

## 相対的な誤差楕円も表示する平面の測量網計算プログラムの開発 ADJUSTMENT PROGRAM OF THE HORIZONTAL SURVEY NETWORK WITH RELATIVE ERROR ELLIPSE

松井佐久夫<sup>#, A)</sup>, 木村洋昭<sup>A)</sup>, 木内淳<sup>B)</sup>

Sakuo Matsui<sup>#, A)</sup>, Hiroaki Kimura<sup>A)</sup>, Jun Kiuchi<sup>B)</sup>

<sup>A)</sup> RIKEN

<sup>B)</sup> SPring-8 Service Co.,Ltd.

### Abstract

The two-dimensional program for the network survey using Schreiber method is written in FBASIC language. It is useful for the survey and simulation because also the graphics results is get at the same time. The memory and speed of PC are improved very much now. Furthermore this program is improved from BASIC language to Excel VBA. After the parameters and survey data are filled in the column of the Excel sheet the calculated results continued in the lower lines by applying Excel VBA (Visual Basic Application). This program can calculate also the relative error ellipse, which is much important to the accelerator.

### 1. はじめに

加速器のアライメント関連技術として測量は欠かせないが、土木や建設に比べ加速器特有の部分は少なく、測量会社に依頼できる。一方、建設前に、最終のアライメントまで必要な精度を得る最適な基準点の位置や数を求めるシミュレーションは欠かせないがこの計算は測量会社への依頼は難しい。

SPring-8 では 23 年前の蓄積リングの建設時から水平面の位置計算には測量の教科書にある観測方程式からシュライバーの消去法を用いる最小二乗法の BASIC プログラム<sup>[1]</sup>を使用してきた。実測値と計算結果からの推定値との差で、入力ミスや測量の不良場所を特定しやすいような部分も加えた。また分散、共分散行列を計算するので、誤差楕円<sup>[2]</sup>、特に加速器に重要な相対的な誤差楕円、また、既知点 1 点ともう一点は x 又は y の一軸のみ固定する計算や同時に図示までできるプログラムを追加した。

幸い、富士通の FBASIC は Windows XP や Windows7 でも走るののでその後の長さ 700m の SACL A や 400m の XSBT の計画段階でのシミュレーションや実際の測量計算にも容量やスピードを気にすることなく使用できている。

レベルは 1 次元でプログラム集<sup>[3]</sup>のを用いている。

最近では加速器でレーザートラッカーが多用されるようになり、そのコントロールのソフトに測量後の 3 次元の計算も含まれるようになってきている。ここではこれらの市販のソフトとの比較も試みた。

### 2. 市販のソフトとの比較

SPring-8 建設時、測量会社の計算にはソフトに保証書がついているが同じ測定値を BASIC プログラムに入力し結果に差がないことを確認した。最近ではレーザートラッカーのソフト New River Kinematics 社の Spatial Analyzer(SA)の USMN(Unified Spatial Metrology Network)を使用しているのので比較し、よく一致していることがわかった。

<sup>#</sup> matsui@spring8.or.jp

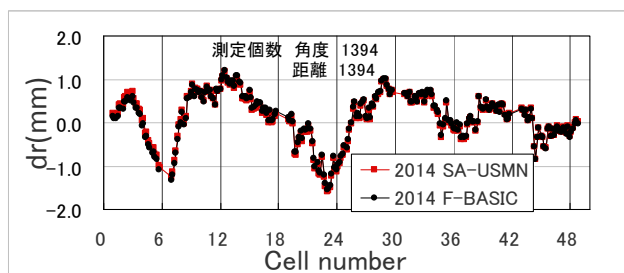


Figure 1: Comparison between SA-USNM and FBASIC.

### 3. 誤差楕円とシミュレーション

#### 3.1 シミュレーション

測定値に似せるために BASIC の一様乱数から下記のプログラムで正規乱数に変換している。

```
*TRANS
' Trans Random Number from flat distribution to Normal one
RN=RND*.5 : PN=SGN(RN) : IF RN<0 THEN RN=-RN
SUM=0 : XTRAN=0 : DXTRAN=0.1
COMP=RN*SQPI/DXTRAN
FOR IRAN=1 TO 30000
    SUM=SUM+EXP(-XTRAN*XTRAN)
    IF SUM>COMP THEN RNN=XTRAN : GOTO 14210
    XTRAN=XTRAN+DXTRAN
NEXT IRAN
14210 BUNP=PN*RNN*SIGMA/.707107
return
```

#### 3.2 絶対的な誤差楕円

水平面内での網平均計算には角度と距離測定の測定エラーから推定される重みを取り込んだ最小二乗法が用いられる。この重みに計算結果から求まる標準偏差を掛けて 2 次元平面で表示したのが誤差楕円である。SACL A 光源棟で角度精度 0.5 秒、距離精度 10  $\mu$  m+0.4ppm の条件で測量値を決めた値から振らせ、それを解析し求められた値を 400 セット繰り返しプロットしたのが Figure2 である。乱数で振らせた模擬の測定値の標準偏差と解析時の測定エラーの値を同じにしたため 2  $\sigma$  の赤い楕円内に大部分が入っているのがわかる。(理論上は 86.5%)

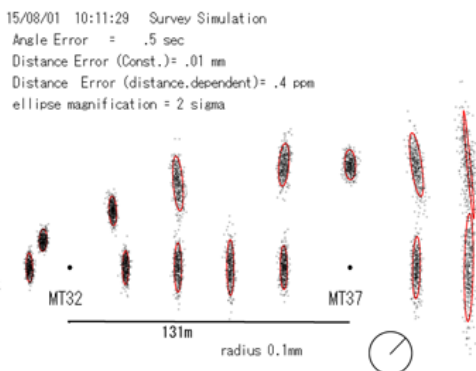


Figure 2: Simulation results of the SACL A undulator hall.

### 3.3 相対的な誤差楕円

観測方程式と測定器のエラーから求められる重みの行列から解を求めるために逆行列を計算する。この時求まる分散共分散行列の対角成分は各点の x または y 座標の分散のみなので 2 次元で傾きを持つ絶対的な誤差楕円は非対角成分も用いて計算される。加速器では軌道の滑らかさが重要なので相対的な誤差楕円の方がより重要になる。教科書<sup>[4]</sup>にある下記の式で求めることができる。プログラムでは簡単のために隣り合う 2 点からのみ計算しているが離れていても共分散の項から求めることができる。

	1	2	3	4
1	$\sigma_N^2$ $\sigma_{NE}$ $\sigma_{NE}$ $\sigma_E^2$			
2	$\sigma_{N1N2}$ $\sigma_{E1N2}$ $\sigma_{N1E2}$ $\sigma_{E1E2}$	$\sigma_N^2$ $\sigma_{NE}$ $\sigma_{NE}$ $\sigma_E^2$		
3	$\sigma_{N1N3}$ $\sigma_{E1N3}$ $\sigma_{N1E3}$ $\sigma_{E1E3}$	$\sigma_{N2N3}$ $\sigma_{E2N3}$ $\sigma_{N2E3}$ $\sigma_{E2E3}$	$\sigma_N^2$ $\sigma_{NE}$ $\sigma_{NE}$ $\sigma_E^2$	
4	$\sigma_{N1N4}$ $\sigma_{E1N4}$ $\sigma_{N1E4}$ $\sigma_{E1E4}$	$\sigma_{N2N4}$ $\sigma_{E2N4}$ $\sigma_{N2E4}$ $\sigma_{E2E4}$	$\sigma_{N3N4}$ $\sigma_{E3N4}$ $\sigma_{N3E4}$ $\sigma_{E3E4}$	$\sigma_N^2$ $\sigma_{NE}$ $\sigma_{NE}$ $\sigma_E^2$

$$\sigma_{\Delta N}^2 = \sigma_{N1}^2 - 2\sigma_{N1N2} + \sigma_{N2}^2$$

$$\sigma_{\Delta E}^2 = \sigma_{E1}^2 - 2\sigma_{E1E2} + \sigma_{E2}^2$$

$$\sigma_{\Delta NE} = \sigma_{N1E1} - \sigma_{N1E2} - \sigma_{E1N2} + \sigma_{N2E2}$$

Figure 3: Covariance matrix and relative error ellipse.

## 4. シミュレーションの応用

### 4.1 測定エラーの評価とシミュレーション

網平均計算には角度と距離の測定エラーの見積が必要になる。測定器は例えば角度 1 秒、距離は 15  $\mu$  m+距離 $\times$ 6  $\mu$  m/m などメーカーから与えられる。

しかし実際には空気の揺らぎ、ターゲットそのものや台の不安定性、など他の要因も加わる。シミュ

レーションで達成できる精度を検討する際に測定の実力値を正確に評価しておくことは欠かせない。Figure 4 は SACL A 光源棟でレーザートラッカーで測定したデータを角度 1 秒、距離は 10  $\mu$  m+距離 $\times$ 2.5  $\mu$  m/m の条件で計算した時の、測定距離対計算値と測定値の差を青でプロットしたものである。赤はシミュレーションで角度 0.5 秒、距離は 10  $\mu$  m+距離 $\times$ 0.4  $\mu$  m/m の条件で測定値を振らせ、同条件で解析したものである。距離依存性は 10  $\mu$  m+距離 $\times$ 2.5  $\mu$  m/m の直線は立ち上がりも早く、実際とかけ離れており、例えば 0.4  $\mu$  m/m 程度が妥当なことを示している。解析の条件によりこの分布は変化するのでシミュレーションのパラメーターをふり実際のエラーに近い値を見つけることが可能と考えられる。

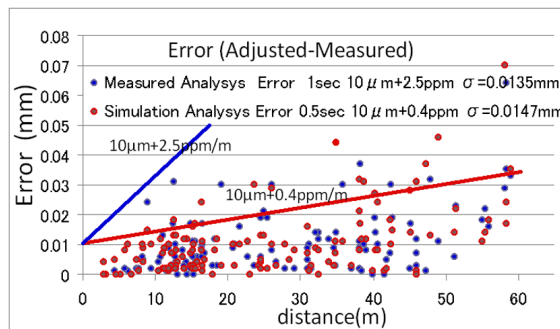


Figure 4: Comparison between measured and simulation.

### 4.2 蓄積リングへの適用

Figure 5 の最大距離 38m の測量網で、蓄積リングの測量環境を整えた場合のライカ製レーザートラッカー AT402 の実力に近いと考えられる測定エラー (角度 0.5 秒、距離 10  $\mu$  m+0.4ppm)、で反対側は一軸のみ固定でシミュレーションした。固定点と反対側の誤差楕円の半径は Figure 6 のように 1  $\sigma$  0.5mm であった。この場合周長のエラーは 0.5mm $\times$ 2 $\pi$  にはならず、20 回のシミュレーションでは 1436m に対して平均値は -0.03mm で分布の rms 値は 0.07mm であった。Figure 5 を棒で組みあげられた網のイメージで考えると Figure 6 で固定点の反対側で例えば直線の中央から左にずれた場合図の上側の斜めの楕円は中心から左下の方向にずれ結果として周長はほとんど変化せずシステムティックなエラーだけが問題になると推測される。この時の相対的な誤差楕円は Figure 7 のように 1  $\sigma$  で 20  $\mu$  m 程度で収まっている。狭い範囲でも密な網と測量機器の精度の結果である。

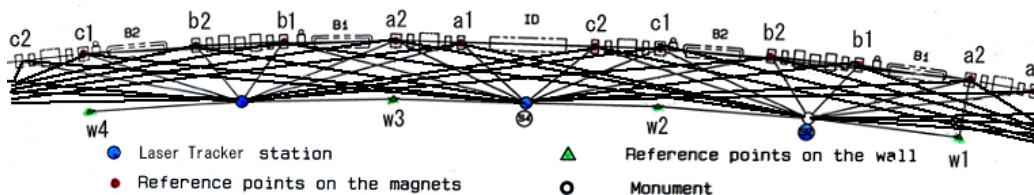


Figure 5: Control network. There are two tracker positions in one cell.

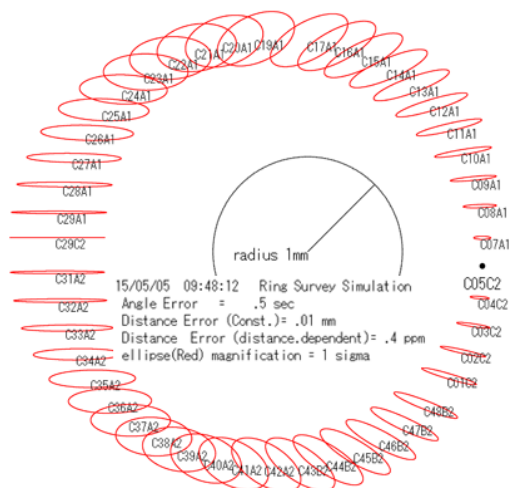


Figure 6: Absolute error ellipse at SPring-8.

15/07/28 00:12:13 Ring Survey Simulation  
Angle Error = .5 sec  
Distance Error (Const.)= .01 mm  
Distance Error (distance.dependent)= .4 ppm  
ellipse(Red) magnification = 1 sigma

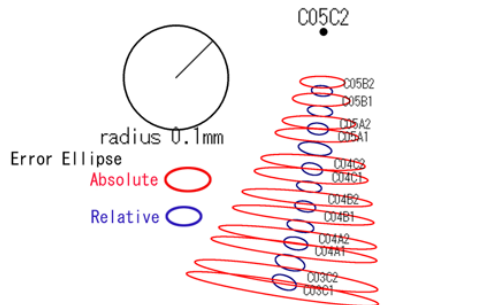


Figure 7: Absolute and relative error ellipse.

## 5. Excel VBA プログラム

### 5.1 Excel シート

Figure8(a),(b),(c)に示すように Excel の 1 枚のシートに必要な数値を入力し、BASIC を Excel VBA に書き直した VBA のプログラムを走らせることで同じシートに計算結果と誤差楕円が出力される。  
解析パラメーター：角度と距離の精度、繰返し回数。  
既知点：2次元 XY で座標を与える。測量点は番号でなくても名前をそのまま使える。BASIC と異なり VBA では大文字小文字の区別がされる。固定点（既知点）は多くても構わないがこのプログラムでは 1 点だけでは計算できない。  
半固定点：1 点は固定で、もう一点までの正確な距離がわからない場合、その点の X 又は Y 座標だけを観測方程式の行列の対応する”列”だけ 0 にすることで固定できるように改修した。  
未知点：XY の近似値を与える。この値が悪いと繰返しが多くなるが、その収束具合は標準偏差の変化（出力ファイルの Reduction ratio）でわかる。  
測定データ部：角度も距離も同時に得られる場合が多いがこのプログラムではまず角度だけ、次に距離だけと別々に入力する。

入力データの区切り：Figure 8(a)のように数字 9999 と上の行と同様にセルを 0 で埋める。各データ間に空白行は入れない。入力が済めば VBA を走らせる。解析時：時刻と実行しているコメントを出力する。結果出力部：計算した日時、計算に用いたパラメーター、既知点、（半固定点）未知点の座標が出力される。また、角度や距離の測定数、各繰返しでの標準偏差とその減衰の比、誤差楕円の長軸、短軸の長さと同軸、さらに計算結果から推測される

	A	B	C	D	E	F	G	H
1	Survey							
2	Analysys Parameters							
3	Angle- Distance Weight							
4	Angle Error ( 1 sigma )			1.000	sec			
5	Distance Error (Consta)			0.01	mm			
6	Distance Error (Distanc			2.5	ppm			
8	Coordinate	right hand (ex.)	+X=West direction	+Y=North direction				
9		fix one coordinate X or Y	Jfix=1 (X) =2(Y) =0(none)					
10		Jfix=1 or 2	→half known point=first Unknown point					
11	JFIX			2				
13	Iteration			3	times			
15			X(m)	Y (m)				
16	Known Poi	Mt32	0.00000	0.00000				
17		9999	0	0				
18	Unknown F	Mt37	130.99995	0.00000				
19		Mt14	-43.11770	0.00000				
20		Mt31	-19.00000	0.00000				
21		Mt33	26.00000	0.00000				
22		Mt34	50.50000	0.00000				
24		9999	0	0				
25	Mrasured	Mt14	T3_1	MtSCSScb0	192.1823			
26	Angle	Mt14	T3_1	BT21_1	355.7378			
27		Mt14	T3_1	BT21_2	204.1411			
29		Mt39	T3_10	W10N	2.0906			
30		9999	0	0				
31	Measured	T3_1	MtSCSSgur	11.084221				
32	Distance	T3_1	MtSCSSbc:	22.632399				
34		Mt37	Mt39	55.500037				
35		9999	0.000	0.000				
37	Analyzing	Iteration=	1	Time=	16:01:52	Making normal equation		
38		Iteration=	1	Time=	16:01:52	Solving equation		
39		Iteration=	2	Time=	16:01:52	Making normal equation		
40		Iteration=	2	Time=	16:01:53	Solving equation		

Fig. 8(a)

42	Result	2015/7/28	16:01:53					
43	Angle Error=		1.0	sec				
44	Distance error(Const)=		0.010	mm				
45	Distance error(distance		2.5	ppm				
47	Iteration times=		3					
49	Position (Known points)							
50	Point	X(m)	Y(m)					
51	Mt32	0.00000	0.00000					
53	Half known point							
54	half known (X(m)	X(mm)	X(m)	Y(m)				
55	Point	Assumed	correction	Adjusted	Fixed Y			
56	Mt37	130.99995	0.034	130.99998	0.00000	0.00000		
57	Unknown Point							
58	Point	Assumed	correction	Adjusted	Assumed	correctio	Adjusted	
59		(m)	(mm)	(m)	(m)	(m)	(m)	
60	Mt14	-43.11770	0.47	-43.11723	0.00000	-0.07	-0.00007	
61	Mt31	-19.00000	-0.08	-19.00008	0.00000	-0.08	-0.00008	
62	Mt33	26.00000	0.02	26.00002	0.00000	0.02	0.00002	
63	Mt34	50.50000	0.14	50.50014	0.00000	0.19	0.00019	
64	Mt35	75.00000	0.06	75.00006	0.00000	0.37	0.00037	
66	Known Point number=	1	Un Known Point numb	49	Total Point	50		
67	Measured Angle number	111	Distance number=	133				
68	Measurement Error (standard deviat	2.0E-06	0.040	sec				
70	Iteration Standard Deviation(rad) and Reduction ratio							
71		1	3.8E-04	0.0E+00				
72		2	2.0E-06	7.2E-02				
73	Error Ellipse (Absolute)							
74	Point	EW(mm)	NS(mm)	SNEW(mm)	Max(mm)	Min(mm)	Angle(deg)	
75	Mt37	0.023	0.000	0.000	0.023	0.000	0.00	
76	Mt14	0.016	0.068	0.010	0.068	0.016	-0.02	
77	Mt31	0.010	0.031	-0.002	0.031	0.010	0.01	
78	Mt33	0.011	0.038	0.006	0.038	0.011	-0.02	

Fig. 8(b)

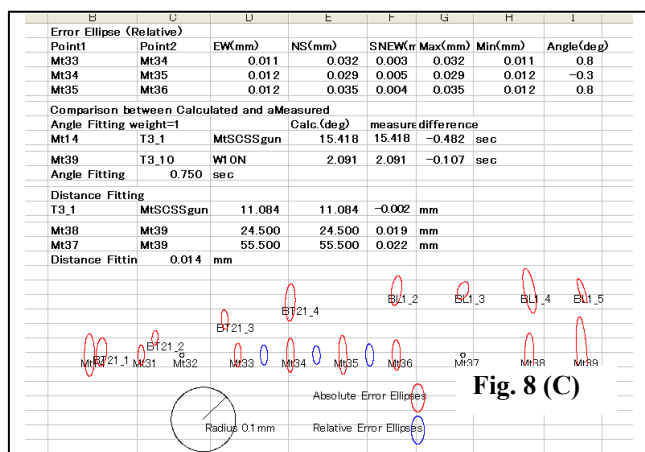


Figure 8: Excel worksheet for the survey calculation.  
(a) Analysis parameters and measured data.  
(b) Calculated Results.  
(c) The values and figures of error ellipses.

角度や距離と実際の測定値との差を出力する。この値が大きいくところでは、入力ミスや測量が良くない、など疑われるのでチェックする。角度だけ、距離だけの平均値も出力される。この2つの平均値は解析する時の角度と距離の精度の設定バランスにも依存する。角度精度を小さくすれば距離の fitting の値は大きくなり逆に距離の精度を小さくすれば角度の fitting の値は大きくなる。

## 5.2 VBA プログラム

宣言部はわずかだが、データの読み込みと絶対的と相対的誤差楕円の計算まで省かずに以下載せる。誤差楕円のグラフィックは結果のみ Figure 8(c)に示した。

ガウスの掃き出し法で連立一次方程式を解く時逆行列を計算するが、行数が多くなるため桁落ち防止のため軸の数値がそれより下の行の中で最大値に交換する Pivotting の改造をした。

### Sub Macro1()

```

Dim i As Integer, i0 As Integer, i1 As Integer, ii As Integer
Dim at As Double, b2 As Double, b4 As Double
mp = 500: mn = 2900: mj = mp * 2:
Dim AL(2900) As Double, a(2900, 2900 / 2) As Double,
Dim QK(2900, 3) As String, QP(500) As String,
pi = 3.1415926535: sqpi = Sqr(pi): ra = pi / 180#: radsec = 3600# / ra
' ----- Weight Parameters , Iteration times, fixed point---
Range("D5").Select
rmt = ActiveCell.Value / radsec: Selection.Offset(1, 0).Select
u1 = ActiveCell.Value * 0.001: Selection.Offset(1, 0).Select
r1 = ActiveCell.Value * 0.000001
jfix = Range("D9"): maxiter = Range("D12")
Range("B16").Select
' ----- Read X, Y Data -----
j = 1
Do Until ActiveCell.Value = "9999"
    QP(j) = ActiveCell.Value: Selection.Offset(0, 1).Select
    XJ(j, 1) = ActiveCell.Value: Selection.Offset(0, 1).Select
    YJ(j, 1) = ActiveCell.Value: Selection.Offset(1, -2).Select

```

```

j = j + 1
Loop
nr = j - 1: ' nr=number of known points
For ii = 1 To nr
    For iter = 2 To maxiter: XJ(ii, iter) = XJ(ii, 1): YJ(ii, iter) = YJ(ii, 1): Next iter
Next ii
Selection.Offset(1, 0).Select
' ----- Assumed X Y data -----
Do Until ActiveCell.Value = "9999"
    QP(j) = ActiveCell.Value: Selection.Offset(0, 1).Select
    XJ(j, 1) = ActiveCell.Value: Selection.Offset(0, 1).Select
    YJ(j, 1) = ActiveCell.Value: Selection.Offset(1, -2).Select
    j = j + 1
Loop
Selection.Offset(1, 0).Select
np = j - 1: nq = np - nr: np=number of total points nq= unknown points
For iter = 1 To maxiter
    For i = 1 To np
        For j = 1 To np: IAN(i, j) = 0: Next j
        Next i
'----- Initialize Matrix A(j,k) , AK(i)-----
n = nq * 2: ' N=twice of unknown points
For i = 1 To mn
    For j = 1 To mn / 2: a(i, j) = 0#: Next j
    P(i) = 0#: AL(i) = 0#: AK(i) = 0#
    Next i
' ----- Read angle data -----
i = 0: l = 0: sigma = rmt
Do
    l = l + 1
    If iter = 1 Then
        QK(l, 1) = ActiveCell.Value: Selection.Offset(0, 1).Select
        QK(l, 2) = ActiveCell.Value: Selection.Offset(0, 1).Select
        QK(l, 3) = ActiveCell.Value: Selection.Offset(0, 1).Select
        ANG(l) = ActiveCell.Value: Selection.Offset(1, -3).Select
        If ANG(l) < 0 Then ANG(l) = ANG(l) + 360#
        End If
    If QK(l, 1) = "9999" Then
        nang = l - 1
    Else
        For j = 1 To np
            If QK(l, 1) = QP(j) Then j1 = j
            If QK(l, 2) = QP(j) Then j0 = j
            If QK(l, 3) = QP(j) Then j2 = j
        Next j
'-- Calculation of Coefficients of observation eq.
c1 = XJ(j0, iter) - XJ(j1, iter): d1 = YJ(j0, iter) - YJ(j1, iter)
If (c1 = 0) * (d1 = 0) Then
    Else: e1 = d1 / (c1 ^ 2 + d1 ^ 2): f1 = -c1 / (c1 ^ 2 + d1 ^ 2)
    End If
c2 = XJ(j0, iter) - XJ(j2, iter): d2 = YJ(j0, iter) - YJ(j2, iter)
If (c2 = 0) * (d2 = 0) Then
    Else: e2 = d2 / (c2 ^ 2 + d2 ^ 2): f2 = -c2 / (c2 ^ 2 + d2 ^ 2)
    End If
t1 = arctan(-c1, -d1)
t2 = arctan(-c2, -d2)
at = ANG(l) * ra
If t2 < t1 Then t2 = t2 + 2 * pi
ANGL(l, iter) = (t2 - t1) / ra: BUN(l) = ANG(l) - ANGL(l, iter)
b2 = (t2 - t1) - at
If IAN(j1, j0) = 1 Then GoTo 10820: ' J1=reference angle

```

PASJ2015 THOM08

```

point
If i = 0 Then GoTo 10710: ' J0=survey instrument position
End If
    For j = 1 To np * 2: a(i, j) = CS(j): Next j
    AL(i) = cl: P(i) = -1# / i1
If QK$(1) = "9999" Then nt = i: Exit Do: ' NT=Angles number
10710: For j = 1 To np * 2: CS(j) = 0#: Next j
    cl = 0#: i1 = 0: IAN(j1, j0) = 1
    i = i + 1
    a(i, j1 * 2 - 1) = e1: a(i, j1 * 2) = f1
    a(i, j0 * 2 - 1) = -e1: a(i, j0 * 2) = -f1
    AL(i) = 0#: P(i) = 1#
    For j = 1 To 2 * np: CS(j) = CS(j) + a(i, j): Next j
    cl = cl + AL(i): i1 = i1 + 1
    i = i + 1
10820: a(i, j2 * 2 - 1) = e2: a(i, j2 * 2) = f2
    a(i, j0 * 2 - 1) = -e2: a(i, j0 * 2) = -f2
    AL(i) = b2: P(i) = 1#
    For j = 1 To np * 2: CS(j) = CS(j) + a(i, j): Next j
    cl = cl + AL(i): i1 = i1 + 1
    i = i + 1
Loop
'----- Read Distance data -----
    i = 1: l = 0
Do
    l = l + 1
    If iter = 1 Then
        QS(1, 1) = ActiveCell.Value: Selection.Offset(0, 1).Select
        QS(1, 2) = ActiveCell.Value: Selection.Offset(0, 1).Select
        SS(1) = ActiveCell.Value: Selection.Offset(1, -2).Select
    End If
If QS(1) = "9999" Then ns = i - 1: ndis = l - 1: Exit Do: ' NS=Distance number
    i0 = i + nt
    For j = 1 To np
        If QS(1, 1) = QP(j) Then j1 = j
        If QS(1, 2) = QP(j) Then j2 = j
    Next j
    c1 = XJ(j2, iter) - XJ(j1, iter): d1 = YJ(j2, iter) - YJ(j1, iter)
    s2 = c1 ^ 2 + d1 ^ 2: s1 = Sqr(s2)
    e1 = c1 / s2: f1 = d1 / s2
    a(i0, 2 * j1 - 1) = -e1: a(i0, 2 * j1) = -f1
    a(i0, 2 * j2 - 1) = e1: a(i0, 2 * j2) = f1
b4 = (s1 - SS(1)) / s1: SSR(1, iter) = s1: SBUN(1) = SS(1) -
SSR(1, iter)
'----- Weight (Distance) -----
    pd = rmt ^ 2 * s2 / (u1 ^ 2 + r1 ^ 2 * s2)
    AL(i0) = b4: P(i0) = pd
    i = i + 1
Loop
'--- components arrange of A
If iter = maxiter Then Exit For
For i = 1 To nt + ns
    For j = 1 To nq * 2: sss = a(i, j + 2 * nr): a(i, j) = sss: Next j
    Next i
    If (jfix = 1) Or (jfix = 2) Then
For i = 1 To mn: a(i, jfix) = 0#: Next i: ' fix X or Y coordinate
    End If
'----- Normal Equations A -----
    n = nq * 2: m = nt + ns
    For i = 1 To n
        For j = 1 To n: S(i, j) = 0#: Next j: AK(i) = 0#
    Next i
    For j = 1 To n
        For k = 1 To n: s3 = 0#
            For i = 1 To m: s3 = s3 + P(i) * a(i, j) * a(i, k): Next i
            S(j, k) = s3
        Next k
    Next j
    For j = 1 To n
        For i = 1 To m: AK(j) = AK(j) + P(i) * a(i, j) * AL(i): Next i
    Next j
'----- Inverse Matrix -----
    For k = 1 To n
        If k <> jfix Then
            smax1 = S(k, k)
        For i = k To n: If Abs(S(k, i)) >= smax1 Then smax1 = Abs(S(k, i)): ich = i
        Next i
        For j = 1 To n
            temp(j) = S(k, j): S(k, j) = S(ich, j): S(ich, j) = temp(j)
        Next j
        tempak = AK(k): AK(k) = AK(ich): AK(ich) = tempak
        t = 1# / S(k, k): S(k, k) = 1#
        For j = 1 To n: S(k, j) = S(k, j) * t: Next j
        For i = 1 To n
            If i <> k Then
                q = S(i, k): S(i, k) = 0#
                For j = 1 To n: S(i, j) = S(i, j) - q * S(k, j): Next j
            End If
        Next i
    End If
    Next k
'----- Calculate X -----
    For i = 1 To n: XX(i, iter) = 0#
        For j = 1 To n: XX(i, iter) = XX(i, iter) + S(i, j) * AK(j): Next j
    Next i
    j = 1
    For i = nr + 1 To np: XJ(i, iter + 1) = XJ(i, iter) - XX(j,
iter): YJ(i, iter + 1) = YJ(i, iter) - XX(j + 1, iter)
    j = j + 2
    Next i
'----- Standard Deviation -----
    For i = 1 To m: V(i) = 0
        For j = 1 To n: V(i) = V(i) + a(i, j) * XX(j, iter): Next j
        V(i) = V(i) + AL(i): 'AL(I) is lack in the textbook program
    Next i
    vv = 0#
    For i = 1 To m: vv = vv + P(i) * V(i) ^ 2: Next i
    SD(iter) = Sqr(vv / (m - n))
    sdifang = 0#: sdifdis = 0#
    For i = 1 To nang: sdifang = sdifang + (ANGL(i, iter) -
ANG(i)) ^ 2: Next i: Angfit(iter) = Sqr(sdifang / nang) * 3600
    For i = 1 To ndis: sdifdis = sdifdis + (SSR(i, iter) - SS(i)) ^
2: Next i: Disfit(iter) = Sqr(sdifdis / ndis) * 1000#
    Next iter
    'Print "ITER="; iter
For i = 2 To maxiter: Reduction(i) = Sqr(SD(i) / SD(i - 1)): Next i
'----- Error ellipse (Absolute) -----
    For i = 1 To nq
        SN_abs(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2 - 1, i * 2 -
1))) * 1000: EW_abs(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2, i *
2))) * 1000
        SNEW_abs(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2 - 1, i *
2))) * 1000 * Sgn(S(i * 2 - 1, i * 2))
        Angle_abs(i) = 0.5 / ra * Atn(2 * SNEW_abs(i) ^ 2 /

```

```
(SN_abs(i)^2 - EW_abs(i)^2) * Sgn(S(i * 2 - 1, i * 2))
  SMAX_abs(i) = Sqr(SN_abs(i)^2 + EW_abs(i)^2 +
Sqr((SN_abs(i)^2 - EW_abs(i)^2)^2 + 4 * SNEW_abs(i)^4) / Sqr(2)
  SMIN_abs(i) = Sqr(Abs(SN_abs(i)^2 + EW_abs(i)^2 -
Sqr((SN_abs(i)^2 - EW_abs(i)^2)^2 + 4 * SNEW_abs(i)^4))) / Sqr(2)
  Next i
  ' -----Error ellipse (Relative) -----
  For i = 1 To nq - 1
    SN_rela(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2 - 1, i * 2 -
1) + S(i * 2 + 1, i * 2 + 1) - 2 * S(i * 2 - 1, i * 2 + 1))) * 1000
    EW_rela(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2, i * 2) +
S(i * 2 + 2, i * 2 + 2) - 2 * S(i * 2, i * 2 + 2))) * 1000
    SNEW_rela(i) = SD(maxiter - 1) * Sqr(Abs(S(i * 2 - 1, i *
2) + S(i * 2 + 1, i * 2 + 2) - S(i * 2, i * 2 + 1) - S(i * 2 - 1, i * 2 +
2))) * 1000 * Sgn(S(i * 2 - 1, i * 2) + S(i * 2 + 1, i * 2 + 2) - S(i
* 2, i * 2 + 1) - S(i * 2 - 1, i * 2 + 2))
    Angle_rela(i) = 0.5 / ra * Atn(2 * SNEW_rela(i)^2 /
(SN_rela(i)^2 - EW_rela(i)^2)) * Sgn(S(i * 2 - 1, i * 2) + S(i *
2 + 1, i * 2 + 2) - S(i * 2, i * 2 + 1) - S(i * 2 - 1, i * 2 + 2)) +
(Sgn(EW_rela(i)^2 - SN_rela(i)^2) + 1) * 45 * ra
    SMAX_rela(i) = Sqr(SN_rela(i)^2 + EW_rela(i)^2 +
Sqr((SN_rela(i)^2 - EW_rela(i)^2)^2 + 4 * SNEW_rela(i)^4) / Sqr(2)
    SMIN_rela(i) = Sqr(Abs(SN_rela(i)^2 + EW_rela(i)^2 -
Sqr((SN_rela(i)^2 - EW_rela(i)^2)^2 + 4 * SNEW_rela(i)^4))) / Sqr(2)
  Next i
```

```
Function arctan(x As Double, y As Double) As Double
' arc tangent
Dim pi As Double
pi = 3.141592653
  arctan = Atn(y / x)
  If x < 0 Then
    arctan = arctan + pi
  ElseIf y < 0 Then
    arctan = arctan + pi * 2
  End If
End Function
```

### 5.3 計算時間の比較

FBASIC のコンパイラと VBA で同様のプログラムを Windows7 のノート PC (Core (TM)i3 CPU 1.33GHz) での 1 回計算時間 ( )内の時間は Windows XP Pentium(R)D CPU 3.4GHz 使用) は Table 1 の通り VBA の方がかなり速いことがわかった。

Table 1: Comparison of Calculation Time between Application FBASIC and That of Excel VBA

場所	測量点	角度	距離	FBASIC	VBA
光源棟	50	111	133	<1 秒	<1 秒
蓄積リ ング	486	1300	1400	26 分 (36 分)	10 分 (19 分)

測量点数が 500、測量の数が角度、距離含め 3000 程度なら観測方程式の行列の要素数は 3000 行×1000 列=300 万にのぼるが 10 分程度で計算できるのでもう少し多くても実用的な範囲内と考えられる。

## 6. まとめ

### 6.1 BASIC 系ソフト (Excel VBA 含む)

- BASIC のプログラムから Excel VBA への変換は同じ文法が使える等スムーズに実現できた。
- Excel (VBA) で解析でき使い勝手が向上した。
- シミュレーションや相対的な誤差楕円の計算など目的や設定が決まっている場合は測量器に付属しているソフトに比べ使いやすい。
- VBA でも計算と同時に誤差楕円も図示できる。
- 今回の計算時間は FBASIC より VBA の方が速かった。
- 外部からダウンロードできるところに VBA のソフトを置き、使用者が使いやすい様に改良していただければと考えている。(それまでは必要ならメールで問い合わせてください。)

### 6.2 シミュレーション結果

ライカ製 AT402 の実力を十分発揮できれば、蓄積リングの架台間の半径方向のエラーは 20 μm(1σ)程度で収まり、また周長のエラーは極わずかでシステムティックエラーのみによることがわかった。

### 6.3 測定値のエラーの評価

空気などの環境やターゲット、測量器の実力から測定値のエラーの見積もりはシミュレーションと実測値の比較からある程度評価が可能だと考えられる。

## 参考文献

- [1] 細野武庸他, “測量叢書 第 1 巻 改訂版 基準点測量”, 日本測量協会 (1992).
- [2] 原田健久, “わかりやすい測量厳密計算法”, 鹿島出版会 (1992).
- [3] 石川甲子男他, “パーソナルコンピュータによる測量計算プログラム”, 山海堂 (1991).
- [4] TJM Kennie and G Petrie, “Engineering Surveying Technology”, p163.