

Npatch
1.0.0

構築: Doxygen 1.8.6

2016 年 03 月 15 日 (火) 18 時 49 分 54 秒

Contents

1	ファイル索引	1
1.1	ファイル一覧	1
2	ファイル詳解	3
2.1	CalcGeo.h ファイル	3
2.1.1	マクロ定義詳解	4
2.1.1.1	CAL_STDVAL	4
2.1.1.2	GEO_ALW_L	4
2.1.1.3	GEO_ALW_V	4
2.1.1.4	GEO_REAL	4
2.1.1.5	INLINE	4
2.1.1.6	PAI	4
2.1.2	関数詳解	4
2.1.2.1	Calc_3dTransAxisPntn	4
2.1.2.2	Calc_3dTransAxisPntnInv	5
2.1.2.3	CalcCrossPointLine	5
2.1.2.4	CalcInProduct	5
2.1.2.5	CalcIntersectionLine	6
2.1.2.6	CalcIntersectionLine2	6
2.1.2.7	CalcLineSize	6
2.1.2.8	CalcLineVec	7
2.1.2.9	CalcNearPosOnLine	7
2.1.2.10	CalcNormalize	7
2.1.2.11	CalcNormalize2	7
2.1.2.12	CalcOutProduct	8
2.1.2.13	CalcPlaneD	8
2.1.2.14	CalcVec	8
2.1.2.15	CalcVecAngle	8
2.1.2.16	CalcVecAngleDegree	9
2.1.2.17	CalcVecMirror	9
2.1.2.18	CalcVecRotate2	9

2.1.2.19	CalcVecSize	10
2.2	CalcGeo_Matrix.h ファイル	11
2.2.1	関数詳解	11
2.2.1.1	Calc_3dMat4Mov	11
2.2.1.2	Calc_3dMat4Multi14	11
2.2.1.3	Calc_3dMat4Multi41	12
2.2.1.4	Calc_3dMat4Multi44	12
2.2.1.5	Calc_3dMat4Rot2	12
2.2.1.6	Calc_3dMat4RotAxis	13
2.2.1.7	Calc_3dMat4TranAxis	14
2.2.1.8	Calc_3dMat4TranAxisInv	14
2.3	FNpt.h ファイル	14
2.3.1	マクロ定義詳解	15
2.3.1.1	INLINE	15
2.3.1.2	NPT_REAL	15
2.3.2	関数詳解	15
2.3.2.1	fnpt_correct_pnt2_	15
2.3.2.2	fnpt_correct_pnt_	16
2.3.2.3	fnpt_cvt_pos_to_eta_xi_	17
2.3.2.4	fnpt_move_vertex_	18
2.3.2.5	fnpt_param_crt_	19
2.4	Npt.h ファイル	19
2.4.1	マクロ定義詳解	20
2.4.1.1	INLINE	20
2.4.1.2	NPT_ALW_V	20
2.4.1.3	NPT_REAL	20
2.4.2	関数詳解	20
2.4.2.1	npt_correct_pnt	20
2.4.2.2	npt_correct_pnt2	21
2.4.2.3	npt_cvt_pos_to_eta_xi	21
2.4.2.4	npt_move_vertex	22
2.4.2.5	npt_param_crt	23
2.5	npt_Version.h ファイル	23
2.5.1	詳解	23
2.5.2	マクロ定義詳解	24
2.5.2.1	NPT_REVISION	24
2.5.2.2	NPT_VERSION_NO	24

Chapter 1

ファイル索引

1.1 ファイル一覧

ファイル一覧です。

CalcGeo.h	3
CalcGeo_Matrix.h	11
FNpt.h	14
Npt.h	19
npt_Version.h	23

Chapter 2

ファイル詳解

2.1 CalcGeo.h ファイル

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "CalcGeo_Matrix.h"
#include <stdbool.h>
```

マクロ定義

- #define GEO_REAL float
- #define INLINE inline
- #define GEO_ALW_L (1.0e-3)
- #define GEO_ALW_V (1.0e-5)
- #define PAI 3.14159265358979323846 /* π の値 */
- #define CAL_STDVAL(x)

関数

- INLINE GEO_REAL CalcVecSize (GEO_REAL vec[3])
- INLINE GEO_REAL CalcLineSize (GEO_REAL pp[3], GEO_REAL lp[3])
- INLINE void CalcNormalize (GEO_REAL vec[3])
- INLINE bool CalcNormalize2 (GEO_REAL vec[3], GEO_REAL vec_o[3])
- INLINE void CalcVec (GEO_REAL pp[3], GEO_REAL lp[3], GEO_REAL vec[3])
- INLINE bool CalcLineVec (GEO_REAL pp[3], GEO_REAL lp[3], GEO_REAL vec[3], GEO_REAL *length)
- INLINE GEO_REAL CalcInProduct (GEO_REAL vec1[3], GEO_REAL vec2[3])
- INLINE void CalcOutProduct (GEO_REAL vec1[3], GEO_REAL vec2[3], GEO_REAL vec_o[3])
- INLINE GEO_REAL CalcVecAngle (GEO_REAL vec1[3], GEO_REAL vec2[3])
- INLINE GEO_REAL CalcVecAngleDegree (GEO_REAL vec1[3], GEO_REAL vec2[3])
- INLINE GEO_REAL CalcPlaneD (GEO_REAL pos[3], GEO_REAL vec[3])
- INLINE bool CalcIntersectionLine (GEO_REAL vec1[3], GEO_REAL d1, GEO_REAL vec2[3], GEO_REAL d2, GEO_REAL pos[3], GEO_REAL vec[3])
- INLINE bool CalcIntersectionLine2 (GEO_REAL vec1[3], GEO_REAL d1, GEO_REAL vec2[3], GEO_REAL d2, GEO_REAL pos[3], GEO_REAL vec[3])
- INLINE void CalcNearPosOnLine (GEO_REAL pnt[3], GEO_REAL pos[3], GEO_REAL vec[3], GEO_REAL pos_x[3])
- INLINE void CalcVecRotate2 (GEO_REAL rot_vec[3], GEO_REAL rad, GEO_REAL veci[3], GEO_REAL veco[3])

- **INLINE** void **CalcVecMirror** (GEO_REAL rot_vec[3], GEO_REAL veci[3], GEO_REAL veco[3])
- **INLINE** bool **CalcCrossPointLine** (GEO_REAL pp1[3], GEO_REAL lp1[3], GEO_REAL pp2[3], GEO_REAL lp2[3], GEO_REAL pos_x1[3], GEO_REAL pos_x2[3])
- **INLINE** void **Calc_3dTransAxisPntn** (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], int n, GEO_REAL pntni[][3], GEO_REAL pntno[][3])
- **INLINE** void **Calc_3dTransAxisPntnInv** (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], int n, GEO_REAL pntni[][3], GEO_REAL pntno[][3])

2.1.1 マクロ定義詳解

2.1.1.1 #define CAL_STDVAL(x)

値:

```
{
    if ( (x) > 1.0 ) {
        (x) = 1.0;
    } else if ( (x) < -1.0 ) {
        (x) = -1.0;
    }
}
```

2.1.1.2 #define GEO_ALW_L (1.0e-3)

2.1.1.3 #define GEO_ALW_V (1.0e-5)

2.1.1.4 #define GEO_REAL float

幾何演算系 実数型の指定

- デフォルトでは、GEO_REAL=float
- コンパイル時オプション-D_REAL_IS_DOUBLE_を付与することで GEO_REAL=double になる

2.1.1.5 #define INLINE inline

2.1.1.6 #define PAI 3.14159265358979323846 /* π の値 */

2.1.2 関数詳解

2.1.2.1 **INLINE** void **Calc_3dTransAxisPntn** (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], int n, GEO_REAL pntni[][3], GEO_REAL pntno[][3])

点列座標を指定された座標系に変換する (ワールド->ローカル)

引数

in	<i>orig</i>	指定座標系の原点となる点
in	<i>x_axis</i>	指定座標系のX軸となるベクトル
in	<i>y_axis</i>	指定座標系のY軸となるベクトル
in	<i>z_axis</i>	指定座標系のZ軸となるベクトル
in	<i>n</i>	点列座標数

in	<i>pnti</i>	点列座標
in	<i>pnto</i>	指定座標系での点列座標

戻り値

なし

2.1.2.2 `INLINE void Calc_3dTransAxisPntInlv (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], int n, GEO_REAL pntni[][3], GEO_REAL pntno[][3])`

点列座標を指定された座標系に逆変換する（ローカル->ワールド）

引数

in	<i>orig</i>	指定座標系の原点となる点（ワールド座標系から見た座標）
in	<i>x_axis</i>	指定座標系のX軸となるベクトル（ワールド座標系から見たベクトル）
in	<i>y_axis</i>	指定座標系のY軸となるベクトル（ワールド座標系から見たベクトル）
in	<i>z_axis</i>	指定座標系のZ軸となるベクトル（ワールド座標系から見たベクトル）
in	<i>n</i>	点列座標数
in	<i>pnti</i>	点列座標（入力）
in	<i>pnto</i>	点列座標（逆変換後）

戻り値

なし

2.1.2.3 `INLINE bool CalcCrossPointLine (GEO_REAL pp1[3], GEO_REAL lp1[3], GEO_REAL pp2[3], GEO_REAL lp2[3], GEO_REAL pos_x1[3], GEO_REAL pos_x2[3])`

2線分の交点（最近点）を求める

引数

in	<i>pp1</i>	線分 1 始点
in	<i>lp1</i>	線分 1 終点
in	<i>pp2</i>	線分 2 始点
in	<i>lp2</i>	線分 2 終点
out	<i>pos_x1</i>	線分 1 上の最近点
out	<i>pos_x2</i>	線分 2 上の最近点

戻り値

終了コード true:成功 false:失敗 (2線分が並行など)

2.1.2.4 `INLINE GEO_REAL CalcInProduct (GEO_REAL vec1[3], GEO_REAL vec2[3])`

ベクトルの内積

引数

in	<i>vec1</i>	ベクトル 1
----	-------------	--------

in	<i>vec1</i>	ベクトル 2
----	-------------	--------

戻り値

内積値

2.1.2.5 **INLINE** bool CalcIntersectionLine (GEO_REAL *vec1*[3], GEO_REAL *d1*, GEO_REAL *vec2*[3], GEO_REAL *d2*, GEO_REAL *pos*[3], GEO_REAL *vec*[3])

2 平面の交線（無限線分）取得

引数

in	<i>vec1</i>	平面 1 の法線ベクトル (正規化済)
in	<i>d1</i>	平面 1 の原点からの距離
in	<i>vec2</i>	平面 2 の法線ベクトル (正規化済)
in	<i>d2</i>	平面 2 の原点からの距離
out	<i>pos</i>	面の交線の通過点 (原点からの最短距離)
out	<i>vec</i>	面の交線のベクトル

戻り値

終了コード true:成功 false:失敗 (2 平面が並行など)

2.1.2.6 **INLINE** bool CalcIntersectionLine2 (GEO_REAL *vec1*[3], GEO_REAL *d1*, GEO_REAL *vec2*[3], GEO_REAL *d2*, GEO_REAL *pos*[3], GEO_REAL *vec*[3])

2 平面の交線（無限線分）取得 CalcIntersectionLine と求め方と通過点の位置が違うのみ

引数

in	<i>vec1</i>	平面 1 の法線ベクトル (正規化済)
in	<i>d1</i>	平面 1 の原点からの距離
in	<i>vec2</i>	平面 2 の法線ベクトル (正規化済)
in	<i>d2</i>	平面 2 の原点からの距離
out	<i>pos</i>	面の交線の通過点 (x-y 平面上、x-z 平面上、y-z 平面上のいずれか)
out	<i>vec</i>	面の交線のベクトル

戻り値

終了コード true:成功 false:失敗 (2 平面が並行など)

2.1.2.7 **INLINE** GEO_REAL CalcLineSize (GEO_REAL *pp*[3], GEO_REAL *lp*[3])

線分の長さ取得

引数

in	<i>pp</i>	始点
in	<i>lp</i>	終点

戻り値

ベクトルのサイズ (長さ)

```
2.1.2.8  INLINE bool CalcLineVec (  GEO_REAL pp[3], GEO_REAL lp[3], GEO_REAL vec[3], GEO_REAL * length )
```

線分のベクトルと長さ取得

引数

in, out	vec	方向ベクトル out:長さ 1 に変更
---------	-----	---------------------

戻り値

true 正常 false 同一点

2.1.2.9 `INLINE void CalcNearPosOnLine (GEO_REAL pnt[3], GEO_REAL pos[3], GEO_REAL vec[3], GEO_REAL pos_x[3])`

点から線分上に垂線を下した点を求める

引数

in	pnt	点座標
in	pos	線分の通過点
in	vec	線分の方向ベクトル (単位ベクトル)
out	pos_x	点から線分に下した垂線との交点

戻り値

終了コード なし

2.1.2.10 `INLINE void CalcNormalize (GEO_REAL vec[3])`

ベクトルを正規化する (長さ 1 のベクトルとする)

引数

in, out	vec	方向ベクトル out:長さ 1 に変更
---------	-----	---------------------

戻り値

なし

2.1.2.11 `INLINE bool CalcNormalize2 (GEO_REAL vec[3], GEO_REAL vec_o[3])`

ベクトルを正規化する (長さ 1 のベクトルとする)

引数

in, out	vec	方向ベクトル out:長さ 1 に変更
---------	-----	---------------------

戻り値

true 正常 false 同一点

2.1.2.12 `INLINE void CalcOutProduct (GEO_REAL vec1[3], GEO_REAL vec2[3], GEO_REAL vec_o[3])`

ベクトルの外積

引数

in	<i>vec1</i>	ベクトル 1
in	<i>vec1</i>	ベクトル 2
out	<i>vec_o</i>	外積ベクトル

戻り値

なし

2.1.2.13 **INLINE GEO_REAL CalcPlaneD (GEO_REAL *pos*[3], GEO_REAL *vec*[3])**

点と方向ベクトルより平面の方程式のDを求める 平面の方程式 $Ax + By + Cz = D$

引数

in	<i>pos</i>	点座標
in	<i>vec</i>	方向ベクトル (正規化済)

戻り値

平面の方程式のD (原点からの距離+-あり)

2.1.2.14 **INLINE void CalcVec (GEO_REAL *pp*[3], GEO_REAL *lp*[3], GEO_REAL *vec*[3])**

線分のベクトル作成 (単位ベクトルではない)

引数

in, out	<i>vec</i>	ベクトル (単位ベクトルではない)
---------	------------	-------------------

戻り値

なし

2.1.2.15 **INLINE GEO_REAL CalcVecAngle (GEO_REAL *vec1*[3], GEO_REAL *vec2*[3])**

ベクトル間の角度 (ラジアン) を求める

引数

in	<i>vec1</i>	ベクトル 1 (単位ベクトル)
in	<i>vec2</i>	ベクトル 2 (単位ベクトル)

戻り値

ベクトル間の角度 (0 ~ π ラジアン) 3次元のため、裏側から見れるため角度は (0 ~ π) で返す
> π の角度を求めるためには見る方向も指定する必要あり

2.1.2.16 **INLINE GEO_REAL CalcVecAngleDegree (GEO_REAL *vec1*[3], GEO_REAL *vec2*[3])**

ベクトル間の角度 (度数) を求める

引数

in	<i>vec1</i>	ベクトル 1 (単位ベクトル)
in	<i>vec2</i>	ベクトル 2 (単位ベクトル)

戻り値

ベクトル間の角度 (0 ~ 180 度) 3 次元のため、裏側から見れるため角度は (0 ~ 180) で返す > 180 度の角度を求めるためには見る方向も指定する必要がある

2.1.2.17 **INLINE** void CalcVecMirror (**GEO_REAL** *rot_vec*[3], **GEO_REAL** *veci*[3], **GEO_REAL** *veco*[3])

ベクトルのミラー 線対称のベクトル取得 ベクトルの回転の 180 度固定版 単位ベクトルである必要はありません

引数

in	<i>rot_vec</i>	回転軸のベクトル
in	<i>veci</i>	入力ベクトル
out	<i>veco</i>	線対称にミラーしたベクトル

戻り値

終了コード なし

2.1.2.18 **INLINE** void CalcVecRotate2 (**GEO_REAL** *rot_vec*[3], **GEO_REAL** *rad*, **GEO_REAL** *veci*[3], **GEO_REAL** *veco*[3])

ベクトルの回転

引数

in	<i>rot_vec</i>	回転軸のベクトル
in	<i>rad</i>	回転角 (rad)
in	<i>veci</i>	入力ベクトル
out	<i>veco</i>	回転後のベクトル

戻り値

終了コード なし

2.1.2.19 **INLINE** **GEO_REAL** CalcVecSize (**GEO_REAL** *vec*[3])

ベクトルのサイズ (長さ) を求める

引数

in	<i>vec</i>	方向ベクトル
----	------------	--------

戻り値

ベクトルのサイズ (長さ)

2.2 CalcGeo_Matrix.h ファイル

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

関数

- **INLINE** void **Calc_3dMat4Multi14** (GEO_REAL matA[4], GEO_REAL matB[4][4], GEO_REAL matC[4])
- **INLINE** void **Calc_3dMat4Multi41** (GEO_REAL matA[4][4], GEO_REAL matB[4], GEO_REAL matC[4])
- **INLINE** void **Calc_3dMat4Multi44** (GEO_REAL matA[4][4], GEO_REAL matB[4][4], GEO_REAL matC[4][4])
- **INLINE** void **Calc_3dMat4Mov** (GEO_REAL mov_x, GEO_REAL mov_y, GEO_REAL mov_z, GEO_REAL matA[4][4])
- **INLINE** void **Calc_3dMat4RotAxis** (GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], GEO_REAL matA[4][4])
- **INLINE** void **Calc_3dMat4TranAxis** (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], GEO_REAL matA[4][4])
- **INLINE** void **Calc_3dMat4TranAxisInv** (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], GEO_REAL matA[4][4])
- **INLINE** void **Calc_3dMat4Rot2** (GEO_REAL pos[3], GEO_REAL vec[3], GEO_REAL th_rad, GEO_REAL matA[4][4])

2.2.1 関数詳解

2.2.1.1 **INLINE** void **Calc_3dMat4Mov** (GEO_REAL *mov_x*, GEO_REAL *mov_y*, GEO_REAL *mov_z*, GEO_REAL *matA*[4][4])

平行移動用の 4×4 変換マトリクスを求める

引数

in	<i>mov_x</i>	X 軸平行移動量
in	<i>mov_y</i>	Y 軸平行移動量
in	<i>mov_z</i>	Z 軸平行移動量
out	<i>mat4</i>	4×4 変換マトリクス (行ベクトル系)

戻り値

終了コード なし

2.2.1.2 **INLINE** void **Calc_3dMat4Multi14** (GEO_REAL *matA*[4], GEO_REAL *matB*[4][4], GEO_REAL *matC*[4])

マトリック演算 (行ベクトル系) 4×1 マトリクスと 4×4 マトリクスの積を求める

引数

in	<i>matA</i>	4×1 マトリクス A
in	<i>matB</i>	4×4 マトリクス B
out	<i>matC</i>	1×4 マトリクス C

戻り値

終了コード なし

2.2.1.3 `INLINE void Calc_3dMat4Multi41 (GEO_REAL matA[4][4], GEO_REAL matB[4], GEO_REAL matC[4])`

4×4マトリクスと4×1マトリクスの積を求める

引数

in	<i>matA</i>	4 × 4 マトリクス A
in	<i>matB</i>	4 × 1 マトリクス B
out	<i>matC</i>	4 × 1 マトリクス C

戻り値

終了コード なし

2.2.1.4 `INLINE void Calc_3dMat4Multi44 (GEO_REAL matA[4][4], GEO_REAL matB[4][4], GEO_REAL matC[4][4])`

4 × 4 マトリクスと 4 × 4 マトリクスの積を求める

引数

in	<i>matA</i>	4 × 4 マトリクス A
in	<i>matB</i>	4 × 4 マトリクス B
out	<i>matC</i>	4 × 4 マトリクス C (演算結果)

戻り値

終了コード なし

2.2.1.5 `INLINE void Calc_3dMat4Rot2 (GEO_REAL pos[3], GEO_REAL vec[3], GEO_REAL th_rad, GEO_REAL matA[4][4])`

回転用 4 × 4 変換マトリクスを求める (任意軸)

引数

in	<i>pos</i>	回転軸上の 1 点の座標
in	<i>vec</i>	回転軸のベクトル
in	<i>th_rad</i>	回転角 (rad)
out	<i>matA</i>	回転用 4 × 4 変換マトリクス (行ベクトル系)

戻り値

終了コード なし

2.2.1.6 `INLINE void Calc_3dMat4RotAxis (GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], GEO_REAL matA[4][4])`

座標軸回転用 4 × 4 変換マトリクスを求める

引数

in	<i>x_axis</i>	新座標系の X 軸となるベクトル
in	<i>y_axis</i>	新座標系の Y 軸となるベクトル
in	<i>z_axis</i>	新座標系の Z 軸となるベクトル
out	<i>matA</i>	4 × 4 変換マトリクス (行ベクトル系)

戻り値

終了コード なし

```
2.2.1.7  INLINE void Calc_3dMat4TranAxis ( GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3],  
                                           GEO_REAL z_axis[3], GEO_REAL matA[4][4] )
```

任意座標軸設定用 4×4 変換マトリクスを求める (ワールド → ローカル座標変換)

引数

in	<i>orig</i>	新座標軸の原点となる点
in	<i>x_axis</i>	新座標系のX 軸となるベクトル
in	<i>y_axis</i>	新座標系のY 軸となるベクトル
in	<i>z_axis</i>	新座標系のZ 軸となるベクトル
out	<i>mat4</i>	4 × 4 変換マトリクス (行ベクトル系)

戻り値

終了コード なし

2.2.1.8 `INLINE void Calc_3dMat4TranAxisInv (GEO_REAL orig[3], GEO_REAL x_axis[3], GEO_REAL y_axis[3], GEO_REAL z_axis[3], GEO_REAL mat4[4][4])`

任意座標軸設定用 4 × 4 変換マトリクスを求める (ローカル → ワールド座標変換)

引数

in	<i>orig</i>	任意座標軸の原点となる点 (ワールド座標系で指定)
in	<i>x_axis</i>	任意座標系のX 軸となるベクトル (ワールド座標系で指定)
in	<i>y_axis</i>	任意座標系のY 軸となるベクトル (ワールド座標系で指定)
in	<i>z_axis</i>	任意座標系のZ 軸となるベクトル (ワールド座標系で指定)
out	<i>mat4</i>	4 × 4 変換マトリクス (行ベクトル系)

戻り値

終了コード なし

2.3 FNpt.h ファイル

```
#include <stdio.h>
#include <stdlib.h>
#include <string>
#include <math.h>
#include "Npt.h"
```

マクロ定義

- `#define NPT_REAL float`
- `#define INLINE inline`

関数

- `void fnpt_param_crt_ (NPT_REAL p1[3], NPT_REAL norm1[3], NPT_REAL p2[3], NPT_REAL norm2[3], NPT_REAL p3[3], NPT_REAL norm3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], int *ret)`
- `void fnpt_cvt_pos_to_eta_xi_ (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL *eta, NPT_REAL *xi)`
- `void fnpt_correct_pnt_ (NPT_REAL *eta, NPT_REAL *xi, NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL pos_o[3])`

- void `fnpt_correct_pnt2_` (`NPT_REAL` `pos`[3], `NPT_REAL` `p1`[3], `NPT_REAL` `p2`[3], `NPT_REAL` `p3`[3], `NPT_REAL` `cp_side1_1`[3], `NPT_REAL` `cp_side1_2`[3], `NPT_REAL` `cp_side2_1`[3], `NPT_REAL` `cp_side2_2`[3], `NPT_REAL` `cp_side3_1`[3], `NPT_REAL` `cp_side3_2`[3], `NPT_REAL` `cp_center`[3], `NPT_REAL` `pos_o`[3])
- void `fnpt_move_vertex_` (`NPT_REAL` `p1`[3], `NPT_REAL` `p2`[3], `NPT_REAL` `p3`[3], `NPT_REAL` `cp_side1_1`[3], `NPT_REAL` `cp_side1_2`[3], `NPT_REAL` `cp_side2_1`[3], `NPT_REAL` `cp_side2_2`[3], `NPT_REAL` `cp_side3_1`[3], `NPT_REAL` `cp_side3_2`[3], `NPT_REAL` `cp_center`[3], `NPT_REAL` `p1_n`[3], `NPT_REAL` `p2_n`[3], `NPT_REAL` `p3_n`[3], `NPT_REAL` `cp_side1_1_n`[3], `NPT_REAL` `cp_side1_2_n`[3], `NPT_REAL` `cp_side2_1_n`[3], `NPT_REAL` `cp_side2_2_n`[3], `NPT_REAL` `cp_side3_1_n`[3], `NPT_REAL` `cp_side3_2_n`[3], `NPT_REAL` `cp_center_n`[3])

2.3.1 マクロ定義詳解

2.3.1.1 #define `INLINE` inline

2.3.1.2 #define `NPT_REAL` float

長田パッチ 関数 (Fortran インターフェース) 実数型の指定

- デフォルトでは、`NPT_REAL=float`
- コンパイル時オプション-`D_REAL_IS_DOUBLE_`を付与することで `NPT_REAL=double` になる

2.3.2 関数詳解

2.3.2.1 void `fnpt_correct_pnt2_` (`NPT_REAL` `pos`[3], `NPT_REAL` `p1`[3], `NPT_REAL` `p2`[3], `NPT_REAL` `p3`[3], `NPT_REAL` `cp_side1_1`[3], `NPT_REAL` `cp_side1_2`[3], `NPT_REAL` `cp_side2_1`[3], `NPT_REAL` `cp_side2_2`[3], `NPT_REAL` `cp_side3_1`[3], `NPT_REAL` `cp_side3_2`[3], `NPT_REAL` `cp_center`[3], `NPT_REAL` `pos_o`[3])

長田パッチ 近似曲面補正

引数

in	<code>pos</code>	入力点座標 (p1p2p3 三角形上の座標)
in	<code>p1</code>	長田パッチ 頂点 1 座標
in	<code>p2</code>	長田パッチ 頂点 2 座標
in	<code>p3</code>	長田パッチ 頂点 3 座標
in	<code>cp_side1_1</code>	長田パッチ p1p2 辺の 3 次ベジェ制御点 1
in	<code>cp_side1_2</code>	長田パッチ p1p2 辺の 3 次ベジェ制御点 2
in	<code>cp_side2_1</code>	長田パッチ p2p3 辺の 3 次ベジェ制御点 1
in	<code>cp_side2_2</code>	長田パッチ p2p3 辺の 3 次ベジェ制御点 2
in	<code>cp_side3_1</code>	長田パッチ p3p1 辺の 3 次ベジェ制御点 1
in	<code>cp_side3_2</code>	長田パッチ p3p1 辺の 3 次ベジェ制御点 2
in	<code>cp_center</code>	長田パッチ 三角形中央の 3 次ベジェ制御点
out	<code>pos_o</code>	出力点座標 (曲面補正後の点)

戻り値

なし

注意

入力点座標が p1p2p3 三角形平面内でない場合、出力座標は保証されない
`pos`,`p1`,`p2`,`p3`,`cp_*`,`pos_o` は同一座標系とする。(cp_* 系は相対座標ではない)

```
2.3.2.2 void fnpt_correct_pnt_( NPT_REAL * eta, NPT_REAL * xi, NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL  
    p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL  
    cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL  
    pos_o[3] )
```

長田パッチ 近似曲面補正 入力： η 、 ξ パラメータ

引数

in	<i>eta</i>	入力点座標 長田パッチ η パラメータ
in	<i>xi</i>	入力点座標 長田パッチ ξ パラメータ
in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>cp_side1_1</i>	長田パッチ p1p2 辺の 3 次ベジェ制御点 1
in	<i>cp_side1_2</i>	長田パッチ p1p2 辺の 3 次ベジェ制御点 2
in	<i>cp_side2_1</i>	長田パッチ p2p3 辺の 3 次ベジェ制御点 1
in	<i>cp_side2_2</i>	長田パッチ p2p3 辺の 3 次ベジェ制御点 2
in	<i>cp_side3_1</i>	長田パッチ p3p1 辺の 3 次ベジェ制御点 1
in	<i>cp_side3_2</i>	長田パッチ p3p1 辺の 3 次ベジェ制御点 2
in	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点
out	<i>pos_o</i>	出力点座標 (曲面補正後の点)

戻り値

なし

注意

η と ξ のパラメータで 3 角形上の座標が決まる (例) 頂点 1 $\eta = 0.0$; $\xi = 0.0$; 頂点 2 $\eta = 1.0$; $\xi = 0.0$; 頂点 3 $\eta = 1.0$; $\xi = 1.0$; 辺 1 の中点 $\eta = 0.5$; $\xi = 0.0$; 辺 2 の中点 $\eta = 1.0$; $\xi = 0.5$; 辺 3 の中点 $\eta = 0.5$; $\xi = 0.5$; 3 角形の重心 $\eta = 2.0/3.0$, $\xi = 0.5*2.0/3.0$;
 p1,p2,p3,cp_*,pos_o は同一座標系とする。(cp_* 系は相対座標ではない)

**2.3.2.3 void fnpt_cvt_pos_to_eta_xi_ (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3],
 NPT_REAL * eta, NPT_REAL * xi)**

長田パッチ 曲面補間 関数 長田パッチ (3 次多項式) η 、 ξ パラメータ取得 入力座標より η 、 ξ パラメータを求める

引数

in	<i>pos</i>	入力点座標 (p1p2p3 三角形上の座標)
in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
out	<i>eta</i>	長田パッチ η パラメータ
out	<i>xi</i>	長田パッチ ξ パラメータ

戻り値

なし

注意

入力点座標が p1p2p3 三角形平面内でない場合、 η 、 ξ のパラメータは保証されない (例) 頂点 1 $\eta = 0.0$; $\xi = 0.0$; 頂点 2 $\eta = 1.0$; $\xi = 0.0$; 頂点 3 $\eta = 1.0$; $\xi = 1.0$; 辺 1 の中点 $\eta = 0.5$; $\xi = 0.0$; 辺 2 の中点 $\eta = 1.0$; $\xi = 0.5$; 辺 3 の中点 $\eta = 0.5$; $\xi = 0.5$; 3 角形の重心 $\eta = 2.0/3.0$, $\xi = 0.5*2.0/3.0$;
 pos,p1,p2,p3 は同一座標系とする。

```
2.3.2.4 void fnpt_move_vertex_( NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3],  
    NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3],  
    NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL p1_n[3], NPT_REAL p2_n[3], NPT_REAL  
    p3_n[3], NPT_REAL cp_side1_1_n[3], NPT_REAL cp_side1_2_n[3], NPT_REAL cp_side2_1_n[3], NPT_REAL  
    cp_side2_2_n[3], NPT_REAL cp_side3_1_n[3], NPT_REAL cp_side3_2_n[3], NPT_REAL cp_center_n[3] )
```

長田パッチ 頂点移動に伴う長田パッチパラメータ更新 長田パッチパラメータの実体は制御点である。頂点座標が移動するとそれに応じてパラメータを更新する必要がある

引数

in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>cp_side1_1</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side1_2</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_side2_1</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side2_2</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_side3_1</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side3_2</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点
in	<i>p1_n</i>	長田パッチ 移動後 頂点 1 座標
in	<i>p2_n</i>	長田パッチ 移動後 頂点 2 座標
in	<i>p3_n</i>	長田パッチ 移動後 頂点 3 座標
out	<i>cp_side1_1_n</i>	長田パッチ 移動後 <i>p1p2</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side1_2_n</i>	長田パッチ 移動後 <i>p1p2</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side2_1_n</i>	長田パッチ 移動後 <i>p2p3</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side2_2_n</i>	長田パッチ 移動後 <i>p2p3</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side3_1_n</i>	長田パッチ 移動後 <i>p3p1</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side3_2_n</i>	長田パッチ 移動後 <i>p3p1</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_center_n</i>	長田パッチ 移動後 三角形中央の 3 次ベジェ制御点

戻り値

なし

注意

長田パッチパラメータ更新は速くないので、可能であれば 頂点移動時に同時に長田パッチパラメータ (制御点) を更新することを推奨する

```
2.3.2.5 void fnpt_param_crt_ ( NPT_REAL p1[3], NPT_REAL norm1[3], NPT_REAL p2[3], NPT_REAL norm2[3],
    NPT_REAL p3[3], NPT_REAL norm3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL
    cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL
    cp_center[3], int * ret )
```

長田パッチ パラメータ生成 関数長田パッチパラメータ生成

引数

in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>norm1</i>	長田パッチ 頂点 1 法線ベクトル (単位ベクトル)
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>norm2</i>	長田パッチ 頂点 2 法線ベクトル (単位ベクトル)
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>norm3</i>	長田パッチ 頂点 3 法線ベクトル (単位ベクトル)
out	<i>cp_side1_1</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side1_2</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side2_1</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side2_2</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side3_1</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side3_2</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点
out	<i>ret</i>	リターンコード =0 正常 !=0 異常

戻り値

戻り値なし

注意

エラー：許容誤差

2.4 Npt.h ファイル

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "CalcGeo.h"
```

マクロ定義

- #define NPT_REAL float
- #define NPT_ALW_V 0.005
- #define INLINE inline

関数

- int npt_param_crt (NPT_REAL p1[3], NPT_REAL norm1[3], NPT_REAL p2[3], NPT_REAL norm2[3], NPT_REAL p3[3], NPT_REAL norm3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3])
- INLINE void npt_cvt_pos_to_eta_xi (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL *eta, NPT_REAL *xi)
- INLINE void npt_correct_pnt (NPT_REAL eta, NPT_REAL xi, NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL pos_o[3])
- INLINE void npt_correct_pnt2 (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL pos_o[3])
- INLINE void npt_move_vertex (NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL p1_n[3], NPT_REAL p2_n[3], NPT_REAL p3_n[3], NPT_REAL cp_side1_1_n[3], NPT_REAL cp_side1_2_n[3], NPT_REAL cp_side2_1_n[3], NPT_REAL cp_side2_2_n[3], NPT_REAL cp_side3_1_n[3], NPT_REAL cp_side3_2_n[3], NPT_REAL cp_center_n[3])

2.4.1 マクロ定義詳解

2.4.1.1 #define INLINE inline

2.4.1.2 #define NPT_ALW_V 0.005

2.4.1.3 #define NPT_REAL float

長田パッチ 関数 (C++/C) 実数型の指定

- デフォルトでは、NPT_REAL=float
- コンパイル時オプション-D_REAL_IS_DOUBLE_を付与することで NPT_REAL=double になる

2.4.2 関数詳解

2.4.2.1 `INLINE void npt_correct_pnt (NPT_REAL eta, NPT_REAL xi, NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL pos_o[3])`

長田パッチ 近似曲面補正 入力： η 、 ξ パラメータ

引数

in	<i>eta</i>	入力点座標 長田パッチ η パラメータ
in	<i>xi</i>	入力点座標 長田パッチ ξ パラメータ
in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>cp_side1_1</i>	長田パッチ p1p2 辺の 3 次ベジエ制御点 1
in	<i>cp_side1_2</i>	長田パッチ p1p2 辺の 3 次ベジエ制御点 2
in	<i>cp_side2_1</i>	長田パッチ p2p3 辺の 3 次ベジエ制御点 1
in	<i>cp_side2_2</i>	長田パッチ p2p3 辺の 3 次ベジエ制御点 2
in	<i>cp_side3_1</i>	長田パッチ p3p1 辺の 3 次ベジエ制御点 1
in	<i>cp_side3_2</i>	長田パッチ p3p1 辺の 3 次ベジエ制御点 2
in	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジエ制御点
out	<i>pos_o</i>	出力点座標 (曲面補正後の点)

戻り値

なし

注意

η と ξ のパラメータで 3 角形上の座標が決まる (例) 頂点 1 $\eta = 0.0$; $\xi = 0.0$; 頂点 2 $\eta = 1.0$; $\xi = 0.0$; 頂点 3 $\eta = 1.0$; $\xi = 1.0$; 辺 1 の中点 $\eta = 0.5$; $\xi = 0.0$; 辺 2 の中点 $\eta = 1.0$; $\xi = 0.5$; 辺 3 の中点 $\eta = 0.5$; $\xi = 0.5$; 3 角形の重心 $\eta = 2.0/3.0$, $\xi = 0.5*2.0/3.0$;
 p1,p2,p3,cp_*,pos_o は同一座標系とする。(cp_* 系は相対座標ではない)

2.4.2.2 `INLINE void npt_correct_pnt2 (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL pos_o[3])`

長田パッチ 近似曲面補正

引数

in	<i>pos</i>	入力点座標 (p1p2p3 三角形上の座標)
in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>cp_side1_1</i>	長田パッチ p1p2 辺の 3 次ベジェ制御点 1
in	<i>cp_side1_2</i>	長田パッチ p1p2 辺の 3 次ベジェ制御点 2
in	<i>cp_side2_1</i>	長田パッチ p2p3 辺の 3 次ベジェ制御点 1
in	<i>cp_side2_2</i>	長田パッチ p2p3 辺の 3 次ベジェ制御点 2
in	<i>cp_side3_1</i>	長田パッチ p3p1 辺の 3 次ベジェ制御点 1
in	<i>cp_side3_2</i>	長田パッチ p3p1 辺の 3 次ベジェ制御点 2
in	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点
out	<i>pos_o</i>	出力点座標 (曲面補正後の点)

戻り値

なし

注意

入力点座標が p1p2p3 三角形平面内でない場合、出力座標は保証されない
pos,p1,p2,p3,cp_*,pos_o は同一座標系とする。(cp_* 系は相対座標ではない)

2.4.2.3 `inline void npt_cvt_pos_to_eta_xi (NPT_REAL pos[3], NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL * eta, NPT_REAL * xi)`

長田パッチ 曲面補間 関数 (インライン展開あり) 長田パッチ (3 次多項式) η 、 ξ パラメータ取得 入力座標より η 、 ξ パラメータを求める

引数

in	<i>pos</i>	入力点座標 (p1p2p3 三角形上の座標)
in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
out	<i>eta</i>	長田パッチ η パラメータ
out	<i>xi</i>	長田パッチ ξ パラメータ

戻り値

なし

注意

入力点座標が p1p2p3 三角形平面内でない場合、 η 、 ξ のパラメータは保証されない (例) 頂点 1 $\eta = 0.0$; $\xi = 0.0$; 頂点 2 $\eta = 1.0$; $\xi = 0.0$; 頂点 3 $\eta = 1.0$; $\xi = 1.0$; 辺 1 の中点 $\eta = 0.5$; $\xi = 0.0$; 辺 2 の中点 $\eta = 1.0$; $\xi = 0.5$; 辺 3 の中点 $\eta = 0.5$; $\xi = 0.5$; 3 角形の重心 $\eta = 2.0/3.0$, $\xi = 0.5*2.0/3.0$;
pos,p1,p2,p3 は同一座標系とする。

```
2.4.2.4  INLINE void npt_move_vertex ( NPT_REAL p1[3], NPT_REAL p2[3], NPT_REAL p3[3], NPT_REAL  
      cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL  
      cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL cp_center[3], NPT_REAL p1_n[3], NPT_REAL p2_n[3],  
      NPT_REAL p3_n[3], NPT_REAL cp_side1_1_n[3], NPT_REAL cp_side1_2_n[3], NPT_REAL cp_side2_1_n[3],  
      NPT_REAL cp_side2_2_n[3], NPT_REAL cp_side3_1_n[3], NPT_REAL cp_side3_2_n[3], NPT_REAL  
      cp_center_n[3] )
```

長田パッチ 頂点移動に伴う長田パッチパラメータ更新 長田パッチパラメータの実体は制御点である。頂点座標が移動するとそれに応じてパラメータを更新する必要がある

引数

in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>cp_side1_1</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side1_2</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_side2_1</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side2_2</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_side3_1</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 1
in	<i>cp_side3_2</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 2
in	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点
in	<i>p1_n</i>	長田パッチ 移動後 頂点 1 座標
in	<i>p2_n</i>	長田パッチ 移動後 頂点 2 座標
in	<i>p3_n</i>	長田パッチ 移動後 頂点 3 座標
out	<i>cp_side1_1_n</i>	長田パッチ 移動後 <i>p1p2</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side1_2_n</i>	長田パッチ 移動後 <i>p1p2</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side2_1_n</i>	長田パッチ 移動後 <i>p2p3</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side2_2_n</i>	長田パッチ 移動後 <i>p2p3</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side3_1_n</i>	長田パッチ 移動後 <i>p3p1</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side3_2_n</i>	長田パッチ 移動後 <i>p3p1</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_center_n</i>	長田パッチ 移動後 三角形中央の 3 次ベジェ制御点

戻り値

なし

注意

長田パッチパラメータ更新は速くないので、可能であれば 頂点移動時に同時に長田パッチパラメータ (制御点) を更新することを推奨する

```
2.4.2.5 int npt_param_crt ( NPT_REAL p1[3], NPT_REAL norm1[3], NPT_REAL p2[3], NPT_REAL norm2[3],
NPT_REAL p3[3], NPT_REAL norm3[3], NPT_REAL cp_side1_1[3], NPT_REAL cp_side1_2[3], NPT_REAL
cp_side2_1[3], NPT_REAL cp_side2_2[3], NPT_REAL cp_side3_1[3], NPT_REAL cp_side3_2[3], NPT_REAL
cp_center[3] )
```

長田パッチ パラメータ生成 関数 (インライン展開なし) 長田パッチパラメータ生成

引数

in	<i>p1</i>	長田パッチ 頂点 1 座標
in	<i>norm1</i>	長田パッチ 頂点 1 法線ベクトル (単位ベクトル)
in	<i>p2</i>	長田パッチ 頂点 2 座標
in	<i>norm2</i>	長田パッチ 頂点 2 法線ベクトル (単位ベクトル)
in	<i>p3</i>	長田パッチ 頂点 3 座標
in	<i>norm3</i>	長田パッチ 頂点 3 法線ベクトル (単位ベクトル)
out	<i>cp_side1_1</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side1_2</i>	長田パッチ <i>p1p2</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side2_1</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side2_2</i>	長田パッチ <i>p2p3</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_side3_1</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 1
out	<i>cp_side3_2</i>	長田パッチ <i>p3p1</i> 辺の 3 次ベジェ制御点 2
out	<i>cp_center</i>	長田パッチ 三角形中央の 3 次ベジェ制御点

戻り値

リターンコード =0 正常 !=0 異常

注意

エラー：許容誤差

2.5 npt_Version.h ファイル

マクロ定義

- `#define NPT_VERSION_NO "1.0.0"`
- `#define NPT_REVISION "20160215_1000"`

2.5.1 詳解

バージョン情報のヘッダーファイル

2.5.2 マクロ定義詳解

2.5.2.1 `#define NPT_REVISION "20160215_1000"`

Npatch ライブラリのリビジョン

2.5.2.2 `#define NPT_VERSION_NO "1.0.0"`

Npatch ライブラリのバージョン