

ニューラルネットワークを用いた偏微分方程式導出・求解を支援する 視覚的分析

小山田 耕二¹⁾

1) 京都大学 学術情報メディアセンター

koyamada.koji.3w@kyoto-u.ac.jp

Visual analysis to support the derivation and resolution of PDEs

using neural networks

Koji KOYAMADA¹⁾

1) Academic Center for Computing and Media Studies, Kyoto Univ.

概要

本稿では、ニューラルネットワーク (NN) を使用した点群データからの高精度偏微分方程式 (PDE) の導出と解法をサポートするために、視覚分析 (VA) アプローチを提案する。点群データは、データが定義されている時空間座標と、それらの座標に関連付けられている物理データで構成される。NN では、時空間座標を入力し、物理データを出力し、点群データを近似する関数を回帰する。これにより、特に方程式を解く場合に、偏微分方程式の残差が損失関数になり、教師データはゼロとなる。また、NN 構造の最適化は重要であるが、自動化が難しく、主に利用者に委ねられている。したがって、点群データから PDE 導出・求解を行う際には、NN モデルを自動的に構築するプロセスと並行して、人間の関与の重要性を強調する VA を開発する。VA では、時空間およびパラメータ空間における高度な視覚化技術と、NN における損失関数の画期的な構築技術から構成される。この方法の有効性を確認するために、専門家による偏微分方程式の導出と解法をサポートするための要件を明確にする。

1 はじめに

本稿では、ニューラルネットワーク (NN) を使用した点群データから偏微分方程式 (PDE) の導出と解法をサポートするために、視覚分析 (VA) 手法を提案する。VA では、データからの知見創出において、モデル構築を自動的に行うプロセスと並行して、人間関与の重要性を強調する。本研究では、VA の構成要素は、点群データ・可視化・NN モデル・知見創出である (図 1 参照)。

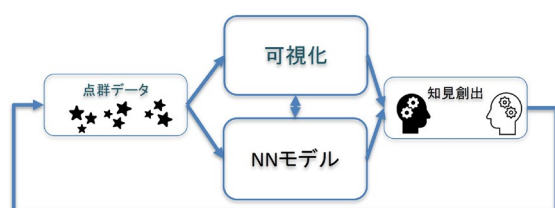


図1 視覚的分析手法

現在、さまざまな現象から取得された点群デ

ータから知見を得る手段として、偏微分方程式 (PDE) の導出に注目されている。PDE は、多変数関数のさまざまな偏導関数の間に関係を表す方程式である。点群データとは、複数の時空間位置で定義された物理量を意味する。たとえば、ナビエ・ストークス方程式は、粘性流体物質の運動を記述する特別の偏微分方程式である。この方程式を効率よく解くためには、対象とする流体现象において、移流効果が無視できるのかどうかを判断して、移流項の省略を行う。この判断のために、取得された流体データから PDE を導出して、移流項を残すべきかどうかを検討することができる。

有限差分法や有限要素法などの伝統的な手法では、PDE を数値的に求解する場合、偏微分項の評価には局所的な補間を前提にしていたので、遠方の影響を取り込むのが困難であった。メッシュフリー解法を代表する粒子法においては、偏微分項の評価は、着目する点の影響半径で設定される球内部という局所情報を利用する[1]。最近は、任

意の連続関数を近似できるという普遍化近似定理 [2、3]に基づき、時空間モデルとしてニューラルネットワーク(Neural Network: NN)モデルを利用することが多くなっている(図 2 参照)。NN モデルでは、偏微分項の評価には大域的な近似を用いることとなる。

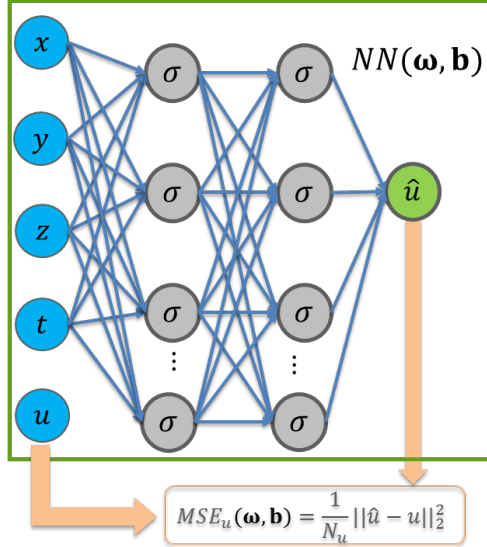


図2 NNモデル

本研究では、NN モデルを使った PDE の導出・求解に注目する。まず、PDE の導出・求解には、与えられた離散点における偏微分項の評価が必要であり、与えられた離散点データを説明する時空間モデルとして、NN を採用する[4]。離散点データとして必要なのは、点群だけなので、点間接続を表すデータは不要で、PDE 求解としては、メッシュフリーに分類される。

NN モデル構築は、損失関数を最小化する NN パラメータ最適化計算である(図 1)。ここで、損失関数は、点群データと NN モデルにより回帰されたデータとの平均二乗誤差 $MSE_u = \|\hat{u} - u\|_2^2 / N_u$ であり、NN モデル誤差と呼ぶ。ここで、 N_u は、PDE の定義される時空間で設定される離散点の数を表す。PDE の導出・求解では、NN モデルにおいて、この離散点で候補となる偏微分項を計算し、この偏微分項から構成される誤差項をどのような NN モデルにどのように組み込むかが高精度化を実現する上で重要である。

2 自動微分について

PDE 導出・求解では、与えられた離散点にお

ける偏微分項の評価が必要であり、時空間離散データを回帰する NN モデルを構築する。その NN モデルに対して自動微分を適用して、PDE を構成する。ここで、自動微分 (Automatic Differentiation: AD) は、コンピュータープログラムで定義された関数の導関数を評価するための一連の手法を意味する[5]。

自動微分 (じどうびぶん、アルゴリズム的微分とも) とは、プログラムで定義された関数を解析し、偏導関数の値を計算するプログラムを導出する技術である。自動微分は複雑なプログラムであっても加減乗除などの基本的な算術演算や基本的な関数 (指数関数・対数関数・三角関数など) のような基本的な演算の組み合わせで構成されていることを利用し、これらの演算に対して連鎖律を繰り返し適用することによって実現される。自動微分を用いることで偏導関数値を少ない計算量で自動的に求めることができる。

ここでは、深層学習で広く使用されている完全接続型 NN の入力変数の 1 階偏導関数を計算する方法について説明する[2]。図 1 は、完全接続型 NN の構造を示す。この NN は、入力層と出力層で構成され、入力層と出力層の間に隠れ層がある。各層のニューロンは、隣接する層のニューロンに完全に接続されるものとする。

k 番目の層の j 番目のニューロン u_j^k は、 $(k-1)$ 番目の層のニューロンに接続される。これは次のように表される。

$$u_j^k = \sigma^{k-1} \left(\sum_{i=1}^{n_{k-1}} w_{i,j}^{k-1} u_i^{k-1} \right) \quad (1)$$

次の式は、この接続プロセスを目的変数 u_j^{L+1} まで再帰的に繰り返すことによって得られる。

$$u_j^{L+1} = \sigma^L \left(\sum_{i=1}^{n_L} w_{i,j}^L \sigma^{L-1} \left(\sum_{i=1}^{n_{L-1}} w_{i,j}^{L-1} \dots \sigma^0 \left(\sum_{i=1}^{n_0} w_{i,j}^0 u_i^0 \right) \dots \right) \right)$$

ここで、 $w_{i,j}^{k-1}$ は、ニューロン u_i^{k-1} から u_j^k までの重みであり、 σ^k と n_k は、それぞれ k 番目の層

の活性化関数とニューロンの数を表す。シグモイド関数 $\sigma(x) = 1/(1 + e^{-x})$ が活性化関数として使用されると仮定する。 k 番目の層のバイアス b_j^k は、 $u_{n_k+1}^k = 1$ と設定することにより、重み $w_{n_k+1,j}^k$ として計算することもできる。重みの総数 n_{weight} 、バイアスの総数 n_{bias} は、

$$\begin{cases} n_{weight} = \sum_{i=1}^L n_i n_{i-1} \\ n_{bias} = \sum_{i=1}^L n_i \end{cases}$$

すべての重み、バイアスを格納する配列をそれぞれ $\omega, \mathbf{b}$ と表す。

以下では、出力層の単位 u_j^{L+1} に対する入力層 u_j^0 の1階導関数 $\frac{\partial u_j^{L+1}}{\partial u_j^0}$ を計算する方法を説明する。

$u_j^0, j = 1, 2, 3, 4$ は、それぞれ、時空間座標 x, y, z, t を表し、 u_1^{L+1} は、説明変数 u を表す。具体的には、 $u_1^{L+1} = u(x, y, z, t)$ $\frac{\partial u_1^{L+1}}{\partial u_1^0} = \frac{\partial u}{\partial x}$ となる。

1 階導関数計算では、連鎖律を適用して、

$\frac{\partial u_1^{L+1}}{\partial u_j^0}$ を分解することにより、反復手順を構築する。 u_j^{k+1} は、 $u_i^0, u_i^1, \dots, u_i^k$ によって計算される。

したがって、 u_j^{k+1} は、 $u_i^0, u_i^1, \dots, u_i^k$ の合成関数と見なすことができ、その偏導関数は連鎖律を使用して計算される。連鎖律による偏導関数の分解をこの合成関数に再帰的に適用することにより、入力変数の偏導関数を計算する。この手順では、 $\frac{\partial u_1^{L+1}}{\partial u_j^0}$ は中間変数 u_i^1 の連鎖律に従って

分解され、その部分式は中間変数 u_i^2 の連鎖律によってさらに分解される。次の式は、この分解プロセスを中間変数 u_j^L まで再帰的に繰り返すことによって得られる。

$$\frac{\partial u_j^{L+1}}{\partial u_i^0} = \sum_m^{n_1} \left(\sum_n^{n_2} \left(\sum_m^{n_3} \left(\dots \left(\sum_m^{n_L} \frac{\partial u_j^{L+1}}{\partial u_m^L} \frac{\partial u_m^L}{\partial u_i^{L-1}} \right) \dots \right) \frac{\partial u_m^3}{\partial u_i^2} \right) \frac{\partial u_m^2}{\partial u_i^1} \right) \frac{\partial u_m^1}{\partial u_i^0}$$

3 PDE 導出について

これまで、我々は自動微分を使って、与えられた時空間離散データより PDE 導出を行う研究を行ってきた[4]。この研究では、与えられ

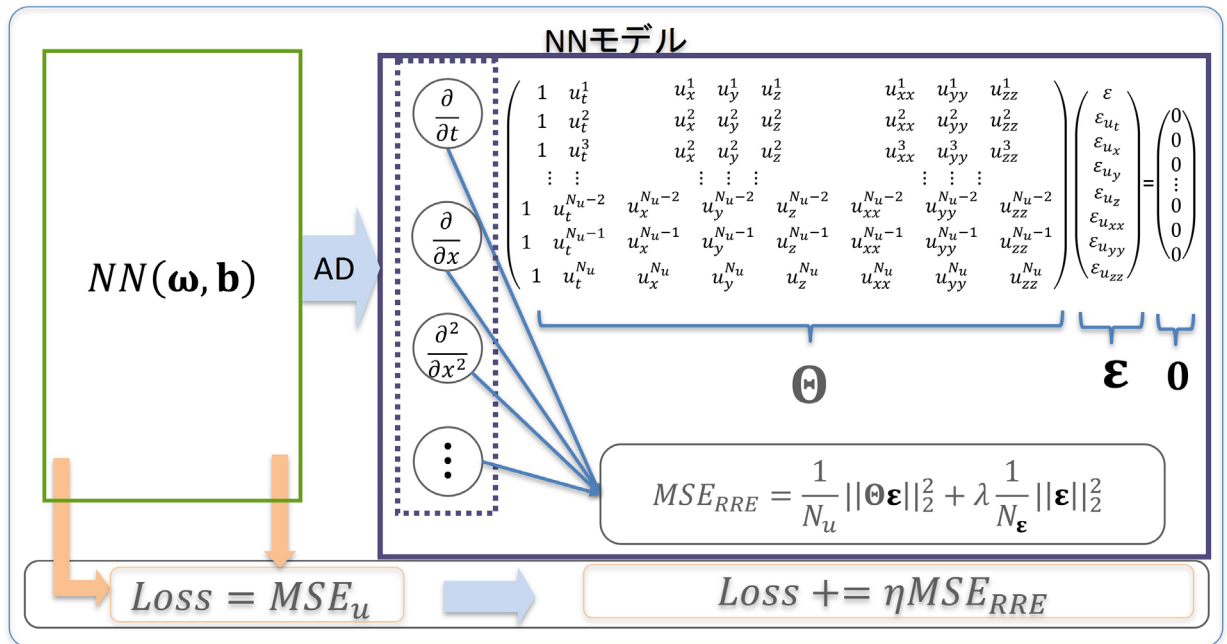


図3. PDE導出・求解のためのNNモデル

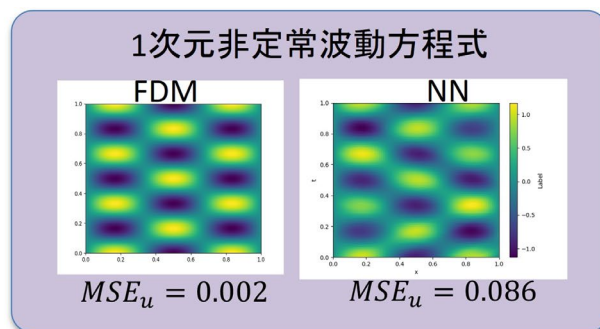
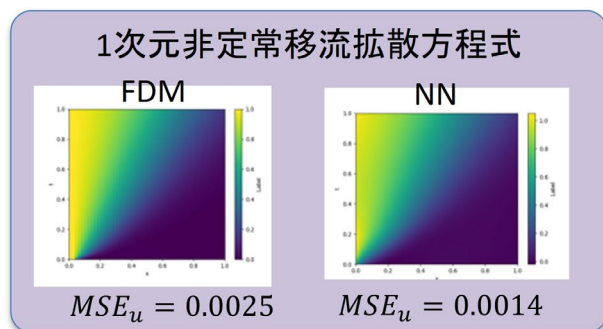


図5 NNとFDMにおけるPDE求解精度比較

た離散点における偏微分項の評価が必要であり、時空間離散データを回帰する NN モデルを構築して、そのモデルに対して自動微分を適用して、偏微分項を計算し、正則化回帰方程式 (Regularized regression equation: RRE) ・ PDE を構成した。NN モデル構築を損失関数の最小化問題と見なし、損失関数に偏微分項から構成される新たな誤差項を加えて、各目的関数 (誤差) を加算し、単一の目的関数を有する最適化問題に変換する (図 3 参照)。

損失関数としては、説明モデルと時空間離散データの誤差：モデル誤差だけでなく、RRE 誤差を統合し、NN モデルを構築する。NN モデル構築は、損失関数を最小化する NN モデルパラメータを最適化する計算プロセスであり、この損失関数を複数の誤差から構成する手法を提案するものである。ここで、NN モデルパラメータは、層数、層ごとのニューロン数、ニューロン間の重み、ニューロン毎のバイアス値から構成される。

4 PDE 求解について

PDE 求解においても、PDE 導出同様、NN モデル構築を損失関数の最小化問題と見なし、損失関数に偏微分項から構成される新たな誤差項を加えて、各目的関数 (誤差) を加算し、単一の目的関数を有する最適化問題に変換する。PDE 求解では、特に、NN モデルにおいて、境界条件において、モデル誤差を定義する。境界上で解が制約される境界条件だけでなく、温度場における熱流束に課される条件のように、解の勾配に境界条件が付される場合がある。この場合は自動微分による勾配計算を行って、誤差が定義される。

次に、この損失関数に PDE 誤差を統合し、NN

モデルを構築する。ここで PDE 誤差を定義するために、PDE を左辺=0 の形式にして、左辺を偏微分項で構成する。NN モデル構築における最適化計算時には、右辺がゼロにならないのでこれを PDE 誤差と定義する。

本研究では、NN モデルを用いた求解手法を厳密解のある PDE に適用する。このような PDE として、1 次元非定常移流拡散方程式や 1 次元非定常波動方程式を取り上げ、有限差分法などの伝統的な手法による解析結果との比較を行った (図 5 参照)。

検証実験で用いた 1 次元非定常移流拡散方程式の形式、特長及び設定した初期条件や境界条件やそれに基づいて導いた厳密解については、[6]を参照してほしい。今後は、次元数を 2 や 3 にした場合、他の PDE の場合、また、ナビアーストックス方程式のように複数の PDE から構成される場合などについて、本手法の有用性を確認する予定である。本手法と同様の提案が Raissi らより提案されているが、彼らは、伝統的 PDE 求解手法との比較を行っていない [7]。

5 PDE 導出・求解を支援する VA

本研究では、これまで述べた自動微分・PDE 導出・求解技術から構成される VA を使って、PDE 導出・求解誤差がどのような要因に支配されるのかを明らかにする。具体的には、NN モデルパラメータ、偏微分項の候補多様性、点群データ数などから構成される多次元データに対する PDE 導出・求解誤差分布を可視化し、専門家による要因分析を支援する。また、時空間偏微分項に対して、Empirical Dynamic Modeling (EDM) [8]を適用して、構成される多次元時系列データから特定されるシステム状態とその遷移を可視化し、偏微分項に関する本質的な組み合わせの探索を支援する。

5.1 粒子レンダリングを利用した点群データ可

視化

この研究では、核融合炉内における、電磁場シミュレーション結果から適切な開始点を設定して計算された磁力線を構成する点に注目して、磁力線束を点群データと見なす。NN そして背景格子を使って粒子個数密度を計算した2つの場合に対して、粒子レンダリング可視化結果を計算時間・画質の観点で比較する。

5.2 NN モデルを利用した高精度な PDE 導出・求解

この研究では、対照実験用に、Loss に AD で構成した誤差を加算しない手法を開発する。具体的には、PDE 導出において、点群データから NN モデル構築のみを行い、その次に、RRE の最適化を行う手法である。この2つの手法において、厳密解をもつ PDE に対して、同じ点群データを生成して PDE を導出し、その PDE 導出誤差を比較する。

5.3 パラメータ空間における NN モデル可視化技術

この研究では、平行座標や次元圧縮等による多次元データ可視化技術を開発し、PDE 導出・求解誤差を最小化するパラメータ探索の観点での有用性の評価を行う。具体的には、意図的に

作成した多次元データを使って、PDE 導出・求解誤差を最小化するパラメータをどの程度正確に、効率よく探索できるかという観点でユーザースタディを実施する。

5.4 VA 技術を使った適用検証

図4に、PDE 導出・求解のための VA ダッシュボード例を示す。3つのウィンドウから構成され、左上では、対象時空間内の点群データの指定、PDE 選択を行い、PDE 導出の場合、偏微分項候補数を選択された PDE の何倍にするかを指定する。また、PDE 求解の場合、境界条件で指定する領域での点群データ数の指定を行う。偏微分項候補について、EDM を使って計算したアトラクタを二次元平面で可視化する。右上では、NN モデルパラメータについての情報を指定する。具体的には、NN の層数の範囲、各層におけるニューロン数の範囲、Loss に追加する RRE/PDE 誤差への重み η を指定する。実行ボタンを押すとこの条件を含む python コードが計算サーバ上で実行され、下のウィンドウで、計算結果(Loss、PDE 導出・求解誤差など)を可視化する。可視化は、時空間・パラメータ空間

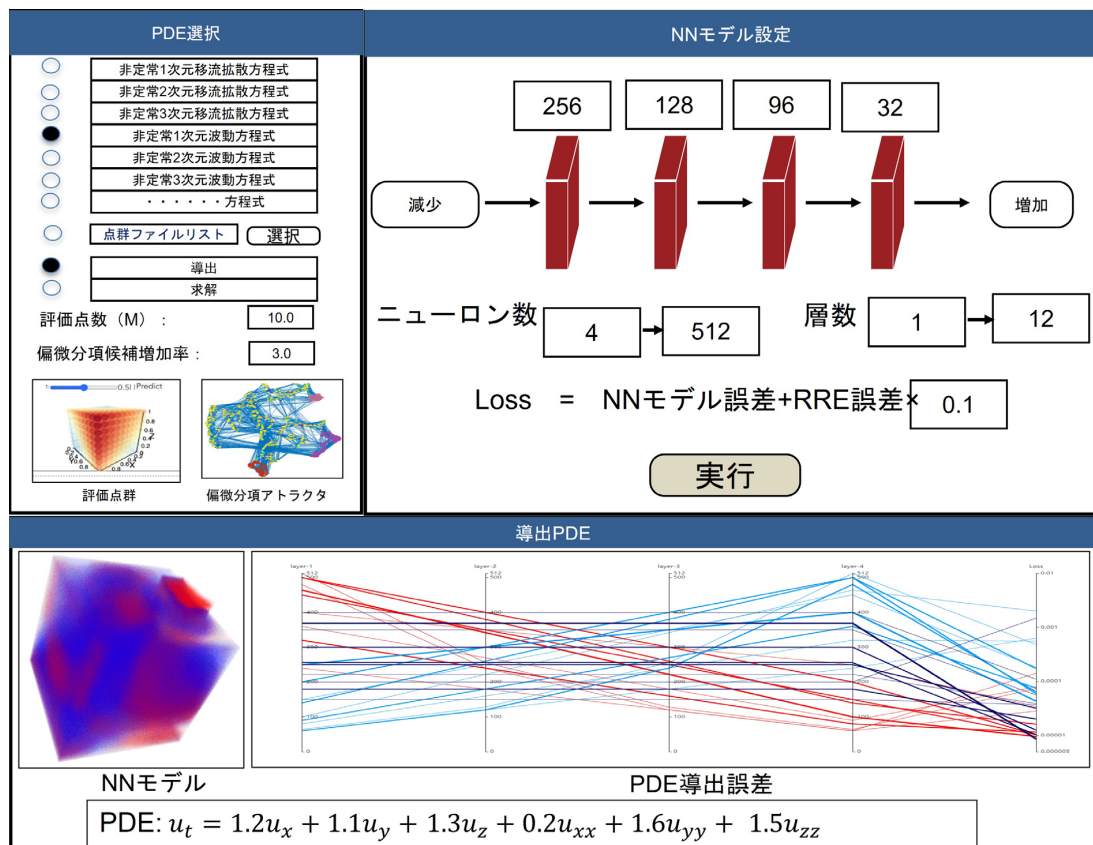


図4. PDE導出・求解を支援するVAダッシュボード

で行う。このダッシュボードを専門家に利用してもらい、以下の観点で評価する予定である。

1. 時空間において、点群データと構築されたNNモデルがどの程度一致しているか明らかにできるか。(5.1、5.2)
2. PDEを構成する偏微分項候補・境界条件多様性がPDE導出誤差にどのような影響を与えるか。(5.3)
3. NNモデルパラメータ・点群データ数がLoss、PDE導出・求解誤差にどのような影響を与えるか(5.3) たとえば、図4右下では、“出力層に向かってニューロン数減少”パターンがLossを小さくすることがわかる。
4. PDE求解誤差について、**硬いPDE**を使って、特に局所補間と大域近似に関して、伝統的手法と比較する。(5.4) たとえば、図4左下では、粒子レンダリングを使って比較結果を確認できることを示す。
5. PDEを構成する偏微分項とそうでない項との違いをEDMにより明らかにできるか。

6 おわりに

本研究では、NNモデルを使ったPDEの導出・求解に注目し、PDE導出・求解誤差がどのような要因に支配されるのかを明らかにするための視覚的分析の要件について述べた。今後は、視覚的分析システムのプロトタイプを開発し、これを専門家に利用してもらい、そのフィードバックを得て、システムの改善に役立てる予定である。

参考文献

- [1] 室谷他、“流体シミュレーションにおける粒子法：MPS法とLSMPS法の数値解析精度の比較”、応用数理、26/2、2016
- [2] G. Cybenko、“Approximations by superpositions of sigmoidal functions”、Mathematics of Control, Signals, and Systems、2(1989)、303 - 314. doi:10.1007/BF0255127
- [3] Liang and R. Srikant、“WHY DEEP NEURAL NETWORKS FOR FUNCTION APPROXIMATION?”、ICLR2017
- [4] K. Koyamada、et al.、“Data-driven derivation of partial differential equations using neural network model”、International Journal of Modeling、Simulation、and Scientific Computing、Vol.12、No.02、2021
- [5] 自動微分 Wikipedia、<https://ja.wikipedia.org/wiki/%E8%87%AA%E5%8B%95%E5%BE%AE%E5%88%86>
- [6] Convection–diffusion equation、https://en.wikipedia.org/wiki/Convection%E2%80%93diffusion_equation
- [7] M. Raissi、et al.、“Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”、Journal of Computational Physics 378 (2019) 686–707
- [8] Hiroaki Natsukawa、Ethan R. Deyle、Gerald M. Pao、Koji Koyamada、George Sugihara:”A Visual Analytics Approach for Ecosystem Dynamics based on Empirical Dynamic Modeling.” IEEE Trans. Vis. Comput. Graph. 27(2): 506-516 (2021)