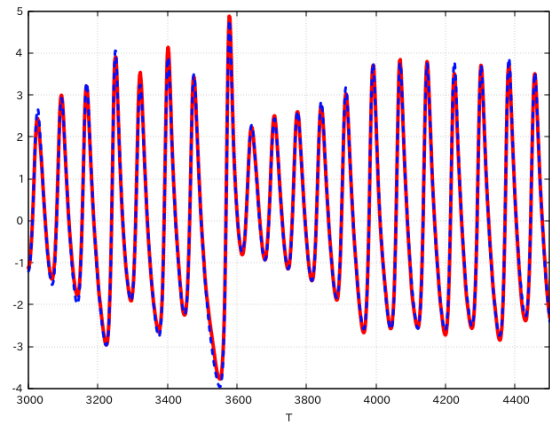


図 4.6: z 座標の推定値と真値

※ ここで青点線は真値，赤線は流体情報処理装置の出力値を示す。

また図 4.7, 4.8 は位相空間上に $(x(t), y(t), z(t))$ とし三次元プロットした図である。同様に良い精度で流体情報処理装置が推定できていると考えられる。

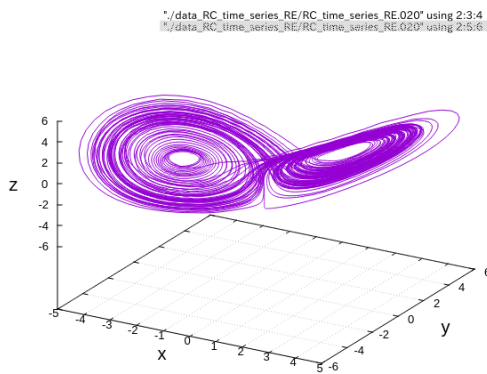


図 4.7: 三次元プロットした図 1

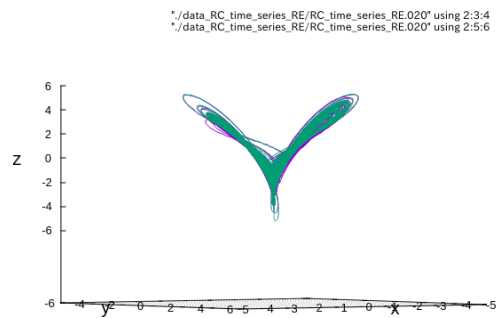


図 4.8: 三次元プロットした図 2

※ ここで緑線は真値，紫線は流体情報処理装置の推定値を示す。

§ 4.2.2 入力 $x(t)$, 出力 $x(t + 10)$

ローレンツモデルの x 座標の $x(t)$ を入力値, $x(t + 10)$ を出力値として学習した。

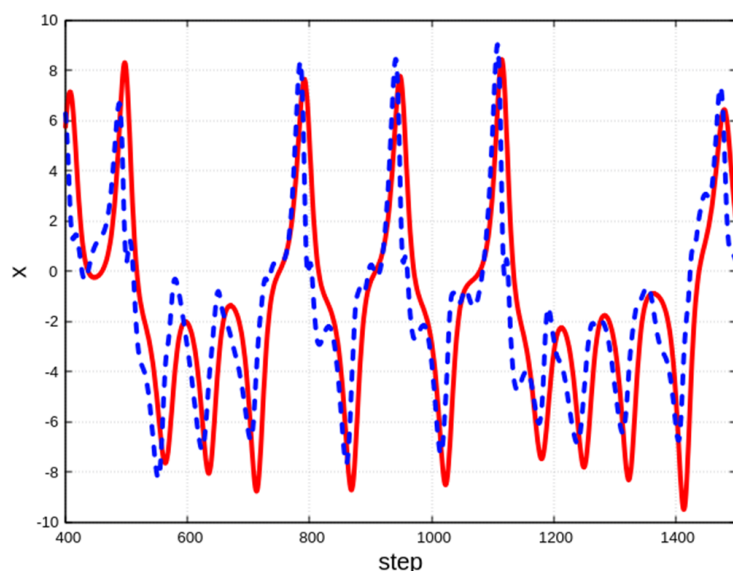


図 4.9: 真値 (青点線) 流体情報処理装置 (赤線) の比較

リザーバーとしてどの領域の流体運動を用いるのが適切か

流体情報処理装置をリザーバーとして用いる際、どの領域の流体運動（今回は具台的には速度）を循環層として用いることが適切であるかを検討した。図 4.10 は循環層の値の取得する場所を徐々に円柱から遠ざけた場合のエラーの推移である。図 4.10 の右図は遠ざけた様子を簡単に示したものである。

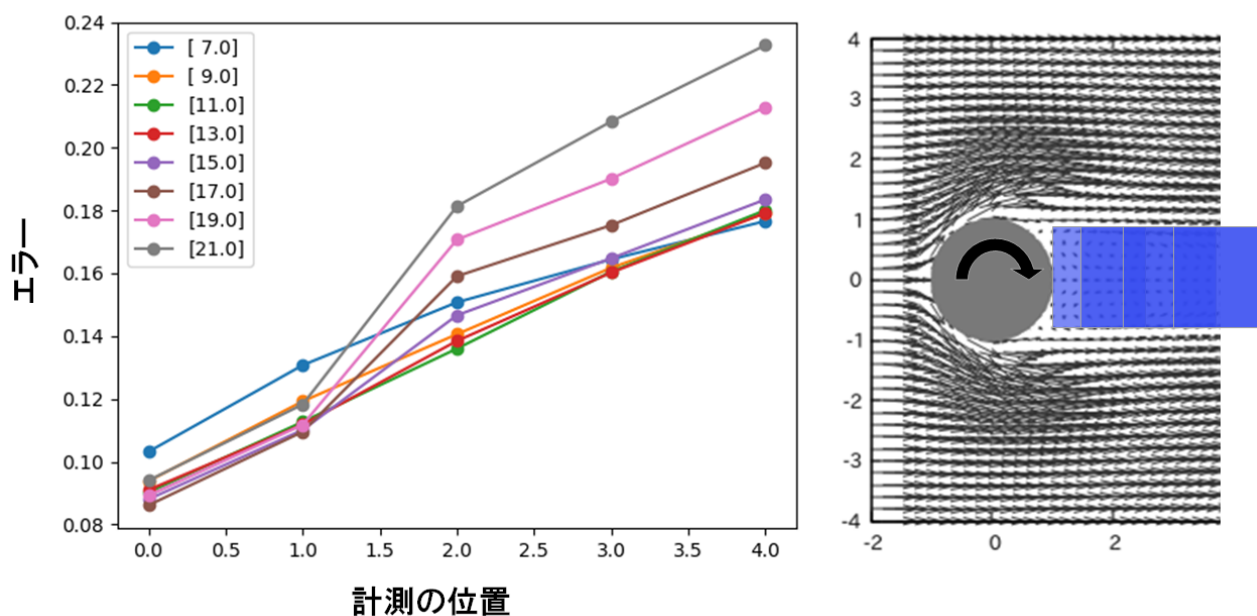


図 4.10: 左：エラーの推移/右：取得する場所の図

図 4.10 の左に注目する. 実線の色と数字 $F = 7.0 \sim 21.0$ 入力信号の大きさを示している. 詳細は表 4.1 に書いてあるので今一度見てほしい. 図 4.10 より円柱に近い場所ではすばやく回したほうがよりエラーが小さく ($F = 15.0 \sim 19.0$), 円柱から遠ざかると遅く回したほうがエラーが小さくなる ($F = 5.0 \sim 11.0$) 傾向があることがわかる. この理由として, 円柱から遠い場合, 円柱をすばやく回しすぎると与えた信号により循環層がカオス状態になり流体情報処理装置の推定精度が下がってしまったと考えられる.

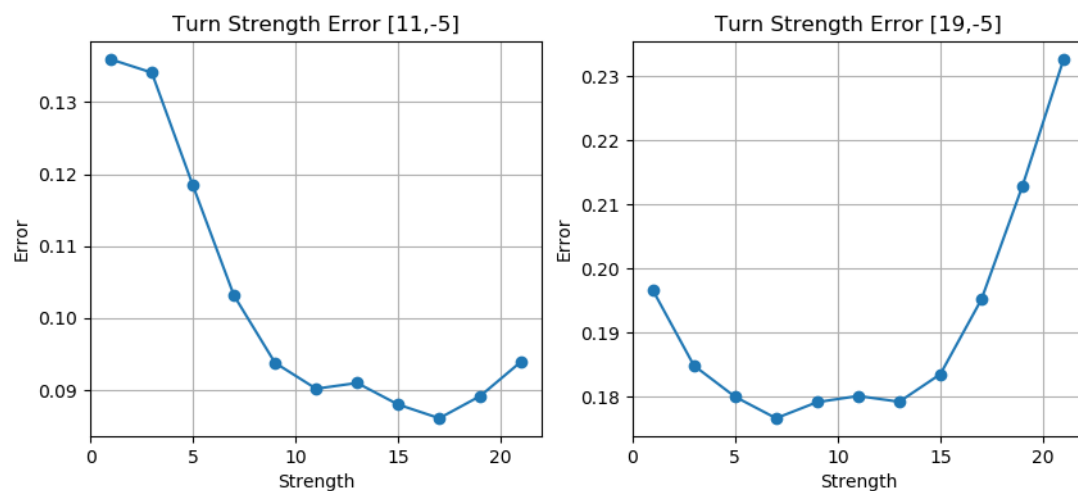


図 4.11: 回す速さによるエラーの変化 (循環層が近い (左)/遠い (右))

循環層として円柱から近い場所と遠い場所での比較

どの速さで回すとエラーが極値を取るかを検討した. 図 4.13 の左図ははより円柱に近い場所, 中央図は円柱から離れた場所を示す. わかりやすく図で循環層として値を抽出した場所は図 4.12 のようになる. (図 4.10 の左図横軸の 0.0, 4.0 に対応)

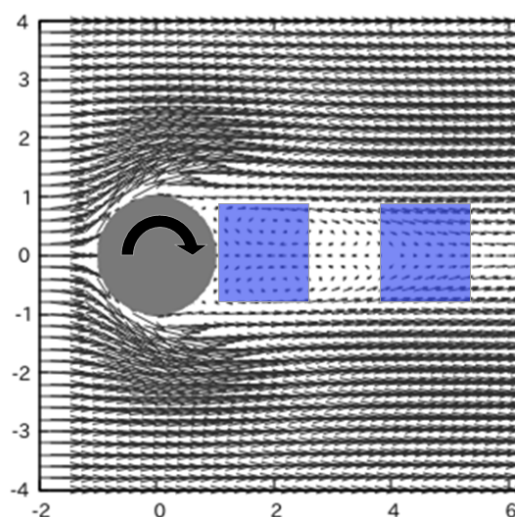


図 4.12: 左:円柱から近い/右:円柱から遠い

図より近い場所では17の強さ,つまり最大円柱表面速度が約8.5となる場合にエラーが最も小さくなる. また遠い場所では7の強さ,つまり最大円柱表面速度が3.5となる場合にエラーが最も小さくなる. これは流体がカオス状態になっていない且つ入力信号の円柱の回転がうまく流体に伝わっているからであると考えられる. 円柱の回転による入力 that うまく流体に伝わっているかどうかは図 4.13 をみるとより明らかに確認することができる.

次に図 4.13 で円柱から近い場合と遠い場合とで入力信号に対して, 循環層の値が入力値に対してどのように反応しているかを示した図である. 近い場合ではより早く回したときのほうが入力信号に鋭敏に反応しているように見える. また遠い場所ではより遅く回したほうが入力値の変化をうまく循環層が表現できているように感じる. 今回は数値的にこの判断をすることはできないが最大リャプノフ指数を変分方程式より求めることが可能になればより詳細な解析が可能になるようだろう.

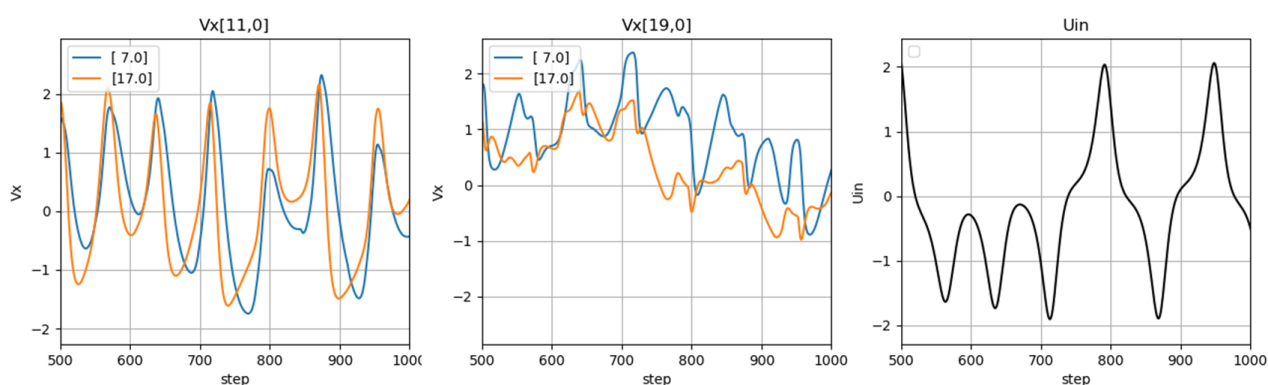


図 4.13: 左:円柱から近い/中央:円柱から遠い/右:入力信号の変化

今回エラーが低くなった理由を「循環層として使用した流体がカオス状態になったため」とした. しかしリャプノフ指数をうまく求めることが現状できていない以上は的確な説明をこの観点からつけることができない. そこで別の味方として「学習した出力重みが

うまく流体の信号を捉えることができているかどうか」という別の観点から考えることにした。最後に図 4.14 は学習した出力重み W^{out} の 100 点の値の大きさを示したカラーマップ（左）とそれぞれの値を分析したグラフ（右）である。

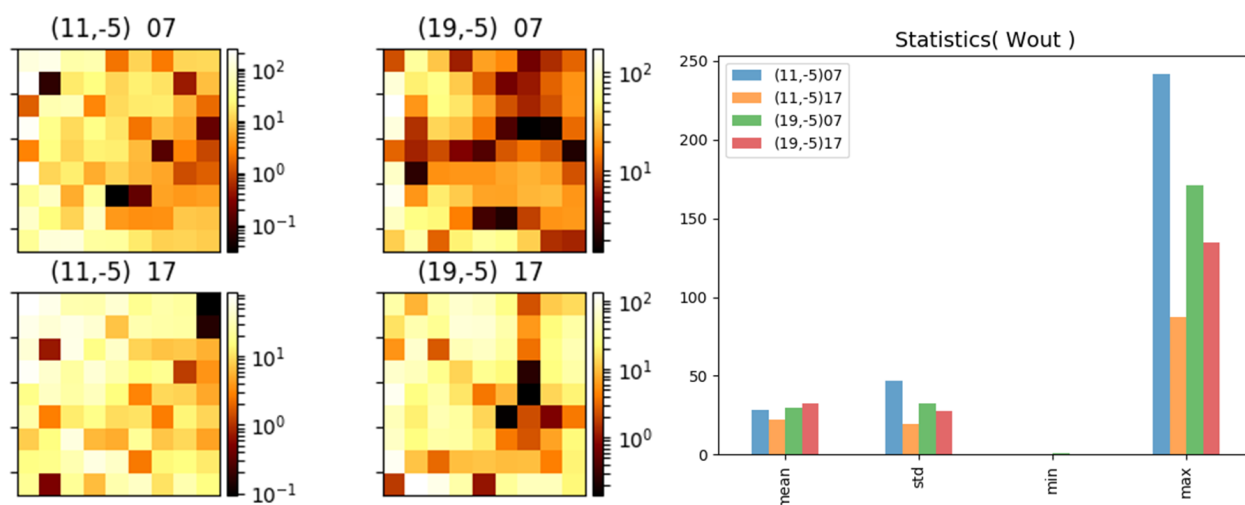


図 4.14: RC の出力値（緑）と真値（紫）の比較

当初この見方をする際「エラーが大きいほど出力重み行列 W^{out} の分散が大きい」との予想をたて解析をすすめた。右のグラフの「std」が分散を示している。確かに円柱から近い場合は成り立っている。エラーの小さい「早く回した場合（オレンジ）」が「遅く回した場合（青）」に比べて分散が小さくなっている。しかし円柱から離れた場合は成り立っていない。グラフを見えてもらえばわかるように円柱から近い場合と同様に「早く回した場合（緑）」が「遅く回した場合（赤）」に比べて分散が小さくなっている。よって今回の計算結果からは出力重みの数値的分散がエラーの大きさに直接寄与しているかどうかは言及することはできなかった。

§ 4.3 Task2:株価予測

全セクションは使用する流体の範囲を小さめの 100 点に限った。今回の株価予測ではより流体の可能性を信じるべく、2000 点を使用した。範囲は図 4.15 の青で示した。前章で重要とされた最大表面角速度はローレンツモデルを参考にし、 $\omega_{max} = 15$ を選定した。この理由としては円柱からの距離によらずエラーが比較的小さくなったことが挙げられる。データの選定・加工の章の方式に沿って Python コードで加工し、Fortran の流体情報処理装置の関数を使用し予測した。

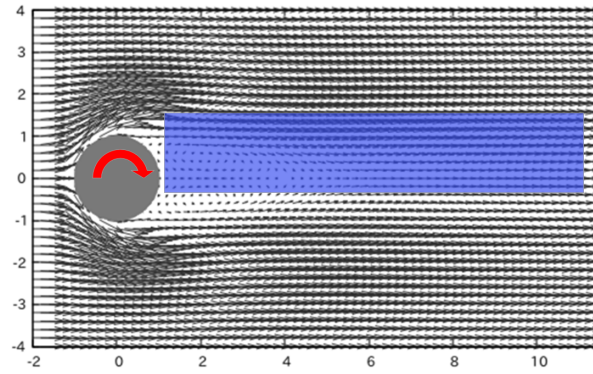
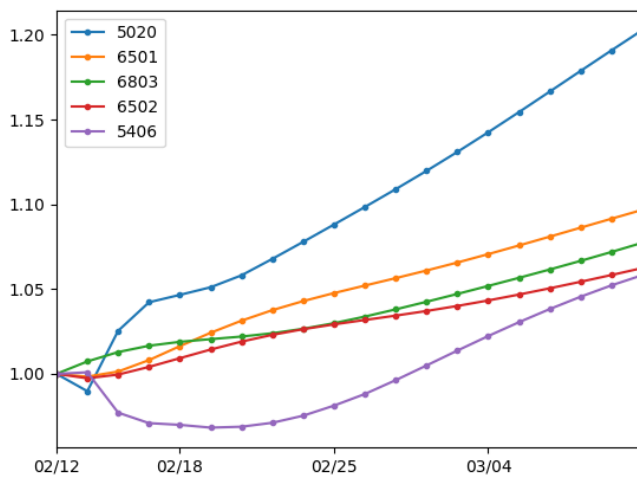


図 4.15: 株価予測の結果（上位 5 銘柄）

株価の予測結果は図 4.16 である。横軸は 2 月 8 日からの時間、縦軸は 2 月 8 日の株価を 1 に規格化し 1 としたときの増加を示している。



銘柄	2 月 28 日
5020	1.10
6501	1.05
6803	1.03
6502	1.03
5406	1.01

表 4.2: 上位 5 銘柄の上昇率

図 4.16: 株価予測の結果（上位 5 銘柄）

これらの結果を踏まえ、予算 1000 万に収まる範囲で株数を適当に調整しポートフォリオ登録をした。プログラム作成当事者の筆者として申し上げることは憚れるが「この株価予測としては不完全」である。まず一つの理由としてはローレンツモデルと株を支配する時間というものが全く異なるからである。つまりローレンツモデルの 1 ステップと株の時系列データの 1 ステップは全くの別物ということである。今回は流体シミュレーションをしているため計算時間が多くかかり FDC で与えられた短い時間では一ステップをどう捉えるかということを検討することが不可能であった。もう一つの理由として株価予測に RC の物理実装の流体情報処理装置を使用するメリットを定量的に評価できていない点である。この点については今後の研究により改善していく予定である。

第5章 結言

前章までに得られた結果をまとめる.

- 1 流体情報処理装置が RC の循環層の役割をなすことが数値計算により実証された.
- 2 株の流動性を考慮し, より身近な株価変動の予測を流体情報処理装置で行った.
- 3 流体情報処理装置を用い株価予測を行った. 定量的評価を満足しているかどうかという観点では, データ加工を経て流体情報処理装置の結果から上位 5 位を選んできたことを考えると妥当だと考えられる.

これからの課題をまとめる.

- 1 リザーバーコンピューターとして流体情報処理装置の性能を数値的に表す.
- 2 単一のデータからの時系列データ予測に頻繁に起こる「結果の考察」という点では今回の流体情報処理装置を使用した場合は不完全である.
- 3 流体情報処理装置の入力信号にかかる重みも学習するようプログラムを組むめるよう, 現在研究中の RNN 及び LSTM の数理的構造理解をさらにすすめる.
- 4 移流が流体情報処理装置に及ぼす影響を解析する.

参考文献

- [1] Joschka Boedecker. *Echo state network reservoir shaping and information dynamics at the edge of chaos*. PhD thesis, 大阪大学, 2011.
- [2] Masanobu Inubushi and Kazuyuki Yoshimura. On the characteristics and structures of dynamical systems suitable for reservoir computing. 電子情報通信学会論文誌 A, Vol. 102, No. 2, pp. 114–120, 2019.
- [3] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, pp. 78–80, 2004.
- [4] Zhixin Lu, Jaideep Pathak, Brian Hunt, Michelle Girvan, Roger Brockett, and Edward Ott. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 27, No. 4, p. 041102, 2017.
- [5] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *CoRR*, Vol. abs/1808.04962, , 2018.
- [6] 真大澤 淳村永・日本銀行金融研究所ディスカッション・ペーパー・シリーズ. Vol. No. 98-J-2, , 1998.