

計算幾何

(Ver.1.3)

2015 年 8 月

株式会社 アイディール

目次

eyemCgSortByArgument.....	1
機能 点群の偏角によるソート.....	1
eyemCgConvexHull	2
機能 点群の凸包.....	2
eyemCgSmallestEnclosingCircle	3
機能 点群の最小包含円.....	3
eyemCgIntersectionOfTwoSegments	4
機能 2線分の交差状態.....	4
eyemCgPointInConvexPolygon.....	6
機能 点の凸多角形内外判定	6
eyemCgIntersectionLineAndConvexPolygon.....	7
機能 直線と凸多角形の交点.....	7
eyemCgIntersectionSegmentAndConvexPolygon.....	8
機能 線分と凸多角形の交点.....	8
eyemCgPointInPolygon	9
機能 点の多角形内外判定	9
eyemCgAreaOfPolygon.....	10
機能 多角形の面積.....	10

eyemCgSortByArgument

機 能 点群の偏角によるソート

形 式

```
#include "eyemLib.h"

int      eyemCgSortByArgument ( int n, EyemOcsDXY taSrcPt[],
                                EyemOcsDXY taDstPt[] );
```

解 説 2次元点群に対して、 y 座標が最小な点 (y 座標が等しい場合は x 座標が最小な点) を始点とし、偏角 (各点と始点を結ぶ線分が x 軸となす角) にしたがって昇順にソートします。なお、偏角が同じ点については、始点からの距離にしたがって昇順にソートします。

引 数

n	データ(座標)点の数です。2以上を入力してください。
$taSrcPt[]$	各点の2次元座標値です。2点以上を入力してください。
$taDstPt[]$	ソート結果の座標値が格納されます。‘ $taSrcPt$ ’と同じ配列を指定することができます。

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgConvexHull

機 能 点群の凸包

形 式

```
#include "eyemLib.h"

int      eyedCgConvexHull ( int n, EyemOcsDXY taPoint[], int *ipCnvxNum
                          EyemOcsDXY taCnvxPt[] );
```

解 説 2次元点群に対して、それらの凸包(点群を含む最小の凸多角形)を求めます。

引 数

n	データ(座標)点の数です。2以上を入力してください。
taPoint[]	各点の2次元座標値です。2点以上を入力してください。
*ipCnvxNum	凸包を構成する座標点の数が格納されます。
taCnvxPt[]	凸包を構成する各点の座標値が格納されます。‘taPoint’と同じ配列を指定することができます。taCnvxPt[0]～taCnvxPt[*ipCnvxNum-1] が有効となります。

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgSmallestEnclosingCircle

機 能 点群の最小包含円

形 式 `#include "eyemLib.h"`
`int eyemCgSmallestEnclosingCircle (int n, EyemOcsDXY taPoint[],`
`EyemOcsDXY *tpCenter, double *dpRadius);`

解 説 2次元点群に対して、それらの最小包含円(点群を含む最小の円)を求めます.

引 数

n	データ(座標)点の数です. 2以上を入力してください.
taPoint[]	各点の2次元座標値です. 2点以上を入力してください.
*tpCenter	最小包含円の中心座標が格納されます.
*dpRadius	最小包含円の半径が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_NOT_ENOUGH_MEM	メモリ不足
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません.

eyemCgIntersectionOfTwoSegments

機 能 2線分の交差状態

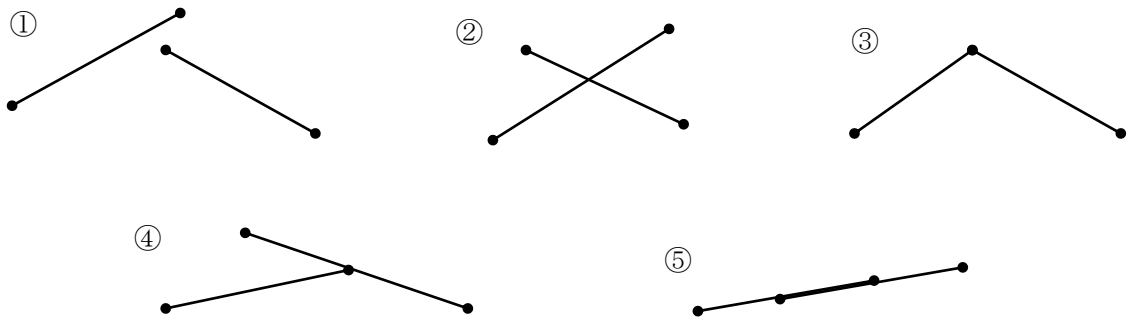
形 式 `#include "eyemLib.h"`
`int eyedCgIntersectionOfTwoSegments (EyemOcsDXY *tpPt1St,`
 `EyemOcsDXY *tpPt1Ed, EyemOcsDXY *tpPt2St, EyemOcsDXY *tpPt2Ed,`
 `int *ipStatus, EyemOcsDXY *tpPoint);`

解 説 2線分が交わっているか否かを判定します。交わっていれば、その交点座標を計算します。

引 数 `*tpPt1St` 第1線分の始点座標.
 `*tpPt1Ed` 第1線分の終点座標.
 `*tpPt2St` 第2線分の始点座標.
 `*tpPt2Ed` 第2線分の終点座標.
 `*ipStatus` 2線分の交差状態結果が格納されます.

EYEM_CG_NOT_INTERSECTION	交差していない(下図①)
EYEM_CG_INTERSECTION	交差している(互いに他をまたぐ) (下図②)
EYEM_CG_END1_CONTACT_END2	線分1の片方の端点と線分2の片 方の端点が接触(一致)(下図③)
EYEM_CG_END1_CONTACT_SGM2	線分1の端点が線分2の端点以外 で接触(下図④)
EYEM_CG_END2_CONTACT_SGM1	線分2の端点が線分1の端点以外 で接触(下図④)
EYEM_CG_SEG1_OVERLAP_SGM2	線分1と線分2が重なる(2線分が同 一直線上にある場合)(下図⑤)

`*tpPoint` 交わっている場合または端点が接触している場合に、その交点座標ま
たは接点座標が格納されます。



戻り値

エラー報告です.

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項

特にありません.

eyemCgPointInConvexPolygon

機 能 点の凸多角形内外判定

形 式

```
#include "eyemLib.h"

int      eyedCgPointInConvexPolygon ( int n, EyemOcsDXY taVertex[],
                                       EyemOcsDXY *tpPoint, int *ipStatus );
```

解 説 指定点が、凸多角形の内側にあるか外側にあるかを調べます。なお、指定点が多角形の辺上または頂点にある場合は“境界上”と判断します。

引 数

n	凸多角形の頂点数です。3以上を入力してください。
taVertex[]	凸多角形の頂点座標です。時計まわり、反時計まわりのいずれかで格納してください。また、3点以上を入力してください。
*tpPoint	指定点の座標です。
*ipStatus	凸多角形に対する指定点の位置の状態が格納されます。

EYEM_CG_OUTER	外側にある(境界を含まない)
EYEM_CG_INNER	内側にある(境界を含まない)
EYEM_CG_BORDER	境界上にある

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgIntersectionLineAndConvexPolygon

機 能 直線と凸多角形の交点

形 式

```
#include "eyemLib.h"

int      eyedCgIntersectionLineAndConvexPolygon( int n, EyemOcsDXY taVertex[],
                                                EyemOcsDABC *tpLine, EyemOcsDXY taPoint[] );
```

解 説 直線と凸多角形の交点を求めます。なお、交点数は2点以下となります。

引 数

n	凸多角形の頂点数です。2以上を入力してください。
taVertex[]	凸多角形の頂点座標です。時計まわり、反時計まわりのいずれかで格納してください。また、2点以上を入力してください。
*tpLine	直線です。
*taPoint[]	直線と凸多角形が交わっている場合に、交点の座標が交点数分(2点以下)格納されます。2次元配列を指定してください。

戻り値 エラー報告です。

0～2	(正常終了)交点数
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgIntersectionSegmentAndConvexPolygon

機 能 線分と凸多角形の交点

形 式

```
#include "eyemLib.h"

int      eyedCgIntersectionSegmentAndConvexPolygon( int n, EyemOcsDXY taVertex[],
                                                    EyemOcsDXY *tpSegmSt, EyemOcsDXY *tpSegmEd, EyemOcsDXY taPoint[] );
```

解 説 線分と凸多角形の交点を求めます。なお、交点数は2点以下となります。

引 数

n	凸多角形の頂点数です。2以上を入力してください。
taVertex[]	凸多角形の頂点座標です。時計まわり、反時計まわりのいずれかで格納してください。また、2点以上を入力してください。
*tpSegmSt	線分始点です。
*tpSegmEd	線分終点です。
*taPoint[]	線分と凸多角形が交わっている場合に、交点の座標が交点数分(2点以下)格納されます。2次元配列を指定してください。

戻り値 エラー報告です。

0～2	(正常終了)交点数
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgPointInPolygon

機 能 点の多角形内外判定

形 式

```
#include "eyemLib.h"

int      eyedCgPointInPolygon( int n, EyemOcsDXY taVertex[],
                                EyemOcsDXY *tpPoint, int *ipStatus );
```

解 説 指定点が、任意の多角形(凸多角形または凹多角形)の内側にあるか外側にあるかを調べます。なお、指定点が多角形の边上または頂点にある場合は“境界上”と判断します。

引 数

n	多角形の頂点数です。3以上を入力してください。
taVertex[]	多角形の頂点座標です。時計まわり, 反時計まわりのいずれかで格納してください。また, 3点以上を入力してください。
*tpPoint	指定点の座標です。
*ipStatus	多角形に対する指定点の位置の状態が格納されます。

EYEM_CG_OUTER	外側にある(境界を含まない)
EYEM_CG_INNER	内側にある(境界を含まない)
EYEM_CG_BORDER	境界上にある

戻り値 エラー報告です。

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません。

eyemCgAreaOfPolygon

機 能 多角形の面積

形 式 `#include "eyemLib.h"`
`int eyemCgAreaOfPolygon(int n, EyemOcsDXY taVertex[], double *dpArea);`

解 説 任意の多角形(凸多角形または凹多角形)の面積を求めます.

引 数

n	多角形の頂点数です. 3以上を入力してください.
taVertex[]	多角形の頂点座標です. 時計まわり, 反時計まわりのいずれかで格納してください. また, 3点以上を入力してください.
*dpArea	多角形の面積が格納されます.

戻り値 エラー報告です.

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

留意事項 特にありません.

改訂履歴

Version No.	内 容
1.0	• 新規発行
1.1	• 2線分交差状態の修正, および凸多角形内外判定関数の追加
1.2	• 直線と凸多角形の交点関数, および線分と凸多角形の交点関数を追加
1.3	• 点の多角形内外判定関数, および多角形の面積関数を追加