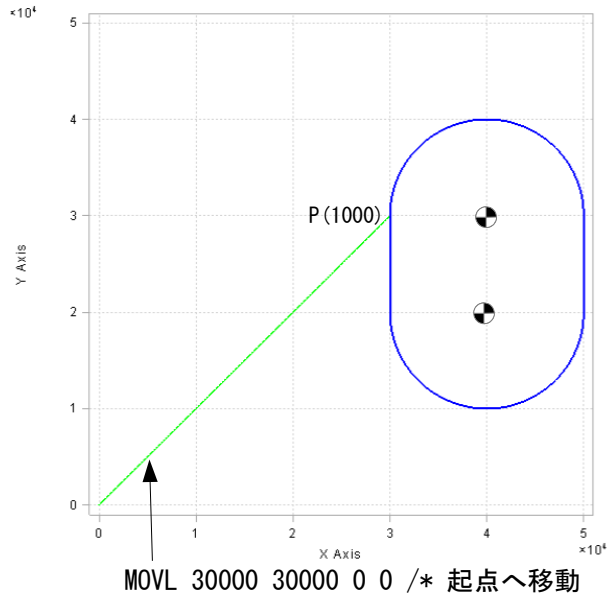
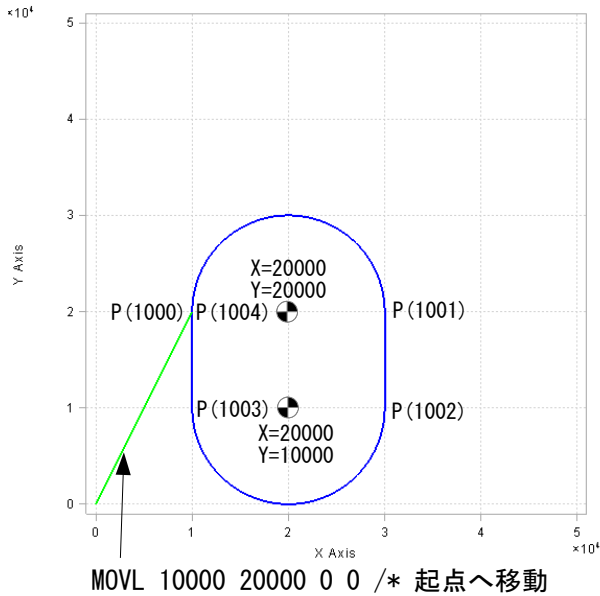


MOV T 2軸円弧補間の動作

各点は絶対値で指定する。
起点に移動後は相対座標移動をする。
(ユーザーズマニュアル MOV T 参照)



各点の設定(絶対座標)

X, Y差分を移動

```

SETP 1000 10000 20000 0 0 /* 起点
SETP 1001 30000 20000 20000 20000 /* XY:絶対座標 UZ:回転中心絶対座標
SETP 1002 30000 10000 0 0
SETP 1003 10000 10000 20000 10000
SETP 1004 10000 20000 0 0
SETP 2000 1004 50 0 0 /* X:最終点 Y:FEED
SETP 2001 CW 0 0 0 /* X:回転方向
SETP 2002 0 0 0 0
SETP 2003 CW 0 0 0
SETP 2004 0 0 0 0
    
```

動作プログラム

```

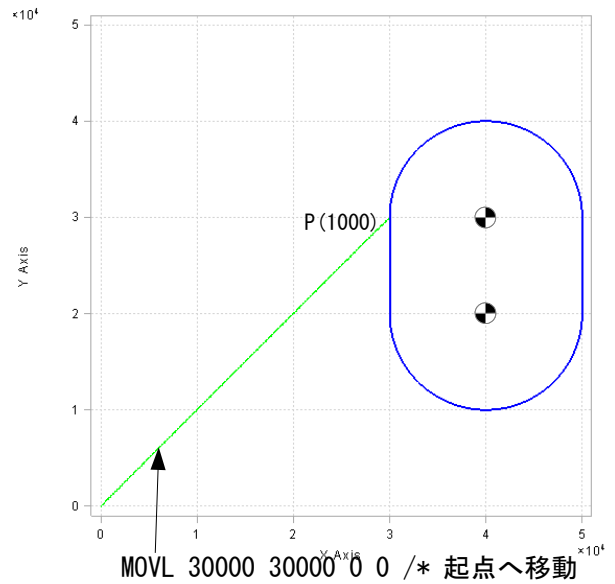
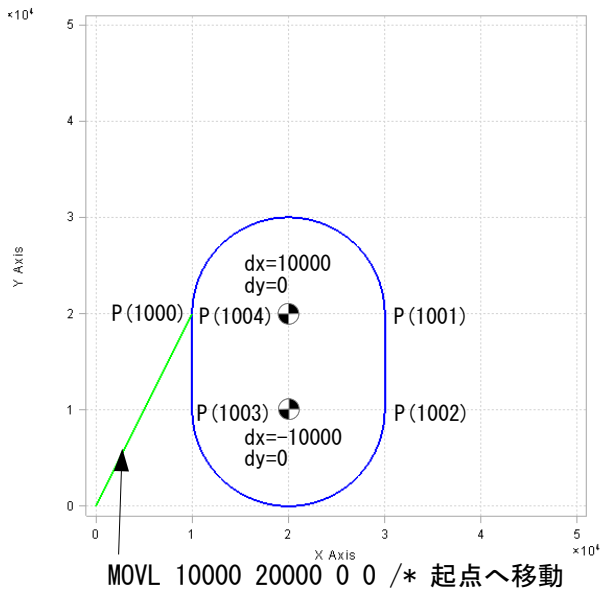
axis=X_A|Y_A
linecolor=green
MOV L 10000 20000 0 0 /* 起点へ移動
WAIT RR(axis)==0

axis=X_A|Y_A
linecolor=green
MOV L 30000 30000 0 0 /* 起点へ移動
WAIT RR(axis)==0

linecolor=blue
FEED axis Y(2000)
DS_DACL
FOR pnt=1001 TO X(2000) /*起点をP(1000)とする
/* 軸 目標点 回転方向
MOV T axis P(pnt) X(pnt+1000)
NEXT pnt
EN_DACL
WAIT RR(axis)==0
    
```

RMVT 2軸円弧補間の動作

相対座標移動



各点の設定(相対座標)

```

SETP 1000 0 0 0 0 /* 起点
SETP 1001 20000 0 10000 0 /* XY: 前の点との相対座標 UZ: 中心の相対座標
SETP 1002 0 -10000 0 0
SETP 1003 -20000 0 -10000 0
SETP 1004 0 10000 0 0
SETP 2000 1004 50 0 0 /* X: 最終点 Y: FEED
SETP 2001 CW 0 0 0 /* X: 回転方向
SETP 2002 0 0 0 0
SETP 2003 CW 0 0 0
SETP 2004 0 0 0 0
    
```

動作プログラム

```

axis=X_A|Y_A
linecolor=green
MOVL 10000 20000 0 0 /* 起点へ移動
WAIT RR(axis)==0
    
```

```

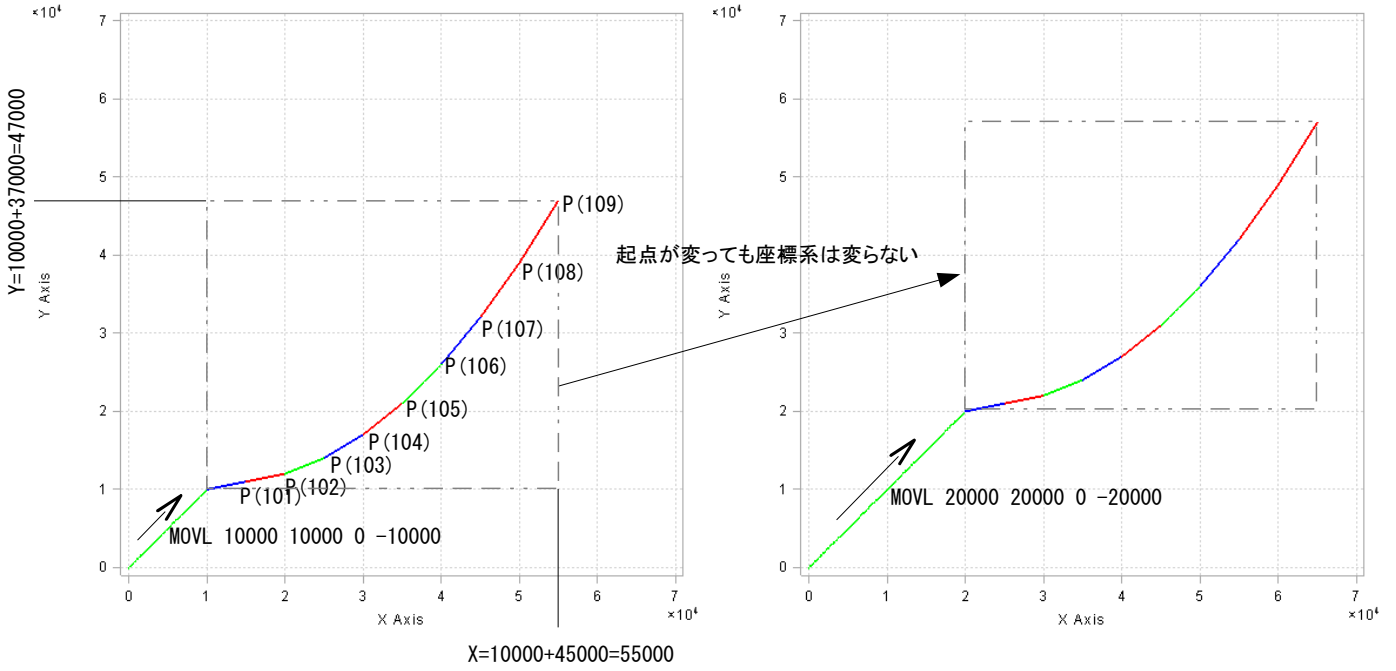
axis=X_A|Y_A
linecolor=green
MOVL 30000 30000 0 0 /* 起点へ移動
WAIT RR(axis)==0
    
```

```

linecolor=blue
FEED axis Y(2000)
DS_DACL
FOR pnt=1001 TO X(2000) /* 起点をP(1000)とする
/* 軸 X座標 Y座標 回転方向 中心dx 中心dy
RMVT axis X(pnt) Y(pnt) X(pnt+1000) U(pnt) Z(pnt)
NEXT pnt
EN_DACL
WAIT RR(axis)==0
    
```

MOVT 3軸連続直線補間の動作 [1]

(各点で色を変えていますが、MOVTの'先読み'仕様のため最後は変わりません)



各点の設定(絶対値指定。Xは増加量一定、Yは増加量が増える)

X, Y, Z差分を移動	X(100)=0	Y(100)=0	Z(100)=0
	X(101)=5000	Y(101)=1000	Z(101)=-450
	X(102)=10000	Y(102)=2000	Z(102)=-460
	X(103)=15000	Y(103)=4000	Z(103)=-470
	X(104)=20000	Y(104)=7000	Z(104)=-480
	X(105)=25000	Y(105)=11000	Z(105)=-500
	X(106)=30000	Y(106)=16000	Z(106)=-450
	X(107)=35000	Y(107)=22000	Z(107)=-500
	X(108)=40000	Y(108)=29000	Z(108)=-450
X(109)=45000	Y(109)=37000	Z(109)=-500	

動作プログラム

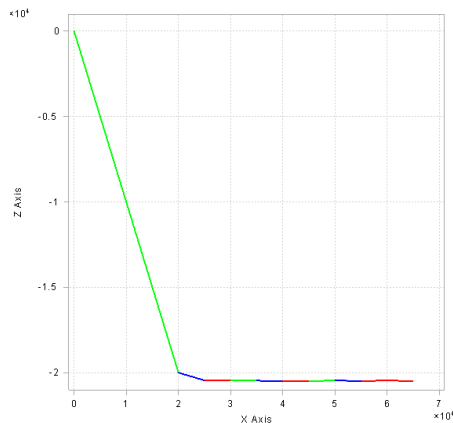
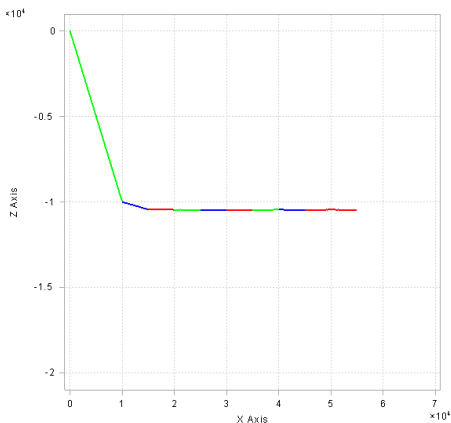
```

linecolor=green
MOVL 10000 10000 0 -10000 /* 起点へ移動
WAIT RR(ALL_A)==0
linecolor=green
MOVL 20000 20000 0 -20000 /* 起点へ移動
WAIT RR(ALL_A)==0
    
```

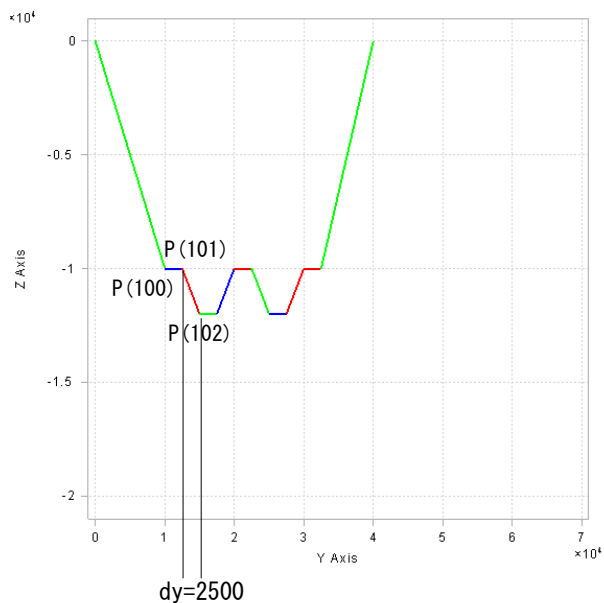
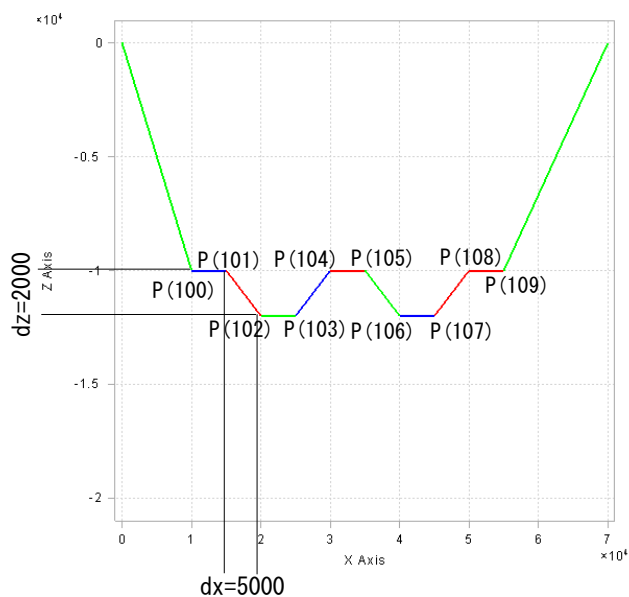
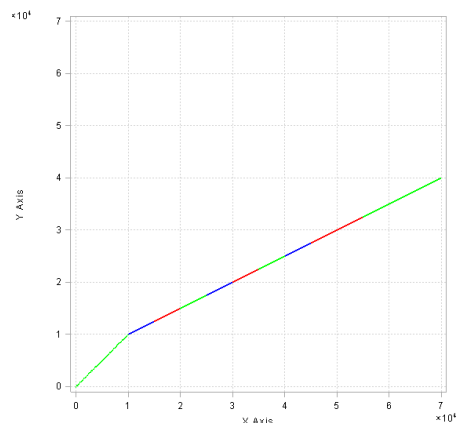
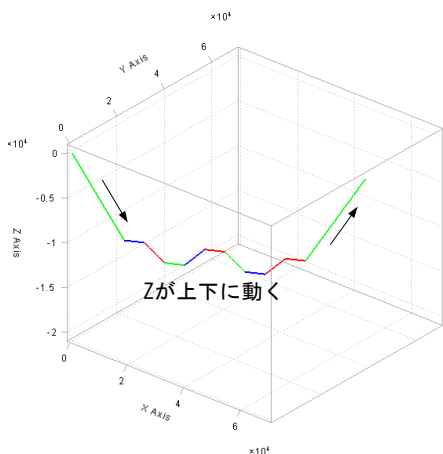
```

DS_DACL
TIME 500
FOR I=101 TO 109 /* 起点をP(100)とする
    MOVT X_A|Y_A|Z_A P(I) /* P(I)-P(I-1)の相対移動
    linecolor=I%3+1
NEXT
EN_DACL
WAIT RR(ALL_A)==0
    
```

Z軸も相対移動

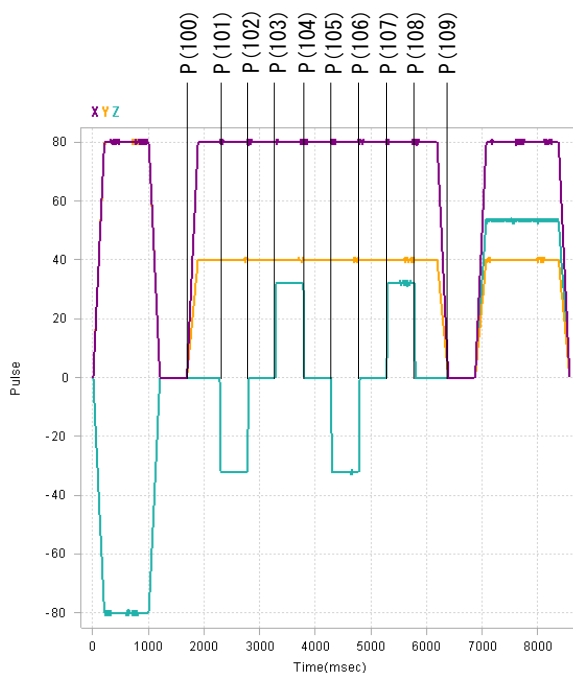


MOVT 3軸連続直線補間の動作 [2]



各点の設定(前の点との相対座標)

X(100)=0	: Y(100)=0	: Z(100)=0
X(101)=5000	: Y(101)=2500	: Z(101)=0
X(102)=10000	: Y(102)=5000	: Z(102)=-2000
X(103)=15000	: Y(103)=7500	: Z(103)=-2000
X(104)=20000	: Y(104)=10000	: Z(104)=0
X(105)=25000	: Y(105)=12500	: Z(105)=0
X(106)=30000	: Y(106)=15000	: Z(106)=-2000
X(107)=35000	: Y(107)=17500	: Z(107)=-2000
X(108)=40000	: Y(108)=20000	: Z(108)=0
X(109)=45000	: Y(109)=22500	: Z(109)=0



連続補間中の加減速は無い

MOVT 3軸連続直線補間の動作 [2] のプログラム(抜粋)

```
PG 0
ACCEL ALL_A 10000
FEED 100
CLRPOS
GOSUB *PointGenerate

linecolor=green
MOVL 10000 10000 0 -10000 /* 起点へ移動
WAIT RR(ALL_A)==0

DS_DACL /* これが無いと連続補間にならない
TIME 500
FOR I=101 TO 109 /* 起点をP(100)とする
  MOVT X_A|Y_A|Z_A P(I) /* P(I)-P(I-1)の相対移動
  linecolor=I%3+1
NEXT
EN_DACL
WAIT RR(ALL_A)==0

linecolor=I%3+1
MOVL 70000 40000 0 0
WAIT RR(ALL_A)==0

END
```

```
*PointGenerate
X(100)=0 : Y(100)=0 : Z(100)=0
X(101)=5000 : Y(101)=2500 : Z(101)=0
X(102)=10000 : Y(102)=5000 : Z(102)=-2000
X(103)=15000 : Y(103)=7500 : Z(103)=-2000
X(104)=20000 : Y(104)=10000 : Z(104)=0
X(105)=25000 : Y(105)=12500 : Z(105)=0
X(106)=30000 : Y(106)=15000 : Z(106)=-2000
X(107)=35000 : Y(107)=17500 : Z(107)=-2000
X(108)=40000 : Y(108)=20000 : Z(108)=0
X(109)=45000 : Y(109)=22500 : Z(109)=0
RETURN
```

AFFIN変換

AFFINコマンドでMOVTの座標を回転します。
AFFINはXY座標が変換対象なので回転中心座標は別の点のXYに入れ、AFFIN後にMOVTの第2,3パラメータ(U, Z座標)としています。

プログラム例

```
P(1000) ~ 元の通過点XY座標
P(1100) ~ 元の回転中心点XY座標
P(1200) ~ AFFIN変換後の通過点と中心点座標
P(1300) ~ AFFIN変換後の中心点(P(1200) ~のUZに合成)

PG 0
ACCEL ALL_A 50000
CLRPOS
GOSUB *PointGenerate
axis=X_A|Y_A
linecolor=green
MOVL P(999) /* 起点へ移動
WAIT RR(axis)==0
linecolor=blue
FEED axis 50

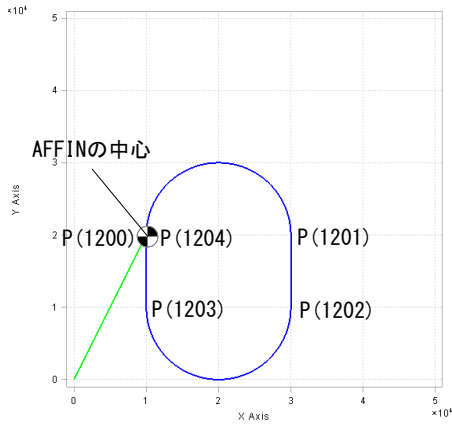
DS_DACL
FOR pnt=1001 TO 1004
  MOVT axis P(pnt+200) U(pnt+100) /* 軸, 通過点, 回転方向
NEXT pnt
EN_DACL
WAIT RR(axis)==0

END

*PointGenerate
FILL P(999) 1000 0 /* 使用するエリアをクリア
/* 通過点の絶対座標
SETP 1000 10000 20000 0 0 /* 起点
SETP 1001 30000 20000 0 0 /* XY:座標
SETP 1002 30000 10000 0 0
SETP 1003 10000 10000 0 0
SETP 1004 10000 20000 0 0
/* 中心点の絶対座標
SETP 1100 0 0 0 0 /* 起点
SETP 1101 20000 20000 CW 0 /* XY:回転中心座標 U:回転方向
SETP 1102 0 0 0 0
SETP 1103 20000 10000 CW 0
SETP 1104 0 0 0 0

SETP 999 10000 20000 /* AFFIN変換中心座標
deg=300000 /* AFFIN変換回転角(10000倍)
PR "deg=" deg
FOR pp=1000 TO 1004
  sp=pp+100 /* 中心点の点番号
  afpp=pp+200 /* AFFIN変換後の通過点の点番号
  AFFIN pp 999 afpp deg /* 通過点のAFFIN変換
  IF U(sp)<>0 THEN
    afsp=pp+300 /* AFFIN変換後の中心点の点番号
    AFFIN sp 999 afsp deg /* 中心点のAFFIN変換
    U(afpp)=X(afsp) /* X中心座標
    Z(afpp)=Y(afsp) /* Y中心座標
  END_IF
  PR "P(" afpp ")=" P(afpp)
NEXT
RETURN
```

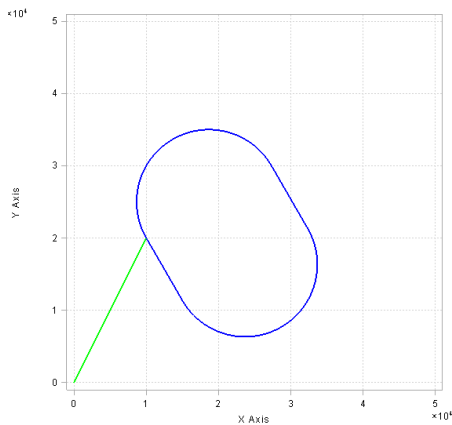
実行結果



0度

deg= 0

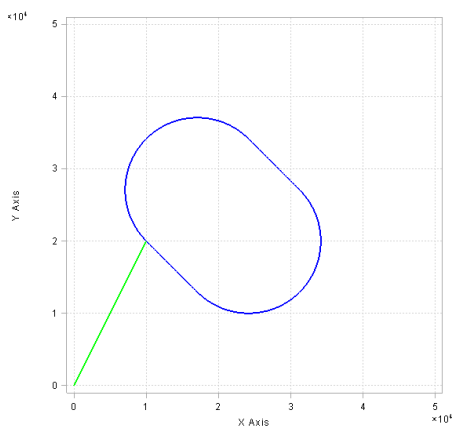
P(1200) = 10000 20000 0 0
 P(1201) = 30000 20000 20000 20000
 P(1202) = 30000 10000 0 0
 P(1203) = 10000 10000 20000 10000
 P(1204) = 10000 20000 0 0



30度

deg= 300000

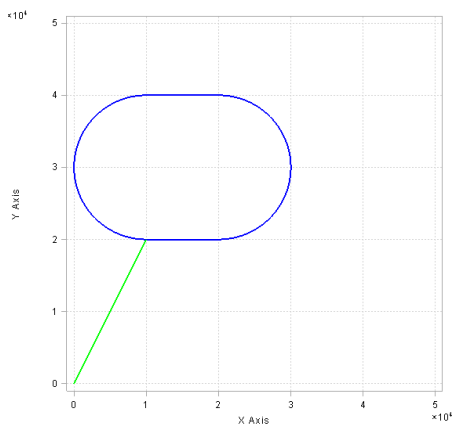
P(1200) = 10000 20000 0 0
 P(1201) = 27320 30000 18660 25000
 P(1202) = 32320 21340 0 0
 P(1203) = 15000 11340 23660 16340
 P(1204) = 10000 20000 0 0



45度

deg= 450000

P(1200) = 10000 20000 0 0
 P(1201) = 24142 34142 17071 27071
 P(1202) = 31213 27071 0 0
 P(1203) = 17071 12929 24142 20000
 P(1204) = 10000 20000 0 0



90度

deg= 900000

P(1200) = 10000 20000 0 0
 P(1201) = 10000 40000 10000 30000
 P(1202) = 20000 40000 0 0
 P(1203) = 20000 20000 20000 30000
 P(1204) = 10000 20000 0 0