

Colaboratory による動的数学文書の作成と数学リメディアル

日本大学・生物資源科学部 濱田 龍義

Tatsuyoshi HAMADA, College of Bioresource Sciences, Nihon University

1 はじめに

2020年4月、新型コロナウイルス感染症の影響から、多くの大学においてオンラインでの対応が必要になった。本稿は2019年に[1]で紹介した内容について、オンラインへの対応を試みた記録である。その過程において、Google Colaboratory による動的ドキュメントの可能性についての考察を行った。Google Colaboratory を用いて作成した動的ドキュメントを利用し、プログラミング教育の導入を行いながら、深層学習等で利用する基礎的な数学概念についての確認を行った。

2 Google Colaboratory について

Google Colaboratory (以下, Colaboratory)¹は Google によって提供されている開発環境である。主要な言語として Python をターゲットとしており, Web ブラウザ上で Python コードを記述, 実行できる。開発環境等をインストールする必要がなく, GPU の利用や, コードの共有が容易であるため, AI の研究者やデータサイエンティストを中心に人気を集めていたが, 最近は教育環境としても注目されている。「Colab ノートブック (図 1)」と呼ばれるコードを記述, 実行できる対話的な環境が用意されており, コードセルにはプログラムのソースコードを, ドキュメントセルには, リッチテキスト (画像, HTML, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$) を記述し, 1つのノートブックとすることができる。



The screenshot shows a Google Colab notebook interface. The top part is a code cell with the following content:

```
## 1.3 Python インタプリタ  
教科書のp.4を開いてください。  
  
Python のバージョンを調べます。  
colabでバージョンを調べるために,ここでは '!' (exclamation mark) を使います。  
  
'!python --version'  
  
と入力してください。**--versionの前には空白(スペース)が必要です.**  
入力したら「セルを実行ボタン」を押してください。  
  
または,  
'Ctrl+Enter'  
もしくは  
'command+return'  
を押しましょう。
```

The bottom part is a text cell with the following content:

1.3 Python インタプリタ
教科書のp.4を開いてください。
Python のバージョンを調べます。 colabでバージョンを調べるために,こ
こでは '!' (exclamation mark) を使います。

!python --version
と入力してください。 --versionの前には空白 (スペース) が必要です。 入力
したら 「セルを実行ボタン」を押してください。
または, Ctrl + Enter もしくは command+return を押しましょう。

Below the cells, the output of the code cell is shown:

```
[1] !python --version  
Python 3.7.15  
  
[2] !cat /etc/issue  
Ubuntu 18.04.6 LTS \n \l
```

図 1: Colab ノートブック

¹<https://colab.research.google.com/>

Colab ノートブックは、Jupyter Notebook によって構築されている。これは IPython と呼ばれる Python のための対話的実行シェルが原型である。現在、Jupyter Notebook は Python だけでなく R や Julia 等、数十の言語をサポートしている。例えば、2019 年には神戸大学の高山信毅氏によって asir カーネルの実装が行われ、Jupyter Notebook 上で数式処理システム Risa/Asir を利用できる [2]。

Jupyter Notebook 自体はオープンソースソフトウェアプロジェクトとして公開されているので、Google のサービスを利用しなくても、適切な環境を用意すれば作成したノートブックを編集、実行が可能である。また、機械学習やデータサイエンティストのためのコミュニティとして Kaggle や、汎用数式処理システム SageMath に特化した CoCalc など、Jupyter Notebook を提供する他のクラウドサービスも複数存在する。

2.1 Google Classroom との連携

日本大学生物資源科学部では、学習支援環境として Google Classroom (以下、Classroom) を活用している。基本的には、すべての講義に対してクラスを作成し、Classroom はクラウドストレージ Google Drive のインターフェースの一種と言って良い。学生に対しては、別途用意された教務システムを通じて Classroom のクラスのパスコードを通知して、各自クラスに参加してもらうという形式をとっている。慣れていない学生は、なかなかクラスに参加できない場合もあるので、教務システムの受講者名簿から学生のメールアドレスを取得し、直接、クラスに招待するというも行っている。

Classroom にはトピックと呼ばれる資料整理のための手段が用意されている。筆者の場合は、全 15 回の講義に対して、各回ごとにトピックを作成して、資料や課題をまとめている。また、「準備」や「IT 環境」、「NEWS」などの話題ごとのトピックを設けて、適宜、新入生が学習を進める上で有益と思われる情報なども公開している。特に、NEWS トピックで、講義に関連する様々なニュース記事を紹介しているが、学生の学習へのモチベーションを上げる上でも、評判が良いようである。

以前は、毎回の課題と A4 用紙 1 枚のみ持込可能にして期末試験を行っていたが、現在は、期末試験を行わず、毎回の課題と Google Form による小テストのみで成績評価を行っている。小テストは 2 週間程度の期限を設けており、教科書やノート、公開している資料等を見ながら回答しても良いので、全体の 30% 程度に抑えている。課題を通して、実際に自分の手を動かして考えることを重視している。

Classroom では「課題」として

- 生徒がファイルを閲覧できる
- 生徒がファイルを編集できる
- 各生徒にコピーを作成

の 3 種類から選んで、Colab ノートブック等のファイルを添付することができる。例えば sample01.ipynb という原本となる Colab ノートブックを添付し、「各生徒にコピーを

作成」を選択すると、各学生の登録された氏名ごとに「氏名 - sample01.ipynb」というファイルが作成され配布される。

原本となる Colab ノートブックのドキュメントセルには解説と、コード見本を掲載し、学生には空白のコードセルへの入力を促し、提出させることで、Classroom 内で容易に課題の確認や採点を行うことができる。Classroom と Colaboratory の親和性が高いことは、オンラインだけでなく、対面の講義においても非常に効果的なようである。

2.2 Markdown 記法

Colaboratory および Jupyter Notebook の利点として、Markdown 記法を採用している点が挙げられる。Markdown 記法は軽量マークアップ言語の一種であり、プレーンテキストから HTML や L^AT_EX 等のマークアップ言語を出力するための簡易記法である。様々な方言があるが、Colaboratory では、以下のようなルールで文書を作成できる。章は#、節は##で開始し、イタリック体は*、ボールド体は**で囲む。短いコードや命令等を入力したい場合は`で囲めば固定幅文字として表示される。また、改行を含むコードは`で囲むことで、ドキュメント内のコードを判別しやすくできる。なお、`の代わりに開始記号として`pythonを用いることで、Pythonの予約語を色分けして表示することができる。シンプルな文法は初心者にもわかりやすい。Colaboratory と Jupyter Notebook で若干異なる部分もあるが、両者ともに MathJax ライブラリを用いた L^AT_EX 記法をサポート (図 2) している点は、数式を交えた文書を作成する上で重要である。

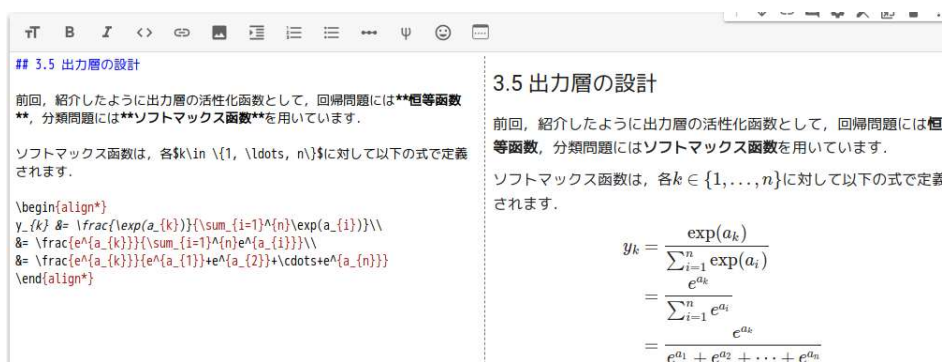


図 2: L^AT_EX 記法をサポート

2.3 Python ライブラリの利用

特にインストールなどの作業を行うことなく、Colaboratory は NumPy, Matplotlib, Pillow(PIL) などの Python の主要なライブラリを利用することができる。教科書として指定している書籍 [3] では、Anaconda と呼ばれる Python パッケージを利用しているが、Anaconda は非常に巨大なパッケージであるため、学生に対して開発環境のインストールを求めることは避けたかった。

数学的概念の可視化を行うため、教科書では扱っていないライブラリについても講義内で紹介している。微分方程式の数値解の求解に科学技術計算用の SciPy (図3), 2変数関数のグラフの対話的描画に Plotly (図4) 等を導入した。後者の2つについても特に追加のインストール作業を必要とせずに Colab ノートブック内で利用可能である。

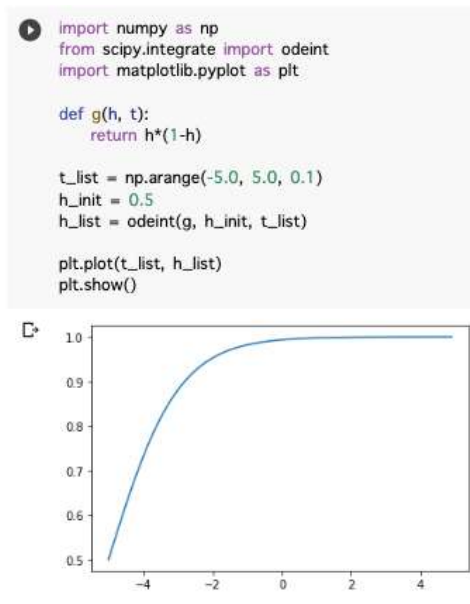


図 3: SciPy:odeint()

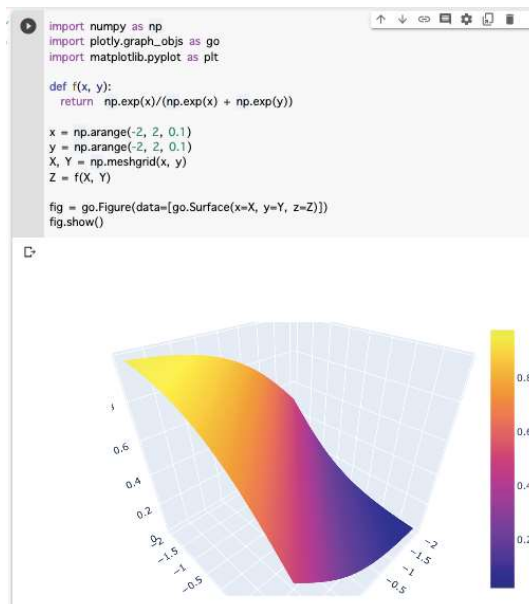


図 4: Softmax function

実際の講義では紹介していないが、数式処理ライブラリ SymPy の利用についても検討している。

2.4 Colab ノートブック内での GeoGebra の利用

Python ライブラリを活用した可視化は、プログラミング教育の観点からは有益な反面、数学の概念についてだけを伝えたい場合には、プログラミング言語やライブラリの利用方法の説明が必要であり、冗長に感じられることもある。そのような場合には、動的数学ソフトウェア GeoGebra の教材を利用している。現在、GeoGebra には Java アプリケーションやスマートフォンアプリなど、様々な利用形態が存在するが、Web アプリケーション <https://www.geogebra.org/> としても提供されている。ここでは、統計数理研究所で 2021 年 12 月に行われた「動的幾何学ソフトウェア GeoGebra の整備と普及」において、山形大学の脇克志氏から紹介された機能を利用している。IPython には display と呼ばれるモジュールがあり、IFrame を用いて HTML のインラインフレーム要素の表示を実現している。src で指定した URL の画面を Colab ノートブック内にインライン表示することができるので、GeoGebra 以外の Web サービスで提供されているコンテンツについても適用可能である。

もちろん、同様のものを Python を用いて実現することも可能だが、対話的な資料を作成する際には GeoGebra を用いることも十分検討に値する。なお、図5の描画については GeoGebra の SlopeField() と、SolveODE() 関数を用いている。

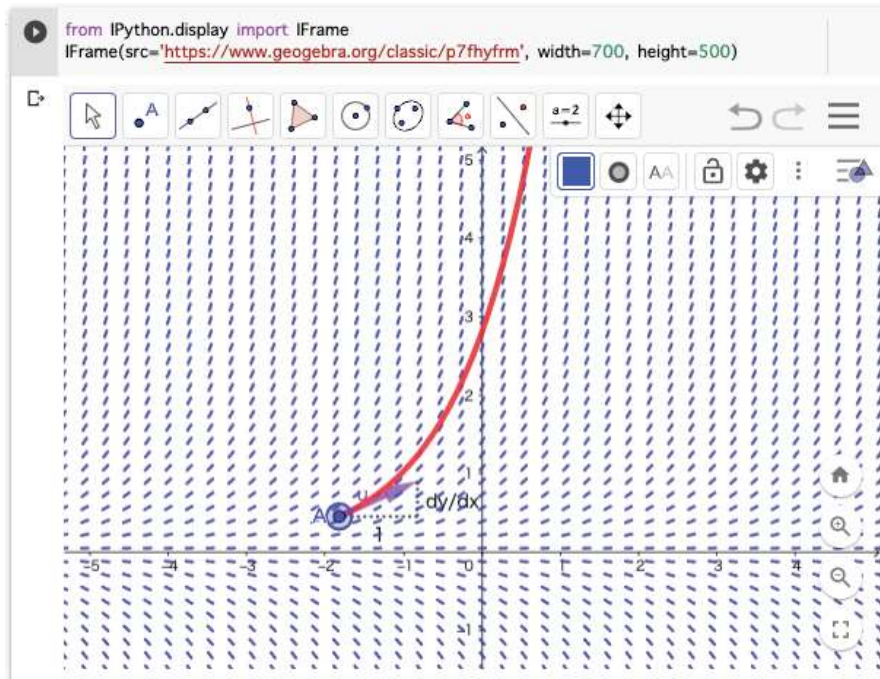


図 5: GeoGebra によるベクトル場と積分曲線の描画

3 数学リメディアルとして

本稿で取り上げている講義では、教科書で取り上げられる項目に関して、高校および大学で学ぶ数学との関連を明確に伝えるように努めた。以下は取り上げたトピックの一部である。

数学 I 「集合と命題」かつ、または、否定、「データの分析」平均値、分散

数学 A 「整数の性質の活用」 n 進法、

数学 II 「図形と方程式」不等式の表す領域、

数学 B 「ベクトル」内積、ベクトル方程式、「確率分布と統計的推測」正規分布、ネイピア数 e 、

数学 III 「式と曲線」、「極限」ネイピア数 e 、「微分法」有理関数、指数関数、合成関数、

線形代数 行列の和、差、積、アダマール積、アダマール商、

解析学 微分方程式、2変数関数

農学系、生命系の多くの他の私立大学と同様に、入試科目として数学 III を課していないため、合成関数の概念や、指数関数の微分積分に慣れていない学生が多い。詳しい調査は行っていないが、ネイピア数 e について聞いたことがないという学生がいることは無視できない。一般に、数学 III を履修していない学生は、数学 B の「確率分布と統計的推測」で無理数の定数として e を取り扱うが、実際には、多くの高校において「確率分布と統計的推測」の授業が実施されていないことも問題を複雑にしている。

4 まとめ

講義を通して、学生からは「線形代数を学ぶと就職活動の際に評価されますか?」, 「これは数学ですか? パソコンの使い方を教えて下さい」, 「ほかにも数学のこともわかり、自分が何をすべきかがわかってよかった。」などの質問や感想があった。

線形代数については必要性について何度も講義内で紹介しているためであろう。現状は、学部として線形代数が必修化されておらず、全体の3割程度しか履修者がいないという状況である。一方で、専門科目において線形代数の利用例は多く、線型回帰モデルや、微分方程式の求解、アミノ酸置換行列の計算等が挙げられる。残念ながら基礎的な数学を学ばない状態で、利用方法だけが紹介されるということが少なからずあるようで憂慮している。

また、コンピュータについて学ぶことをワードプロセッサや表計算等のアプリケーションの利用方法を学ぶことと誤解している方もいるが、数学やニューラルネットワークについて意欲的に学ぼうとしている学生も少なからずおり、この点については頼もしい限りである。

歴史的に、生物資源科学および農学は統計学を多用する分野であり、その重要性は十分に認識されているが、一方で、統計学を学ぶための基礎である線形代数や解析学、確率論などの重要性については、まだまだ伝えきれていないと感じている。動的ドキュメントを用いた課題解決型学習を通して、基礎的な数学概念の確認を進め、線形代数や解析学、統計学入門などの講義への接続となるよう期待している。

令和5年度からは、本稿で紹介している講義「ネットワーク入門」を廃止し、新たに「データサイエンスの基礎」として再編する予定である。これは入学者定員210名のバイオサイエンス学科の必修教養科目として位置づけられており、生物資源科学に関連する事例を積極的に取り組みつつ進めていきたいと考えている。

参考文献

- [1] 濱田龍義: ニューラルネットワーク作成による数学学習の動機付け, 京都大学数理解析研究所講究録 2105, pp.89–98, 2019
- [2] 高山信毅: Jupyter notebook への asir カーネルの実装 (実装報告), 数式処理, 27(1) <http://www.math.kobe-u.ac.jp/HOME/taka/2019/jupyter-jssac.pdf>
- [3] 斎藤康毅: ゼロから作る Deep Learning – Python で学ぶディープラーニングの理論と実装, O'reilly Japan, 2016