

# SnapPy と HIKMOT の使い方

正井 秀俊 (東京大学大学院数理科学研究科)

## 1 はじめに

SnapPy は Jeff Weeks 氏による SnapPea を python 上に Nathan Dunfield 氏 と Marc Culler 氏 が移植し、管理、拡張されているプログラムである。SnapPy は 3次元多様体の四面体分割から、Gluing 方程式と呼ばれる多様体が双曲になるための要請をまとめた式を立式し、それを Newton 法で解くことにより 3次元多様体の双曲構造を（存在すれば）近似計算で求めることができるプログラムである。本論文では SnapPy の基本的な使い方と、著者を含む開発チームで実装された HIKMOT と呼ばれる SnapPy を厳密化したプログラムの使い方を解説する。SnapPy では任意の四面体分割された 3次元多様体を扱うことができるが、ここでは簡単のため対象を結び目補空間に限定する。また、OS は Mac OS Xであることを仮定する。SnapPy は [1] のホームページ (<http://www.math.uic.edu/t3m/SnapPy/installing.html>) からダウンロードできるパッケージを用いて解説する。続いて HIKMOT を用いる際には SnapPy, HIKMOT を python にインストールする必要がありその方法についても簡単に解説する。

## 2 結び目の補空間の四面体分割

結び目とは円周  $S^1$  の 3次元球面  $S^3$  への埋め込みの像のアmbientイソトピー類を指す。本論文では同値類とその適当な表現を区別せず、特に断りをなく表現を同じ記号で表す。結び目  $K$  が双曲であるとは補空間  $S^3 \setminus K$  がある有限体積双曲多様体と同相になることを言う。同相な有限体積双曲多様体は等長であることが知られており (Mostow-Prasad 剛性) 結び目補空間に同相な双曲多様体の不変量はそのまま結び目の不変量となる。結び目の 2次元平面への射影で交点が横断的で、2重点であるものを図式という。SnapPy は結び目の図式からその補空間の四面体分割を計算することができる。はじめに、結び目の入力方法、補空間の四面体分割の計算方法を解説する。

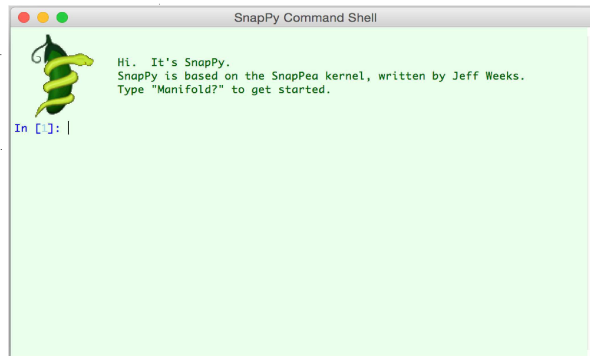


図 1: SnapPy 起動画面

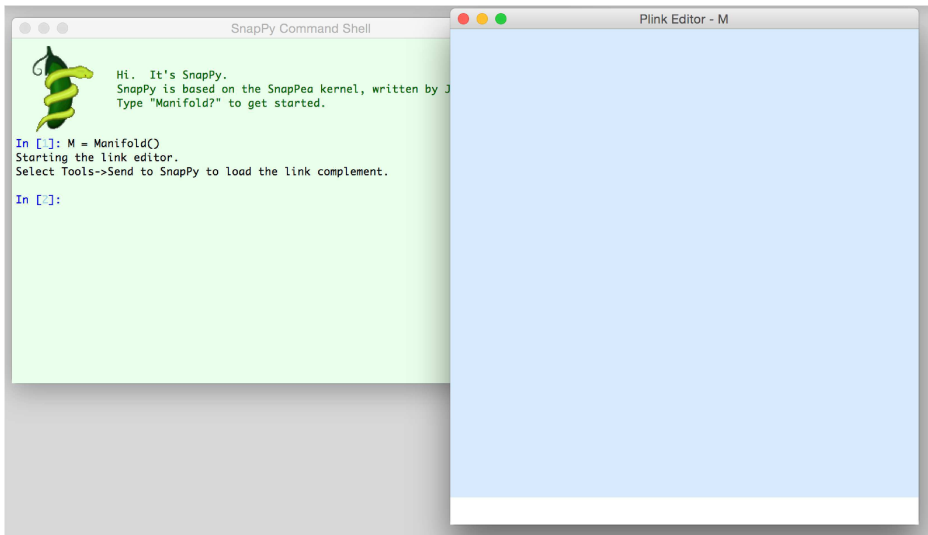


図 2: Plink 起動画面

図 1 は SnapPy の起動画面である。基本的には python のインタプリタ<sup>1</sup>と同様に動く。「M = Manifold()」と入力する<sup>2</sup>と図 2 のように新しい水色のウィンドウが立ち上がる (Plink と呼ばれる)。このウィンドウが結び目の描画画面である。Plink 上では

- マウスクリックにより区分的に直線で得られる結び目の絵を描くことができる。また、
- ループになるように絵を描いたのち、交点をクリックすることで交差交換ができる。

結び目を描いた後、図 3 のように「Tools」の「Send to SnapPy」を選ぶことで結び目補空間の四面体分割を SnapPy に計算させ、「M = Manifold()」としていた場合には変数「M」にその情報が格納される。「M = Manifold()」の「M」は変数であり、任意の文字列に変えても良い。「Manifold()」は SnapPy に備わっている関数であり、3次元多様体の四面体分割、双曲構造を計算する関数である。「Manifold('4.1')」のように括弧の中に情報を渡すこともできる (引数という)。引数を渡さなければ Manifold 関数は Plink ウィンドウを立ち上げる。交点数の小さい結び目には交点数の順に名前が付いており、「4.1」は 4 交点を持つ一つ目の結び目を指す。SnapPy は 11 交点までの結び目のデータなど様々なデータを持っている。詳しくは SnapPy に「Manifold?」と入力すると説明が出力される。

変数「M」に四面体分割の情報を格納すると、SnapPy は自動で Gluing 方程式を立式し Newton 法で近似解を計算する。図 4 の [2] のように「M」と入力すると「unnamed link(0,0)」と M の“名前”が出力される。最後の「(0,0)」は M がトラスカusp を一つ持ち、そのカusp はデー手術をされていないということの意味する。デー手術をされていた場合には手術の係数が出力される。次に、「M」、「」（ピリオド）、「tab キー」の順に打つと図 4 の [3] のような出力が得られる。もう一度「tab キー」を打つと M について計算できる様々な関数のリストが得られる。各項目について最後にクエスチョンマーク「?」をつけて入力すると、その関数の機能の説明が出力される。図 5 は「M.volume?」と入力した際の出力である。実際に計算をする際には「M.volume()」と最後に括弧をつける必要があることに注意する。SnapPy には様々な機能があるが、それを GUI で出力してくれるのが browse() 関数である。SnapPy を初めて使う際にはまず、「M.browse()」を試しどのような情報が計算できるのかを概観することをお勧めする。

<sup>1</sup>プログラミングに不慣れな読者はあまり気にする必要はなく、これから解説するコマンドで動くと考えれば良い。

<sup>2</sup>最後に Enter キーを押す。頭にスペースなどは入ってはいけない。

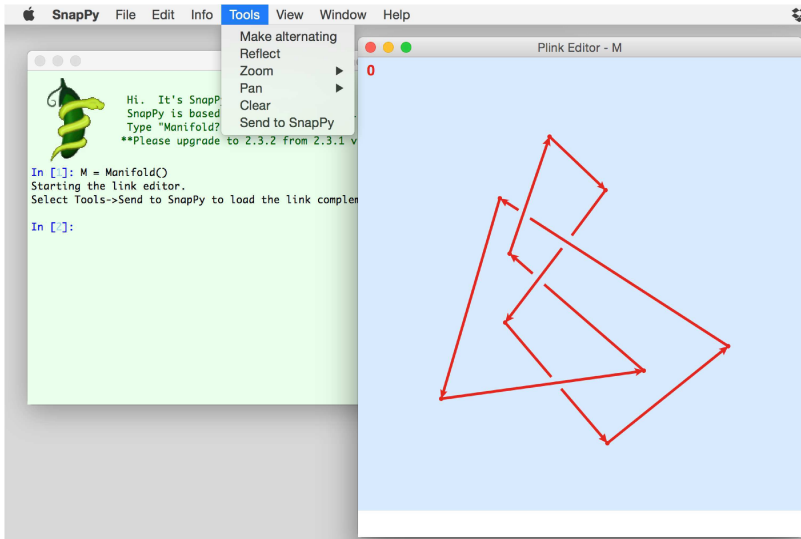


図 3: Plink で描いた結び目の図式を SnapPy に送る

### 3 HIKMOT

SnapPy では Newton 法による近似計算を行っているため、たとえ「SnapPy が双曲構造をみつけた」としてもそれは、数学的には与えられた多様体が双曲構造持つことの証明にはならない。論文 [2] では区間演算を用いて、数学的に厳密に双曲性の証明を行うプログラムを作成した。詳しい内容は論文に譲ることにし、今回はそのインストール方法、使い方を説明する。記述を様々な OS に対応させると、煩雑になるため、今回は Mac OS X に限定し、できるだけ具体的にインストール方法<sup>3</sup>を説明する。

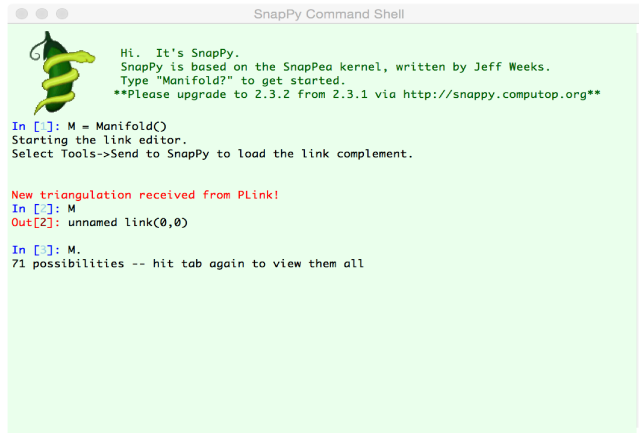
#### 3.1 HIKMOT のインストール

Mac でプログラムなどのインストールを行う際には、まず初めに Xcode (<https://developer.apple.com/jp/xcode/index.html>) をインストールする必要がある。これは HIKMOT のみならず、GAP など他のパッケージをインストールする際にも必要であるものである。

Xcode のインストールののち、HIKMOT は次のような手順でインストールできる。Mac の場合はアプリケーションフォルダの中のユーティリティーフォルダにあるターミナルを起動する。

1. Homebrew ([http://brew.sh/index\\_ja.html](http://brew.sh/index_ja.html)) という、「プログラムインストール用のプログラム」をインストールする。ホームページにあるようにターミナルに「`ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`」と入力すれば良い。
2. ターミナルに「`brew update`」と入力し、その後「`brew install boost`」と入力する (boost というパッケージがインストールされる)

<sup>3</sup>この方法は 2016 年 1 月現在有効である。今後も大きな変更はないと思われるため、詳しい説明はしばらくは意味があると思われる。



```

SnapPy Command Shell

Hi. It's SnapPy.
SnapPy is based on the SnapPea kernel, written by Jeff Weeks.
Type "Manifold?" to get started.
**Please upgrade to 2.3.2 from 2.3.1 via http://snappy.computop.org**

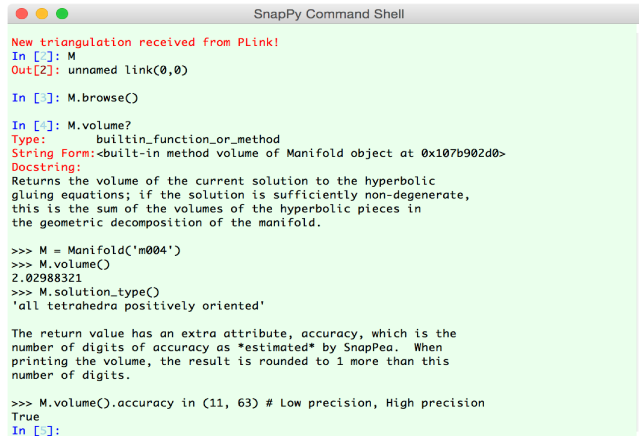
In []: M = Manifold()
Starting the link editor.
Select Tools->Send to SnapPy to load the link complement.

New triangulation received from PLink!
In []: M
Out[2]: unnamed link(0,0)

In []: M.
71 possibilities -- hit tab again to view them all

```

図 4: tab キーでその機能一覧を表示



```

SnapPy Command Shell

New triangulation received from PLink!
In []: M
Out[2]: unnamed link(0,0)

In []: M.browse()

In []: M.volume?
Type:          builtin_function_or_method
String Form: <builtin method volume of Manifold object at 0x107b902d0>
Docstring:
Returns the volume of the current solution to the hyperbolic
gluing equations; if the solution is sufficiently non-degenerate,
this is the sum of the volumes of the hyperbolic pieces in
the geometric decomposition of the manifold.

>>> M = Manifold('m004')
>>> M.volume()
2.02988321
>>> M.solution_type()
'all tetrahedra positively oriented'

The return value has an extra attribute, accuracy, which is the
number of digits of accuracy as "estimated" by SnapPea. When
printing the volume, the result is rounded to 1 more than this
number of digits.

>>> M.volume().accuracy in (11, 63) # Low precision, High precision
True
In []:

```

図 5: 関数の後ろに「?」をつけてヘルプを表示

3. SnapPy を python モジュールとしてインストールする。 <http://www.math.uic.edu/t3m/SnapPy/installing.html> の「Python Modules for Macintosh or Windows」項目に従い、まず `setuptools` (<https://pypi.python.org/pypi/setuptools>) をインストールする。その後はターミナルに「`python -m easy_install -U snappy`」と入力すれば良い。
4. HIKMOT のホームページ (<http://www.oishi.info.waseda.ac.jp/~takayasu/hikmot/>) から HIKMOT をダウンロードし、得られた `.zip` ファイルをダブルクリックで解凍する。
5. ターミナル上で「`cd`」コマンドを用いて解凍されたフォルダに移動する。例えばターミナルに「`cd Downloads/hikmot.v1.0.1`」と入力する。
6. その後ターミナルに「`sudo python setup.py install`」と入力することでインストールできる。(アカウントのパスワードを求められるので入力する)

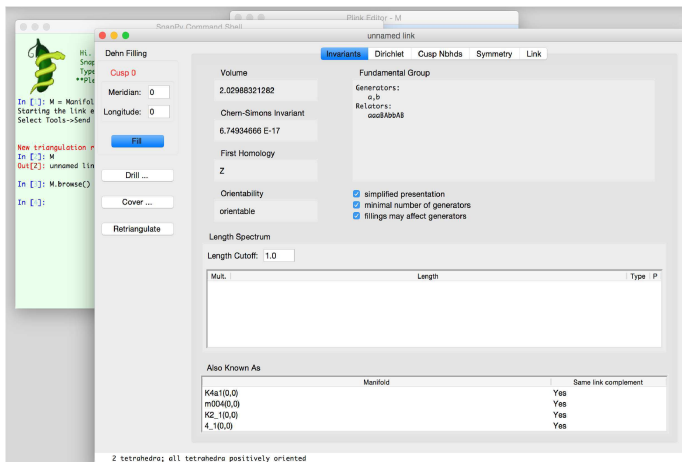


図 6: browse 機能

### 3.2 HIKMOT の使い方

```

hidetoshimasai - python2.7 - 80x24
Last login: Thu Jan 14 11:23:43 on ttys001
miffy115:~ hidetoshimasai$ python
Python 2.7.10 [Anaconda 2.3.0 (x86_64)] (default, May 28 2015, 17:04:42)
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
>>> import snappy
>>> import hikmot
>>> M = snappy.Manifold('4_1')
>>> hikmot.verify_hyperbolicity(M)
(True, [[([0.49999999999999977, 0.50000000000000034])+([0.86602540378443837, 0.86602540378443893])], ([0.49999999999999972, 0.50000000000000034])+([0.86602540378443837, 0.86602540378443893])])
>>>

```

図 7: HIKMOT の使い方

1. インストールの時と同様にターミナルを起動する。
2. ターミナルに `python` と打ち `python` を起動する。
3. 「`import snappy`」と「`import hikmot`」と入力し（順不同）SnapPy と HIKMOT をインポートする。
4. 「`M = snappy.Manifold()`」などと入力し変数 `M` に四面体分割の情報を入力する。今回は `python` の上で `snappy` を動かしているため `snappy` を頭につける必要があるがこの `snappy.Manifold()` は上で説明した `Manifold()` と同じものである。
5. 「`hikmot.verify_hyperbolicity(M)`」とすると変数 `M` に格納された四面体分割された多様体が双曲構造を持つかの判定が出来る。

図7は出力結果である。出力の初めの「True」は与えられた多様体 `M` が厳密に双曲構造を持つことを証明できたことを意味する。後ろの数値は厳密な双曲構造を含む区間を表すデータである。ここで、HIKMOT は双曲構造が存在することを証明することができるが、HIKMOT が双曲構造を見つけれなかった場合にも双曲構造が存在しないことの証明にはならないことに注意する。

## 謝辞

「計算代数システムによる新しい数学の開拓と発展」にお招きくださった世話人の方々に感謝いたします。本研究は JSPS 特別研究員奨励費の援助を受けております。

## 参考文献

- [1] M. Culler, N. Dunfield, and J. Weeks, *SnapPy, a computer program for studying the topology of 3-manifolds*, Available at <http://snappy.computop.org>.
- [2] N. Hoffman, K. Ichihara, M. Kashiwagi, H. Masai, S. Oishi, and A. Takayasu, Verified computations for hyperbolic 3-manifolds, *Exper. Math.*, 25 (2016), Issue 1, 66-78.