

# “STACK 入門から徹底活用まで” ～ STACK を使った多肢選択問題の活用 ～

吉富賢太郎（大阪公立大学）

MSFD 39 © 東京理科大学 1号館 3F 136 教室  
2026.03.22(日) 14:00～

- ① STACK を local で動かそう
- ② AI の影響
- ③ STACK を使った問題設計と多肢選択問題
- ④ まとめ

- 検索のコツ ... “STACK” “eassessment”  
<https://stack-assessment.org>  
ドキュメント充実  
<https://docs.stack-assessment.org/en/>  
日本語化 (/ja/) ... 名古屋大学の中村研  
ただし、本当の”document” はソースコードにあり (笑)
- github stack math で検索 <https://github.com/math>  
STACK, goemaxima 等の DL はココ!

# STACKのメリットデメリット

- メリット 1 : 即時フィードバック (重要)  
→ 学修にはフィードバックが大事 (紙だと 1 週間かかる)
- メリット 2 : 自動採点 (あまり本質じゃない)  
→ オンラインテスト利用 (紙でやって採点は自動とかも) は便利
- デメリット 1 : コンテンツ開発に慣れるまでが?  
→ 大分解消してきた (サンプルとかが充実・既存を複製して改造が楽)  
<https://xml.mathedu.jp/>  
[https://stack.mathedu.jp/sc/ky\\_linear\\_algebra.txt](https://stack.mathedu.jp/sc/ky_linear_algebra.txt)
- デメリット 2 : 重い  
→ goemaxima で maxima の利用を軽く  
100 人規模の同時使用程度・全学課題利用等はほぼ問題なし (?)

# STACKのメリットデメリット

- メリット 1 : 即時フィードバック (重要)  
→ 学修にはフィードバックが大事 (紙だと 1 週間かかる)
- メリット 2 : 自動採点 (あまり本質じゃない)  
→ オンラインテスト利用 (紙でやって採点は自動とかも) は便利
- デメリット 1 : コンテンツ開発に慣れるまでが?  
→ 大分解消してきた (サンプルとかが充実・既存を複製して改造が楽)  
<https://xml.mathedu.jp/>  
[https://stack.mathedu.jp/sc/ky\\_linear\\_algebra.txt](https://stack.mathedu.jp/sc/ky_linear_algebra.txt)
- デメリット 2 : 重い  
→ goemaxima で maxima の利用を軽く  
100 人規模の同時使用程度・全学課題利用等はほぼ問題なし(?)

# STACKのメリットデメリット

- メリット 1 : 即時フィードバック (重要)  
→ 学修にはフィードバックが大事 (紙だと 1 週間かかる)
- メリット 2 : 自動採点 (あまり本質じゃない)  
→ オンラインテスト利用 (紙でやって採点は自動とかも) は便利
- デメリット 1 : コンテンツ開発に慣れるまでが?  
→ 大分解消してきた (サンプルとかが充実・既存を複製して改造が楽)  
`https://xml.mathedu.jp/`  
`https://stack.mathedu.jp/sc/ky\_linear\_algebra.txt`
- デメリット 2 : 重い  
→ goemaxima で maxima の利用を軽く  
100 人規模の同時使用程度・全学課題利用等はほぼ問題なし (?)

# STACKのメリットデメリット

- メリット 1 : 即時フィードバック (重要)  
→ 学修にはフィードバックが大事 (紙だと 1 週間かかる)
- メリット 2 : 自動採点 (あまり本質じゃない)  
→ オンラインテスト利用 (紙でやって採点は自動とかも) は便利
- デメリット 1 : コンテンツ開発に慣れるまでが?  
→ 大分解消してきた (サンプルとかが充実・既存を複製して改造が楽)  
<https://xml.mathedu.jp/>  
[https://stack.mathedu.jp/sc/ky\\_linear\\_algebra.txt](https://stack.mathedu.jp/sc/ky_linear_algebra.txt)
- デメリット 2 : 重い  
→ goemaxima で maxima の利用を軽く  
100 人規模の同時使用程度・全学課題利用等はほぼ問題なし (?)

## ★ 基本環境の導入 (http:8088)

- ・ Mac: homebrew 等で webserver, php, mysql, maxima, (git) 等を入れる.
- ・ Linux: 容易 ・ Windows: Linux を入れて同様 ww

※ ただし, 今回はちょっと失敗して動かなくなりました (汗)

## ★ Moodle の導入

Moodle を DL → moodle.org のサイトにしたがって導入

5.1 はちょっとセキュリティがらみで設定変更が必要

## ★ STACK の導入

STACK: github サイト → git clone

question plugin として Moodle に認識させ INSTALL

goemaxima を Docker で動かす (8080)

# localhost での導入

## ★ 基本環境の導入 (http:8088)

- ・ Mac: homebrew 等で webserver, php, mysql, maxima, (git) 等を入れる.
- ・ Linux: 容易
- ・ Windows: Linux を入れて同様 ww

※ ただし, 今回はちょっと失敗して動かなくなりました (汗)

## ★ Moodle の導入

Moodle を DL → moodle.org のサイトにしたがって導入

5.1 はちょっとセキュリティがらみで設定変更が必要

## ★ STACK の導入

STACK: github サイト → git clone

question plugin として Moodle に認識させ INSTALL

goemaxima を Docker で動かす (8080)

## ★ 基本環境の導入 (http:8088)

- ・ Mac: homebrew 等で webserver, php, mysql, maxima, (git) 等を入れる.
- ・ Linux: 容易
- ・ Windows: Linux を入れて同様 ww

※ ただし, 今回はちょっと失敗して動かなくなりました (汗)

## ★ Moodle の導入

Moodle を DL → moodle.org のサイトにしたがって導入

5.1 はちょっとセキュリティがらみで設定変更が必要

## ★ STACK の導入

STACK: github サイト → git clone

question plugin として Moodle に認識させ INSTALL

goemaxima を Docker で動かす (8080)

## ★ オンライン自動フィードバック (採点) システムの基本構造

- 問題変数 ... 問題内容を制御・記述するための変数定義  
default で 80 以上の定義済み変数  
user 変数を乱数を使って設定
- 問題文 ... 問題文と解答欄のデザイン.  
... 一問一答式の単純計算, 多肢選択, 問題を読みながら穴埋め, 何でも  
... 問題文自体 (解答欄以外) の内容を問題変数で定義しておくこともできる.
- 解答判定とフィードバック  
... フィードバック: 学習利用としては最も重要  
... **PRT** (二分木による解答判定) の各ノードでフィードバックと採点

## ★ オンライン自動フィードバック (採点) システムの基本構造

- 問題変数 ... 問題内容を制御・記述するための変数定義  
default で 80 以上の定義済み変数  
user 変数を乱数を使って設定
- 問題文 ... 問題文と解答欄のデザイン.  
... 一問一答式の単純計算, 多肢選択, 問題を読みながら穴埋め, 何でも  
... 問題文自体 (解答欄以外) の内容を問題変数で定義しておくこともできる.
- 解答判定とフィードバック  
... フィードバック: 学習利用としては最も重要  
... PRT (二分木による解答判定) の各ノードでフィードバックと採点

## ★ オンライン自動フィードバック (採点) システムの基本構造

- 問題変数 ... 問題内容を制御・記述するための変数定義  
default で 80 以上の定義済み変数  
user 変数を乱数を使って設定
- 問題文 ... 問題文と解答欄のデザイン.  
... 一問一答式の単純計算, 多肢選択, 問題を読みながら穴埋め, 何でも  
... 問題文自体 (解答欄以外) の内容を問題変数で定義しておくこともできる.
- 解答判定とフィードバック  
... フィードバック: 学習利用としては最も重要  
... PRT (二分木による解答判定) の各ノードでフィードバックと採点

## ★ オンライン自動フィードバック (採点) システムの基本構造

- 問題変数 ... 問題内容を制御・記述するための変数定義  
default で 80 以上の定義済み変数  
user 変数を乱数を使って設定
- 問題文 ... 問題文と解答欄のデザイン.  
... 一問一答式の単純計算, 多肢選択, 問題を読みながら穴埋め, 何でも  
... 問題文自体 (解答欄以外) の内容を問題変数で定義しておくこともできる.
- 解答判定とフィードバック  
... フィードバック: 学習利用としては最も重要  
... **PRT** (二分木による解答判定) の各ノードでフィードバックと採点

★ 多肢選択問題 (MCQ) は数学教育において無用か？

→ 適当に作れば無用 vs 適切に創りこめば有用

∴ 多肢選択問題は、学生の誤答に関する知見の集合

★ 適切なフィードバックを備え、

★ 適切に誤答が構成された MCQ

→ 学習や評価に有効 (のはず)

メリット： 学生の入力が容易 (入力 of 知識が不要)。

デメリット： 教員の教育経験が必要。問題開発も手間がかかる。

→ テンプレートを後で詳解します

# 多肢選択問題

★ 多肢選択問題 (MCQ) は数学教育において無用か？

→ 適当に作れば無用 vs 適切に創りこめば有用

∴ 多肢選択問題は、学生の誤答に関する知見の集合

★ 適切なフィードバックを備え、

★ 適切に誤答が構成された MCQ

→ 学習や評価に有効 (のはず)

メリット： 学生の入力が容易 (入力 of 知識が不要) .

デメリット： 教員の教育経験が必要. 問題開発も手間がかかる.

→ テンプレートを後で詳解します

# 多肢選択問題

★ 多肢選択問題 (MCQ) は数学教育において無用か？

→ 適当に作れば無用 vs 適切に創りこめば有用

∴ 多肢選択問題は、学生の誤答に関する知見の集合

★ 適切なフィードバックを備え、

★ 適切に誤答が構成された MCQ

→ 学習や評価に有効 (のはず)

メリット： 学生の入力が容易 (入力 of 知識が不要)。

デメリット： 教員の教育経験が必要。 問題開発も手間がかかる。

→ テンプレートを後で詳解します

# 多肢選択問題

★ 多肢選択問題 (MCQ) は数学教育において無用か？

→ 適当に作れば無用 vs 適切に創りこめば有用

∴ 多肢選択問題は、学生の誤答に関する知見の集合

★ 適切なフィードバックを備え、

★ 適切に誤答が構成された MCQ

→ 学習や評価に有効 (のはず)

メリット： 学生の入力が容易 (入力 of 知識が不要).

デメリット： 教員の教育経験が必要. 問題開発も手間がかかる.

→ テンプレートを後で詳解します

## 形成的評価 (Formative Assessment)

- 学習の**途中**で実施
- 目的：学習の**改善**・理解の深化
- フィードバックが中心
- 誤りや思考過程を重視

## 成績評価 (Summative Assessment)

- 学習の**終了時**に実施 (and/or 授業時に行う小テスト)
- 目的：到達度の**判定**・認定
- 点数・評価が中心
- 比較可能性・公平性を重視

# AIには全部解けてしまうか？

大学初年次数学レベル ... ほぼ正解する.

ただし, 解説は教員の意図通りとは限らない.

解けない例 1. 行列式が 89 の 3 桁 (以上) の整数成分の 2 次行列の例を答えなさい.

出題意図: 大きい成分の 2 次行列式の計算の逆を考えさせる.

→ AI(Gemini, ChatGPT 共通) ... 行列式が 89 でないものを答えたり,  
挙句の果てには, 「小数も可能なら」とか言い出す.

→ 理論的には, 問い詰めれば気付くが, それでも, できない

解けない例 2. 今年の某公立大学の入試問題のある問題 ... 典型的な誤答で満点でない

- **Bloom (1969)** : Mastery Learning
- **Black & Wiliam (1998)** : フィードバックが学習成果を大きく改善
- **Hattie (2009)** : フィードバックは効果量が多い

## 形成的評価の本質

- 現在の理解状態の可視化
- 次の学習行動への橋渡し
- 学習者のメタ認知の促進

→ AI へのプロンプト次第で適切なフィードバックの可能性

→ 学生に AI と対話させることでメタ認知が促進する可能性

- **Bloom (1969)** : Mastery Learning
- **Black & Wiliam (1998)** : フィードバックが学習成果を大きく改善
- **Hattie (2009)** : フィードバックは効果量が多い

## 形成的評価の本質

- 現在の理解状態の可視化
- 次の学習行動への橋渡し
- 学習者のメタ認知の促進

AI は協力的で強力なツール

- AI へのプロンプト次第で適切なフィードバックの可能性
- 学生に AI と対話させることでメタ認知が促進する可能性

★ 主な役割 ... 単位認定・成績付与・進級／卒業判定

## 重視される観点

- 信頼性 (reliability)
- 妥当性 (validity)
- 公平性 (fairness)

→ AIに任せる(採点させる)のはまだ**無理**

→ 従来のオンラインテストはほとんど**無効**

★ 主な役割 ... 単位認定・成績付与・進級／卒業判定

## 重視される観点

- 信頼性 (reliability)
- 妥当性 (validity)
- 公平性 (fairness)

AIは成績評価の敵!?(T T)

→ AIに任せる(採点させる)のはまだ**無理**

→ 従来のオンラインテストはほとんど**無効**

オンライン問題を利用した

- ・ 形成的評価 (オンライン演習)
- ・ 成績評価 (オンラインテスト・紙で配布+オンライン採点)

→ オンラインテストや課題としてのオンライン演習は AI の影響を受ける

→ AI は活用しつつ, 内省に利用させる等

- 途中の計算過程を入力させるもの [自己学習 (形成)・課題 (成績)]  
例. ガウス消去法や行列式の計算において, 過程で現れた行列を入力してもらう  
AI → 課題としての利用は困難
- 文脈を読み取り自分で考える問題 (計算はツール利用可) [課題 (成績)]  
AI → 課題としての利用は困難
- 概念理解が必要な多肢選択問題 [オンラインテスト (成績)・対面演習 (形成)]  
AI → オンラインテストとしての利用は (そのままでは) 困難

Q. MCQ を時間制限つきにすればどうか?

- 途中の計算過程を入力させるもの [自己学習 (形成)・課題 (成績)]  
例. ガウス消去法や行列式の計算において, 過程で現れた行列を入力してもらう  
AI → 課題としての利用は困難
- 文脈を読み取り自分で考える問題 (計算はツール利用可) [課題 (成績)]  
AI → 課題としての利用は困難
- 概念理解が必要な多肢選択問題 [オンラインテスト (成績)・対面演習 (形成)]  
AI → オンラインテストとしての利用は (そのままでは) 困難

Q. MCQ を時間制限つきにすればどうか?

- 途中の計算過程を入力させるもの [自己学習 (形成)・課題 (成績)]  
例. ガウス消去法や行列式の計算において, 過程で現れた行列を入力してもらう  
AI → 課題としての利用は困難
- 文脈を読み取り自分で考える問題 (計算はツール利用可) [課題 (成績)]  
AI → 課題としての利用は困難
- 概念理解が必要な多肢選択問題 [オンラインテスト (成績)・対面演習 (形成)]  
AI → オンラインテストとしての利用は (そのままでは) 困難

Q. MCQ を時間制限つきにすればどうか?

- 途中の計算過程を入力させるもの [自己学習 (形成)・課題 (成績)]  
例. ガウス消去法や行列式の計算において, 過程で現れた行列を入力してもらう  
AI → 課題としての利用は困難
- 文脈を読み取り自分で考える問題 (計算はツール利用可) [課題 (成績)]  
AI → 課題としての利用は困難
- 概念理解が必要な多肢選択問題 [オンラインテスト (成績)・対面演習 (形成)]  
AI → オンラインテストとしての利用は (そのままでは) 困難

Q. MCQ を時間制限つきにすればどうか?

# 多肢選択問題の実装

	Moodle 標準	STACK(標準)	STACK(裏技?)
ランダム化	× <sup>(*1)</sup>	○	○
フィードバック	○	×	○ <sup>(*2)</sup>
部分点	△	×	○ <sup>(*2), (*3)</sup>

(\*1) 大量自動生成とランダム出題を組み合わせる方法がある (長坂, 吉富)

(\*2) [吉富 ATCM2022]

(\*3) 部分点は, PRT の得点に変数を使えるようになって簡単にできるようになった.

# STACK による MCQ(標準版と裏技版)

標準版:

STACK の多肢選択問題は、解答欄 (解答変数) の「解答例」 (Teacher's answer) として、選択肢のリストを渡す。以下に例を示す:

例. ta1:[[1, false, “誤答選択肢 1”], [2, true, “正解選択肢”],  
[3, false, “誤答選択肢 2”], ...];

STACK のドキュメントによる正解判定の方法: mcq\_\_correct 関数を使う。

mcq\_\_correct(ta1) はラジオボタンの場合は 2, チェックボックス (正解が 2 つ以上の場合あり) は [2] というリストを返す

→ ans1 と mcq\_\_correct の出力の比較で正誤判定

注: ta1 として [[“選択肢 1”, true], “選択肢 2”, false]] のような形式も使うことができる (ここでは用いない).

# STACK による MCQ(標準版と裏技版)

裏技版:(ta1 の形式と ans1 に入る値の構造を利用する)

ta1: として, 正解のパターンも複数ある場合も含め:

```
[[1, true, "正解パターン 1-1"],[2, true, "正解パターン 1-2"], [101, true, "正解パ  
ターン 2-1"],[102, true, "正解パターン 2-2"], ..., [1001, false, "誤答パターン 1-1"],  
[1002, false, "誤答パターン 1-2"], ... [2001, false, "誤答パターン 2-1"], [2002,  
false, "誤答パターン 2-2"], ...,]
```

のようナリストを用意する:

→ 選択したオプションは [101, 2002] のようナリストで格納される

→ どのパターンから選んだかが判別可能→ フィードバック・部分点を設定 (→ PRT)

- `stack_include()` で自作ライブラリや問題テンプレートの活用.
- `maxima` の `values()` 関数ので、何が定義されていたか確認できる.
- 問題変数で言語を参照可能 (`%_STACK_LANG`) → 多言語化も問題変数で制御  
(`stack_include()` と併用で)
- `parsons` を使うと順序並び換えや Drag & Drop の数式対応版として使える.
- `[[reveal]]` block や `[[if]]` block により JS を簡易的に利用可能

# 具体的な問題例・実践例

デモ

- stack\_include を用いたライブラリの読み込みについて説明した.
- stack\_include を使うことでMCQなどの定型化ができることを説明した.
- 計算負荷のない or 少ない多肢選択問題 (MCQ)10~20 題で構成された小テストを適切な時間制限で実施することで, 学生への概念定着や誤概念検出を試みる手法について説明した (効果については, 検証継続中).
- 他にも, 簡単な数値入力で解答できる問題などを用いて上記のような小テスト構成を用いれば, 成績評価としても一定の効果が期待される.

※ 時間制限 MCQ によるテストに学生が AI でチート対抗できる可能性について  
→ AI 自身の分析ではまだ当分は大丈夫そう? (長坂氏)

- stack\_include を用いたライブラリの読み込みについて説明した.
- stack\_include を使うことでMCQなどの定型化ができることを説明した.
- 計算負荷のない or 少ない多肢選択問題 (MCQ)10~20 題で構成された小テストを適切な時間制限で実施することで, 学生への概念定着や誤概念検出を試みる手法について説明した (効果については, 検証継続中).
- 他にも, 簡単な数値入力で解答できる問題などを用いて上記のような小テスト構成を用いれば, 成績評価としても一定の効果が期待される.

※ 時間制限 MCQ によるテストに学生が AI でチート対抗できる可能性について  
→ AI 自身の分析ではまだ当分は大丈夫そう? (長坂氏)

告知: 数式処理学会 第17期第3回 教育分科会



<https://www.jssac.org/Education/Conf/edu202603.html>



懇親会参加の方は **3/23(明日)** までに登録をお願いします

※ 鶴橋で焼き肉の予定！