

oniprog ライブラリの説明書

最終更新 2013/06/16

Oniprog

はじめに

このマニュアルはおおよそ何がプログラム中にあるかを把握することを目的とします. 詳細については, ソースを読解してください. また, ユニットテストも書いてあります. そのテストを見れば, 各クラスや関数のおおよその使い方がわかることを意図しています.

Todo 事項

- 制約付きドロネー分割への対応
- 3次元ドロネー分割への対応
- 2次元メッシュ分割への対応
- 3次元描画エンジンの実装
- クリッピング計算の実装

GeoLib

Point2.h

2次元点クラス

たとえば, `GeoLib::get<1>(poi)` にて Y 座標を得ることができる.

Point3.h

3次元点クラス

Math.h

CrossProduct_2	2次元内積計算
DotProduct_2	2次元外積計算
Determinant2x2	2x2の行列式計算
Determinant3x3	3x3の行列式計算
Orient_2	

Orient2 計算. 3点 P1,P2,P3 があり, P1 から P2 に向かう直線に対して, P3 が左にあるとき正の値, 右にあるとき負の値, 直線上にあるとき 0 となる判定関数. ただし, 後述する OrientExact_2 を普通は用いる方がよい.

Orient_3

4点 P1,P2,P3,P4 があるとき, P1,P2,P3 が作る三角形の法線方向に点 P4 があるとき正の値. 反対方向に点があるとき負の値となり, 面上に点があるとき 0 となる関数.

DeterminantAbs2x2	2x2の行列式ですべての値を絶対値で計算する
DeterminantAbs3x3	3x3の行列式ですべての値を絶対値で計算する
OrientAbs_2	Orient2を絶対値で計算したもの
OrientExact_2	適応型無誤差演算を用いた Orient_2
InCircle	

4点 P1,P2,P3,P4 があるとき, 点 P1,P2,P3 で定義される円に対して, 点 P4 が内部にあるとき正となる. ただし, P1,P2,P3 の Orient_2 が正の値のとき. 負の値をとるときは, 解の符号が逆転する.

InCircleExact	InCircleの適応型無誤差演算版
----------------------	---------------------------

Rational.h

boost::multiprecision を用いて Rational 型を定義している。GeoLib::static_rational_cast<double>(val)にてこの型の値を浮動小数点数に変換できる。

Distance.h

GetDistancePP_2 **2 点間の 2 次元距離を計算する**

CrossPoint.h

IsCrossSegment_2 **2 線分に交点があるかを判定する。**

GetCrossLine_2 **2 直線の交点を計算する。**

Delaunay.h

Delaunay2D **2 次元ドロネー分割クラス**

Apply メソッドに点群(たとえば Point_2)を渡せば、それをドロネー分割する。

結果は、begin, end で得ることができる。

無誤差演算を用いて、摂動をかけているので、ロバストな実装になっています。

CrossSegment.h

CrossSegment_2T

交差する線分同士に、交点を追加します。また、近距離にある点を統合します。ただし、平行(に近い関係にある)2 線分の処理は入っていません。(手抜きです。リクエストがあればがんばって入れるかも)

Apply メソッドにたとえば std::vector<Point_2>のイテレータを渡します。なお、2 つ組で扱われます。2 つとも同じ点のときは、点をあらし、2 点が異なるときは、それは線分を表している、という意味になります。

計算結果は、制約付きドロネー分割のクラスに渡しやすいように、ポインタとなっています。

以上