

OpenGL による結び目の近傍抜き出しソフトウェア

A tool for construction of knots using OpenGL

谷 研究室 澄川 知弘
Tomohiro Sumikawa

概要

結び目をダイアグラムで扱い三角形分割をして平面に埋め込み、近傍を描画する

1 はじめに

自明性判定は結び目理論の基本的問題である。自明性判定とは自明な結び目との同型判定である。結び目の自明性判定において結び目の近傍を抜き出した空間を得ることは非常に有用である。そこで入力した結び目の近傍を抜き出し描画するソフトウェアを作成した。

1.1 グラフィカルな操作

結び目理論は数学の一分野である。一般に数学者はプログラマであるとは限らない。そこでソフトウェアの入力を視覚的にすることで扱いを容易にした。コンピュータを扱う機会の少ないものにとっても簡単に操作をすることが可能である。

1.2 可視化

数学者は実際に結び目を描いて同型性判定を考える。今まで結び目の近傍を見ることができたソフトウェアでプログラマ以外に対し操作が容易なものはなかった。故にソフトウェアの出力を映像で出力することとしたが、非常に有用である。

1.3 移植性

数学者は様々なコンピュータ環境で研究をしていることであろう。今まで結び目の近傍を見ることができたソフトウェアでどこでも稼動するものはなかった。そこでソフトウェアの開発ツールにはオープン仕様であるものだけを用いた。

2 球面

2.1 定義

n 次元球面は以下で定義される。

$$S^n = (x_0, x_1, \dots, x_n) \quad s.t.$$

$$x_0, x_1, \dots, x_n \in \mathbb{R}$$

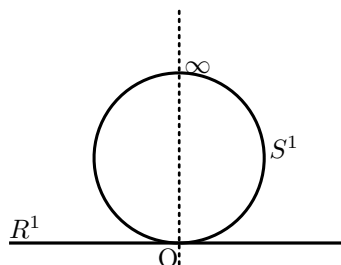
$$x_0^2 + x_1^2 + \dots + x_n^2 = 1$$

S^n は n 次元球面

2.1.1 球面とユークリッド空間の関係

$\infty \in S^n$ は無限遠の 1 点

$$R^n \cup \{\infty\} = S^n$$



3 結び目

3.1 結び目とは

直感的には、1 本の紐の両端を繋いだ輪である。正確には、空間内の自分自身では決して交わらないような閉曲線である。結び目は太さを持たず、簡単に変形できるゴムのようなものであり、切ってまた結び直すことは出来ないが、伸ばしたり縮めたり、ねじったり曲げることができる。ある結び目を変形させてできたものと、変形させる前の結び目とは区別をしない。つまり、変形してできるものは同じものと判断する。

3.2 定義

$$K : S^1 \rightarrow S^3$$

K は $Knot$

3.3 折線

線分は二つの点のペアである。 p_x, p_y を端点とする線分を $|p_x p_y|$ または $\overline{p_x p_y}$ などと表す。

折線は連続した線分の列である。

$$s_1 = |p_0, p_1|, s_2 = |p_1, p_2|, \dots, s_n = |p_{n-1}, p_n|$$

で $R = \{s_0 s_1 \dots s_{n-1}\}$ は折線.

頂点のリストを用いて $K(p_0 p_1 \dots p_n)$ とも表す.

3.4 単純閉折線

閉じた折線を閉折線とよぶ. $K(p_0 p_1 \dots p_n)$ のとき $p_0 = p_n$ となる折線が閉折線である. 閉折線は多角形である.

連続する線分の共有の端点以外に共有点を持たない折線を単純折線という. 折線上で隣り合っていない線分同士がどこかで接していない折線が単純折線である.

閉折線でありかつ単純折線なものは単純閉折線である.

3.5 結び目の表現

有限個の線分からなる単純閉折線として結び目を表現する.

4 結び目理論の基本的問題

4.1 同型判定

同型判定は結び目理論の基本的問題のひとつである.

4.2 同型の直観的イメージ

結び目は伸縮自在なゴム紐のようなものとして想像される. 伸ばす. 縮める. ひねる. 絡ませる. などの変形によって作ることのできる結び目は全て同型な結び目と思ふことができる.

4.3 同型の定義

$$\begin{aligned} K, K' &: \text{Knot} \\ \forall x &\in K \\ \exists h(x) &\in K' \\ \text{s.t.} & \quad h: S^3 \rightarrow S^3 \\ \implies & \quad K, K' \text{ は同型} \end{aligned}$$

4.3.1 同型判定のクラス

NP 困難よりも難しいとして知られている.

4.4 自明な結び目

$$\begin{aligned} K &: \text{Knot} \\ K &\subset S^2 \\ \implies & \quad K \text{ は自明} \end{aligned}$$

交差や絡みを持たないことが自明のイメージである.

4.5 自明性判定

自明性判定は結び目理論の基本的な問題のひとつである. 自明性判定は自明な結び目との同型判定と考えることができる.

4.6 自明性判定のクラス

NP に属するとして知られている.

5 複体

5.1 単体

$r+1$ 個の位置ベクトル $\forall a_0, a_1, \dots, a_r \in \mathbb{R}^n$ を各々 S^n の $r-1$ 次元以下の部分空間に含まれないとする.

$$\left\{ \sum_{i=0}^r \lambda_i a_i \mid \lambda_i \in \mathbb{R}, \sum_{i=0}^r \lambda_i = 1, \lambda_0, \lambda_1, \dots, \lambda_r \geq 0 \right\}$$

を n 次元単体と呼ぶ.

5.1.1 n 次元単体の例

代表的な n 次元単体は次の図形である.

0 次元単体: 点

1 次元単体: 線分

2 次元単体: 三角形

3 次元単体: 四面体

4 次元単体: 五胞体

5.2 定義

以下を満たすとき K は n 次元複体である.

1) 有限個の n 次元単体からなる

2) $\alpha \in K, \delta \in \alpha$

s.t.

$\alpha: n$ 次元単体

$\delta: n$ 次元未満の単体

$\implies \delta \in K$

3) $\forall \alpha, \beta \in K$

s.t.

$\alpha, \beta: n$ 次元単体

$\delta = \alpha \cap \beta$

$\implies \delta: n$ 次元未満の単体

5.2.1 n 次元複体のイメージ

n 次元複体は n 次元単体を密に並べたもののように見える.

6 近傍

6.1 空間の表現

このソフトウェアでは 3 次元複体として空間を表現する。

空間に含まれる線分の集合として結び目を描く。

6.2 定義

近傍は結び目に隣接した 3 次元単体の集合である。

7 重心細分

7.1 分割

次を満たす $\{\Delta_0, \Delta_1, \dots, \Delta_n\}$ は I の分割である。

- 1) $\Delta_i \in I$
 $s.t. \forall i \in \{0, 1, \dots, n\}$
- 2) $\Delta_i \cap \Delta_j = \phi$
 $s.t. i \neq j, \forall i, j \in \{0, 1, \dots, n\}$
- 3) $\Delta_0 \cup \Delta_1 \cup \dots \cup \Delta_n = I$

7.2 重心

r 次元単体の重心は $\frac{\sum_{i=0}^r a_i}{r}$ 記号 g で表す。

7.3 三角形への重心細分

7.3.1 記号の定義

位置ベクトル a, b, c により定義した 2 次元単体を Δabc と表す。

その重心は $\frac{a+b+c}{3}$ である。

7.3.2 定義

$\Delta a \frac{a+b}{2} g, \Delta a \frac{c+a}{2} g, \Delta b \frac{b+c}{2} g, \Delta b \frac{b+a}{2} g, \Delta c \frac{c+a}{2} g, \Delta c \frac{b+c}{2} g$ は Δabc の分割である。
 それらを Δabc の重心細分と呼ぶ。

7.4 四面体への重心細分

位置ベクトル a, b, c, d により定義される 3 次元単体の重心は $\frac{a+b+c+d}{4}$ である。

a, b, c, d により定義される 3 次元単体は $\Delta abc, \Delta abd, \Delta acd, \Delta bcd$ を含んでいる。

$\Delta abc, \Delta abd, \Delta acd, \Delta bcd$ を重心細分することで得られた各 2 次元複体の各頂点へのベクトルと 3 次元複体の重心へのベクトルにより新たな 3 次元複体を定義できる。その集合は a, b, c, d により定義される 3 次元単体の分割になっている。

その集合を a, b, c, d により定義される 3 次元単体の重心細分と呼ぶ。

7.5 近傍と重心細分

Hass *et al.* は結び目を含む空間全体を 2 回重心細分してから近傍を考えている。

ソフトウェアもそれに沿い近傍を定義。

8 ソフトウェアについて

8.1 入力

3 次元である結び目を 2 次元の射影図に投影したものを GUI で入力する。交点における辺の上下情報を抽出。

8.2 出力

入力から得られるデータを元に、結び目の近傍の映像をウィンドウ上に出力する。

8.3 開発言語

今回ソフトウェアを開発するにあたり、どこでも誰でも使用できることに重きを置き、LEDA ライブラリを使わずに OpenGL の機能だけを使った C 言語により開発を行う。グラフィックライブラリが使えないため、全て座標計算により処理していく。

8.4 LEDA

グラフ理論、ネットワーク問題、幾何学計算などの分野におけるアルゴリズムを提供する有償の C++ クラスライブラリ。

8.5 OpenGL

3 D グラフィックスのためのオープン仕様のプログラムインターフェイス。OpenGL に準拠した C 言語のライブラリに GLUT がある。

参考文献

- [1] Colin C. Adams
 The Knot Book — an elementary introduction to the mathematical theory of knots— (1994)
- [2] 床井 浩平
 GLUT による OpenGL 入門 (2005)
- [3] Joel Hass, Jeffrey C. Lagarias and Nicholas Pippenger
 "The Computational Complexity of Knot and Link Problems" Journal OF THE ACM Vol. 46 pp.185-211 (March 1999)