Sage における次数 2 のベクトル値ジーゲル保型形式の計算と開発環境について

竹森 翔

北海道大学

導入

この講演では，主に 2 つの内容について話す．

(1) Sage での次数 2 のベクトル値保型形式の計算について．とくに，ベクトル値ジーゲル保型形式のなす加群の構造定理について．

(2) Sage のプログラムの開発環境について．とくに，Emacs 内で Sage を動かしたり，Sage のプログラムを編集したりするための elisp package について．

次数 2 のベクトル値ジーゲル保型形式に関する記号

j: 非負整数,

$\mathbf{Sym}(j)$: $\mathbf{GL}_2(\mathbb{C})$ の次数 j の対称テンソル積表現

ここで，u_1, u_2 を不定元とし，$\mathbf{Sym}(j)$ の表現空間を u_1, u_2 についての $\mathbb{C}$ 係数 j 次斉次多項式のなす $j+1$ 次元の空間と同一視する.

$\mathfrak{H}_2$: 次数 2 のジーゲル上半空間を次で定義する.

$$\mathfrak{H}_2 := \left\{ Z \in M_2(\mathbb{C}) \;\middle|\; Z = {}^tZ,\ \mathrm{Im}\, Z > 0 \right\}.$$

次数 2 のベクトル値ジーゲル保型形式に関する記号 (2)

k, j: 非負整数，$u = (u_1, u_2)$ とおく．
$F = F(Z, u) : \mathfrak{H}_2 \to \mathrm{Sym}(j)$ を正則関数とする．
F が次の 3 つの条件を満たすとき，F を次数 2，ウェイト $\det^k \otimes \mathrm{Sym}(j)$ のベクトル値ジーゲル保型形式とよぶ．

$$F(Z + S,\ u) = F(Z,\ u), \quad \forall\, S = {}^tS \in M_2(\mathbb{Z}),$$

$$F(-Z^{-1},\ u) = (\det Z)^k F(Z,\ uZ),$$

$$F({}^t\alpha Z \alpha,\ u) = (\det \alpha)^{-k}\, F(Z,\ u\alpha^{-1}), \quad \forall\, \alpha \in \mathrm{GL}_2(\mathbb{Z}).$$

注意 1

- この 3 つの条件は，F が $\mathrm{Sp}_2(\mathbb{Z})$ の作用で不変であるという条件のいいかえである．
- 3 つめの条件で，$\alpha = -1_2$ とすると，$j : odd \Rightarrow F = 0$ が分かる．

次数 2 のベクトル値ジーゲル保型形式に関する記号 (3)

$A_{k,\,j}$: ウェイトが $\det^k \otimes \mathrm{Sym}(j)$ のベクトル値ジーゲル保型形式のなす $\mathbb{C}$ ベクトル空間 (有限次元であることが知られている) とし,

$$A^0_{\mathrm{Sym}(j)} := \bigoplus_{k:even} A_{k,\,j}, \quad A^1_{\mathrm{Sym}(j)} := \bigoplus_{k:odd} A_{k,\,j},$$

とおく. とくに, $j = 0$ のとき, $A := A^0_{\mathrm{Sym}(0)}$ とおく.
関数の通常の積によって, A は環になり, $A^0_{\mathrm{Sym}(j)}$ と $A^1_{\mathrm{Sym}(j)}$ は A 加群とみなせる.

ベクトル値ジーゲル保型形式の加群の構造定理

$A_{k,\,j}$ の元のフーリエ係数を計算するには，(おそらく) 加群 $A^{0}_{\mathrm{Sym}(j)}$，$A^{1}_{\mathrm{Sym}(j)}$ の構造を明示的に知る必要がある．(Hecke 固有値を知りたいだけなら，van der Geer, Carel Faber, Bergström らによる方法もある．ただし，ウェイトに制限がつく．)

$A^{0}_{\mathrm{Sym}(j)}$，$A^{1}_{\mathrm{Sym}(j)}$ の構造定理は，$j \leq 8$ のときに知られている．以下では，知られている構造定理の紹介，$A^{0}_{\mathrm{Sym}(10)}$ の構造定理の紹介 (講演者の結果) と Sage での $A_{k,\,j}$ の元の計算例を与える．

ベクトル値ジーゲル保型形式の加群の構造定理 (2)

まず，A の構造定理について述べる．

定理 2 (井草)

$$A = \mathbb{C}[\varphi_4, \varphi_6, \chi_{10}, \chi_{12}],$$

$\varphi_4, \varphi_6, \chi_{10}, \chi_{12}$ は $\mathbb{C}$ 上代数的独立．

ここで，φ_4, φ_6 はウェイト 4，6 のジーゲル・アイゼンシュタイン級数，χ_{10}, χ_{12} はウェイト 10，12 のカスプ形式である．

Rankin-Cohen 型微分作用素

$f \in A_{k,0}$, $g \in A_{l,0}$ とし，$2\pi i \{f, g\}_{\mathrm{Sym}(2)}$ を

$$\left(kf\frac{\partial g}{\partial z_1} - lg\frac{\partial f}{\partial z_1}\right)u_1^2 + \left(kf\frac{\partial g}{\partial z_2} - lg\frac{\partial f}{\partial z_2}\right)u_1u_2 + \left(kf\frac{\partial g}{\partial z_3} - lg\frac{\partial f}{\partial z_3}\right)u_2^2$$

と置くと，$\{f, g\}_{\mathrm{Sym}(2)} \in A_{k+l,2}$ となる (T. Satoh)．ただし，$Z = \begin{pmatrix} z_1 & z_2 \\ z_2 & z_3 \end{pmatrix} \in H_2$．この微分作用素は Rankin-Cohen 型微分作用素の例になっている．Rankin-Cohen 型微分作用素は保型形式を構成するのに役立つ．伊吹山氏らによって，Rankin-Cohen 型微分作用素の一般的な理論が構築されている．

ベクトル値ジーゲル保型形式の加群の構造定理 (3)

定理 3

加群 $A^{\epsilon}_{\mathrm{Sym}(j)}$ の free resolution は次で与えられる.

加群 M	proved by	free resolution
$A^{0}_{\mathrm{Sym}(2)}$	佐藤	$0 \to A \to A^4 \to A^6 \to M \to 0$
$A^{1}_{\mathrm{Sym}(2)}$	伊吹山	$0 \to A \to A^4 \to M \to 0$
$A^{0}_{\mathrm{Sym}(4)}, A^{1}_{\mathrm{Sym}(4)}$	伊吹山	階数 5，自由
$A^{0}_{\mathrm{Sym}(6)}$	伊吹山	階数 7，自由
$A^{1}_{\mathrm{Sym}(6)}$	van Dorp	階数 7，自由
$A^{0}_{\mathrm{Sym}(8)}, A^{1}_{\mathrm{Sym}(8)}$	喜友名	階数 9，自由
$A^{0}_{\mathrm{Sym}(10)}$	T.	$0 \to A^2 \to A^{13} \to M \to 0$

ベクトル値ジーゲル保型形式の加群の構造定理 (4)

注意 4

- これらの結果において，生成元や関係式などは明示的に与えられている．
- ほとんどの生成元は，Rankin-Cohen 型微分作用素で構成される．

$A^0_{\mathrm{Sym}(10)}$ の計算は Sage のパッケージ “degree2” (`https://github.com/stakemori/degree2`) で行なった．
例として，$A^0_{\mathrm{Sym}(2)}$ の計算を Sage でやってみる．

Emacs での Sage のプログラムの開発について

ジーゲル保型形式の計算に使ったパッケージ “degree2” は，GNU Emacs で作った．

Emacs 内で Sage を実行したり，Sage のコードを編集したりするための elisp (Emacs lisp) package は主に 2 つある．

- sage-mode：2007 年頃に作られた．作者は Nick Alexander，現在のメンテナーは Ivan Andrus．Sage の optional package として提供されている．
- sage-shell-mode：2012 年頃に作られた．作者は講演者．MELPA からインストールできる (後で説明)．

ここでは，sage-shell-mode の紹介や sage-mode との比較をする．

sage-shell-mode を作った動機

- 非同期処理 (ユーザーの入力をロックしないような処理) を適切に行う.
- python-mode への依存を少なくする.
- auto-complete，helm (または anything) のプラグインを作る.

最初は，sage-mode 用の auto-complete のプラグインを作ろうとしたが，非同期に補完候補を集める関数がうまく動かなかった.

python-mode について

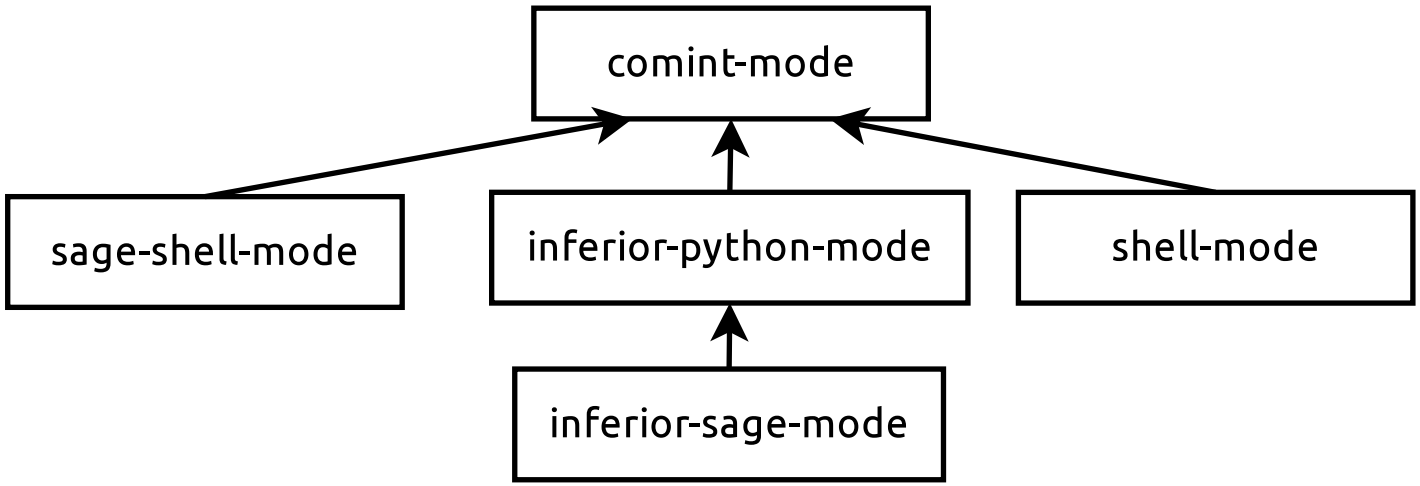
python-mode と呼ばれている major mode は少なくとも 3 つある.

ファイル名	作者	備考
python.el	Dave Love	Emacs 24.1 まで標準添付
python.el	Fabiàn E. Gallina	Emacs 24.2 から標準添付
python-mode.el	Tim Peters	ソースは 2 万 6 千行以上

それぞれが，python-mode と inferior-python-mode (Python のインタプリターを実行するための mode) を提供している.

major mode のヒエラルキー

Emacs では既存の major mode から派生した major mode を簡単に作ることができる.



comint は command interpreter の略. inferior-sage-mode は sage-mode で提供されている major mode.

sage-mode との比較

sage-mode が持っていて sage-shell-mode が持っていない機能は,

- Sage Notebook の中でのように，出力を LaTeX でタイプセットし，Emacs 内でインライン表示する機能.
- グラフなどの画像を Emacs 内でインライン表示する機能.
- Doctest や build の機能.

Sage Wiki にも，sage-mode との比較がある.
`http://wiki.sagemath.org/SageModeComparison`
Sage Wiki には書いていないが，sage-shell-mode は SageTeX のサポートも少しだけある.

MELPA

Emacs 24 からはパッケージ管理機能 (package.el) が標準添付されている．(Emacs 23 ユーザーは package.el をインストールする必要がある)．package.el は 複数のリポジトリを扱かうことができ，MELPA は非公式のリポジトリの 1 つである．

sage-shell-mode のインストール

sage-shell-mode は MELPA からインストールできる．上の設定の元で，

M-x package-install RET sage-shell-mode RET

とすればよい．もし，Emacs 内で

```
(executable-find "sage")
```

を評価して，`nil` 以外が返ってくるならば特別な設定は必要ない (executable-find は シェルコマンドの which のようなもの)．

sage-shell-mode のインストール (2)

もし上のコードが nil を返すならば Emacs に 環境変数 SAGE_ROOT を伝える必要がある.

```
(setq sage-shell:sage-root "/path/to/sage/root/")
```

Sage の中での環境変数 SAGE_ROOT を調べるには，Sage の中で次のコードを評価すればよい.

```
import os; print os.environ["SAGE_ROOT"]
```

エイリアス

sage-mode との名前の衝突を避けるために，
sage-shell-mode は冗長な名前を使っている．以下の elisp のコードでより短い名前が定義される．このコードを評価する前までは，sage-mode との併用が可能である．

```
(dolist (c sage-shell:func-alias-alist)
  (defalias (cdr c) (car c)))
(dolist (c sage-shell:var-alias-alist)
  (defvaralias (cdr c) (car c)))
```

例えば，sage-shell:run-sage には run-sage というエイリアスが定義され，sage-shell:sage-mode には sage-mode というエイリアスが定義される．

Emacs 内での Sage の起動

M-x sage-shell:run-sage とすると，Emacs 内で Sage のプロセスを走らせることができる．**M-x sage-shell:run-new-sage** で複数のプロセスを走らせることもできる．

キー	説明
RET	現在の行を評価
TAB	補完
空行で C-d	Sage を終了する
M-n, M-p	入力履歴を行き来する
C-c C-c	プログラムの実行を中断する

他のインタプリタ

sage-shell-mode は gp, gap, singular などの他のインタプリタのサポートも少しだけしている．例えば，マジックコマンド%gap によって gap のインタプリタに移ったときや，カーソルが `gap.eval()` の括弧の中にあるときは，gap のコマンドを補完する．
ただし，`gap_console()` で gap のインタプリタに移ることは想定していない．

編集用の major mode

拡張子が.sageのファイルを開くと自動的にsage-shell:sage-modeになる (エイリアスはsage-mode). sage-shell:sage-modeはpython-modeとほぼ同じで，違いはいくつかのキーバインドだけである．主なキーバインドは以下の通り.

キー	説明
C-c C-l	現在のファイルをロード
C-c C-r	リージョンをプロセスに送る
C-c C-c	バッファをプロセスに送る
C-c C-z	プロセスのバッファを表示する

SageTeX

Emacs 内で Sage を起動すると，環境変数 TEXINPUTS に
`$SAGE_ROOT/local/share/texmf/tex/generic/sagetex`
が追加される SageTeX を使っている TeX のバッファを訪れているときに，**M-x sage-shell-sagetex:load-file** によって，”ファイル名.sagetex.sage ”が Sage のプロセスにロードされる．
Sage の起動やパッケージのロードにかかる時間が節約できるという利点があるが，”ファイル名.sagetex.sage ”と Sage のシェルは名前空間を共有することになる．

他の機能

最近追加した機能として

- pdb のサポート (Gallina の inferior-python-mode と同じ)
- エラーが起きて，スタックトレースが表示されたときに，エラーが起こった行へのリンクを表示する機能
- TeX のコンパイルをし，SageTeX のファイルを Sage のプロセスにロードし，もう一度コンパイルするコマンド.

がある.

プラグイン

sage-shell-mode には auto-complete，helm，anything 用のプラグインがある．全て MELPA からインストールできる．

- auto-complete-sage：自動補完のためのプラグイン．
- helm-sage：helm 用のプラグイン．
- anything-sage：helm-sage の anything 版．

リポジトリ

sage-shell-mode などのリポジトリや簡単なマニュアルは, 全て GitHub にある.

- sage-shell-mode
  `https://github.com/stakemori/sage-shell-mode`
- auto-complete-sage
  `https://github.com/stakemori/auto-complete-sage`
- helm-sage
  `https://github.com/stakemori/helm-sage`
- anything-sage
  `https://github.com/stakemori/anything-sage`