

## ChatGPT によるラグランジュ法の教材自動作成

学習院大学経済学部経営学科 白田 由香利  
和洋女子大学全学教育センター 桑名 杏奈

**内容：**経営数学教育で生成 AI を活用する事例として、ワークショップ(WS)を行い、自分のノート PC 上で ChatGPT に、教材となる演繹推論過程グラフを描画させます。演繹推論法による ChatGPT の活用については、ジャーナル論文[1]を参照してください。前半で、経営数学教育での ChatGPT の活用について報告いたします。ここでは、ラグランジュに未定乗数問題を使って ChatGPT に演繹推論過程グラフを作成させる過程について報告をします。後半は、WS を行います。無料版 ChatGPT でも正しく稼働するように、シンプルな数学問題を解かせます。以下に掲載したコマンドプロンプトをそのまま打ち込んで頂きます。ChatGPT は統計的 AI であるので、セッションごとに異なる対応をします。それにどう対処するかをWSで習得して頂きます。ChatGPT がコントロールできず、望むような演繹推論過程グラフを描画するプログラムを作成させることができなかつた場合でも、こちらで予め用意した Python+Graphviz プログラムを動かすことで、描画までの流れを体験できます。WSの目的は、会員の皆様に、ChatGPT による教材作成を体験して頂き、現場での教材作成コストを軽減することです。ChatGPT を活用することで、大量の高品質教材の作成を目指します。

**研究[1]の目的：**経営数学教育における生成 AI の活用を実践する。本論文は、経営数学教育における生成 AI の利用方法として、次の二点を提案する：**(1) 問題解法および学生向け可視化教材の自動生成**、**(2) 生成 AI を生徒役とすることで、教師の教授スキルを向上させること**。我々の研究チームは、ChatGPT を用いて上記の研究を進めている。ChatGPT はニューラルネットモデルに基づく AI であり、数学問題解法において誤りをおかすことがあるが、近年、記号処理的 AI の組み込みや Advanced Data Analysis といった拡張機能により、方程式の正確な解決が可能になっている。しかし、ChatGPT の経営数学の文章題に対する読解力は発展途上にあり、解法のための正しい演繹推論を行うことが困難な場合も少なくない。正しい演繹推論に到達すれば、ChatGPT はその正しい解法プランを可視化して提示することは可能である。こうした可視化教材の自動生成は、筆者が 10 年前にシステムを構築したが、生成 AI の登場により、システム構築をしなくても、容易に教材作成が可能になった。これは教育現場にとって大きな進展である。また、筆者が提案する生成 AI の活用法では、AI に教師の役割を担わせるのではなく、教師役はあくまで人間が果たす。本提案では、生成 AI が生徒役を演じ、それを相手に教育実習生などの人間教師が指導法のスキル向上を図ることを目標とする。本稿では、ChatGPT によって作成された可視化教材と具体的

な ChatGPT への指示/教授法を示し, その効果について探求する.

#### 参加者が用意する物:

- ノート PC. Python がインストールされたもの.
- WiFi. 足りないパイソンパッケージがあった場合, その場でインストールします.  
特に, graphviz のインストールには注意が必要です. Conda ではインストールがうまくいきません. "pip install graphviz"で, 私の場合成功しました. 私は conda と pip を併用したくなかったのですが, 仕方なく pip でインストールしました. そこは自己責任でインストールしてください.
- 無料の「Google Colaboratory」でも, 行えます. ブラウザ上で動くので, 無料の Google アカウントと WiFi だけあれば体験可能です. graphviz のインストールは不要です.
- 有料版 ChatGPT ライセンス. 無料ライセンス版でも体験可能です. ⇒問題文を入力して, 式を立てるところは, 無料版でもかなり正しくできます. 方程式を解くなどの計算部分については, Wolfram プラグインを使っても正解を得られないかもしれません.

#### 参考文献

- [1] 白田由香利:「経営数学教育における生成 AI の活用方法」, 日本経営数学会誌, Vol. 42, No.1-2, 2024 年 3 月.
- [2] Dowling, Edward T. *Mathematics for economists*. McGraw-Hill, 1980.

- 無料版 ChatGPT でできないこと：正確な計算ができない場合がある。
- Google Colab でできないこと：  
有向グラフ内の TeX 数式を整形できない (TeX ソースコードが表示されます)。

例 1 (距離を表す式から速度を求める問題)

■ChatGPT (無料版) での作業■

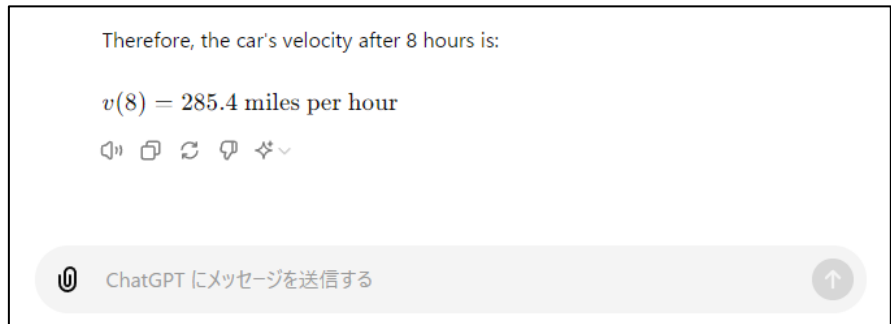
以下を ChatGPT にそのままコピペ

A racing car travels a distance of  $s(t)=12.7t^3 -143t^2+135t$  miles, where  $t$  represents the time in hours.  
Find the car's velocity after 8 hours.

Answer: 285.4 miles per hour

出典: David Ebner, "Math Word Problems The Easy Way," Barron's, Educational Series, Inc., 2002.

「 $s(t)$ の式を  $t$  で微分して、 $t$  に 8 を代入する」という手順を踏んで、以下の通り正しい答えを出力してくれると思います。



グラフを描く準備のため、以下を ChatGPT にそのままコピペします。

What is the given data and the unknown of this problem?  
Solve the problem step by step using deductive reasoning to find the unknown value.  
Please list the given data with the exact values and the equations that have appeared so far, using latex notation.  
Let's draw the deductive reasoning graph. Please write a program using Python and graphviz which summarize dependencies among equations and variables. That is the deductive reasoning graph.  
In the graph, the circle includes the equation or the variable with the exact value.

グラフを描くための Python プログラムを出力してくれます。これをコピーして Google Colab に貼り付けます。

## Python Program to Create Deductive Reasoning Graph:

Here's the Python program using `graphviz` to summarize the dependencies among equations and variables:

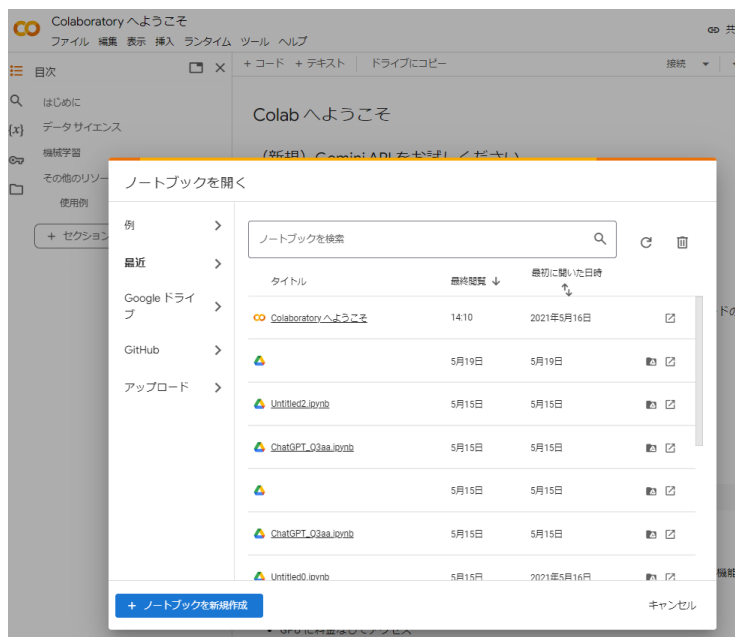
```
python 📄 コードをコピーする  
  
from graphviz import Digraph  
  
# Create a new directed graph  
dot = Digraph()  
  
# Define the nodes with their equations or values  
dot.node("s(t)", "s(t) = 12.7t^3 - 143t^2 + 135t")  
dot.node("v(t)", "v(t) = s'(t) = 38.1t^2 - 286t + 135")  
dot.node("t=8", "t = 8")  
dot.node("v(8)", "v(8) = 285.4")  
  
# Define the edges showing dependencies  
dot.edge("s(t)", "v(t)", label="differentiate")  
dot.edge("v(t)", "v(8)", label="substitute t = 8")  
dot.edge("t=8", "v(8)")  
  
# Render the graph  
dot.render("deductive_reasoning_graph", format="png", view=True)
```

### ■ Google Colaboratory での作業 ■

<https://colab.research.google.com/?hl=ja> に無料のアカウントでログイン。


graphviz は組み込まれているようですので、「pip install graphviz」をする必要はありません。

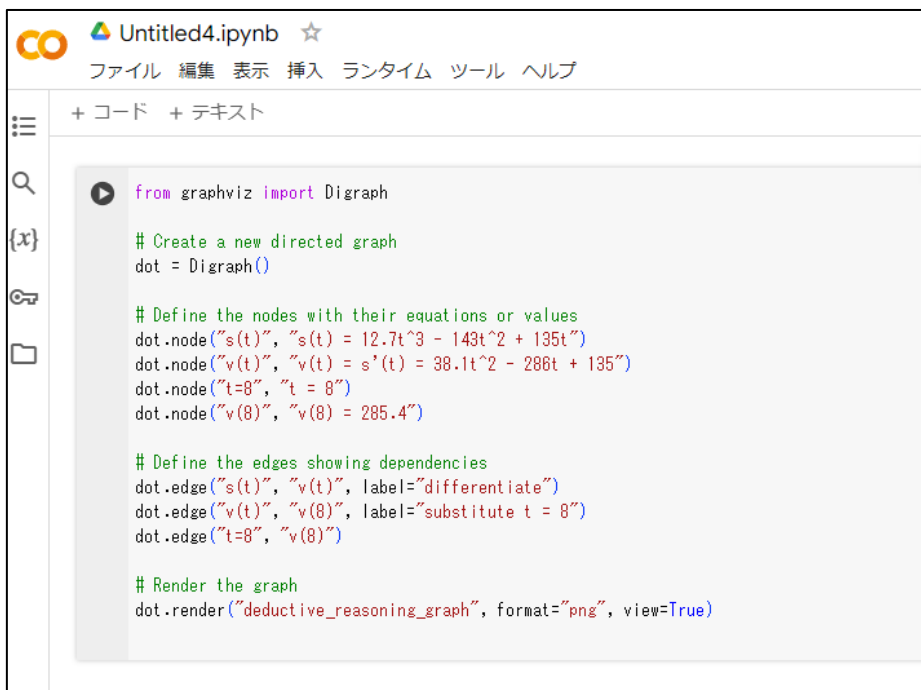
左下の「ノートブックを新規作成」をクリック



灰色背景の枠（セル）に、ChatGPT が作ってくれた Python プログラムをコピペします。  
もし灰色背景の枠が出ていなかったら、「+コード」というボタンをクリックしてください。



貼り付けできたら、セルの左にあるボタン  をクリックします。



実行はされますが、下部に「～.png」という画像のファイル名らしきものだけ出てきて、グラフは描けません。

```
▶ from graphviz import Digraph


# Create a new directed graph
dot = Digraph()

# Define the nodes with their equations or values
dot.node("s(t)", "s(t) = 12.7t^3 - 143t^2 + 135t")
dot.node("v(t)", "v(t) = s'(t) = 38.1t^2 - 286t + 135")
dot.node("t=8", "t = 8")
dot.node("v(8)", "v(8) = 285.4")

# Define the edges showing dependencies
dot.edge("s(t)", "v(t)", label="differentiate")
dot.edge("v(t)", "v(8)", label="substitute t = 8")
dot.edge("t=8", "v(8)")

# Render the graph
dot.render("deductive_reasoning_graph", format="png", view=True)
```

↔ 'deductive\_reasoning\_graph.png'

Python プログラム最下行に、一行、「dot」と書き足して、再度  をクリックすると、グラフが描けます。★ChatGPT に色々お願いしてみたのですが、最後の行に dot を書き足してもらえないでいました。仕方ないので Python に直接書き込みます。

```

from graphviz import Digraph

# Create a new directed graph
dot = Digraph()

# Define the nodes with their equations or values
dot.node("s(t)", "s(t) = 12.7t^3 - 143t^2 + 135t")
dot.node("v(t)", "v(t) = s'(t) = 38.1t^2 - 286t + 135")
dot.node("t=8", "t = 8")
dot.node("v(8)", "v(8) = 285.4")

# Define the edges showing dependencies
dot.edge("s(t)", "v(t)", label="differentiate")
dot.edge("v(t)", "v(8)", label="substitute t = 8")
dot.edge("t=8", "v(8)")

# Render the graph
dot.render("deductive_reasoning_graph", format="png", view=True)
dot

```

★有向グラフの中の数式は、「 $t^3$ 」などの形ではなく「 $t^3$ 」のままになっています。「dot2tex」パッケージを使えば LaTeX と組み合わせると「 $t^3$ 」などの形にできるそうですが、Google Colab 上でこの問題を解決する方法は、わかりませんでした。「 $t^3$ 」の形のまま進めてください。

★Google Colab の上部、「+テキスト」をクリックして出てきた枠に、「\$」記号で挟んだ TeX 数式をコピーすると、数式を整形することはできます。整形した数式をグラフ内部に書き戻す方法はわかりません。数式を確認するためにお使いください。

例 2 (利付債の PV (Present Value) を計算する)

以下を ChatGPT にそのままコピペ

Solve the problem step by step using deductive reasoning.  
An 8% coupon, 10-year maturity bond with par value of \$100 paying semiannual coupon payments of \$4 each. Suppose that the interest rate is 3% annually. Find the present value of the bond. Find the exact bond PV.

利率 :  $r = 1 + \frac{0.03}{2} = 1.015$

クーポン (全部で 20 回分) の PV の合計 :  $\frac{4}{r} + \frac{4}{r^2} + \frac{4}{r^3} + \dots + \frac{4}{r^{20}} = 68.67455514 \dots$  ①

額面償還分の PV :  $\frac{100}{r^{20}} = 74.24704182 \dots$  ②

最終解答 : ① + ② = 142.921597

Step に分けて計算してくれます。①の計算で少しだけ誤差が出ます。

4. Calculate the total present value of the bond:

$$PV = 69.27308 + 74.23418 = 143.50726$$

Therefore, the exact present value of the bond is:

143.51

This result shows the present value of the bond, considering semiannual coupon payments and the current market interest rate.

計算の誤差をなくすために、以下を ChatGPT に入力してみます。しかし、残念ながら無料版では改善されませんでした。計算は誤っていますが、考え方は正しく出力されているのでこのまま進めます。★場合によっては、計算を行う Python プログラムを出力してくれるかもしれません。ここでは無視して進めます。

The calculation is not correct. Please use the plugin for the calculation.

グラフを描く準備のため、以下を ChatGPT にそのままコピペ

What is the given data and the unknown of this problem?  
Solve the problem step by step using deductive reasoning to find the unknown value.  
Please list the given data with the exact values and the equations that have appeared so far, using latex notation.  
Let's draw the deductive reasoning graph. Please write a program using Python and graphviz which summarize dependencies among equations and variables. That is the deductive reasoning graph.  
In the graph, the circle includes the equation or the variable with the exact value.

グラフを描くための Python プログラムを出力してくれます。




Here's the Python code to create the deductive reasoning graph:

```
python 📄 コードをコピーする  
  
from graphviz import Digraph  
  
# Create a new directed graph  
dot = Digraph()  
  
# Define the nodes with their equations or values  
dot.node("r_annual", "r_{\text{annual}} = 3\%")  
dot.node("r_semiannual", "r_{\text{semiannual}} = 1.5\%")  
dot.node("T", "T = 10")  
dot.node("n", "n = 20")  
dot.node("PV_coupons", "PV_{\text{coupons}} = C \times \left(1 - \frac{1}{(1 + r_{\text{semiannual}})^n}\right)")  
dot.node("PV_par", "PV_{\text{par}} = \frac{F}{(1 + r_{\text{semiannual}})^n}")  
dot.node("PV_bond", "PV_{\text{bond}} = PV_{\text{coupons}} + PV_{\text{par}}")
```

### ■ Google Colaboratory での作業 ■

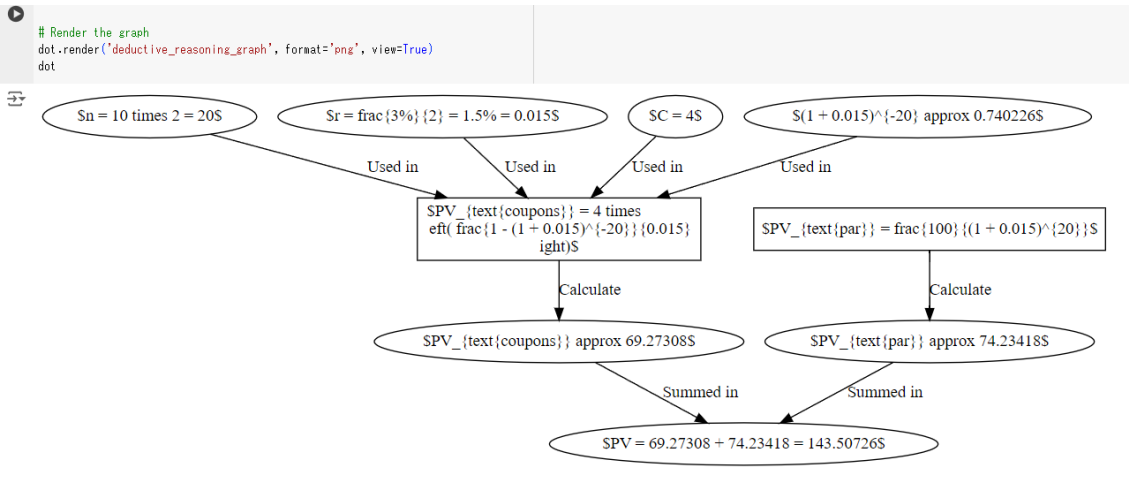
「+コード」というボタンをクリックしてください。新しく灰色背景の枠（セル）が出てきます。ChatGPT が作ってくれた Python プログラムをコピペします。



例 1 と同様、最下行に `dot` と一言追記して、セルの左にあるボタン  をクリックします。

```
# Render the graph  
dot.render("deductive_reasoning_graph", format="png", view=True)  
dot
```

数式が整形されないので見にくいですが、考え方は問題なくグラフ化されます。



以上