

ダイアグラムから作成する 結び目の近傍の抜き出し ソフトウェア

谷研究室 加藤 浩平
 田村 俊
2007年2月10日

目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

目次

- ◆ 導入 (結び目とは?・研究背景・目的)
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

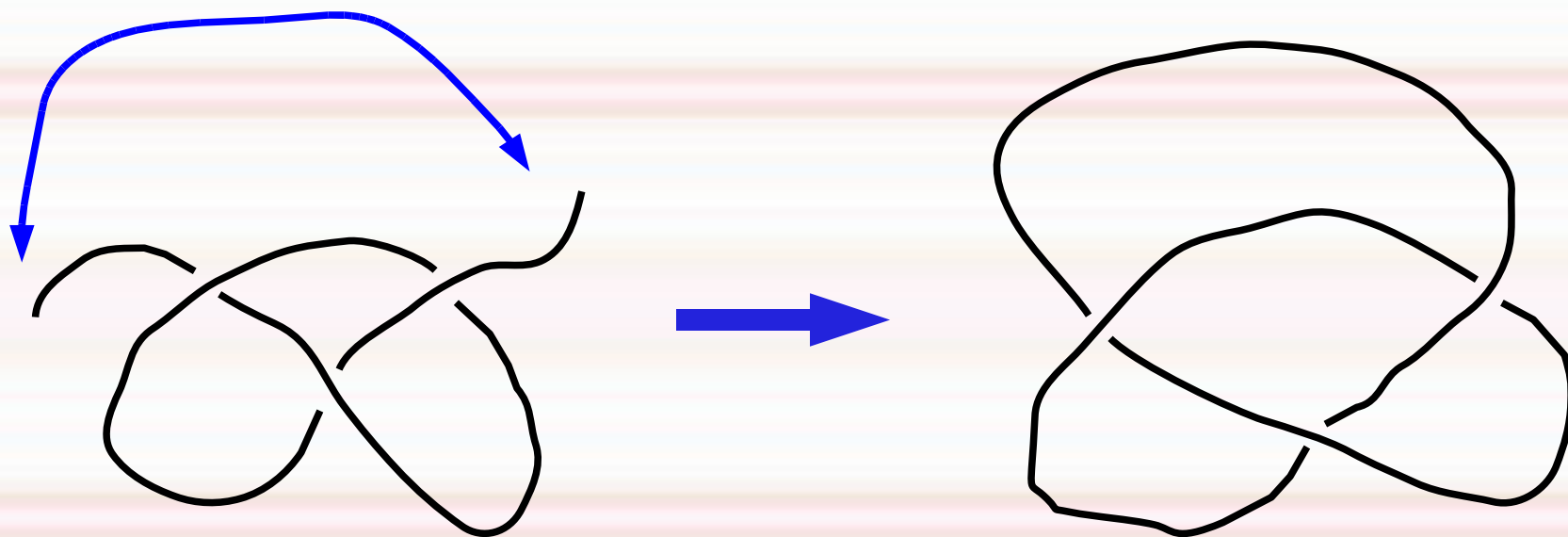
結び目の定義

3次元空間上の自己交差しない閉曲線



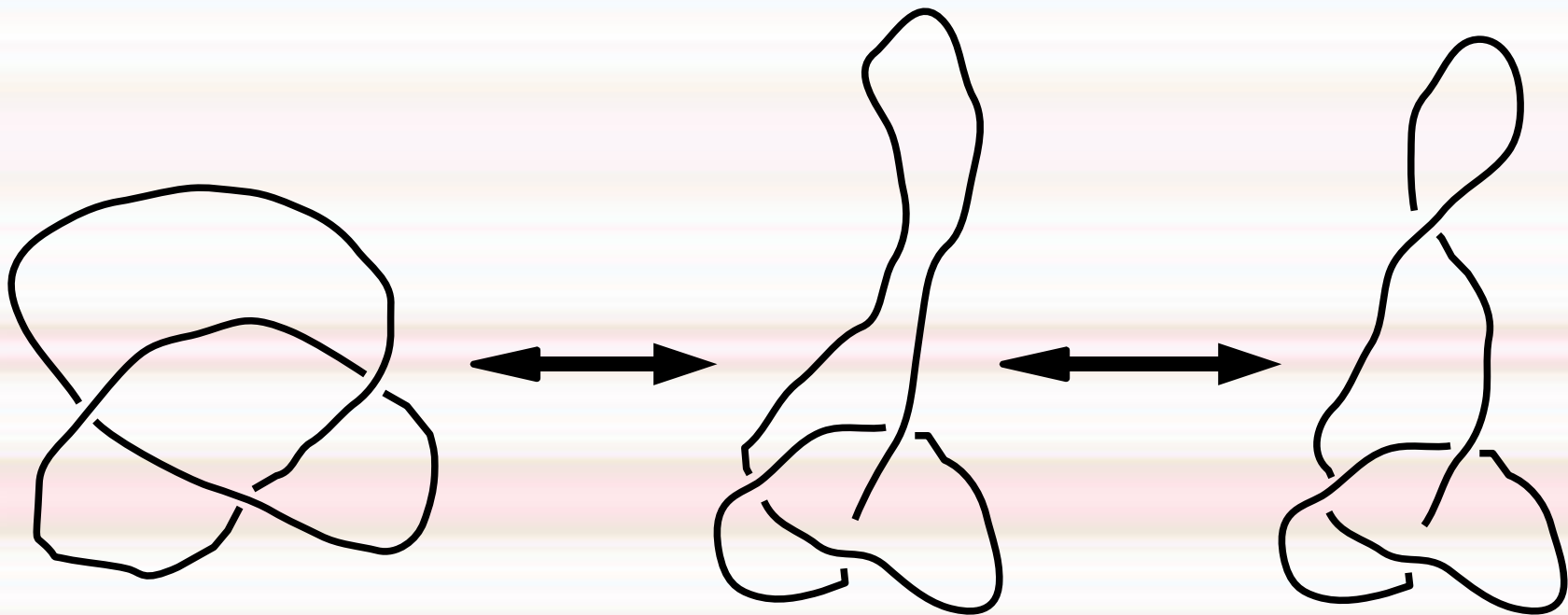
結び目のイメージ

- ・一本の紐の両端点を結んだ輪のイメージ

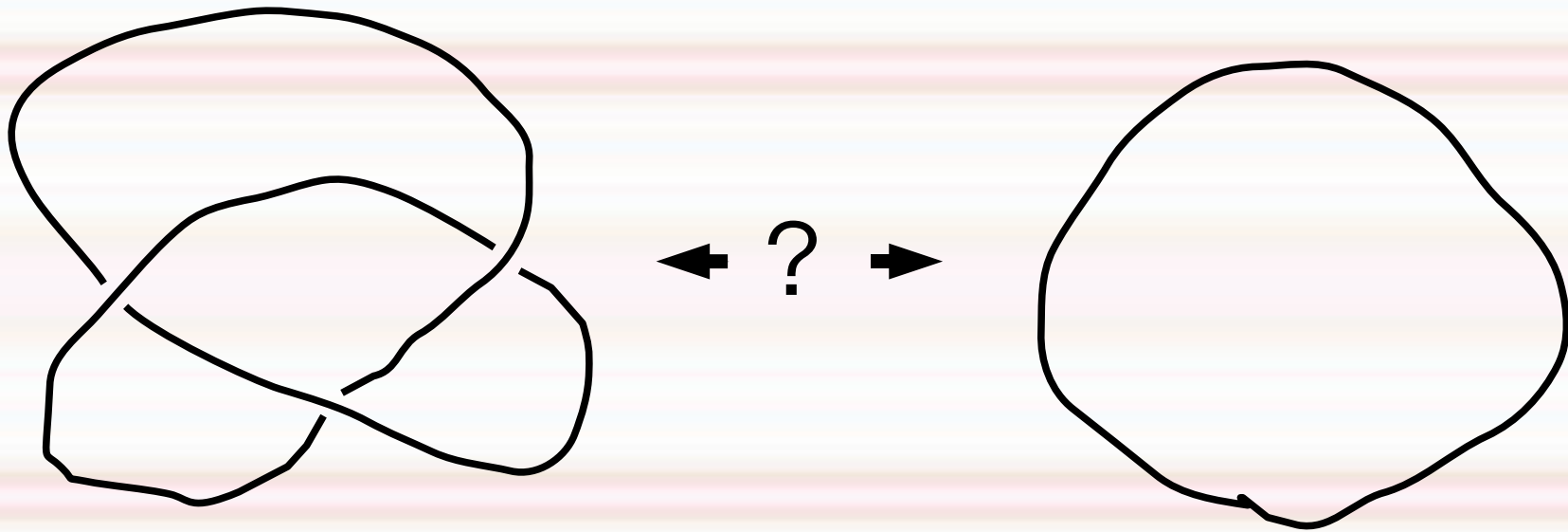


結び目の同値性

- ・切って再びつなげる事は不可能
- ・自在に伸縮可能



結び目理論の基本的問題



交点のない結び目に変形できるか

目的

結び目の近傍を抜き出すこと

結び目理論の応用

- ◆ 量子コンピュータ
- ◆ 遺伝子
- ◆ 暗号
- ◆ 生化学
- ◆ 合成化学
- ◆ 統計力学

図 1

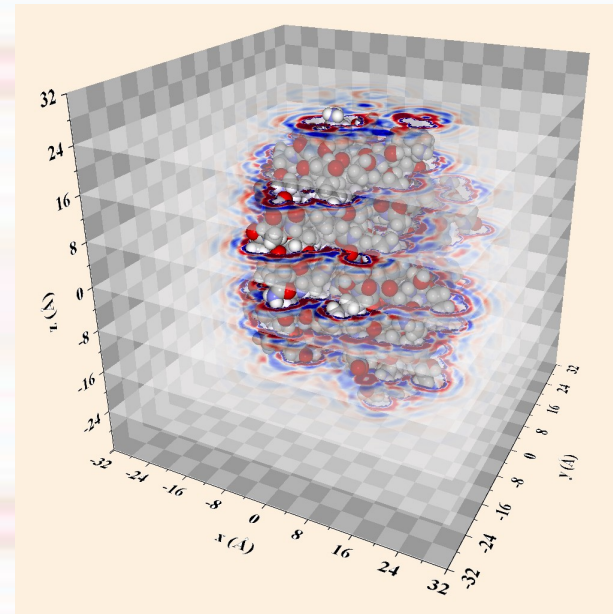
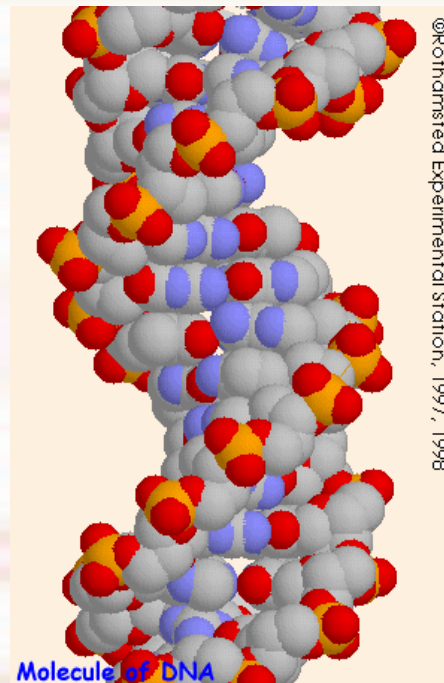


図 2



※図 1 <http://www.bio.is.ritsumei.ac.jp/~t-imai/research.html>

※図 2 <http://www.rothamsted.ac.uk/notebook/courses/guide/dnast.htm>

目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

計算の流れ

結び目の平面埋め込み

H. DE FRAYSSEIX達のアルゴリズム(1990)

3次元表示(結び目)

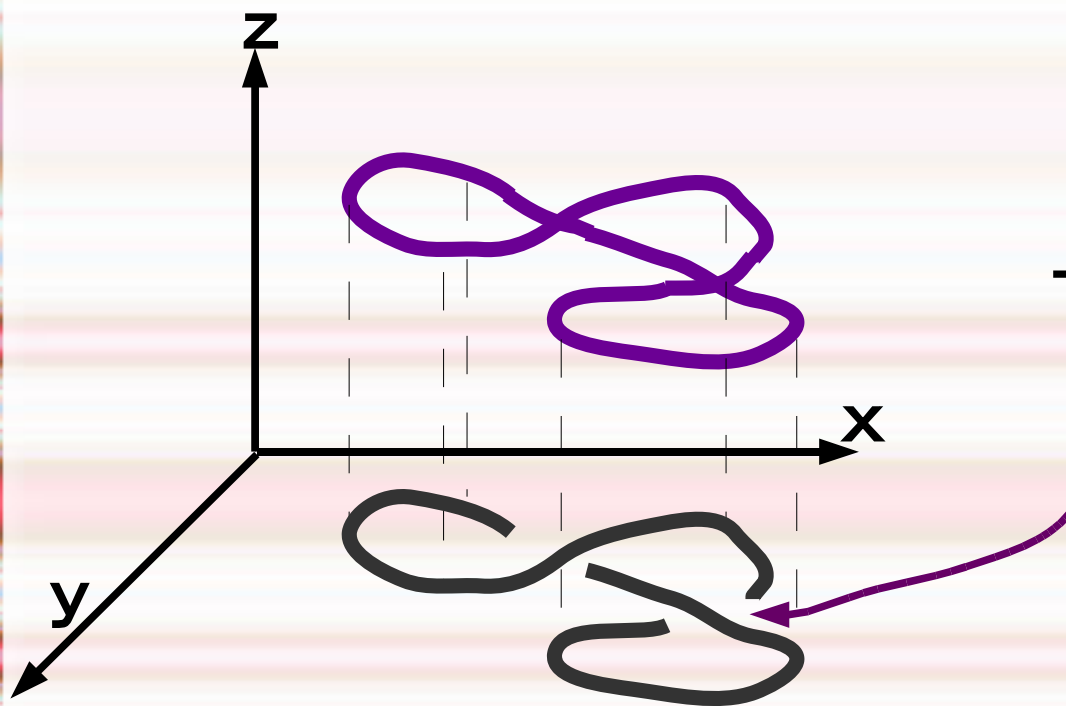
近傍抜き出し

JOEL HASS達のアルゴリズム(1999)

3次元表示(近傍)

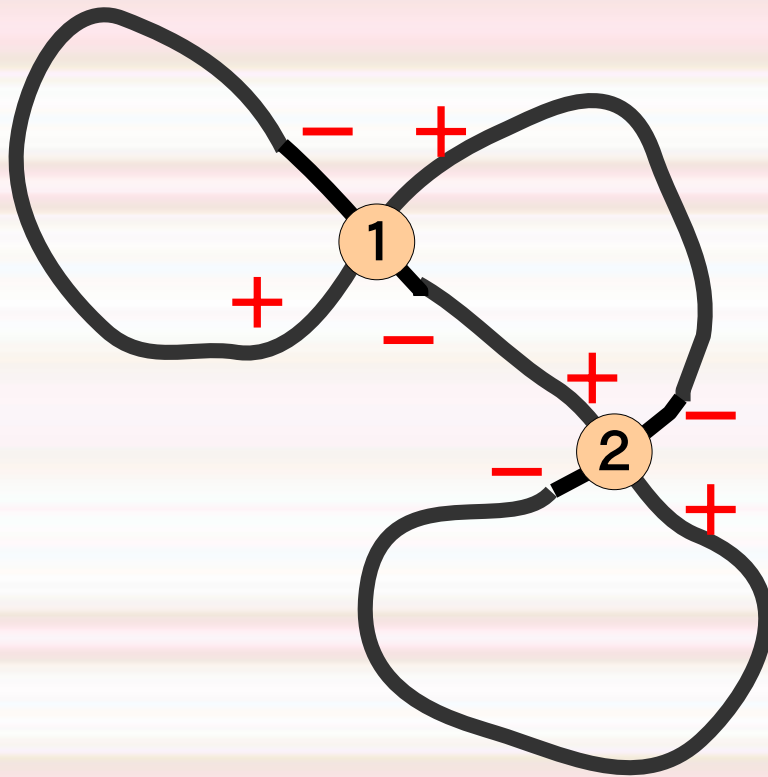
ダイアグラム(平面への射影図)の定義

- ・ 多重点は横断的な 2 重点のみを持ち
 - ・ 交点に交差の上下と辺の出る順番の情報をもつ
- グラフとして考える

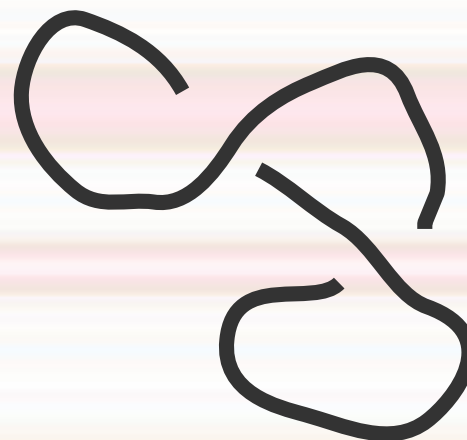


上下関係を入れる

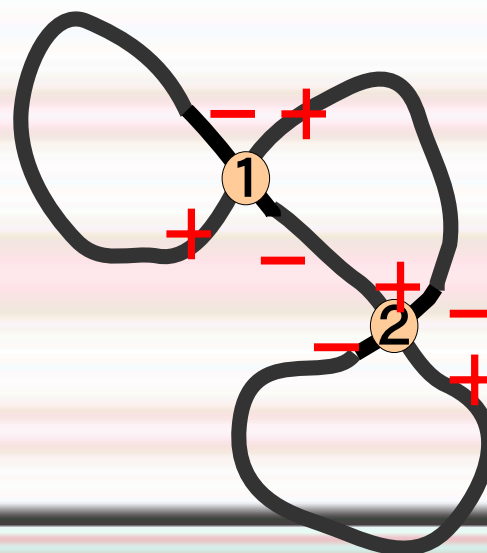
ダイアグラム(平面への射影図)



数学者は結び目を描くときダイアグラムで描く

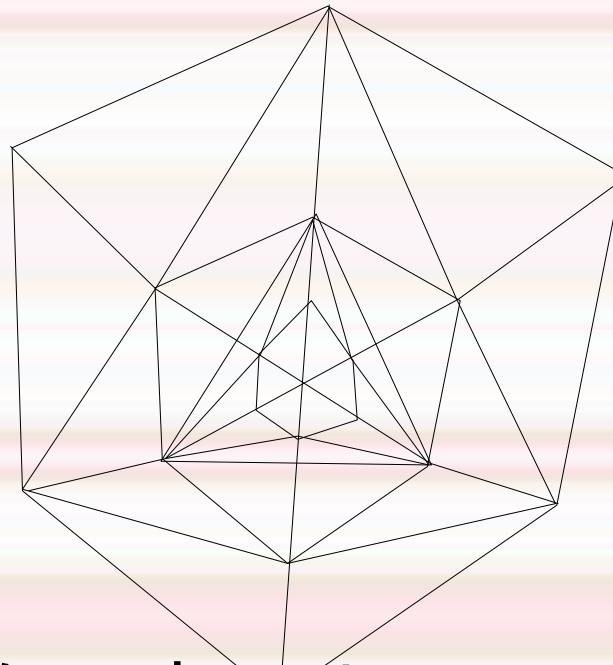


入力データをダイアグラムとしたソフトウェアを作成する



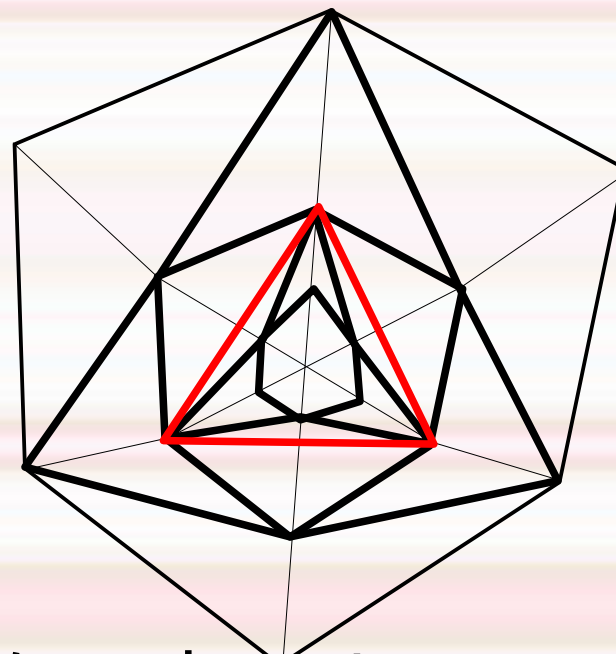
空間の考え方

三次元空間上では四面体が密になって並んでいるもの考える



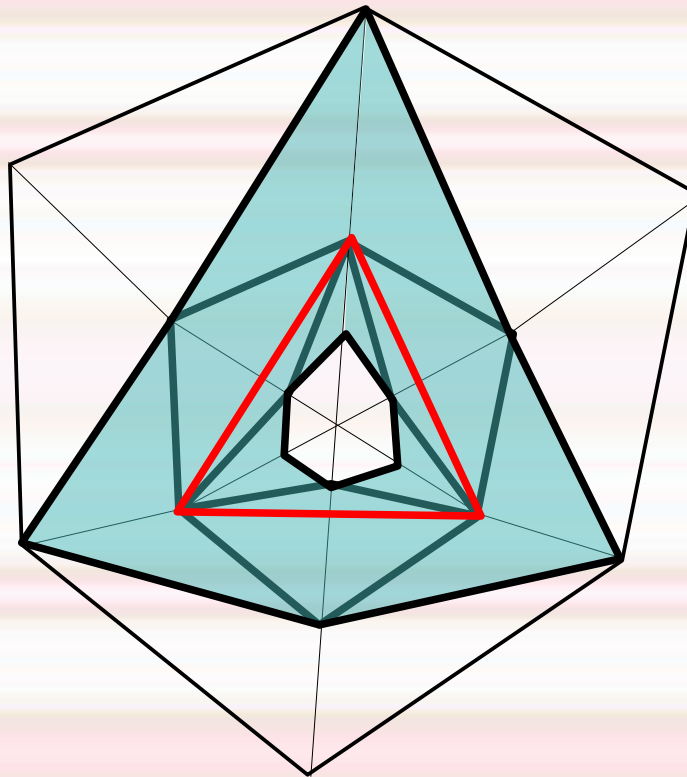
三次元空間上の辺に結び目を描く

近傍



三次元空間上の辺に結び目を描く

近傍



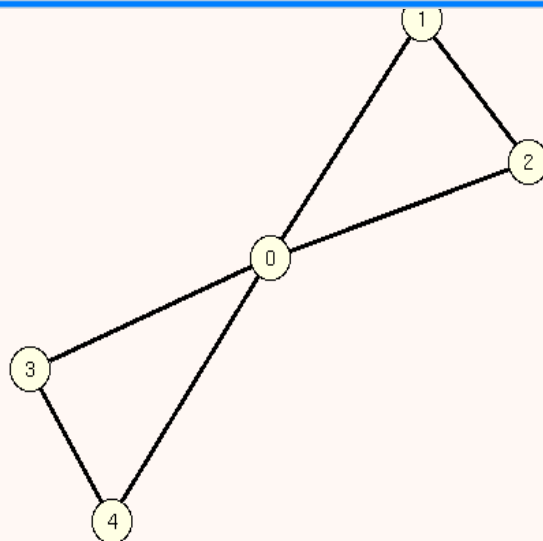
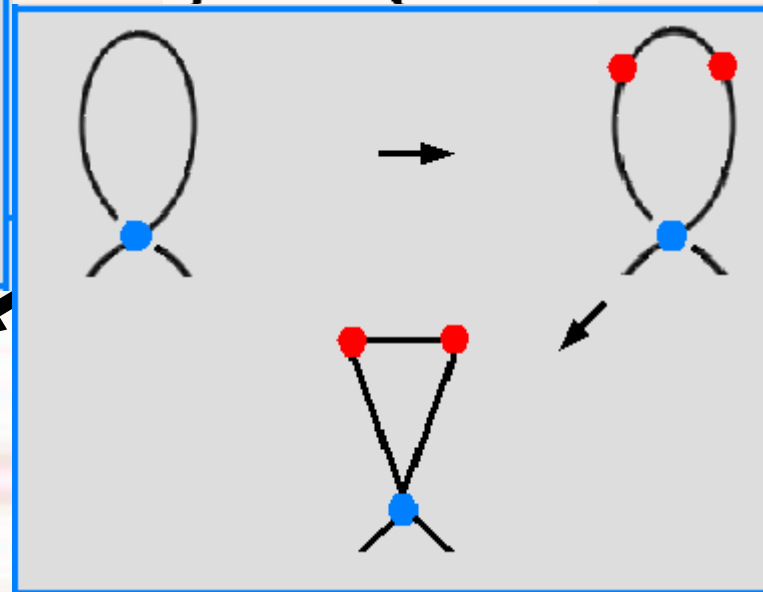
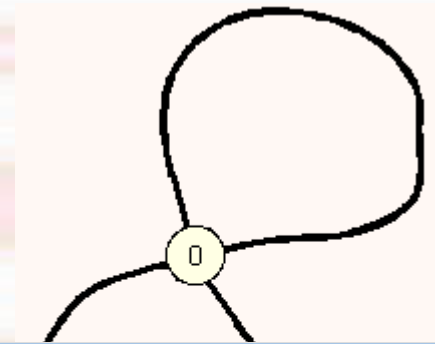
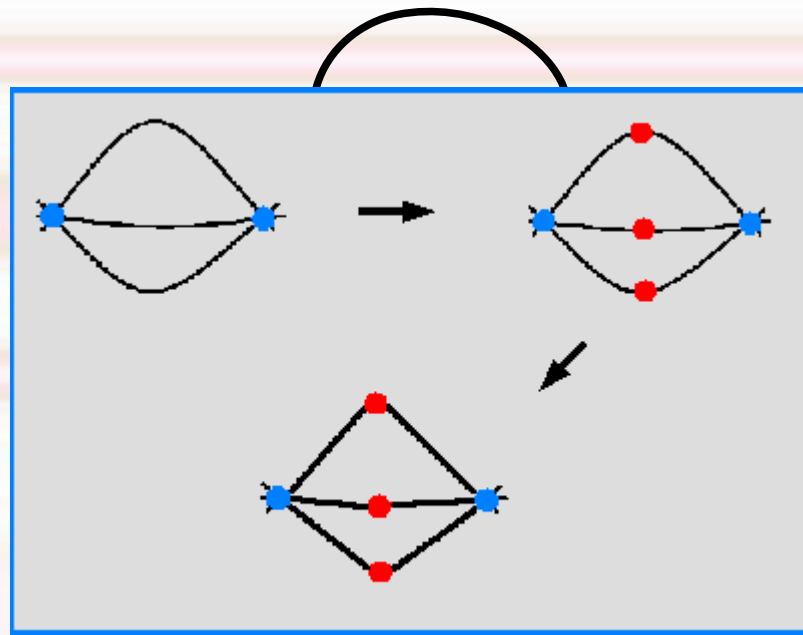
結び目に隣接した四面体の集合 = 結び目の近傍

	田村	加藤
結び目の平面埋め込み	C++	
3次元表示(結び目)	OpenGL(C++)	Java3D(Java)
近傍抜き出し		Java
3次元表示(近傍)	OpenGL(C++)	Java3D(Java)

目次

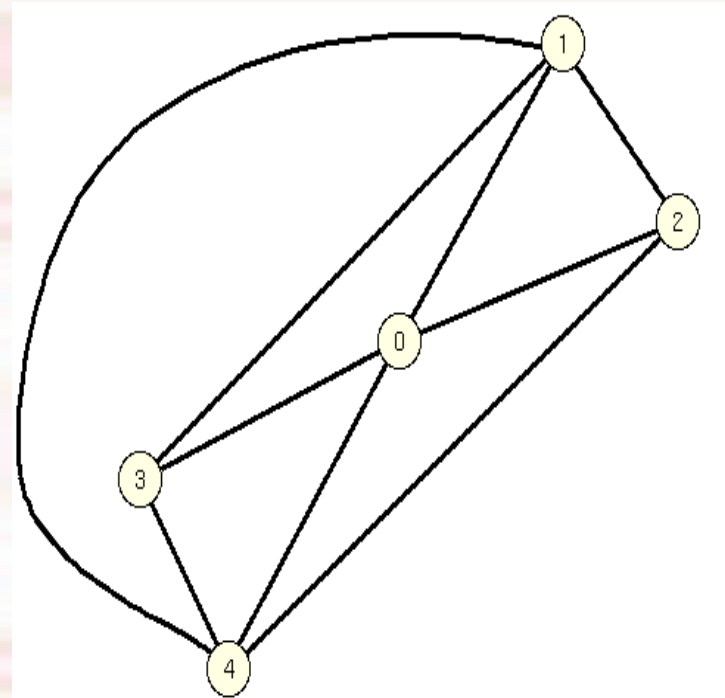
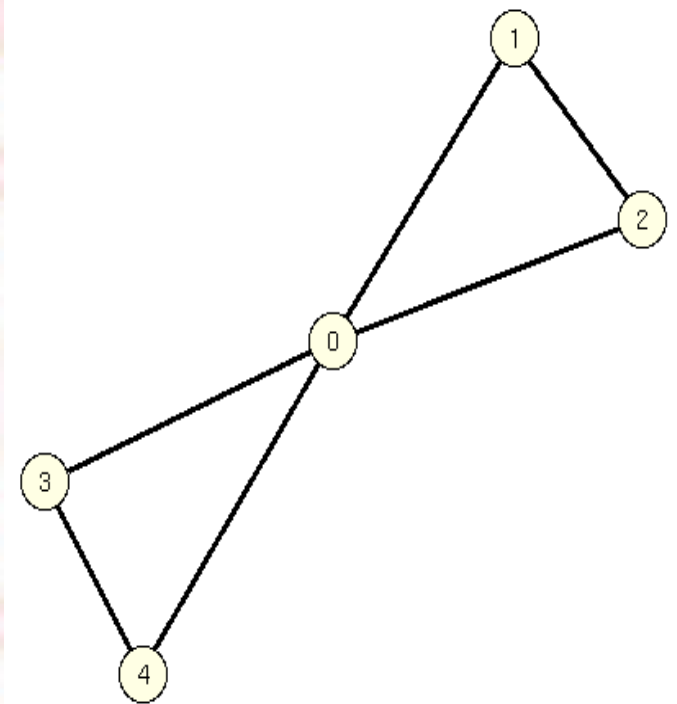
- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ **結び目の平面埋め込みと3次元表示**
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

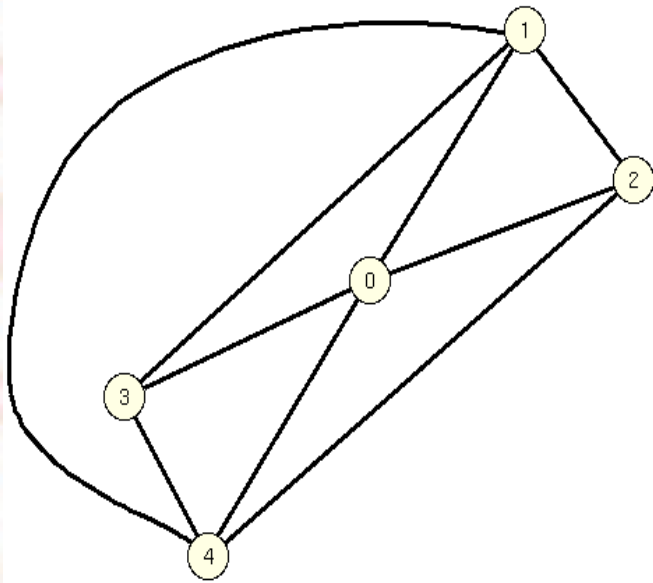
入力したいダイアグラム



辺の分割

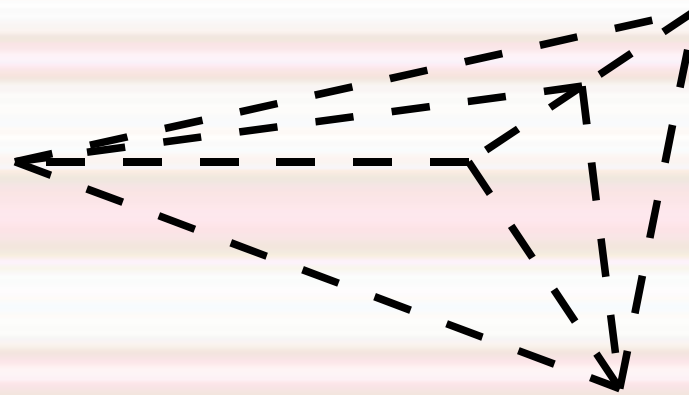
三角形分割





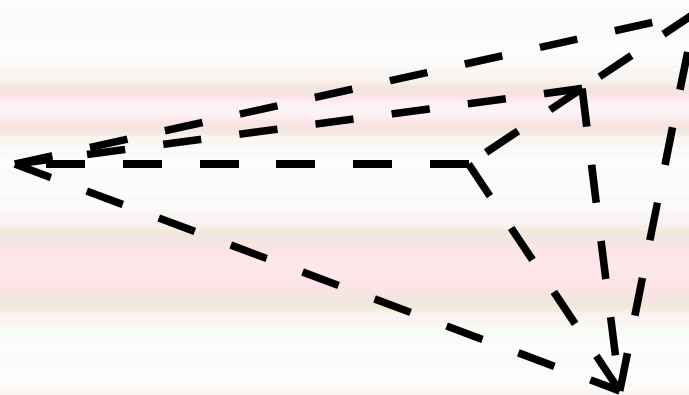
H. DE FRAYSSEIX 達のアルゴリズム (1990)

全ての辺が直線かつ交差しないように頂点に
 x, y 座標を与える

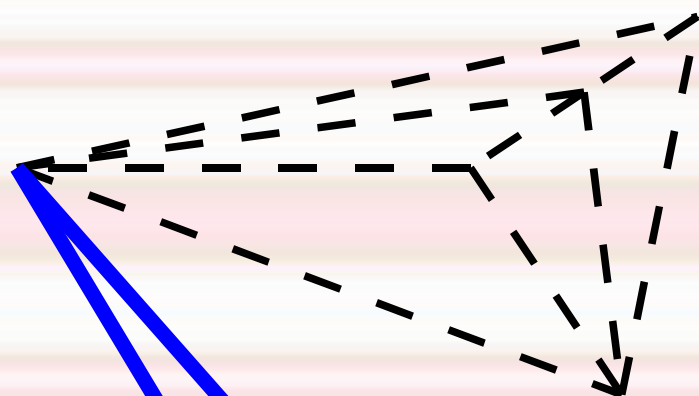


$$z=1$$

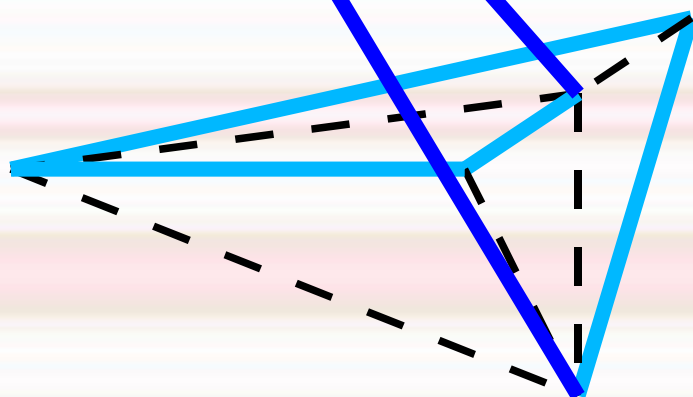
結局、座標を差の情報からアップの場合は $z \neq 1$ - それの
値を送るに出力する



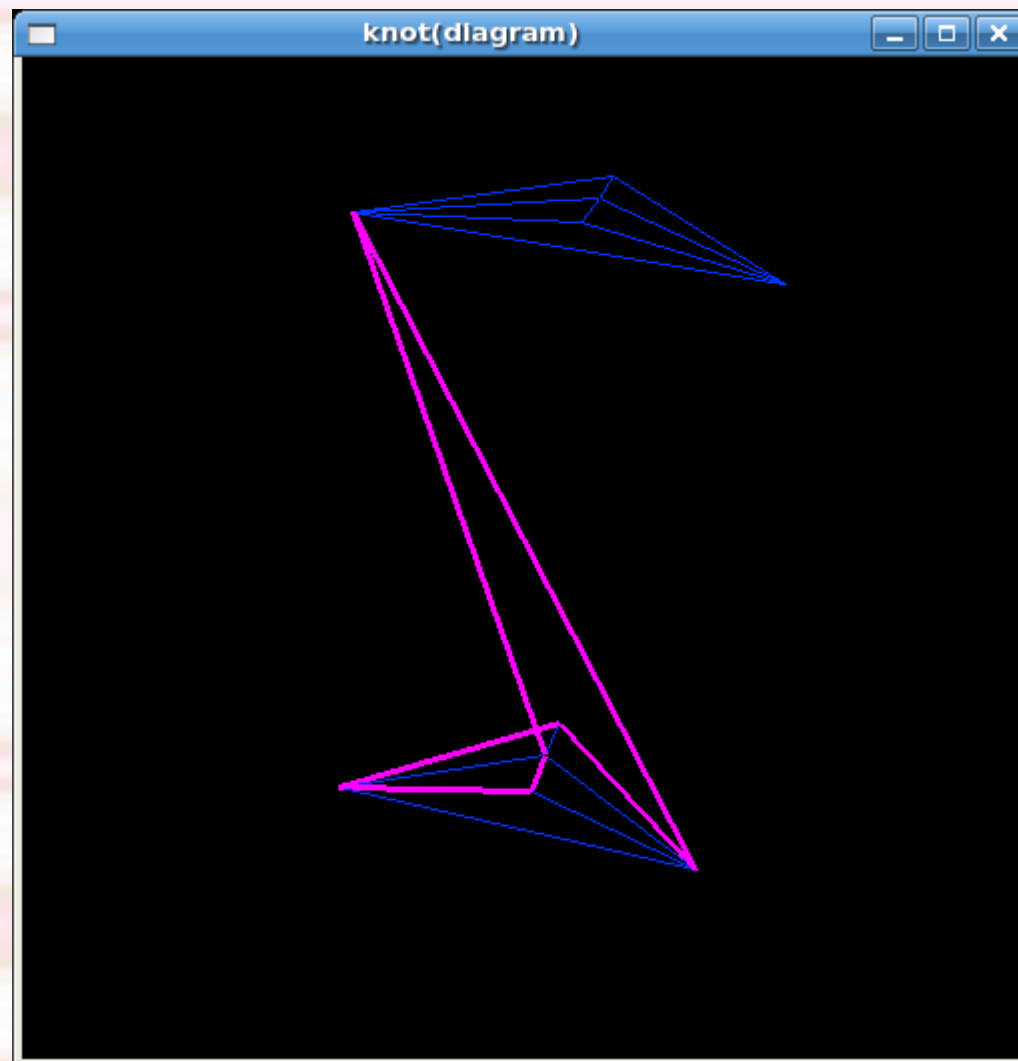
$$z=-1$$



$z=1$

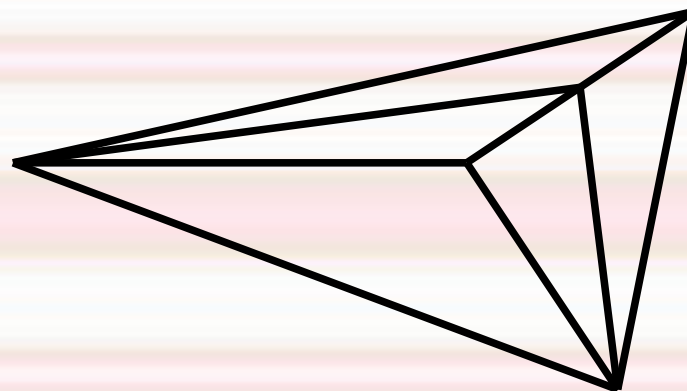


$z=-1$

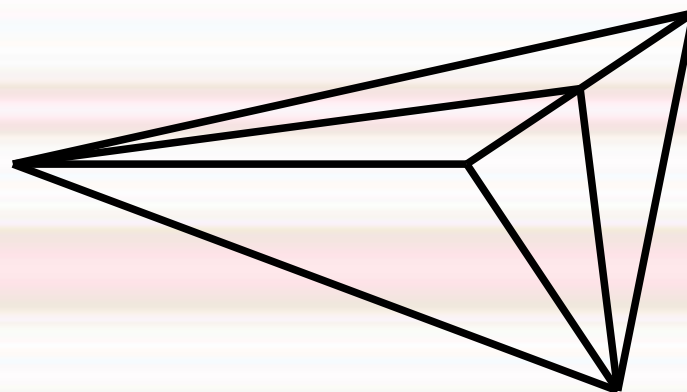


目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

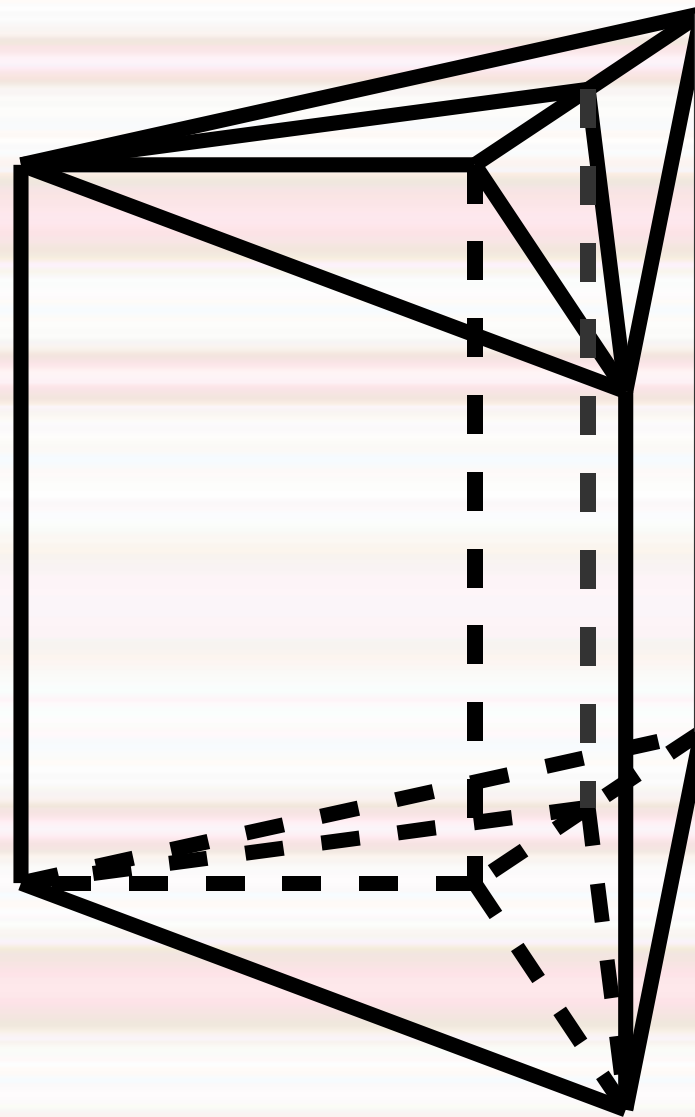


$z=1$



$z=-1$

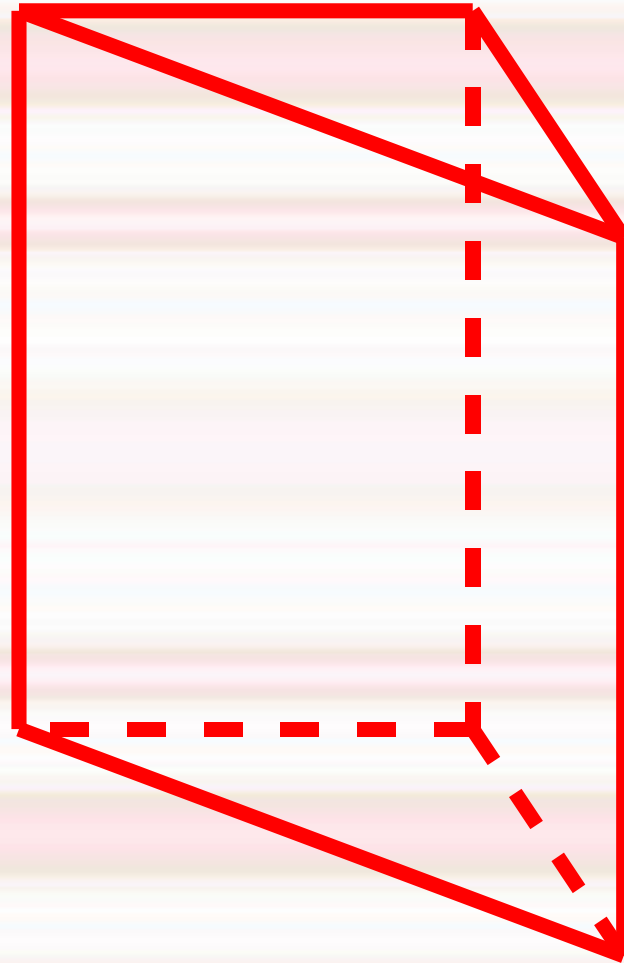
三角柱をつくる



$z=1$

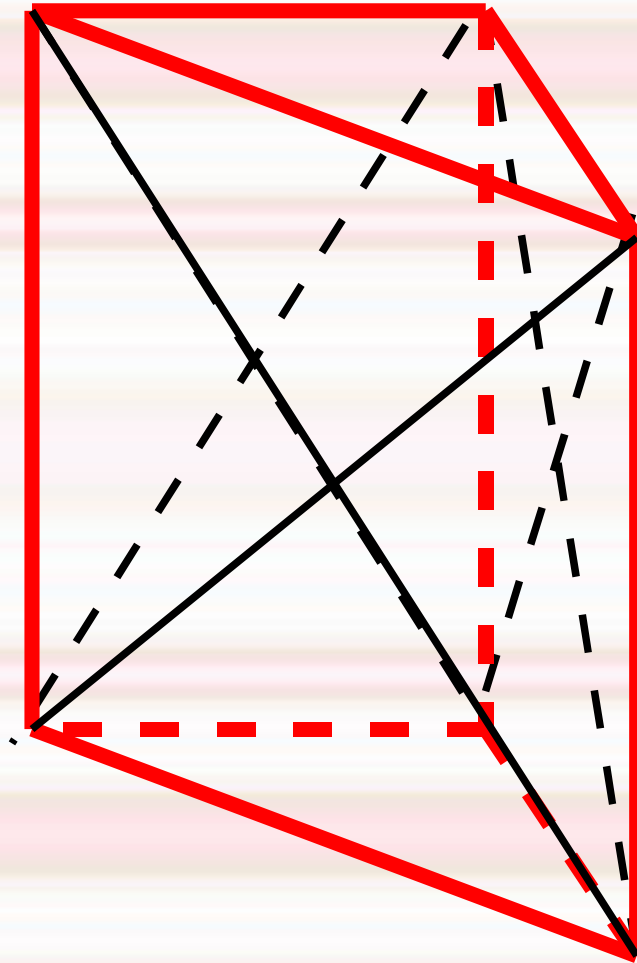
$z=-1$

三角柱を四面体に分割する



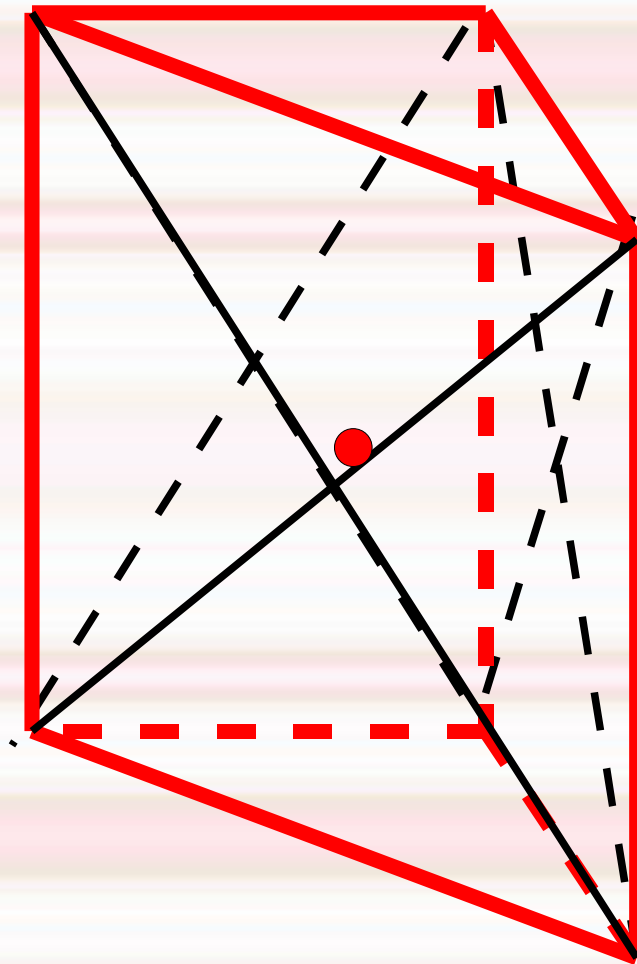
- ・側面に対角線を引く

三角柱を四面体に分割する



・三角柱の重心
を求める

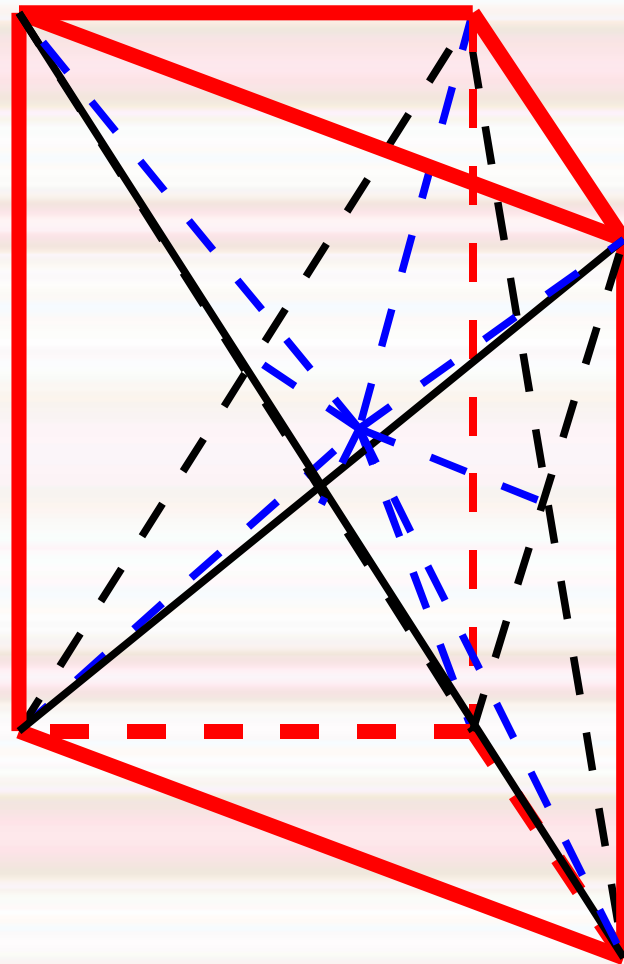
三角柱を四面体に分割する



- ・ 三角柱の重心と点と対角線の交点を結ぶ

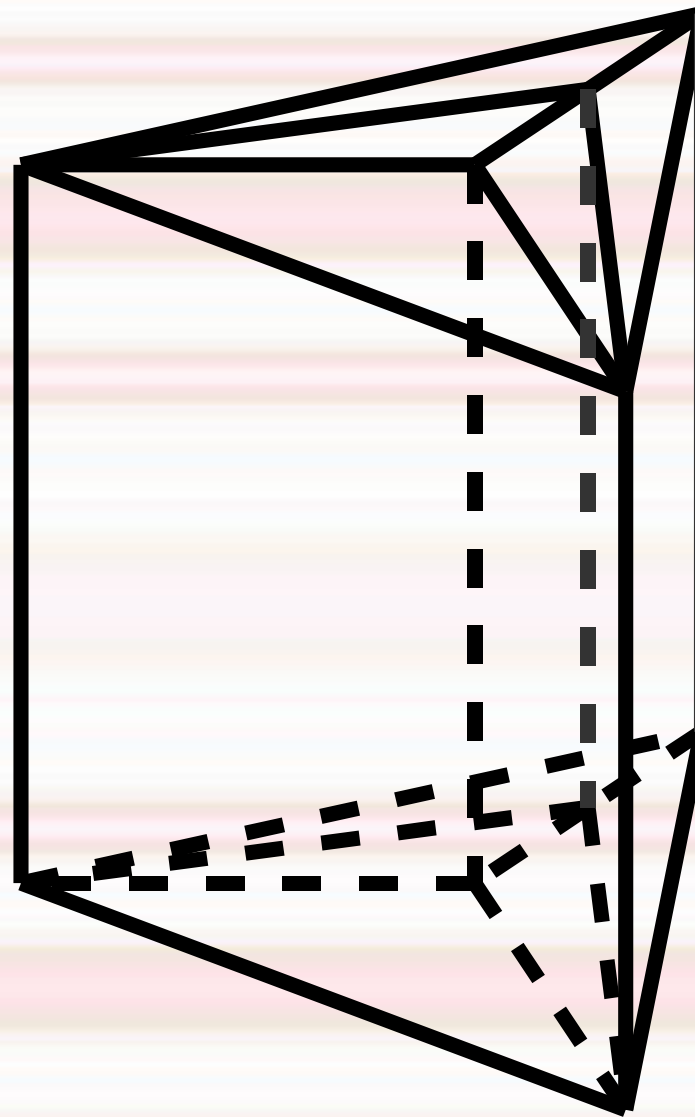
- ・ 14個の四面体に分割できる

三角柱を四面体に分割する

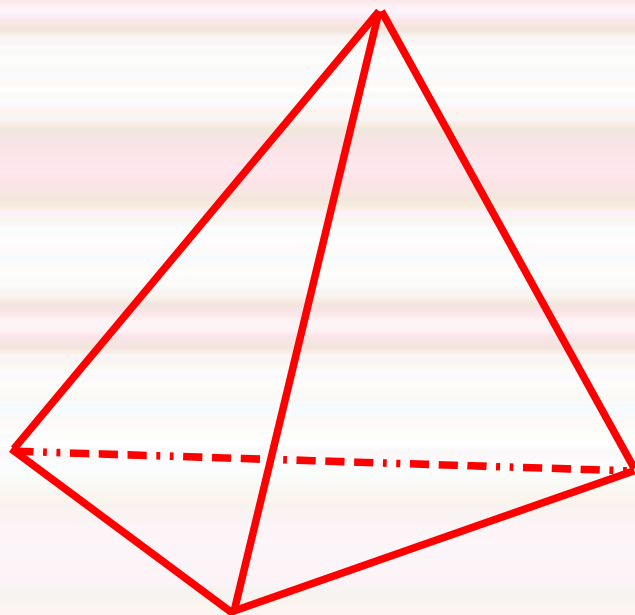


- ・ 三角柱の重心と点と各三角形の頂点を結ぶ
- ・ 14個の四面体に分割できる

三角柱をつくる

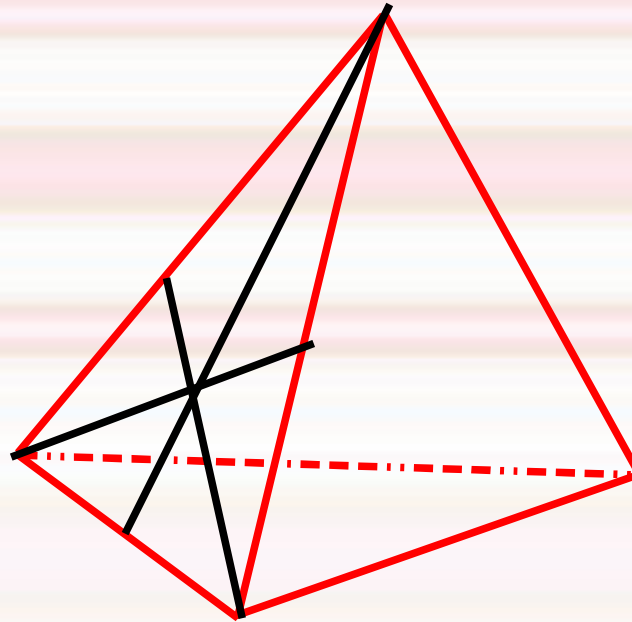


四面体を24個の四面体に分割する

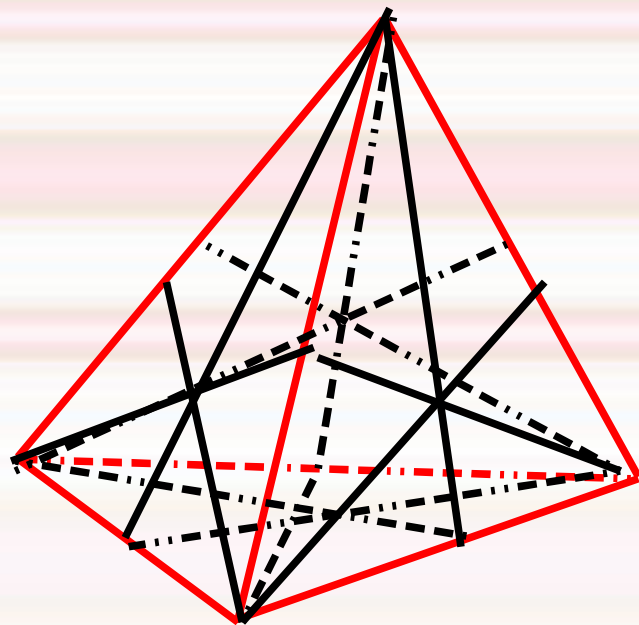


- ・ 四面体を構成する面を6つの三角形に分ける

四面体を24個の四面体に分割する

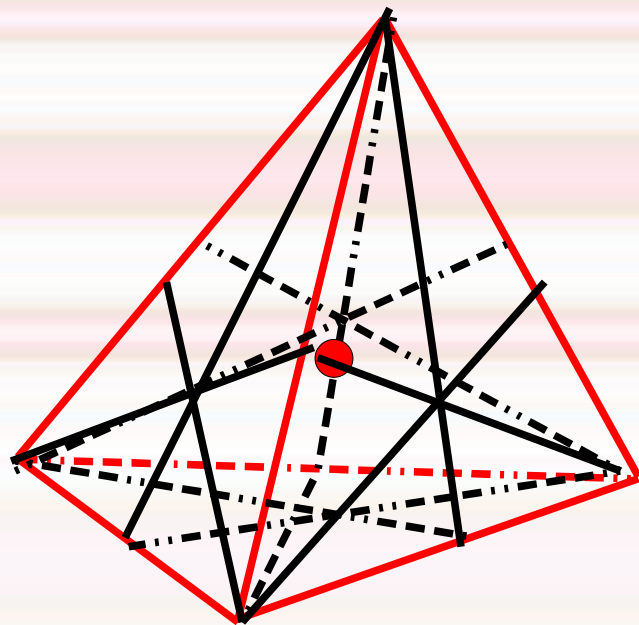


四面体を24個の四面体に分割する



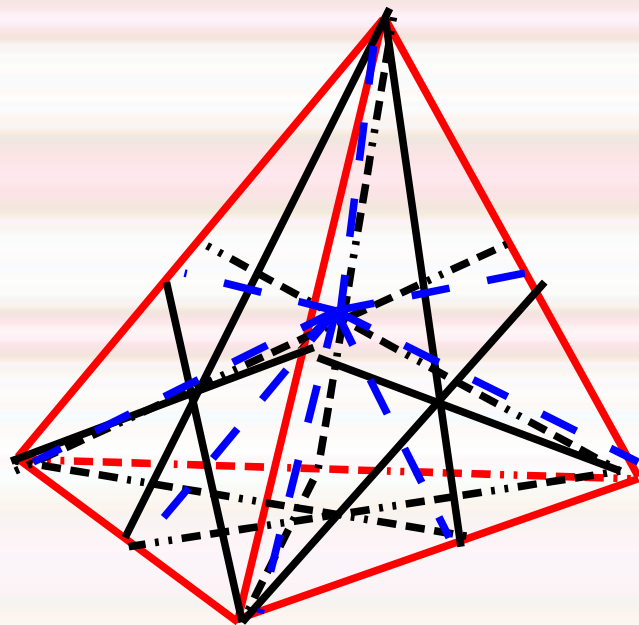
- ・ 四面体の重心を求める

四面体を24個の四面体に分割する

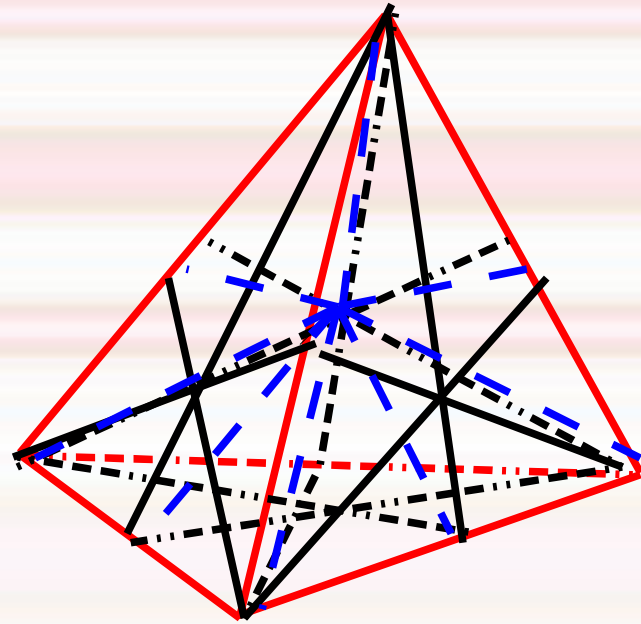


- ・ 四面体の重心と各三角形の頂点を結ぶ
- ・ 24個の四面体が出る

四面体を24個の四面体に分割する



- ・できた四面体を再び今の工程で24個の四面体に分割する



結び目の辺と隣接した四面体だけを抜取る

目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

*OpenGL*と*Java3D*の特徴

OpenGL : 実行時間が速い
OSに依存しない(OSごとにコンパイル)
グラフィックボードに依存する

Java3D : 実行時間が遅い
一度コンパイルしてしまえばどのOSでも
実行出来る
グラフィックボードに依存しない

OpenGLとJava3Dの 近傍の表示時間の比較

ダイアグラムの交 点数・四面体の数		OpenGL(C++)	Java3D(Java)
1個	111168	2.094(s)	6.210(s)
4個	498720	9.843(s)	19.777(s)
8個	1029120	19.249(s)	37.132(s)

目次

- ◆ 導入（結び目とは？・研究背景・目的）
- ◆ 計算の流れ
- ◆ 結び目の平面埋め込みと3次元表示
- ◆ 四面体分割と近傍の抜き取り
- ◆ デモンストレーション
- ◆ 比較
- ◆ 考察・今後の課題

考察

ダイアグラムの数値を入力することで近傍を求めることができたので数学者が紙に描いたダイアグラムの近傍を出力することはできた

近傍を求めることができたのでそこから交点のない結び目と同値かを判定するための基盤ができた

今後の課題

直接画面に描いた結び目からも近傍が求められるように対応させる

終