

# Oniprog(おにっぷ) ライブラリ説明書

最終更新 2014/11/10

Oniprog

はじめに .....	3
TODO 事項.....	3
GEOLIB.....	4
POINT2.H .....	4
POINT3.H .....	4
MATH.H.....	4
<i>CrossProduct_2</i> 2次元外積計算 .....	4
<i>DotProduct_2</i> 2次元内積計算.....	4
<i>Determinant2x2</i> 2x2 の行列式 .....	4
<i>Determinant3x3</i> 3x3 の行列式 .....	4
<i>Orient_2</i> .....	4
<i>Orient_3</i> .....	4
<i>DeterminantAbs2x2</i> 2x2 の行列式ですべての値を絶対値で計算したもの .....	4
<i>DeterminantAbs3x3</i> 3x3 の行列式ですべての値を絶対値で計算したもの .....	4
<i>OrientAbs_2</i> <i>Orient2</i> を絶対値で計算したもの.....	4
<i>OrientExact_2</i> 適応型無誤差演算を用いた <i>Orient_2</i> .....	4
<i>InCircle</i> .....	5
<i>InCircleExact</i> .....	5
RATIONAL.H .....	5
DISTANCE.H .....	5
<i>GetDistancePP_2</i> 2点間の2次元距離を計算する.....	5
CROSSPOINT.H.....	5
<i>IsCrossSegment_2</i> 2線分に交点があるかを判定する. ....	5

<i>GetCrossLine_2</i> 2 直線の交点を計算する. ....	5
DELAUNAY.H.....	5
<i>Delaunay2D</i> 2 次元ドロネー分割クラス.....	5
CROSSSEGMENT.H .....	5
<i>CrossSegment_2T</i> .....	5

## 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
- クリッキング計算の実装

# Geolib

---

## Point2.h

2次元点クラス.

例えば, `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次元外積計算

DotProduct\_2                        2次元内積計算

Determinant2x2                      2x2 の行列式

Determinant3x3                      3x3 の行列式

### Orient\_2

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` を普通は用いるとよい.

### 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, 面上に点があるとき 0 となる判定関数.

DeterminantAbs2x2    2x2 の行列式ですべての値を絶対値で計算したもの

DeterminantAbs3x3    3x3 の行列式ですべての値を絶対値で計算したもの

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 が正の値のときであり，負の値となるときは，解の符号が逆転する．

## InCircleExact

InCircle の適応型無誤差演算バージョン

## Rational.h

boost::multiprecision を用いて Rational 型を定義している．無誤差演算のためのクラス．GeoLib::static\_rational\_cast<double>(val)にてこの型の値を浮動小数点値に変換できる．

## 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，交点を追加する．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）

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、点を表し、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クラスに渡しやすいように、ポインタとなります。

以上