

D.3 講義のための補足ノート—東京大学大学院集中講義: 2007-05-30

Leck1/3.tex

単位取得予定者は feedback form を提出してください。用紙が足りないときは足してください。それ以外の人への提出も歓迎 (project に feedback してます)。

資料: <http://cc1.math.kobe-u.ac.jp/pukiwiki.php?KnxK2007> (cc1 の名前は時々かわります。cc1 へはリンクしないで下さい。)

D.3.1 今回の概要と数学の題材

(1) 微分の章から グラフの作成. (2) Polytope のいろはと polymake (3) できたら, Lorentz モデル, Xaos を見せる.

D.3.2 追加の章: Polytope のいろはと polymake

Polytope のいろは. 黒板で説明.

1. 凸包, polytope, polyhedron の定義. V 表現, H 表現.
2. $\text{face}_w(P)$ の定義. facet の定義.
3. Normal cone, Normal fan の定義と定理.

例 D.1 点の集合の凸包を計算し, その facet の法線ベクトルを計算せよ.

Polymake

Todo: polymake 開発の歴史. アルゴリズムの歴史.

Polymake を実行するには, ターミナルから以下を (一度) 実行しておく.

```
export PATH=/usr/local/lib/polymake/bin:$PATH
```

polymake では, 入力点は原点をとらない超平面の上においておく必要がある. 下の例では $x_1 = 1$ に点をのせてある. このファイル名を p-conv0.txt とする. Et/p-conv0.txt

```
POINTS
1 0 0
1 2 0
1 0 3
1 1 1
```

```
polymake p-conv0.txt FACETS
```

と入力することにより, 次の出力ファイル (同じ名前の出力 p-dim.txt) Et/p-conv.txt

```

_application polytope
_version 2.2
_type RationalPolytope

POINTS
1 0 0
1 2 0
1 0 3
1 1 1

FACETS
0 1 0
3 -3/2 -1
0 0 1

AFFINE_HULL

```

を得る.

アルゴリズムについての覚え書き.

1. n 次元の凸包. Edelsbrunner の coloring algorithm “E.Edelsbrunner, *Algorithms in Combinatorial Geometry* (Springer-Verlag 1986)” [?].
2. G.Ziegler の本, “Günter M.Ziegler, *Lectures on Polytopes* (Graduate Texts in Mathematics 152, Springer-Verlag New York 1995)” [2] V 表現と H 表現の変換. Fourier-Motzkin algorithm.
3. CDD ライブラリ [3] およびアルゴリズムの説明 (“The double description is revisited”) . “Komei Fukuda cdd Home Page (http://www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd.html)”

次元の計算. このファイル名を p-dim.txt とする. Et/p-dim0.txt

```

POINTS
1 0 0 3
1 2 0 3
1 0 3 3
1 1 1 3

```

polymake p-dim.txt DIM

と入力することにより, 次の出力ファイル Et/p-dim.txt

```

_application polytope
_version 2.2
_type RationalPolytope

POINTS
1 0 0 3
1 2 0 3
1 0 3 3
1 1 1 3

DIM
2

```

を得る.

Volume 計算.

D.3.3 フィードバック : Q and A

Q. 疑問. Parallels で仮想マシンから internet につながりません.

A. 答え. 仮想マシンは特段の理由がない限り Host-Only で利用してください. Parallel 側は hostonly にして, DHCP を有効にします. Mac 側は システム環境設定, 共有, インターネット で共有を有効にします. 詳しくは, 英語のマニュアル “Parallels Desktop for Mac User Guide” を “Host-Only Networking with Internet Sharing” で検索してください. なお, bridged で動く仮想マシンを同時動作させると動作が変になるようです.

Q. asir を使うとき glib3 がないと言われます.

A. glib は X11 を直接よびだす感じの asir の命令 draw_obj をそのまま使うのは使いにくいので, 作成された 2 次元専用の簡易グラフィックライブラリです. knoppix のバージョンが古いとこのエラーがでますが, openxm package を remove, apt-get する必要もないでしょう. glib3 は Mac 版に対応するためのもので, linux 版では

```
sudo cp /usr/local/OpenXM/lib/asir/glib /usr/local/OpenXM/lib/asir/glib3
```

で OK です.

Q. 有限体上の 2 次形式の lattice の base を計算するソフトは?

A. いままで各種 meeting で注意深く聞いていなかったものでしりません. Kash/Kant グループの Michael Pohst, Sage の William Stein, Magma group の誰かに聞くと知ってるかも? 調べてみます.

Q. knoppix/math でのユーザインタフェースのプロジェクトはどんなものがありますか?

A. 最近のニューフェースは sage_notebook です. たとえば sage_maxima と入力してください. web サービスとも一体化してます. (ひっそり webasir っていうのもあります). jsmath を用いて数式をタイプセットします.

もうすこし昔からあるプロジェクトは texmacs です. エンジンとのインタフェースを書くのは以外に簡単なので, トライしてみたら.

あとは, web での数学計算情報サービスはどんどん増えていますね. 日本では casio の研究所が関数の数値計算のサービスをしています. たとえば wikipedia の zeta 関数の日本語版の外部リンクに “ゼータ関数の計算” がありますがこれをクリックすると, こちらへ移動します. pari を裏で使えばもっと精度が高いのができそうな気がします (計算資源の問題を回避する必要があるのでしょうか.)

以下神戸大講義版から予測される質問のコピー.

Q. PATH 環境変数ってなんですか?

A. 一般にシェルからコマンドを起動できないときはまず, set コマンドで環境変数 PATH の値を確認します. シェルは環境変数 PATH に書かれているディレクトリ (フォルダ) からコマンド (プログラム) を探します. たとえば PATH の値が

```
/usr/sbin;/usr;/usr/bin
```

の場合は /usr/sbin, /usr, /usr/bin の順番にコマンドを探します. ls /usr/sbin とかをやってみて, kterm がなければすくなくとも標準の PATH にはこのようなファイルは存在していないということですね.

PATH にフォルダ (たとえば /usr/X11R6/bin) を追加するには csh 系の場合は

```
set path=(/usr/X11R6/bin $path)
```

を .cshrc ファイルに書いておきます. bash 系の場合は,

```
PATH=/usr/X11R6/bin:$PATH
export PATH
```

を `.bashrc` ファイルに書いておきます。unix では個人の環境設定はホームディレクトリ直下の “.” ではじまるファイル (ドットファイル) を編集することにより変更できます。これらのファイルはテキストファイルなので emacs などのテキストエディタで編集します。たとえば `.cshrc` を編集したいときは, emacs `.cshrc` で OK。

システム全体の環境設定は `/etc` 以下に書いてあるファイルを編集することにより変更できます。くわしくは unix の本をみてください。

Q. Asir でプログラムを書いて $\zeta(2)$ の値の高精度計算を試みましたが、時間とメモリを大量消費する。A. 有理数計算を多用すると約分に gcd 計算が多用されてメモリと時間を多用します。マニュアルの `ctrl("bigfloat",1); setprec` を参照し, `bigfloat` を用いて, かつ和を小さい方から足していくとすこしよくなります。

$\zeta(s)$ の計算には特別な高速アルゴリズムがあると思います (私しらず)。たとえば数論システムの pari では, 次のように計算します。

```
gp                unix shell より pari/gp を起動.
\p 200            有効桁数の設定.
zeta(2)
sqrt(6*zeta(2))  たしかに pi の値が...
```

Asir には pari がくみこまれていて,

```
setprec(200);
eval((pari(zeta,2)*6)^(1/2));
```

Q. `openxm fep asir` がうまくうごきません。

A. ホームの `.feprc` (`rc` は run command の略) を設定してますか?

Q. maxima で作成した図が保存できません。A. `plot_format` オプションを使います。たとえば

```
plot2d(sin(x), [x,0,10], [plot_format,ps]);
```

`gv` (`ghostview`) が立ち上がって図を表示するので, そこから save document します。詳しくは, “中川, Maxima 入門” を参照。

Q. Maxima の Todd-Coxeter アルゴリズムを試してみました。B 型のワイル群の位数を計算してみました。

A. Todd-Coxeter アルゴリズムの実装が一番しっかりしてるのは, GAP です。