数理解析研究所講究録 第2105巻 2019年 52-58

数式処理を sympy で教えてみた (涙)

関西学院大学・理工学部　西谷　滋人
Shigeto R. Nishitani, Department of Informatics, Kwansei Gakuin University

1　はじめに

2004 年より関西学院大学理工学部情報科学科の 3 回生に対して，Maple を使って数式処理演習を行なってきました．そこでの演習形態の特徴は，アクティブラーニングの一形態となる**反転授業** [1] です．あらかじめ，作成したテキストを配布しておき，学生が自身の PC に install しているソフトを使って予習してきてもらいます．そして，演習中にはそれと関連する課題を仕上げます．さらにその作業の進め方は，**ペア作業から，ペア試験，さらにペア評価**へと変遷してきました．これらの内容はすでに RIMS 研究集会で報告しており, また報告書に記載しているので，そちらを参照ください [2, 3, 4]．アンケート結果を見る限り，学生の反応はいいのですが，問題は，このような手間暇かけている割に，スキルの定着が低いという現実でした．そこで，本年度は使用言語を python にかえて，同じテキスト内容，演習方式，試験方式，試験レベルで行い，比較しました．

最終的なテストの成績で比較すると表 1 の通りであり，pair 方式のテストで 7 点，個別 personal 試験で 15 点の減少という結果でした．演習内容，試験方式，試験レベルについて詳しく述べ，学生からの訴えについて考察を加えます．

表 1: 17,18 年度の各試験の平均点．

year	pair	personal
2017	77.41	63.05
2018	70.54	48.54

2　演習内容

テキストは一般公開するため，github にあげて [5]，nbviewer から見れるように設定しています [6]．これらは，長年改定を加えてきた Maple のテキストを python の sympy および scipy(numpy をふくむ) ライブラリを利用するように command を変更しました．

学生 PC は関西学院大学が提供する windows7 で動く PC に anaconda, jupyter notebook を install し FireFox で動くようにしています．jupyter notebook の使用法は少し慣れが必要ですが，web browser の通常の操作であり，GUI(graphical user interface) での直観的な入力・編集が可能であるため，学生には馴染みやすいようでした．ただし，

- IE で jupyter notebook が反応しない (browser の再起動)

- 起動時の home directory がどこかわからない
- printout の一部が欠ける (IE のバグ，FireFox で解決)
- init_session() が効かない (パッチが出ていたが，適用稼働に自信がないため見送り，再起動で対応)

などの些細な問題が当初ありましたが，概ね順調に授業は進行しました．対応策を括弧内に示しています．授業の進め方は，

- テキストの単元を指定して，次の週までに予習
- 固定したペアで課題を解き，
- ペアで一つのレポートを提出．

です．

3　試験形式と内容

試験はペアで自由に相談ができるペア試験を中間試験とし，一切の相談ができない個別 personal 試験を最終試験として課しました．

表 2: 各年度の試験内容とその配点．

年度	内容 (配点)	平均点
2012	微分 (10+10)+積分 (10+10)+線形代数 (10+10)+センター (20+20)	86
2013	微積分 (10+15)+線形代数 (10+15)+センター (25+25)	71
2014	微分 (10+10)+積分 (10+10)+線形代数 (10+10)+センター (40)	69
2015	微分 (15+10)+積分 (15+10)+線形代数 (15+10)+センター (5+20)	80
2016	微積分 (15+15)+線形代数 (15+15)+センター (10+30)	57
2017	微積分 (15+15)+線形代数 (15+15)+センター (10+30)	62
2018	微積分 (15+15)+線形代数 (15+15)+センター (10+30)	49

内容と配点は表 2 の通りです．センターと書かれたのは，センター試験の数 IA あるいは IIB の穴埋め問題をそのまま使った問題です．2012，13 年度は 2 問ずつ出していました．しかし，2014 年度にその一部を簡単には解けない数値に変更した問題を試しました．その後，2015 年度以降は，センター試験のままと改変した問題の 2 問を出し，その配点を示してます．今年度は，そのままの問題は「2015 年度大学入試センター試験　追試　数学 II・B 第 2 問 (2) 」を使い，その一部を変更しました．変更前は，

> 座標平面上の放物線 $y = 1 - x^2$ を C とする．
>
> $\frac{1}{2} < b \leq 1$ として，放物線 C 上の 2 点 Q(-1,0) と R$(1-b, 2b-b^2)$ を通る直線を m とする．

という冒頭の箇所に対して，

> 前問の放物線 C の方程式を $y = 1 - 0.5x^2$ として問題を解け．放物線 C 上の 2 点は $\mathrm{Q}(-\sqrt{2}, 0)$ と $\mathrm{R}(\sqrt{2} - b, 1 - (\sqrt{2} - b)^2)$ と読み替えよ．また，S_2 を求めるときの範囲は $\sqrt{2} - b \leq x \leq b$ と読み替えよ．また，数値解となるので，答えはかっこによらず小数点となる．

としています．これによって，

- センター試験のままの問題で解答を確認しながら，
- 数式処理のスクリプトを書き，
- 変更を加えた未知の問題に適用しながら，
- 汎用性の高いスクリプトに仕上げる

という，数式処理で必要な試行錯誤のスキルを身につけることを狙っています．

4　試験結果

図 1 に 12-17 年度の最終個別試験の結果を示しました．ペア試験，評価を始めた当初は，非常に高い平均点でしたが，より実践的なスキルが必要となるセンター試験改変問題を 2014 年に導入して以降ここ数年は，ある程度のところで落ち着いていました．点数分布を見ても，2012-15 にかけては，ほぼ全員がスキルを身につけていると判断できる，高得点に分布が偏る指数関数的な分布です．センター試験改変問題は，難しいですが，数式処理のスキルを最も評価できます．そこでその配点を増やしたこの 2 年間は 40-70 点にピークがくる正規分布的な得点分布となっています．

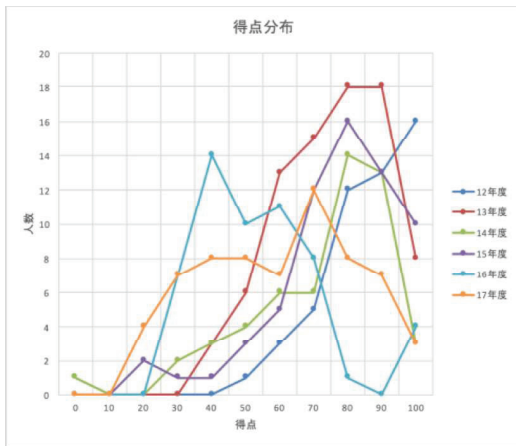

図 1: 12-17 年度の最終個別試験の結果．

これらの結果から，問題の難易度はある程度落ち着いていると判断しています．次に，17 年と 18 年だけを比較すると図 2 のようになります．ペア試験の結果では，それほど顕著な差が認められません．ところが，最終個別 personal 試験では，大きな差が認められます．特に，17 年度にはいなかった 0 点，10 点台が極端に増えています．

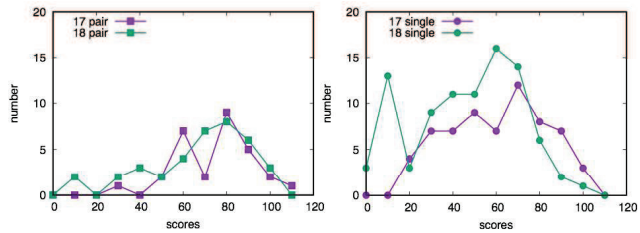

図 2: 17,18 年度のペア，個別試験の得点分布．

そこで 30 点以下が出てきたペア (一部スリーマンセル) の傾向を分類すると，表 3 の通りになります．

表 3: ペア (一部スリーマンセル) による得点の変化と傾向の分類．

group	pair	personal	group	pair	personal	group	pair	personal
A	85	75 0 30	B	15	55 15	C	70	10 10
A	65	75 50 10	B	40	0 40	C	90	10 50
A	65	70 10	B	15	45 55 15	C	75	50 10 45
A	85	20 90 60	B	35	30 65	C	90	30 10 15
A	95	30 75 60				C	80	10 10
A	55	60 30						
A	70	25 30 70				D	35	15 0
A	95	30 25 75						
A	60	60 30						

A group できるのと，できないのが組んで，できないのが自覚できてなかった．

B group できてなくて，頑張ったんやけど，片方がダメやった．

C group できてて，油断した

D group できてなくて，二人とも諦めた

と分類できます．

ペア試験と個別試験の結果の相関図をとると図 3 のようになります．17 年と 18 年で傾向は変わりませんが，全体的に左下が増え，ペア試験結果の点数にかかわらず，個別試験の点数が悪い学生が増えています．

平均点の減少は pair で 7 点，personal で 15 点でしたが，その原因としては，やる気・python・問題の難易度がその候補として，すぐにあげられます．もう少し裏読みすると，

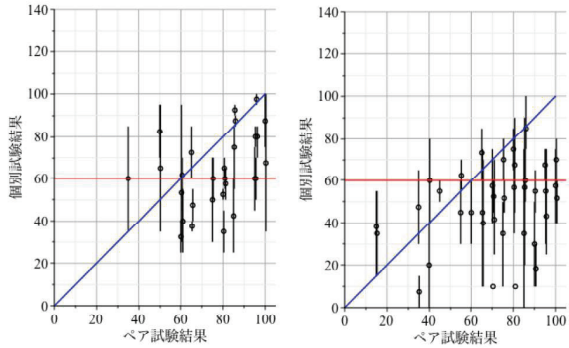

図 3: 17,18 年度のペア，個別試験の相関図.

- ペアの点数が以外と良かったので，甘く見た
- 問題に，コマンド一発で解ける問題がなかった.
- pair 試験から，personal まで通常一月とるが，今回は 2 週間しかなかった.
- 前日がポーランド戦やった

なども考えられます．確定したことは言えず，もう少し試行を重ねて，数年間での平均や動向の様子を見る必要があるでしょう.

5　学生アンケート

一方，授業終了後のアンケートには顕著な傾向が見られました．python そのものに対する肯定・否定が明瞭であった人数を出すと否定 5 に対して，肯定は 24 でした．否定的な意見では，

- python を使うのをやめたほうがいいと思います。maple に戻したほうがいいです。数式処理実習を行う上で今回のアプリケーションは二重積分すらもしっかり解いてくれないものであったので計算処理をさせる授業を行っているのにあまりにも脆弱ではないのかと思うからです。過去の演習問題の答えを見る限り maple では、そのあたりのことはしっかりしていたのでちゃんと先生がこのソフトを使って 1 セメスターの間生徒にシステム上の疑問は起こさないように、たとえ起きてもすぐ対応できるようなソフトを使ってほしいと思います。
- 演習全般については、プログラミングが苦手な私にとってはパイソンが少し難しく感じた。過去の試験の解答が Maple の答えしかなかったのでパイソンではどうすればよいのかわからない問題も少しあった。
- Python は理解するのが難しかったが、ペアワークによって、その難しさよりも、授業の楽しさの方が上回った。

などです．それに対して肯定的な意見は，

- python に少しでも関われてよかった。
- python という時代に即した言語を学習することができてとても良かった。また、教材が GitHub で管理されていることから、直接修正依頼を送ることができ他の授業と比べてスムーズにミスの修正をしていただくことができた。
- はじめて python について学んだがとりあえずコマンドがとても多いと思った。勉強しても新しいものがどんどんでてきてコマンドがないと実行できない操作が多くこの点が c 言語と大きく違っていると感じた。

でした．特に python が使えるだけで嬉しいとか,「最新の言語」,「AI」など，マスコミで取り上げられている反応をそのまま表明している回答が多数見られました．

6　まとめ

私の印象をまとめると，

- Cons(否定箇所)
  - plot3d や数式処理が甘い
  - mathjax なんでネットがないと, 汚いまま
  - 二重積分 (累次積分) は確かに悲惨だ，でも本当に必要か？
  - コマンドの help が見にくい, 探しにくい
- Pros(肯定箇所)
  - jupyter notebook が便利
  - latex, html 変換が default で綺麗
  - mark down が使える
  - key bind が使える
  - 本格的な前後処理とスムーズにつなげる
  - deep learning, 機械学習，データの前処理，plot が充実している
  - 世界中の人が使いまくってる

などがあります．python あるいは sympy の数式処理の性能というより，jupyter notebook の編集作業の性能に対する好感触や周辺環境の充実が強みです．まとめると，

- Maple は性能は高いが，表示・変換や key 操作の制約がきつく，有償ソフトとか，本格的な coding がしんどいなどの理由から，将来使えそうになく，覚えようという意識は低くて，触らなくなって，そのうち忘れちゃう．

- 一方，sympy は性能は高くないが，学生の食いつきは驚くほどいい．

この差は，学習で最も大切とされる「動機」に訴えるんでしょうね [7]．マスコミのおかげで，python はとっても役に立つように見えるんですよ，学生には．私の本音は，「Maple が jupyter notebook で動くと嬉しい」んですが...

参考文献

[1] 溝上慎一：「序」，森朋子・溝上慎一（編）『アクティブラーニング型授業としての反転授業　理論編』，ナカニシヤ出版，2017.

[2] 西谷滋人：「アクティブラーニングにおけるチーム評価の導入」，京都大学数理解析研究所講究録 1909, pp.223–232, 2014.

[3] 西谷滋人：「Maple 版ルフィの仲間たちに試練を!! –ペア評価による数式処理ソフト教育–」，京都大学数理解析研究所講究録 1865, pp.62–71, 2013.

[4] 西谷滋人・廣岡愛未：「パターンとペアプロの数式処理ソフト学習への適用」，京都大学数理解析研究所講究録 1735, pp.127–139, 2011.

[5] daddygongon：「python による数学入門」，`https://github.com/daddygongon/jupyter_num_calc`

[6] daddygongon：「python による数学入門 (nbviewer)」，`https://nbviewer.jupyter.org/github/daddygongon/jupyter_num_calc/blob/master/numerical_calc/README.ipynb`

[7] アーリック・ボーザー：Learn Better，英治出版，2018.