

free な数式処理ソフト Sage の 学部教育での活用事例

木村巖*

本稿では、フリーな統合数学ソフト Sage を理学部数学科 2 年生（前期）の線形代数学の講義で使い、授業改善に役立てた事例を報告する。

1 Sage とは

Sage は、フリーの統合数学ソフトである。フリーソフトウェアとして開発・公開が行われており、数値計算に止まらず、数式処理やグラフィックの描画を行うことができる。

プロジェクトのリーダーは、W. Stein 氏（Washington 大）である。商用の統合数学ソフト、Mathematica, Maple, Matlab, Magmaなどを置き換えることを目標に掲げている。

Sage は現時点で、Apple MacOS X, Linux などの各種 Unix 互換 OS, Microsoft Windows*¹ 上で実行できる。また、Apple iOS デバイスや Google Android 端末の上で UI 部分を走らせることができる（実際の計算は、Sage の公開サーバなどを用いる）。

プロジェクトのウェブページは <http://www.sagemath.org/> である。検索する際には、sagemath で検索するとよい。

Sage の開発は、もともとは数論幾何、特に、Stein 氏の専門である、代数体・有限体上の楕円曲線や Abel 多様体、保型形式といった対象の具体的な計算を目的に始まった。数論研究者向けの Sage の紹介として、拙稿もご覧いただければ幸いである [Kim12]。

プロジェクトが進展するにつれ、さまざまな数値計算・数式処理・グラフィックの機能が追加され、現在では群論、環論（Gröbner 基底）や代数幾何、組合せ論や表現論の専門的な計算の他、2D/3D グラフィックやレイトレーシングによる描画ができる統合数学ソフトとなった。

プロジェクトのモットーの一つが、「車輪を再発明せず、車を作ろう！」である。既存のオープンソース・フリーソフトを組み合わせ、統合数学ソフトウェアを構築している。例えば、数式処理は Maxima/Singular、線形代数は Linbox や LAPACK、代数体の数論は pari-gp や Kash、群論には GAP などである。これらを、Python という、非常に普及したスクリプト言語で統合

* 富山大学大学院理工学研究部（理学）

*¹ Windows 上での実行は、実際には Oracle VirtualBox 上で Linux の仮想マシンを起動し、そこで Linux 版の Sage を実行している

している。また、ノートブック風のフロントエンドはウェブブラウザで実現している。Sage に内蔵されたウェブサーバを経由して、ブラウザと Sage とがクライアント・サーバをなし、計算結果の表示には、HTML, CSS, jsmath といった既存の技術を用いている。

オープンソースソフトウェアとして、開発過程も広く公開されている。Sage のソースコードは分散バージョン管理システム^{*2}に保存され、不具合などはバグトラックスystemで管理されている。また、開発者やユーザのメーリングリストは Google Groups 上におかれていて、活発に議論されている。さらに、数ヶ月に一度の割合で、開発者等による会合 Sage Days が世界各地で開催されている。

国内でも Sage は広く使われるようになってきている。使うだけではなく、不具合の修正や機能の追加など、開発への参画も意図して、2012 年 5 月には Sage Days が九州大学で開催された^{*3}。その時の様子については、横山氏・沼田氏による報告 [横沼] を参照されたい。

2 Sage を使った線形代数の授業

Sage のような CAS (Computer Algebra System) を、学部の線形代数の授業で使う理由はなんだろうか。今回の対象は、富山大学理学部数学科の 2 年生が、前期に受講する線形代数学の講義である。受講対象者は約 50 名強。内容は、実数体・複素数体上の多項式の復習をしてから、固有値と固有ベクトル、内積空間、内積空間上の自己共役・正規作用素とそれらの対角化、までである (ちなみに教科書は Axler [Axl97] で、Chapter 4 から Chapter 7 の途中までだった。) また、Sage を使って、2D/3D のグラフィックや計算例を提示することが主で、受講者に Sage を使わせるということは、講義の際にはしていない。

まず、3 次元のベクトル空間における幾何的な直感を助けるために、グラフィックによる部分空間の提示を行った (図 1)。

受講者らが学んだ高校のカリキュラムでは、空間における平面は、座標軸に直交するものしか扱っていない (一般の平面の方程式も含まれない)。板書で、あまり上手でない平面の絵を見せられても、直感的に把握しづらいかもしれない。Sage を使って提示すれば、拡大縮小の他、回転させることもでき、大いに理解を助けることができる。

また、線形変換 $f: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ の図示例として、「ネコ写し」(図 2) を提示した。直線と半円で構成されたネコ (のように見える絵) を、 $\pi/4$ 回転し $\sqrt{2}$ 倍拡大したネコ (斜めになっているの絵) に写し、重ねて描画したものである。鼻の頭のあたりに原点があり、そこは固定されている。同様に、線形変換の図示例として、上と同じ線形変換により、格子点がどのように移動するかを

^{*2} これまで Mercurial で管理されてきたが、Git への移行が計画されている。

^{*3} 横山俊一氏、沼田泰英氏の主催、<http://www.stat.t.u-tokyo.ac.jp/~numata/html/sage/days/201206/index.ja.html>

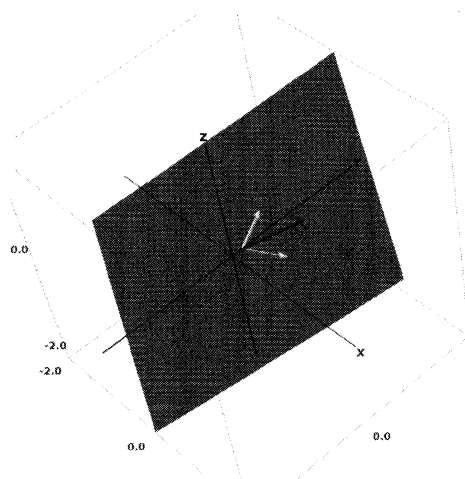


図1 $\mathbb{R}^3$ における平面と、ベクトルの図示.

図示したのが、図3である.

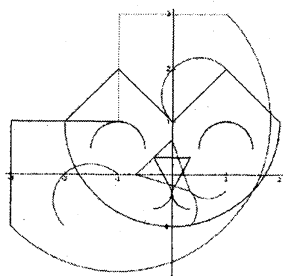


図2 ネコ写し

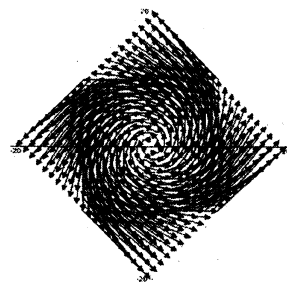


図3 平面上の格子点の移動.

Sage は数式処理システムでもあるので、その性質を活かした例も紹介した. n 次以下の実係数多項式の全体 P_n に、内積を

$$(f(x), g(x)) := \int_0^1 f(x)g(x) dx \quad f(x), g(x) \in P_n,$$

と入れる. 基底 $\{1, x, x^2, \dots, x^n\}$ に Gram-Schmidt の直交化を行うことができる. また, $\sin(x)$ を, 例えばたかだか 5 次の多項式で近似する際に, 上の内積 (但し積分区間は $[-\pi, \pi]$) から導かれるノルムに対して, 直交射影の方法を使い, よい近似を得ることができる. これらの手続きを, Sage のノートブックとして実現することができた (付録を参照).

以上のように, グラフィックや具体例を, Sage を用いて提示しながら講義を行い, 期末試験の際にアンケートを行った. 設問は, 「Sage や Mathematica^{*4}のような CAS を用いた数学の勉

^{*4} 本講義では Mathematica は取り上げなかったが, 富山大学では Mathematica のフローティングライセンスを購入しており, 受講者らは他の講義などで見ているはずである.

強に興味があるか？」である。回答は自由記述式とした。有効回答は 13 で多くないが、その内訳は

- ある (5 名)
- ない (8 名)

であった。記述から「ない」とした理由を拾うと、「数学だけでも難しいのに、数式処理のようなものを使うと、よりややこしくなる」ということであった。

これはもっともな回答であり、CAS を教育に用いる際には、数学の内容を伝える授業なのか、それとも CAS の使い方を伝える授業なのかをはっきりさせる必要がある。

一方、Sage に興味を示した 5 名を対象に、夏休み期間中に Sage のハンズオンセミナーを開催した。Microsoft Window のパソコン (1 名は自分のノートパソコンを持ち込み、それ以外はこちらで用意したデスクトップパソコン) に、Sage をインストールするところから始めた。実際には、仮想化ソフトの Oracle VirtualBox のインストール、Windows 用 Sage のディスクイメージのダウンロード、インストールなどを経て、ブラウザを経由して Sage のノートブックを使うところまで経験した。また、Sage 公開サーバの <http://www.sagenb.org/> の使用も体験してもらった。

Sage がフリーソフトである利点は、この様に、興味を持ってくれた人に、すぐに使ってもらえ、また DVD などに焼いて無償で配布できることである。

一方、課題となるのは、上述のように、数学その物に加えて、CAS のような環境で数学を行うことのハードルの高さ、ユーザの多い Microsoft Windows へのインストールの手間、さらに、初心者向けの日本語での文献が少ないことが挙げられる。

3 まとめ

本稿では、*free* な統合数学ソフト Sage の簡単な紹介を行い、これを用いた線形代数の講義の事例を報告した。教員がグラフィックや計算例を提示する為には、十分高機能である一方で無償であることなど、メリットが大きい。一方、多くの受講者にとっては、自ら使うには、最初はやや敷居が高いかもしれないことが分かった。しかし、意欲的な受講者には、数学を深く学ぶための格好のツールであり、今後教育の現場で、ますます Sage は使われていくものと思う。

末筆ながら、RIMS 研究集会「数学ソフトウェアと教育—数学ソフトウェアの効果的利用に関する研究—」でお世話いただきました皆様、特に研究集会へお誘いいただきました、中村泰之先生 (名古屋大学大学院情報科学研究科) に御礼申し上げます。

参考文献

- [Axl97] Sheldon Axler, *Linear algebra done right*, second ed., Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1997. MR 1482226 (98i:15001)
- [Kim12] Iwao Kimura, 数論研究者のための *Sage*, Proceedings of the Symposium on Algebraic Number Theory and Related Topics, RIMS Kôkyûroku Bessatsu, B32, Res. Inst. Math. Sci. (RIMS), Kyoto, 2012, pp. 125–144.
- [横沼] 横山俊一 and 沼田泰英, *Sage Days in Japan* 開催報告, 数式処理研究の新たな発展, RIMS Kôkyûroku, Res. Inst. Math. Sci. (RIMS), Kyoto, to appear.

付録：Sage notebookの例

Gram-Schmidt orthogonalization

This is an example of Gram-Schmidt orthogonalization in a vector spaces defined by polynomials and integrals. (この付録は、すべてSage notebook上で書いたものである。)

変数 x を定義する.

```
x=var('x')
```

内積を定義する (念頭に置いているのは、たかだか n 次以下の実係数多項式の全体がなす実ベクトル空間) .

$$\langle a, b \rangle := \int_0^1 a(x)b(x) dx.$$

```
def innerprod_integration_0_1(a, b):
    return integrate(a*b, (x, 0, 1))
```

ノルムを定義する：

$$\|a\| := \sqrt{\langle a, a \rangle}.$$

```
def mynorm(a, inner_product_function):
    return sqrt(inner_product_function(a, a))
```

試しに、 $1, x, x^2$ のノルムを計算してみる.

```
mynorm(1, innerprod_integration_0_1)
```

1

```
mynorm(x, innerprod_integration_0_1)
```

$\sqrt{\frac{1}{3}}$

```
mynorm(x^2, innerprod_integration_0_1)
```

$$\sqrt{\frac{1}{5}}$$

Gram-Schmidtの直交化法.

内積空間 $(V, \langle \cdot, \cdot \rangle)$ を考える. 一次独立なベクトルの組 $(v_1, \dots, v_m)$ が与えられたとき, 次のようにして orthonormal system $(e_1, \dots, e_m)$ を定める手続きを, Gram-Schmidtの直交化法というのだった.

$$e_j := \frac{f_j}{\|f_j\|}, \quad f_j = v_j - \sum_{k=1}^{j-1} \langle v_j, e_k \rangle e_k, \quad (j = 1, \dots, m)$$

```
def gram_schmidt(vs, inner_product_function):
    ret = []
    for v in vs:
        f = v - sum([inner_product_function(v, r)*r for r in ret])
        ret.append(f/(mynorm(f, inner_product_function)))
    return(ret)
```

$P_3(\mathbf{R})$, たかだか3次の実係数多項式全体に, 上で定義した内積で, Gram-Schmidtをやってみる.

```
rs = gram_schmidt([1,x,x^2, x^3], innerprod_integration_0_1)
```

```
for r in rs:
    print r.factor()
```

```
1
(2*x - 1)*sqrt(3)
(6*x^2 - 6*x + 1)*sqrt(5)
(2*x - 1)*(10*x^2 - 10*x + 1)*sqrt(7)
```

Orthogonal projectionの計算

ベクトル空間 V の部分空間 U に対して, 射影 $P_U: V = U \oplus U^\perp \rightarrow U$ を orthogonal projection (直交射影) というのだった. $f \in V$ の直交射影 $P_U(f)$ は, $(e_1, \dots, e_m)$ が U の orthonormal basis のとき, 次のようにして計算できる:

$$P_U(f) := \sum_{j=1}^m \langle f, e_j \rangle e_j.$$

```
def orthogonal_projection(f, bases, inner_product_function):
    """This function computes the orthogonal projection of f onto the subspace
    spanned by bases"""
```

```
return(sum([inner_product_function(f, base)*base for base in bases]))
```

$\sin(x)$ の直交射影による多項式近似.

```
def innerprod_integration_minus_pi_pi(a, b):
    return integrate(a*b, (x, -pi, pi))
```

```
sin_approx=orthogonal_projection(sin(x), gram_schmidt([1,x,x^2,x^3,x^4,x^5],
innerprod_integration_minus_pi_pi), innerprod_integration_minus_pi_pi)
```

```
sin_approx.expand().simplify_full()
```

$$\frac{21(33(\pi^4 - 105\pi^2 + 945)x^5 - 30(\pi^6 - 125\pi^4 + 1155\pi^2)x^3 + 5(\pi^8 - 153\pi^6 + 1485\pi^4)x)}{8\pi^{10}}$$

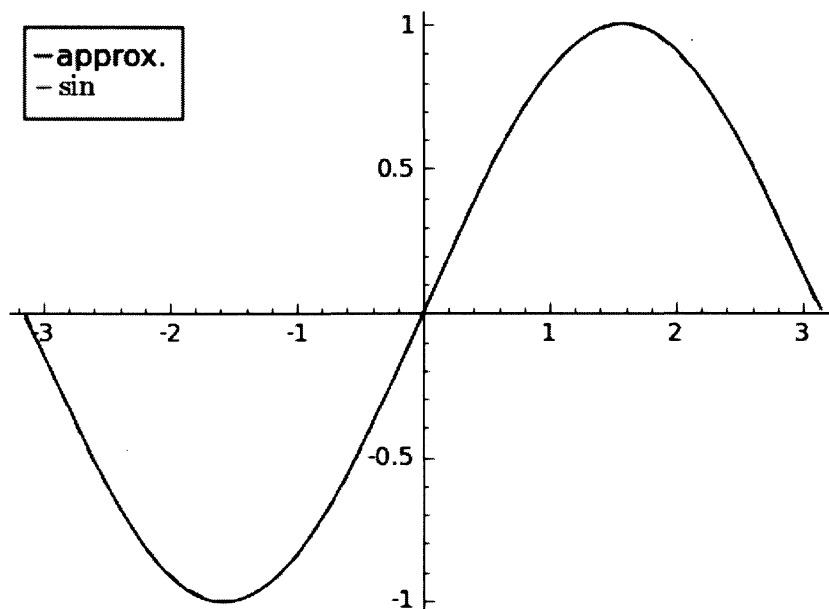
係数を近似値で見ると次のようになる：

```
sum([RR(c[0])*x^c[1] for c in sin_approx.coefficients()])
```

$$0.00564311797634677 x^5 - 0.155271410633428 x^3 + 0.987862135574673 x$$

青線が多項式近似, ダッシュの赤線が $\sin(x)$. 重なっていてほとんど誤差なし.

```
(plot(sin_approx, (x, -pi, pi), legend_label='approx.')+plot(sin(x), (x, -pi,
pi), color='red', linestyle='--', legend_label='$\sin$')).show(figsize=5)
```



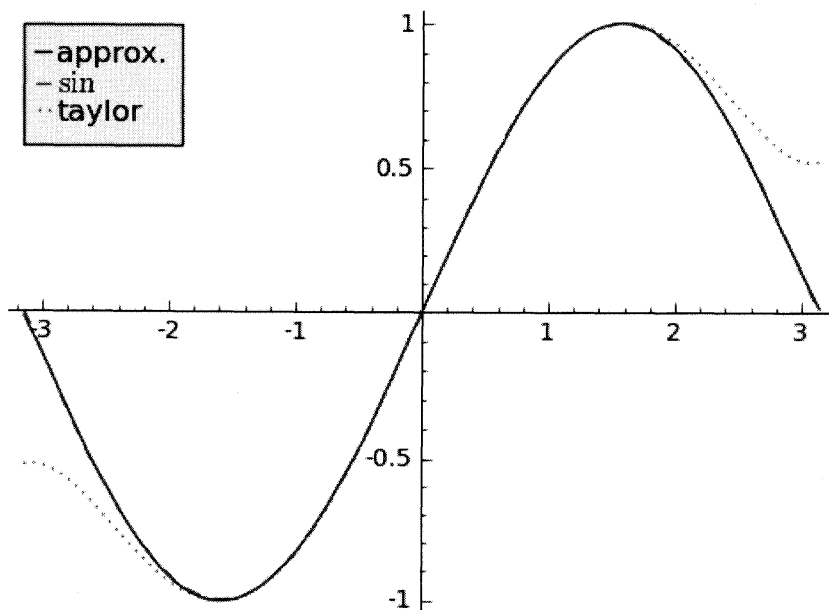
テイラー展開との比較.

```
taylor(sin(x), x, 0, 5)
```

$$\frac{1}{120}x^5 - \frac{1}{6}x^3 + x$$

直交射影による多項式近似（青の実線）， $\sin(x)$ （赤のダッシュ線），Taylor展開による近似（緑の点線）を重ねてプロットする．Taylor展開では，区間の端での近似があまりよくないことがわかる．

```
(plot(sin_approx, (x, -pi, pi), legend_label='approx.')+plot(sin(x), (x, -pi, pi), color='red', linestyle='--', legend_label='$\sin$')+plot(taylor(sin(x), x, 0, 5), (x, -pi, pi), color='green', linestyle=":", legend_label='taylor')).show(figsize=5)
```



以上.