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](#_Toc403377528)

[Todo事項 3](#_Toc403377529)

[Geolib 4](#_Toc403377530)

[Point2.h 4](#_Toc403377531)

[Point3.h 4](#_Toc403377532)

[Math.h 4](#_Toc403377533)

[CrossProduct\_2 2次元外積計算 4](#_Toc403377534)

[DotProduct\_2 2次元内積計算 4](#_Toc403377535)

[Determinant2x2 2x2の行列式 4](#_Toc403377536)

[Determinant3x3 3x3の行列式 4](#_Toc403377537)

[Orient\_2 4](#_Toc403377538)

[Orient\_3 4](#_Toc403377539)

[DeterminantAbs2x2 2x2の行列式ですべての値を絶対値で計算したもの 4](#_Toc403377540)

[DeterminantAbs3x3 3x3の行列式ですべての値を絶対値で計算したもの 4](#_Toc403377541)

[OrientAbs\_2 Orient2を絶対値で計算したもの 4](#_Toc403377542)

[OrientExact\_2 適応型無誤差演算を用いたOrient\_2 4](#_Toc403377543)

[InCircle 5](#_Toc403377544)

[InCircleExact 5](#_Toc403377545)

[Rational.h 5](#_Toc403377546)

[Distance.h 5](#_Toc403377547)

[GetDistancePP\_2 2点間の2次元距離を計算する 5](#_Toc403377548)

[CrossPoint.h 5](#_Toc403377549)

[IsCrossSegment\_2 2線分に交点があるかを判定する． 5](#_Toc403377550)

[GetCrossLine\_2 2直線の交点を計算する． 5](#_Toc403377551)

[Delaunay.h 5](#_Toc403377552)

[Delaunay2D 2次元ドロネー分割クラス 5](#_Toc403377553)

[CrossSegment.h 5](#_Toc403377554)

[CrossSegment\_2T 5](#_Toc403377555)

# 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

・クリッギング計算の実装

# 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