一、作業名稱

Relative Orientation(swing-swing method) 相對方位(雙像旋轉法) [共面式解法]

二、你的班級、姓名、學號

測量115 蔡意嫺 F64116126

三、作業目的

了解共面條件，並通過雙像旋轉法、解析法相對方位，利用影像來重建物體的三維物空間及光束。可以深入理解如何利用影像數據來重建空間模型，並學習應用攝影測量原理來解決實際問題。

四、解析法相對方位(AR)的意義及其用途

解析法相對方位是一種透過精密的數學計算和嚴密的平差方法，來確定立體像對相對位置的技術。核心概念是利用共面條件來建立兩張影像間的相對方位，並以此來推導出立體三維模型的參數。解析法相對方位的目的在於重建立體三維模型，通過計算出兩張影像的外方位元素，ex.旋轉角度，實現兩張影像在三維空間中的相對對位。

在實際應用，解析法相對方位在建築測繪、古蹟數位化等相關領域，有重要的用途。其能夠提供精確的三維坐標數據，並為後續三維建模和其他相關應用，提供重要數據。通過這種方法，能提高數據處理的效率，也能提升重建模型的細節和精度，對於需要高精度任務很重要。

五、你的計算程式使用的計算步驟及流程圖

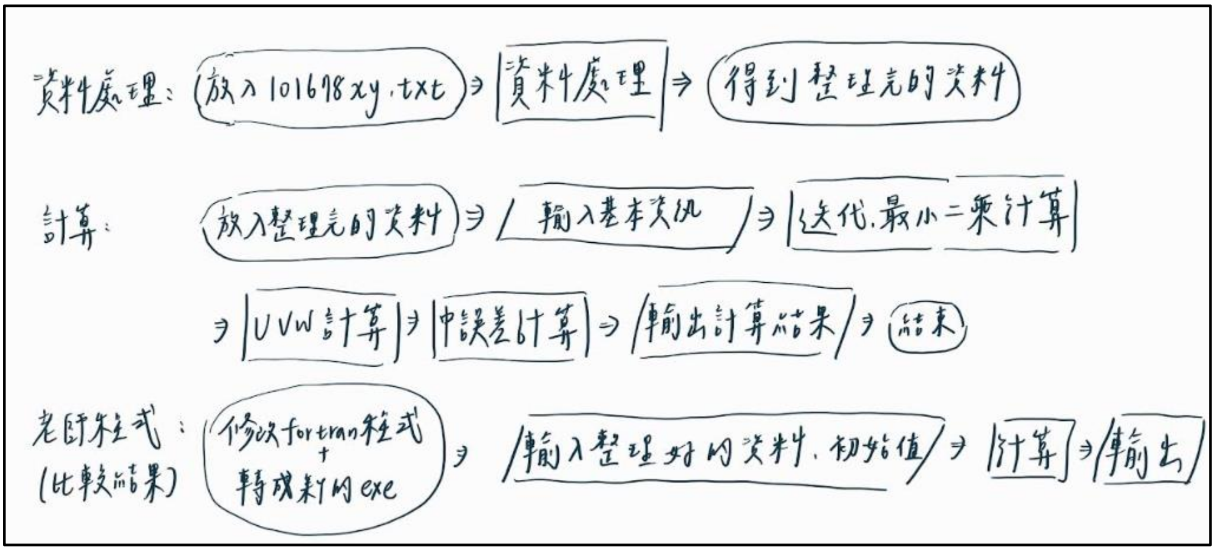
由於作業延後一周繳交，故有時間準備自己的程式碼和修改老師的，因此下列將兩者都介紹。

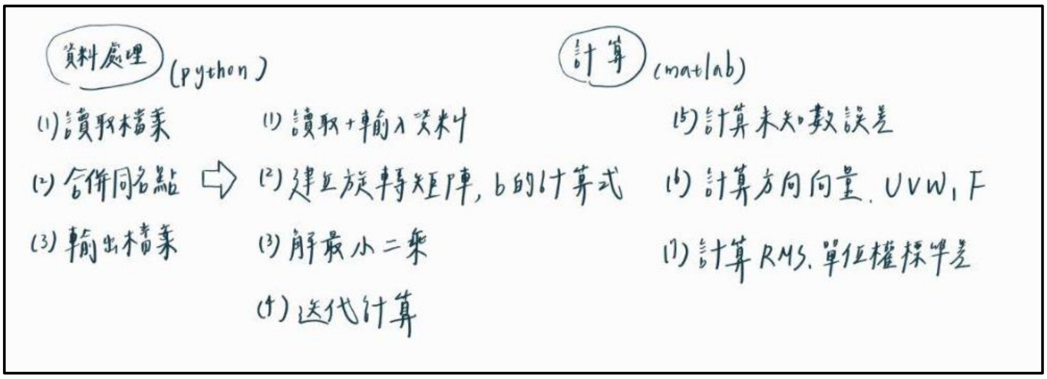
詳細的程式算步驟在第七點步驟5.，這裡只列大綱步驟。

1.使用python做前資料處理，得到一個新的txt檔

2.匯入matlab，做迭代、UVW等的計算，得到計算結果

3.編輯老師的fortran，重新得到一個exe，做輸入、計算、輸出，並得出結果來比較



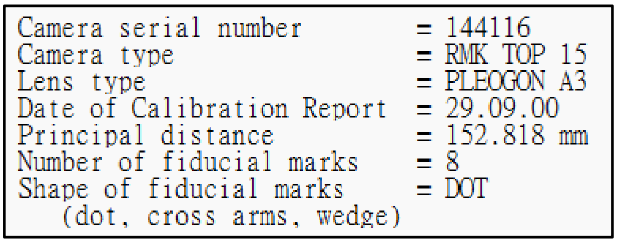


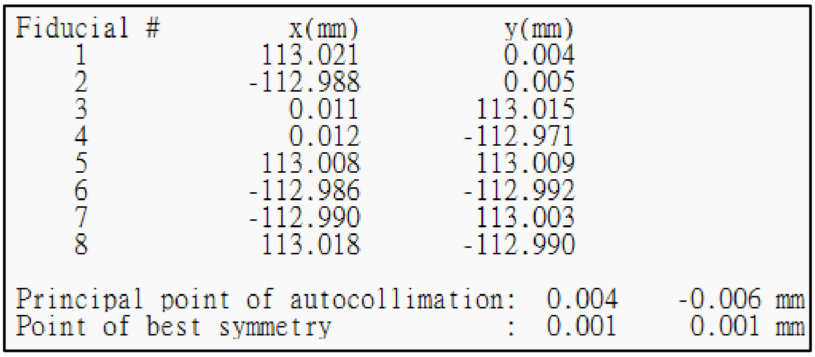
六、你給定5個未知數近似值的做法

為了使計算更簡單，減少不必要的誤差，將五個未知數，phie_L、kapa_L、omega_R、phie_R、kapa_R，皆設為0(degree)。

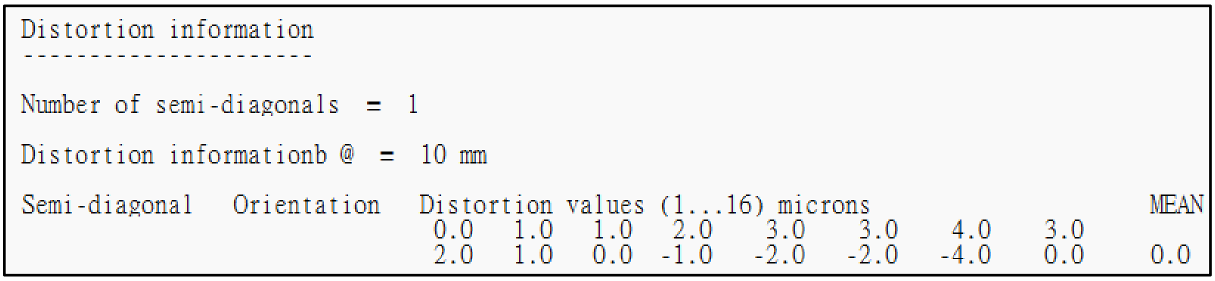
七、作業步驟1至步驟10的AR操作內容及其相關數據

步驟1.相機(VIRTUOZO CAMERA)率定參數檔


```
Camera serial number           = 144116
Camera type                    = RMK TOP 15
Lens type                      = PLEOGON A3
Date of Calibration Report     = 29.09.00
Principal distance             = 152.818 mm
Number of fiducial marks       = 8
Shape of fiducial marks        = DOT
   (dot, cross arms, wedge)
```



```
Fiducial #          x(mm)          y(mm)
    1             113.021          0.004
    2            -112.988          0.005
    3               0.011        113.015
    4               0.012       -112.971
    5             113.008        113.009
    6            -112.986       -112.992
    7            -112.990        113.003
    8             113.018       -112.990

Principal point of autocollimation:  0.004     -0.006 mm
Point of best symmetry             :  0.001      0.001 mm
```



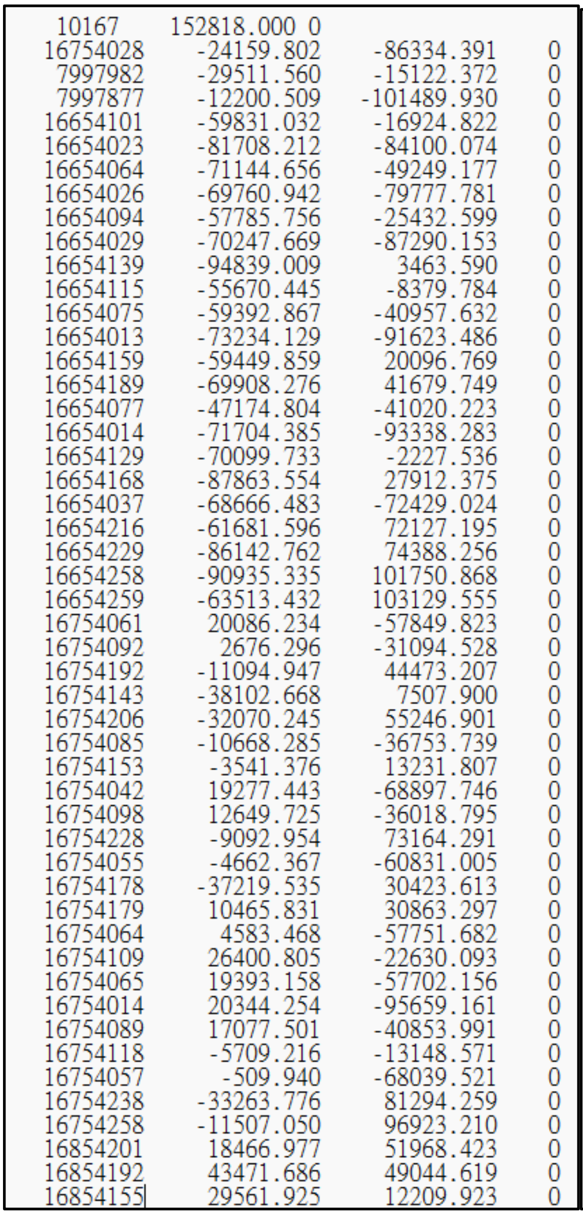
```
Distortion information
----------------------

Number of semi-diagonals  =  1

Distortion informationb @  =  10 mm

Semi-diagonal   Orientation   Distortion values (1...16) microns                       MEAN
                               0.0   1.0   1.0   2.0    3.0    3.0    4.0    3.0
                               2.0   1.0   0.0  -1.0   -2.0   -2.0   -4.0    0.0       0.0
```

步驟2.像坐標量測值

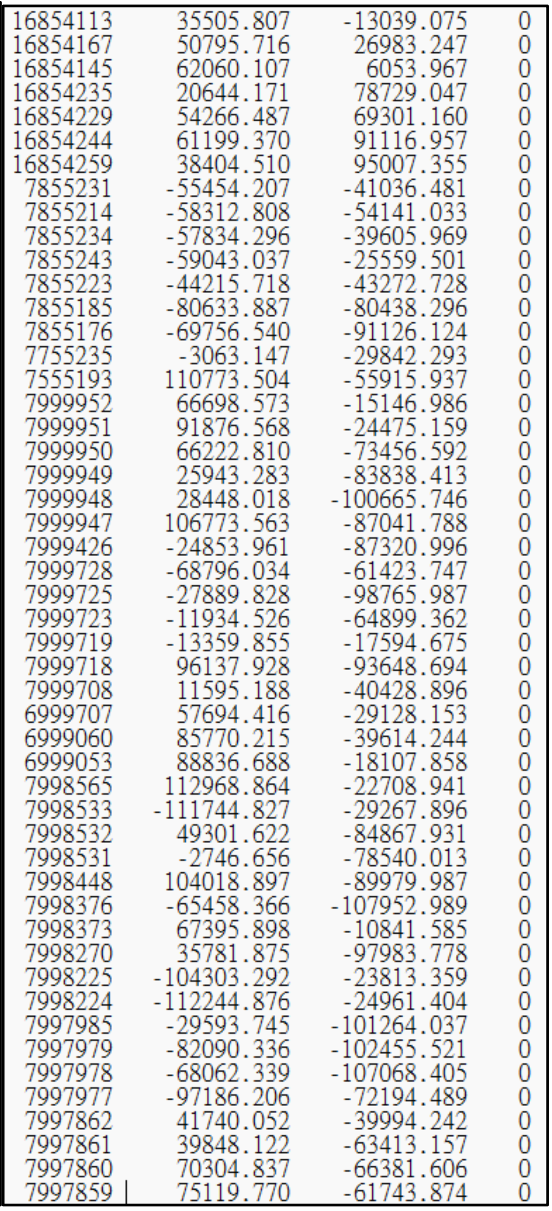
下圖第一行第一個值是第一張影像名稱(10167)、第二個值是焦距(152812.000μm)；第二行第一個值是點位名稱(16754028)、第二個值是x(-24159.802μm)、第三個值是y(-86.334.391μm)、第四個值是code。

接下來每行依序這個方式排列，直至-99停止，接著是下一張圖的資料(10168)。

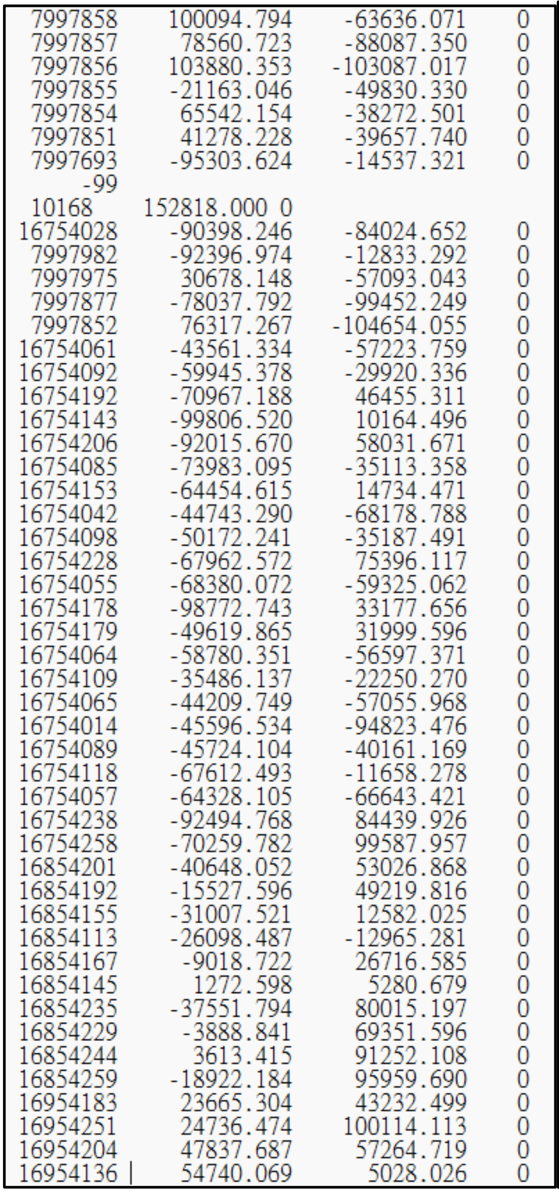
下四圖為兩兩左右排列。



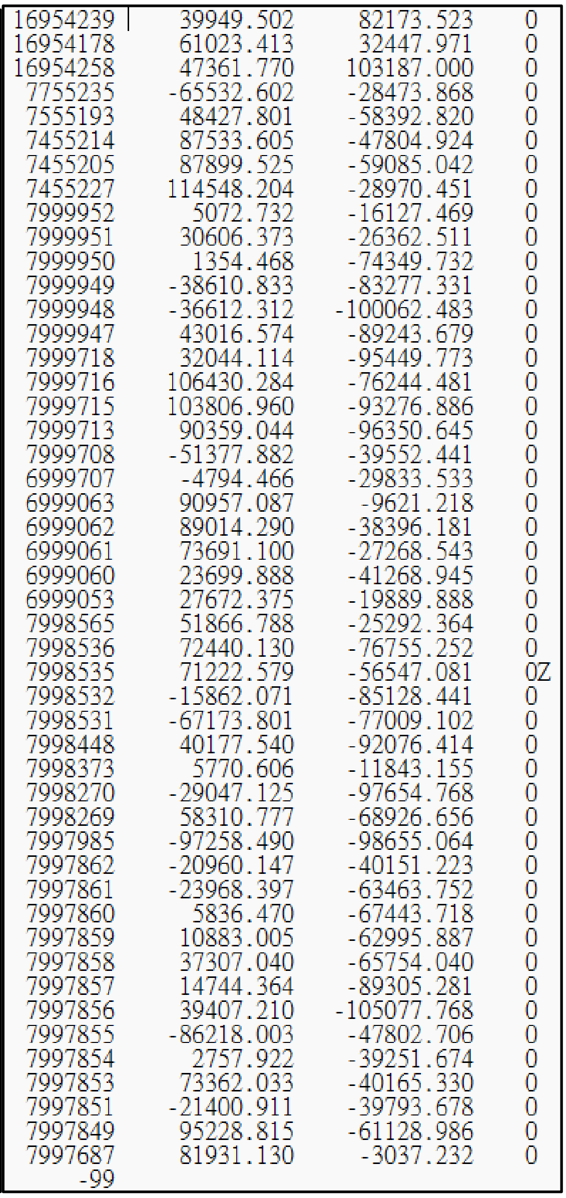
```
   10167     152818.000 0
16754028      -24159.802      -86334.391      0
 7997982      -29511.560      -15122.372      0
 7997877      -12200.509     -101489.930      0
16654101      -59831.032      -16924.822      0
16654023      -81708.212      -84100.074      0
16654064      -71144.656      -49249.177      0
16654026      -69760.942      -79777.781      0
16654094      -57785.756      -25432.599      0
16654029      -70247.669      -87290.153      0
16654139      -94839.009        3463.590      0
16654115      -55670.445       -8379.784      0
16654075      -59392.867      -40957.632      0
16654013      -73234.129      -91623.486      0
16654159      -59449.859       20096.769      0
16654189      -69908.276       41679.749      0
16654077      -47174.804      -41020.223      0
16654014      -71704.385      -93338.283      0
16654129      -70099.733       -2227.536      0
16654168      -87863.554       27912.375      0
16654037      -68666.483      -72429.024      0
16654216      -61681.596       72127.195      0
16654229      -86142.762       74388.256      0
16654258      -90935.335      101750.868      0
16654259      -63513.432      103129.555      0
16754061       20086.234      -57849.823      0
16754092        2676.296      -31094.528      0
16754192      -11094.947       44473.207      0
16754143      -38102.668        7507.900      0
16754206      -32070.245       55246.901      0
16754085      -10668.285      -36753.739      0
16754153       -3541.376       13231.807      0
16754042       19277.443      -68897.746      0
16754098       12649.725      -36018.795      0
16754228       -9092.954       73164.291      0
16754055       -4662.367      -60831.005      0
16754178      -37219.535       30423.613      0
16754179       10465.831       30863.297      0
16754064        4583.468      -57751.682      0
16754109       26400.805      -22630.093      0
16754065       19393.158      -57702.156      0
16754014       20344.254      -95659.161      0
16754089       17077.501      -40853.991      0
16754118       -5709.216      -13148.571      0
16754057        -509.940      -68039.521      0
16754238      -33263.776       81294.259      0
16754258      -11507.050       96923.210      0
16854201       18466.977       51968.423      0
16854192       43471.686       49044.619      0
16854155       29561.925       12209.923      0
```



```
16854113       35505.807      -13039.075      0
16854167       50795.716       26983.247      0
16854145       62060.107        6053.967      0
16854235       20644.171       78729.047      0
16854229       54266.487       69301.160      0
16854244       61199.370       91116.957      0
16854259       38404.510       95007.355      0
 7855231      -55454.207      -41036.481      0
 7855214      -58312.808      -54141.033      0
 7855234      -57834.296      -39605.969      0
 7855243      -59043.037      -25559.501      0
 7855223      -44215.718      -43272.728      0
 7855185      -80633.887      -80438.296      0
 7855176      -69756.540      -91126.124      0
 7755235       -3063.147      -29842.293      0
 7555193      110773.504      -55915.937      0
 7999952       66698.573      -15146.986      0
 7999951       91876.568      -24475.159      0
 7999950       66222.810      -73456.592      0
 7999949       25943.283      -83838.413      0
 7999948       28448.018     -100665.746      0
 7999947      106773.563      -87041.788      0
 7999426      -24853.961      -87320.996      0
 7999728      -68796.034      -61423.747      0
 7999725      -27889.828      -98765.987      0
 7999723      -11934.526      -64899.362      0
 7999719      -13359.855      -17594.675      0
 7999718       96137.928      -93648.694      0
 7999708       11595.188      -40428.896      0
 6999707       57694.416      -29128.153      0
 6999060       85770.215      -39614.244      0
 6999053       88836.688      -18107.858      0
 7998565      112968.864      -22708.941      0
 7998533     -111744.827      -29267.896      0
 7998532       49301.622      -84867.931      0
 7998531       -2746.656      -78540.013      0
 7998448      104018.897      -89979.987      0
 7998376      -65458.366     -107952.989      0
 7998373       67395.898      -10841.585      0
 7998270       35781.875      -97983.778      0
 7998225     -104303.292      -23813.359      0
 7998224     -112244.876      -24961.404      0
 7997985      -29593.745     -101264.037      0
 7997979      -82090.336     -102455.521      0
 7997978      -68062.339     -107068.405      0
 7997977      -97186.206      -72194.489      0
 7997862       41740.052      -39994.242      0
 7997861       39848.122      -63413.157      0
 7997860       70304.837      -66381.606      0
 7997859       75119.770      -61743.874      0
```



```
 7997858    100094.794    -63636.071     0
 7997857     78560.723    -88087.350     0
 7997856    103880.353   -103087.017     0
 7997855    -21163.046    -49830.330     0
 7997854     65542.154    -38272.501     0
 7997851     41278.228    -39657.740     0
 7997693    -95303.624    -14537.321     0
     -99
 10168     152818.000 0
16754028    -90398.246    -84024.652     0
 7997982    -92396.974    -12833.292     0
 7997975     30678.148    -57093.043     0
 7997877    -78037.792    -99452.249     0
 7997852     76317.267   -104654.055     0
16754061    -43561.334    -57223.759     0
16754092    -59945.378    -29920.336     0
16754192    -70967.188     46455.311     0
16754143    -99806.520     10164.496     0
16754206    -92015.670     58031.671     0
16754085    -73983.095    -35113.358     0
16754153    -64454.615     14734.471     0
16754042    -44743.290    -68178.788     0
16754098    -50172.241    -35187.491     0
16754228    -67962.572     75396.117     0
16754055    -68380.072    -59325.062     0
16754178    -98772.743     33177.656     0
16754179    -49619.865     31999.596     0
16754064    -58780.351    -56597.371     0
16754109    -35486.137    -22250.270     0
16754065    -44209.749    -57055.968     0
16754014    -45596.534    -94823.476     0
16754089    -45724.104    -40161.169     0
16754118    -67612.493    -11658.278     0
16754057    -64328.105    -66643.421     0
16754238    -92494.768     84439.926     0
16754258    -70259.782     99587.957     0
16854201    -40648.052     53026.868     0
16854192    -15527.596     49219.816     0
16854155    -31007.521     12582.025     0
16854113    -26098.487    -12965.281     0
16854167     -9018.722     26716.585     0
16854145      1272.598      5280.679     0
16854235    -37551.794     80015.197     0
16854229     -3888.841     69351.596     0
16854244      3613.415     91252.108     0
16854259    -18922.184     95959.690     0
16954183     23665.304     43232.499     0
16954251     24736.474    100114.113     0
16954204     47837.687     57264.719     0
16954136 |   54740.069      5028.026     0
```



```
16954239 |   39949.502     82173.523     0
16954178     61023.413     32447.971     0
16954258     47361.770    103187.000     0
 7755235    -65532.602    -28473.868     0
 7555193     48427.801    -58392.820     0
 7455214     87533.605    -47804.924     0
 7455205     87899.525    -59085.042     0
 7455227    114548.204    -28970.451     0
 7999952      5072.732    -16127.469     0
 7999951     30606.373    -26362.511     0
 7999950      1354.468    -74349.732     0
 7999949    -38610.833    -83277.331     0
 7999948    -36612.312   -100062.483     0
 7999947     43016.574    -89243.679     0
 7999718     32044.114    -95449.773     0
 7999716    106430.284    -76244.481     0
 7999715    103806.960    -93276.886     0
 7999713     90359.044    -96350.645     0
 7999708    -51377.882    -39552.441     0
 6999707     -4794.466    -29833.533     0
 6999063     90957.087     -9621.218     0
 6999062     89014.290    -38396.181     0
 6999061     73691.100    -27268.543     0
 6999060     23699.888    -41268.945     0
 6999053     27672.375    -19889.888     0
 7998565     51866.788    -25292.364     0
 7998536     72440.130    -76755.252     0
 7998535     71222.579    -56547.081     0Z
 7998532    -15862.071    -85128.441     0
 7998531    -67173.801    -77009.102     0
 7998448     40177.540    -92076.414     0
 7998373      5770.606    -11843.155     0
 7998270    -29047.125    -97654.768     0
 7998269     58310.777    -68926.656     0
 7997985    -97258.490    -98655.064     0
 7997862    -20960.147    -40151.223     0
 7997861    -23968.397    -63463.752     0
 7997860      5836.470    -67443.718     0
 7997859     10883.005    -62995.887     0
 7997858     37307.040    -65754.040     0
 7997857     14744.364    -89305.281     0
 7997856     39407.210   -105077.768     0
 7997855    -86218.003    -47802.706     0
 7997854      2757.922    -39251.674     0
 7997853     73362.033    -40165.330     0
 7997851    -21400.911    -39793.678     0
 7997849     95228.815    -61128.986     0
 7997687     81931.130     -3037.232     0
     -99
```

步驟3.

共面條件式:

$BX = XL2 - XL1$

$BY = YL2 - YL1$

$ZX = ZL2 - ZL1$

XL、YL、ZL為攝影中心座標

$$\begin{vmatrix} B_X & B_Y & B_Z \\ D_1 & E_1 & F_1 \\ D_2 & E_2 & F_2 \end{vmatrix} = B_X(E_1F_2 - E_2F_1) + B_Y(F_1D_2 - F_2D_1) + B_Z(D_1E_2 - D_2E_1) = 0$$

線性化方程式:

$$H = H_0 + \left(\frac{\partial H}{\partial \varphi}\right)_0 d\varphi + \left(\frac{\partial H}{\partial K}\right)_0 dK + \left(\frac{\partial H}{\partial \omega'}\right)_0 d\omega' + \left(\frac{\partial H}{\partial \varphi'}\right)_0 d\varphi' + \left(\frac{\partial H}{\partial K'}\right)_0 dK'$$

觀測方程式的一般通式:

$$b_1 d\varphi + b_2 dK + b_3 d\omega' + b_4 d\varphi' + b_5 dK' = -H_0 + v_H$$

每一個偏微分係數的詳細公式:

$$b_1 = \begin{vmatrix} B_x & B_y & B_z \\ \left(\frac{\partial D_1}{\partial \varphi}\right) & \left(\frac{\partial E_1}{\partial \varphi}\right) & \left(\frac{\partial F_1}{\partial \varphi}\right) \\ D_2 & E_2 & F_2 \end{vmatrix} \quad b_2 = \begin{vmatrix} B_x & B_y & B_z \\ x_1(m_{21})_1 - y_1(m_{11})_1 & x_1(m_{22})_1 - y_1(m_{12})_1 & x_1(m_{23}) - y_1(m_{13}) \\ D_2 & E_2 & F_2 \end{vmatrix}$$

$$b_3 = \begin{vmatrix} B_x & B_y & B_z \\ D_1 & E_1 & F_1 \\ 0 & -F_2 & E_2 \end{vmatrix} \quad b_4 = \begin{vmatrix} B_x & B_y & B_z \\ D_1 & E_1 & F_1 \\ \left(\frac{\partial D_2}{\partial \varphi'}\right) & \left(\frac{\partial E_2}{\partial \varphi'}\right) & \left(\frac{\partial F_2}{\partial \varphi'}\right) \end{vmatrix}$$

$$b_5 = \begin{vmatrix} B_x & B_y & B_z \\ D_1 & E_1 & F_1 \\ x_2(m_{21})_2 - y_2(m_{11})_2 & x_2(m_{22})_2 - y_2(m_{12})_2 & x_2(m_{23}) - y_2(m_{13}) \end{vmatrix}$$

步驟4.寫出你給定5個未知數近似值的做法。

如前項，第六點(六、你給定5個未知數近似值的做法)所述。

步驟5.寫出你的(程式採用的)計算步驟。

(1)資料處理(python)

1.讀取檔案、分類點號、xy

2.合併同名點

3.單位從μm轉成mm

4.輸出檔案(格式:點名、左片x、左片y、右片x、右片y)

(2)計算(matlab)

1.讀取檔案(前項python輸出)

2.輸入初始值(phie_L、kapa_L、omega_R、phie_R、kapa_R)

3.輸入基線長、收斂條件值、相機焦距

4.計算自由度

5.定義基本算式:旋轉矩陣、 b值

6.計算最小二乘、迭代增量、未知數迭代後值、未知數中誤差

7.輸出迭代結果、自由度

8.用迭代結果，計算UVW、F值、vol值及其RMS和單位權中誤差

9.輸出UVW結果、F值、vol值

10.輸出剩餘結果

(3)老師程式修改、比較(fortran)

不是主要程式，不多做解釋

1.修改老師程式排版、創作者欄

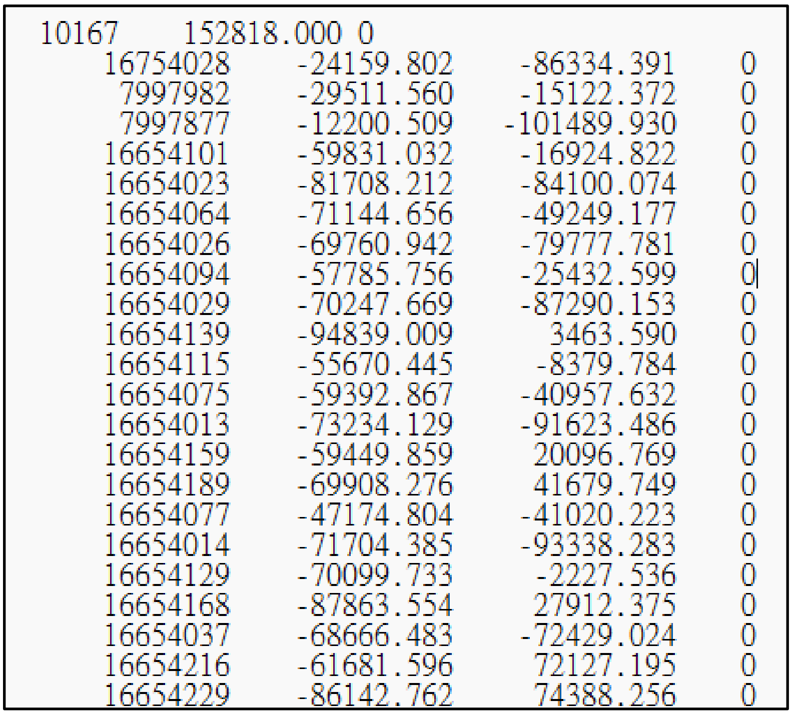
2.輸入像座標量測值、未知數近似值

3.程式計算

4.產出計算結果報表

<u>步驟6.寫出你如何準備/設計你的計算程式的輸入檔</u>

(1)先將101678xy.txt像座標量測值，以-99為界，依10167和10168的影像，按照一樣的格式，複製貼上分成兩個檔案(ex.data1.txt 、data2.txt)

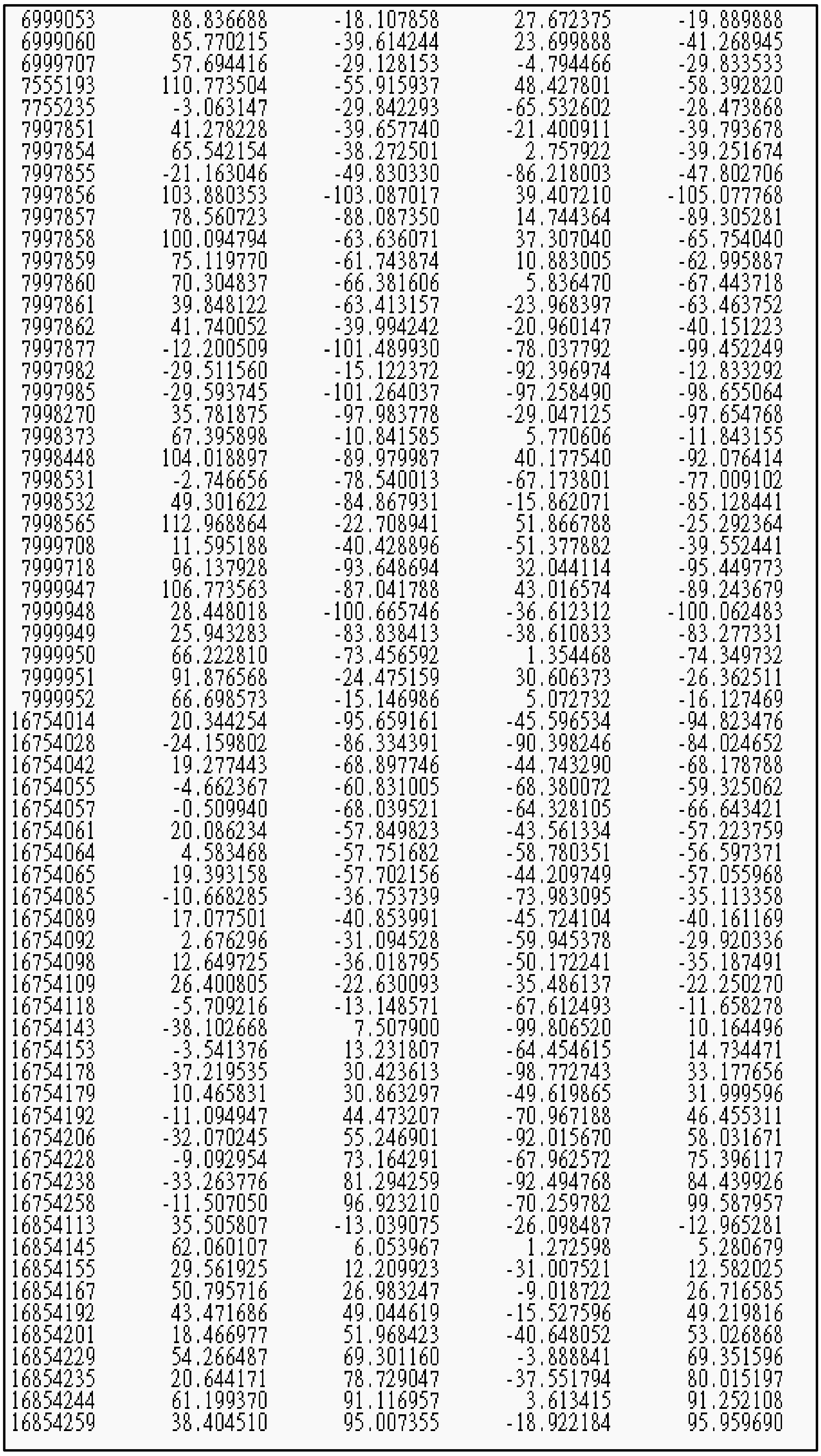


```
 10167     152818.000 0
     16754028     -24159.802     -86334.391     0
      7997982     -29511.560     -15122.372     0
      7997877     -12200.509    -101489.930     0
     16654101     -59831.032     -16924.822     0
     16654023     -81708.212     -84100.074     0
     16654064     -71144.656     -49249.177     0
     16654026     -69760.942     -79777.781     0
     16654094     -57785.756     -25432.599     0
     16654029     -70247.669     -87290.153     0
     16654139     -94839.009       3463.590     0
     16654115     -55670.445      -8379.784     0
     16654075     -59392.867     -40957.632     0
     16654013     -73234.129     -91623.486     0
     16654159     -59449.859      20096.769     0
     16654189     -69908.276      41679.749     0
     16654077     -47174.804     -41020.223     0
     16654014     -71704.385     -93338.283     0
     16654129     -70099.733      -2227.536     0
     16654168     -87863.554      27912.375     0
     16654037     -68666.483     -72429.024     0
     16654216     -61681.596      72127.195     0
     16654229     -86142.762      74388.256     0
```

(2)用python程式碼，定義一個函數，讀取兩個檔案，到-99為止，並將n、x、y三行，存在名為data的陣列裡面(程式第一步驟)

(3)用python程式碼，定義一個函數，合併同名像點，並把他寫入一個新的檔案(ex.output.txt)(程式第二步驟)

(4)獲得output檔，該檔有兩隻照片同點位的x、y，將原來的微米轉成毫米(*0.001)，做為輸入檔(程式第三步驟)。下圖，由左到右分別是點號、10167的xy、10168的xy。



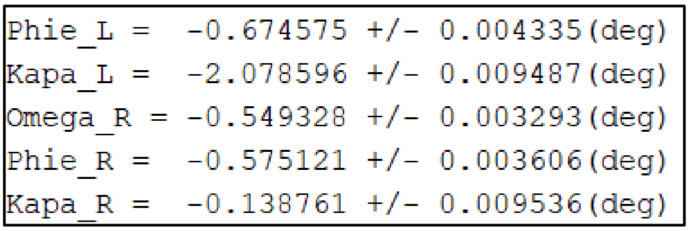
```
 6999053       88.836688      -18.107858       27.672375      -19.889888
 6999060       85.770215      -39.614244       23.699888      -41.268945
 6999707       57.694416      -29.128153       -4.794466      -29.833533
 7555193      110.773504      -55.915937       48.427801      -58.392820
 7755235       -3.063147      -29.842293      -65.532602      -28.473868
 7997851       41.278228      -39.657740      -21.400911      -39.793678
 7997854       65.542154      -38.272501        2.757922      -39.251674
 7997855      -21.163046      -49.830330      -86.218003      -47.802706
 7997856      103.880353     -103.087017       39.407210     -105.077768
 7997857       78.560723      -88.087350       14.744364      -89.305281
 7997858      100.094794      -63.636071       37.307040      -65.754040
 7997859       75.119770      -61.743874       10.883005      -62.995887
 7997860       70.304837      -66.381606        5.836470      -67.443718
 7997861       39.848122      -63.413157      -23.968397      -63.463752
 7997862       41.740052      -39.994242      -20.960147      -40.151223
 7997877      -12.200509     -101.489930      -78.037792      -99.452249
 7997982      -29.511560      -15.122372      -92.396974      -12.833292
 7997985      -29.593745     -101.264037      -97.258490      -98.655064
 7998270       35.781875      -97.983778      -29.047125      -97.654768
 7998373       67.395898      -10.841585        5.770606      -11.843155
 7998448      104.018897      -89.979987       40.177540      -92.076414
 7998531       -2.746656      -78.540013      -67.173801      -77.009102
 7998532       49.301622      -84.867931      -15.862071      -85.128441
 7998565      112.968864      -22.708941       51.866788      -25.292364
 7999708       11.595188      -40.428896      -51.377882      -39.552441
 7999718       96.137928      -93.648694       32.044114      -95.449773
 7999947      106.773563      -87.041788       43.016574      -89.243679
 7999948       28.448018     -100.665746      -36.612312     -100.062483
 7999949       25.943283      -83.838413      -38.610833      -83.277331
 7999950       66.222810      -73.456592        1.354468      -74.349732
 7999951       91.876568      -24.475159       30.606373      -26.362511
 7999952       66.698573      -15.146986        5.072732      -16.127469
16754014       20.344254      -95.659161      -45.596534      -94.823476
16754028      -24.159802      -86.334391      -90.398246      -84.024652
16754042       19.277443      -68.897746      -44.743290      -68.178788
16754055       -4.662367      -60.831005      -68.380072      -59.325062
16754057       -0.509940      -68.039521      -64.328105      -66.643421
16754061       20.086234      -57.849823      -43.561334      -57.223759
16754064        4.583468      -57.751682      -58.780351      -56.597371
16754065       19.393158      -57.702156      -44.209749      -57.055968
16754085      -10.668285      -36.753739      -73.983095      -35.113358
16754089       17.077501      -40.853991      -45.724104      -40.161169
16754092        2.676296      -31.094528      -59.945378      -29.920336
16754098       12.649725      -36.018795      -50.172241      -35.187491
16754109       26.400805      -22.630093      -35.486137      -22.250270
16754118       -5.709216      -13.148571      -67.612493      -11.658278
16754143      -38.102668        7.507900      -99.806520       10.164496
16754153       -3.541376       13.231807      -64.454615       14.734471
16754178      -37.219535       30.423613      -98.772743       33.177656
16754179       10.465831       30.863297      -49.619865       31.999596
16754192      -11.094947       44.473207      -70.967188       46.455311
16754206      -32.070245       55.246901      -92.015670       58.031671
16754228       -9.092954       73.164291      -67.962572       75.396117
16754238      -33.263776       81.294259      -92.494768       84.439926
16754258      -11.507050       96.923210      -70.259782       99.587957
16854113       35.505807      -13.039075      -26.098487      -12.965281
16854145       62.060107        6.053967        1.272598        5.280679
16854155       29.561925       12.209923      -31.007521       12.582025
16854167       50.795716       26.983247       -9.018722       26.716585
16854192       43.471686       49.044619      -15.527596       49.219816
16854201       18.466977       51.968423      -40.648052       53.026868
16854229       54.266487       69.301160       -3.888841       69.351596
16854235       20.644171       78.729047      -37.551794       80.015197
16854244       61.199370       91.116957        3.613415       91.252108
16854259       38.404510       95.007355      -18.922184       95.959690
```

步驟7.寫出你如何準備/設計你的計算程式的輸出檔。

1.matlab:

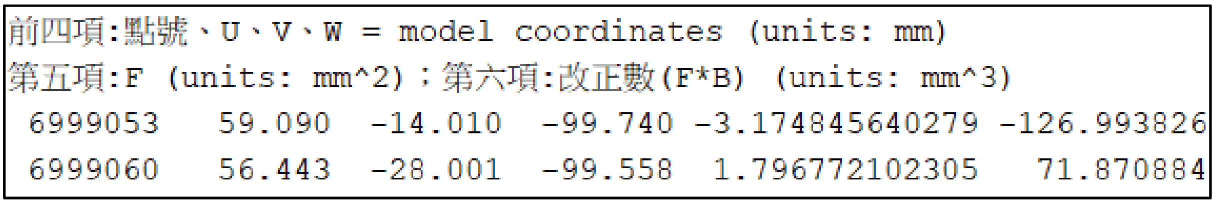
按照老師的輸出格式，並加上一些空行和輸出格式幫助排版。

(1)輸出初始值和未知數結果，取到小數點後6位，用matlab語法"%.6f"，視有效位數狀況更改。



```
Phie_L =  -0.674575 +/- 0.004335(deg)
Kapa_L =  -2.078596 +/- 0.009487(deg)
Omega_R = -0.549328 +/- 0.003293(deg)
Phie_R =  -0.575121 +/- 0.003606(deg)
Kapa_R =  -0.138761 +/- 0.009536(deg)
```

(2)輸出UVW和F、vol值等，因為是大型表格式資料，採用靠右邊，且取到小數點後3位的matlab語法"%8.3f"、%15.12f，視有效位數而定。



```
前四項:點號、U、V、W = model coordinates (units: mm)
第五項:F (units: mm^2)；第六項:改正數(F*B) (units: mm^3)
 6999053   59.090  -14.010  -99.740 -3.174845640279 -126.993826
 6999060   56.443  -28.001  -99.558  1.796772102305   71.870884
```

(3)其餘大概按照老師的fortran檔輸出，詳細的輸出資料在第九點

2.fortran:

用原有老師的檔案，做排版更改。

(1)下載fortran可以使用的編譯器(mingw64)

(2)將編譯器與電腦做連結，將連結貼到系統環境

(3)針對需要更改排版的部分，更改原本老師的ro.f程式檔，並將他存檔

(4)使用命令提示字元，將ro.f檔轉成能夠執行的exe檔

(5)運行並依次輸入需要的資料，產出輸出檔

步驟8.寫出你如何給定你的計算收斂條件。

因像坐標觀測值的最小單位為μm，故收斂條件須採用小於等於1μm的數值，因此設定收斂條件為10^-8（m）。

步驟9.依據步驟5至步驟8來撰寫你的計算程式。

詳細請看附錄、程式碼。

步驟10.使用10167、10168的全部同名像點對（corresponding image point pairs)來計算。

詳細請看附錄、程式碼及第九點。

八、你給定5個未知數近似值 = ?

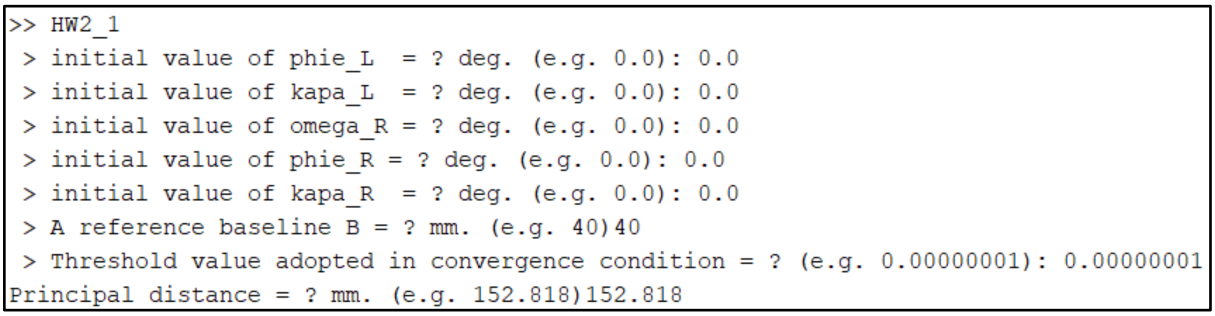
Initial values of 5 R.O. parameters (units: degree):

```
phie_L =  0.000000 : phie  angle of the left  photo
kapa_L =  0.000000 : kapa  angle of the left  photo
omega_R = 0.000000 : omega angle of the right photo
phie_R =  0.000000 : phie  angle of the right photo
kapa_R =  0.000000 : kapa  angle of the right photo
```

九、你的計算成果報表內容要點、成果數據及分析

1.計算程式輸出成果

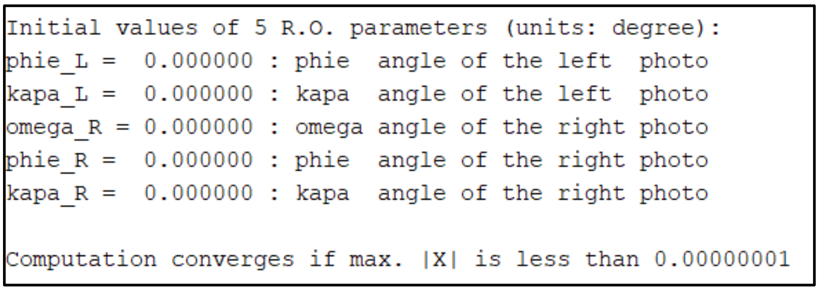
(1)輸入資料



```
>> HW2_1
 > initial value of phie_L  = ? deg. (e.g. 0.0): 0.0
 > initial value of kapa_L  = ? deg. (e.g. 0.0): 0.0
 > initial value of omega_R = ? deg. (e.g. 0.0): 0.0
 > initial value of phie_R = ? deg. (e.g. 0.0): 0.0
 > initial value of kapa_R  = ? deg. (e.g. 0.0): 0.0
 > A reference baseline B = ? mm. (e.g. 40)40
 > Threshold value adopted in convergence condition = ? (e.g. 0.00000001): 0.00000001
Principal distance = ? mm. (e.g. 152.818)152.818
```

輸入phie_L、kapa_L、omega_R、phie_R、kapa_R的初始近似值:0.0(deg)、基線長40(mm)、迭代閥值0.00000001(rad)、相機焦距152.818(mm)。這邊的相機焦距也是有給好的定值，因為我的處理輸入資料和計算是分開的兩個程式檔，為了讓使用者的彈性更高及更方便，這邊的焦距也是讓使用者自己輸入，旁邊括號有附上範例輸入，可以全部照著輸入就好。

(2)基本資訊



```
Initial values of 5 R.O. parameters (units: degree):
phie_L =  0.000000 : phie  angle of the left  photo
kapa_L =  0.000000 : kapa  angle of the left  photo
omega_R = 0.000000 : omega angle of the right photo
phie_R =  0.000000 : phie  angle of the right photo
kapa_R =  0.000000 : kapa  angle of the right photo

Computation converges if max. |X| is less than 0.00000001
```

這邊將輸入資料做一個小整理並輸出，其中將一開始的角度值從radians轉成degree，以及將輸入閥值也顯示出來。

(3)迭代成果

避免截圖太小，使用複製貼上

Iteration Results: (units: radians)

Iteration 1: d_phie(L)=-0.009663755, d_kapa(L)=-0.037222807, d_omega(R)=-0.009922448, d_phie(R)=-0.008423786, d_kapa(R)=-0.003140444

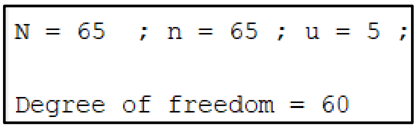
Iteration 2: d_phie(L)=-0.002111431, d_kapa(L)=0.000946962 , d_omega(R)=0.000335599 , d_phie(R)=-0.001616201, d_kapa(R)=0.000719813

Iteration 3: d_phie(L)=0.000001624 , d_kapa(L)=-0.000002503, d_omega(R)=-0.000000726, d_phie(R)=0.000002229 , d_kapa(R)=-0.000001197

Iteration 4: d_phie(L)=0.000000000 , d_kapa(L)=0.000000000 , d_omega(R)=0.000000000 , d_phie(R)=-0.000000000, d_kapa(R)=0.000000000

總共迭代了4次，數值都是由大變小，顯示結果很理想，在第三次迭代時，數值已經全部都在0.00001以下，已經非常小了，再過一次迭代已經全部達到標準:10^-8以下，從結果來看，就算使用10^-7或10^-9當成標準也會在第四次迭代時停止。

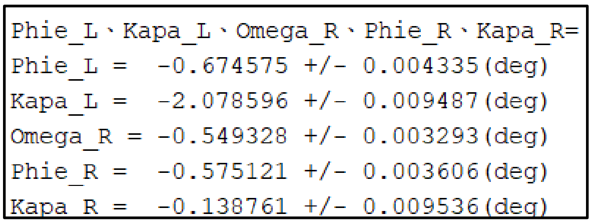
(4)自由度



```
N = 65  ; n = 65 ; u = 5 ;

Degree of freedom = 60
```

N為像點，共65個；n為觀測值，一組像點一組觀測值，一樣共65個；接下來是未知數phie_L、kapa_L、omega_R、phie_R、kapa_R，共五個角度。因此自由度計算:65-5為60。

(5)未知數計算成果

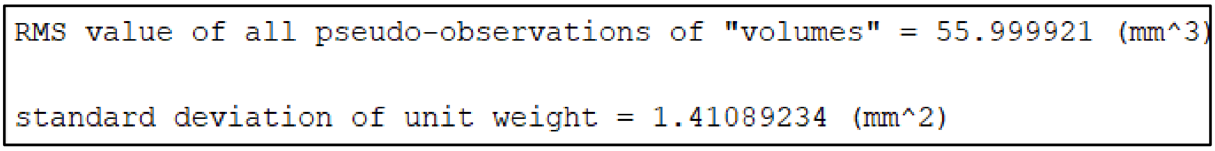


```
Phie_L、Kapa_L、Omega_R、Phie_R、Kapa_R=
Phie_L =   -0.674575 +/- 0.004335(deg)
Kapa_L =   -2.078596 +/- 0.009487(deg)
Omega_R = -0.549328 +/- 0.003293(deg)
Phie_R =   -0.575121 +/- 0.003606(deg)
Kapa_R =   -0.138761 +/- 0.009536(deg)
```

上圖的計算成果分別是Phie_L的-0.674575(deg)、Kapa_L的-2.078596(deg)、Omega_R的-0.549328(deg)、Phie_R的-0.575121(deg)、Kapa_R的-0.138761(deg)，除了Kapa_L距離原來的初始值0.0有點遠，其他的角度都在-1~1(deg)之間，看下來整體數據滿準的。另外，中誤差都非常小，小於0.01(deg)，是十分準確的結果，中誤差以Kapa_L和Kapa_R最高，但也只是0.0095左右，整體數據準確，以Kapa的結果誤差最大。

(6)F的RMS、單位權

報告排版問題放在這邊，原位置在最後面

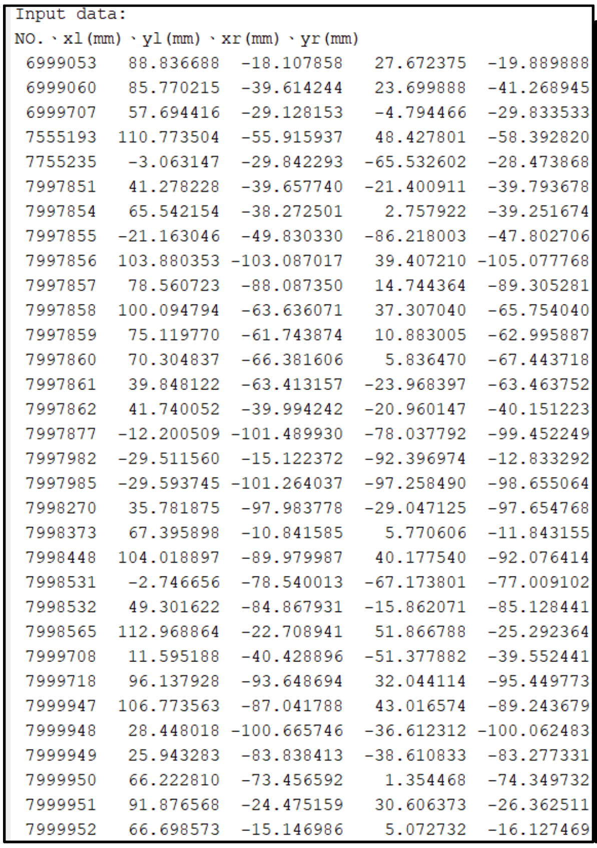


```
RMS value of all pseudo-observations of "volumes" = 55.999921 (mm^3)

standard deviation of unit weight = 1.41089234 (mm^2)
```

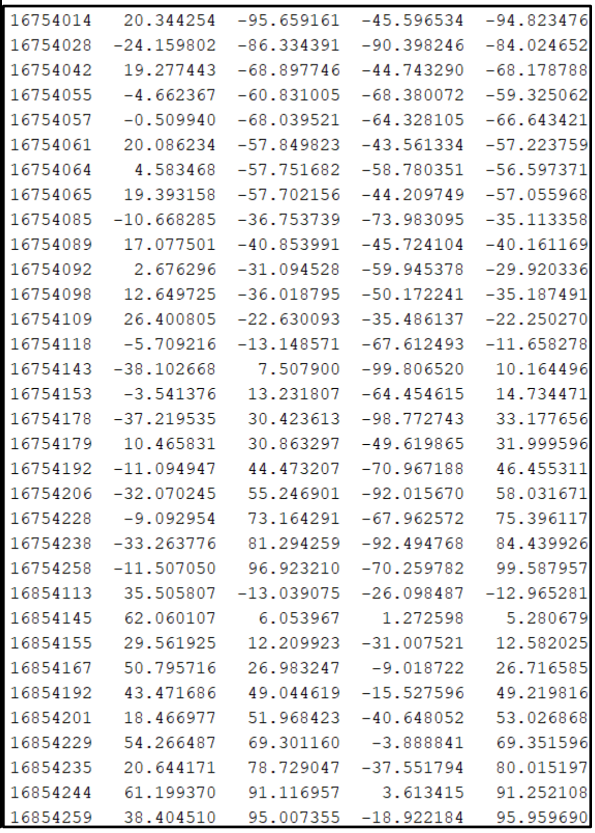
RMS=55.999921(mm^3)，為根據F*B的體積的計算值得出的均方根，可以用來評估模型的準確度。接著是單位權中誤差1.41089234(mm^2)，用F計算而成，可以乘3倍，做觀測值的剔錯(後面會提到)。單位權中誤差在觀測值最大有到三位數，可以計算到1.4，是非常準確的。

(7)輸入data

就是10167xy的同名點整理，所以截的有點小



```
Input data:
NO.、xl(mm)、yl(mm)、xr(mm)、yr(mm)
 6999053   88.836688  -18.107858   27.672375  -19.889888
 6999060   85.770215  -39.614244   23.699888  -41.268945
 6999707   57.694416  -29.128153   -4.794466  -29.833533
 7555193  110.773504  -55.915937   48.427801  -58.392820
 7755235   -3.063147  -29.842293  -65.532602  -28.473868
 7997851   41.278228  -39.657740  -21.400911  -39.793678
 7997854   65.542154  -38.272501    2.757922  -39.251674
 7997855  -21.163046  -49.830330  -86.218003  -47.802706
 7997856  103.880353 -103.087017   39.407210 -105.077768
 7997857   78.560723  -88.087350   14.744364  -89.305281
 7997858  100.094794  -63.636071   37.307040  -65.754040
 7997859   75.119770  -61.743874   10.883005  -62.995887
 7997860   70.304837  -66.381606    5.836470  -67.443718
 7997861   39.848122  -63.413157  -23.968397  -63.463752
 7997862   41.740052  -39.994242  -20.960147  -40.151223
 7997877  -12.200509 -101.489930  -78.037792  -99.452249
 7997982  -29.511560  -15.122372  -92.396974  -12.833292
 7997985  -29.593745 -101.264037  -97.258490  -98.655064
 7998270   35.781875  -97.983778  -29.047125  -97.654768
 7998373   67.395898  -10.841585    5.770606  -11.843155
 7998448  104.018897  -89.979987   40.177540  -92.076414
 7998531   -2.746656  -78.540013  -67.173801  -77.009102
 7998532   49.301622  -84.867931  -15.862071  -85.128441
 7998565  112.968864  -22.708941   51.866788  -25.292364
 7999708   11.595188  -40.428896  -51.377882  -39.552441
 7999718   96.137928  -93.648694   32.044114  -95.449773
 7999947  106.773563  -87.041788   43.016574  -89.243679
 7999948   28.448018 -100.665746  -36.612312 -100.062483
 7999949   25.943283  -83.838413  -38.610833  -83.277331
 7999950   66.222810  -73.456592    1.354468  -74.349732
 7999951   91.876568  -24.475159   30.606373  -26.362511
 7999952   66.698573  -15.146986    5.072732  -16.127469
```

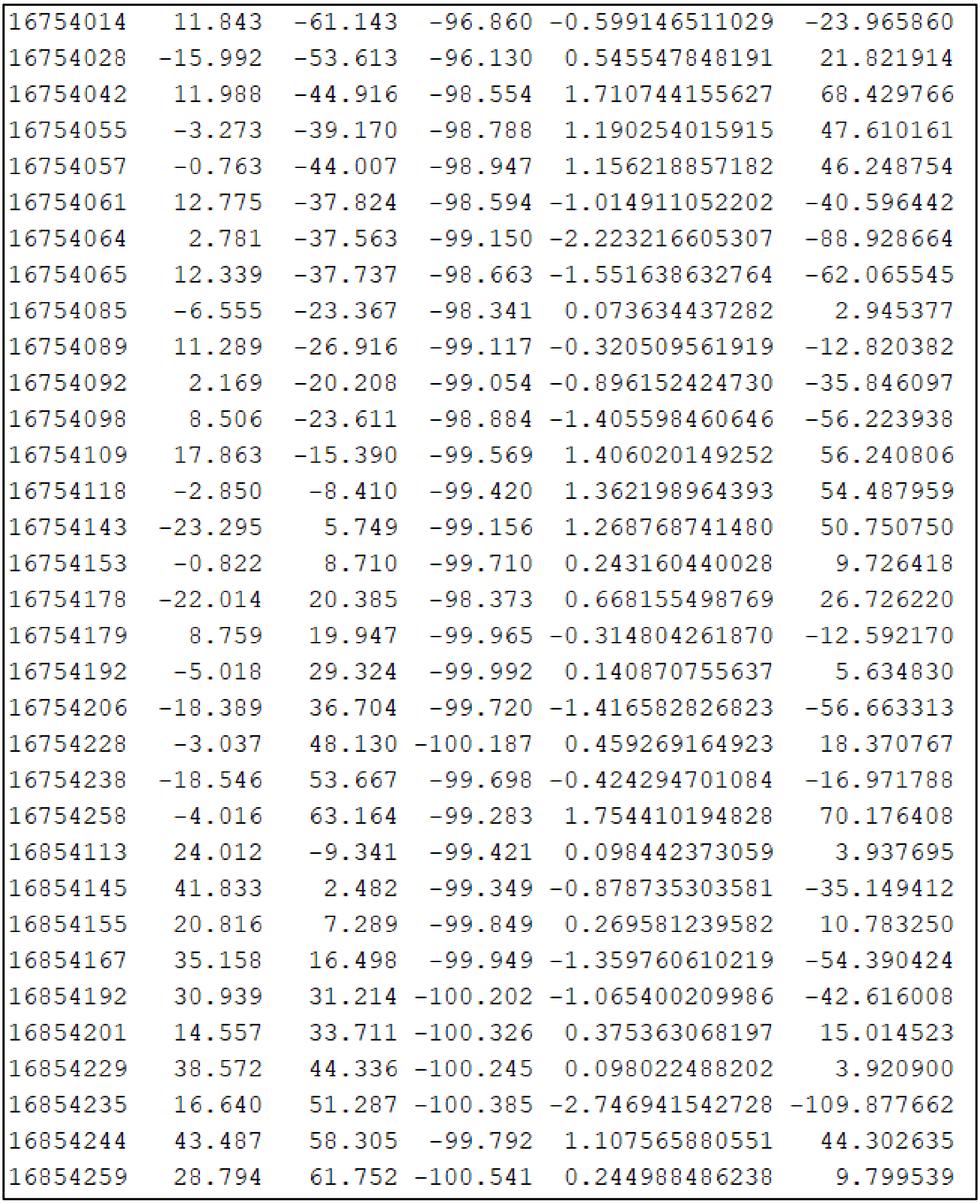


```
16754014   20.344254  -95.659161  -45.596534  -94.823476
16754028  -24.159802  -86.334391  -90.398246  -84.024652
16754042   19.277443  -68.897746  -44.743290  -68.178788
16754055   -4.662367  -60.831005  -68.380072  -59.325062
16754057   -0.509940  -68.039521  -64.328105  -66.643421
16754061   20.086234  -57.849823  -43.561334  -57.223759
16754064    4.583468  -57.751682  -58.780351  -56.597371
16754065   19.393158  -57.702156  -44.209749  -57.055968
16754085  -10.668285  -36.753739  -73.983095  -35.113358
16754089   17.077501  -40.853991  -45.724104  -40.161169
16754092    2.676296  -31.094528  -59.945378  -29.920336
16754098   12.649725  -36.018795  -50.172241  -35.187491
16754109   26.400805  -22.630093  -35.486137  -22.250270
16754118   -5.709216  -13.148571  -67.612493  -11.658278
16754143  -38.102668    7.507900  -99.806520   10.164496
16754153   -3.541376   13.231807  -64.454615   14.734471
16754178  -37.219535   30.423613  -98.772743   33.177656
16754179   10.465831   30.863297  -49.619865   31.999596
16754192  -11.094947   44.473207  -70.967188   46.455311
16754206  -32.070245   55.246901  -92.015670   58.031671
16754228   -9.092954   73.164291  -67.962572   75.396117
16754238  -33.263776   81.294259  -92.494768   84.439926
16754258  -11.507050   96.923210  -70.259782   99.587957
16854113   35.505807  -13.039075  -26.098487  -12.965281
16854145   62.060107    6.053967    1.272598    5.280679
16854155   29.561925   12.209923  -31.007521   12.582025
16854167   50.795716   26.983247   -9.018722   26.716585
16854192   43.471686   49.044619  -15.527596   49.219816
16854201   18.466977   51.968423  -40.648052   53.026868
16854229   54.266487   69.301160   -3.888841   69.351596
16854235   20.644171   78.729047  -37.551794   80.015197
16854244   61.199370   91.116957    3.613415   91.252108
16854259   38.404510   95.007355  -18.922184   95.959690
```

輸入值是從第一組處理資料的程式(python)得出，從左到右分別是:點號、左片x(mm)、左片y(mm)、右片x(mm)、右片y(mm)。

(8)UVW、F、vol

```
前四項:點號、U、V、W = model coordinates (units: mm)
第五項:F (units: mm^2)；第六項:改正數(F*B) (units: mm^3)
 6999053   59.090  -14.010  -99.740 -3.174845640279 -126.993826
 6999060   56.443  -28.001  -99.558  1.796772102305   71.870884
 6999707   37.846  -20.222  -98.603  0.424722409185   16.988896
 7555193   72.770  -39.458  -99.821  3.042874850182  121.714994
 7755235   -1.523  -19.300  -99.290 -1.323019159055  -52.920766
 7997851   27.049  -26.736  -99.030  0.397456429258   15.898257
 7997854   42.714  -26.329  -98.546 -0.580930932410  -23.237237
 7997855  -13.313  -30.853  -96.327  1.303923396567   52.156936
 7997856   66.666  -69.887  -99.234 -0.369388262166  -14.775530
 7997857   50.535  -59.560  -99.565  1.043735275645   41.749411
 7997858   65.393  -44.175  -99.658 -2.264449206856  -90.577968
 7997859   47.858  -41.318  -97.447  0.705102427546   28.204097
 7997860   44.614  -44.129  -97.375  1.533692345585   61.347694
 7997861   25.410  -41.889  -98.470 -3.455885244506 -138.235410
 7997862   27.336  -26.961  -99.010 -0.443896212899  -17.755849
 7997877   -8.957  -64.268  -97.371  2.143591842165   85.743674
 7997982  -18.107   -9.004  -98.205  0.051102544557    2.044102
 7997985  -19.429  -61.863  -94.657 -0.116684767872   -4.667391
 7998270   22.008  -64.220  -98.663 -0.602634472627  -24.105379
 7998373   44.713   -8.636  -98.864  1.733931915164   69.357277
 7998448   67.239  -61.474  -99.482 -0.115497869212   -4.619915
 7998531   -2.444  -50.503  -98.491  1.150060200485   46.002408
 7998532   30.697  -55.397  -97.401 -2.664318283454 -106.572731
 7998565   75.010  -17.650  -99.797  1.844835728018   73.793429
 7999708    7.718  -26.434  -98.870 -0.827575551042  -33.103022
 7999718   61.853  -63.558  -99.334 -0.831038341903  -33.241534
 7999947   69.092  -59.594  -99.428  1.530625969302   61.225039
 7999948   17.156  -65.607  -98.455 -1.032148628210  -41.285945
 7999949   15.926  -54.666  -98.421 -0.542484083129  -21.699363
 7999950   41.730  -48.440  -97.160  0.812181048117   32.487242
 7999951   61.047  -18.298  -99.903 -2.012160218332  -80.486409
 7999952   44.265  -11.445  -99.111  2.791018199339  111.640728
```



```
16754014   11.843  -61.143  -96.860 -0.599146511029  -23.965860
16754028  -15.992  -53.613  -96.130  0.545547848191   21.821914
16754042   11.988  -44.916  -98.554  1.710744155627   68.429766
16754055   -3.273  -39.170  -98.788  1.190254015915   47.610161
16754057   -0.763  -44.007  -98.947  1.156218857182   46.248754
16754061   12.775  -37.824  -98.594 -1.014911052202  -40.596442
16754064    2.781  -37.563  -99.150 -2.223216605307  -88.928664
16754065   12.339  -37.737  -98.663 -1.551638632764  -62.065545
16754085   -6.555  -23.367  -98.341  0.073634437282    2.945377
16754089   11.289  -26.916  -99.117 -0.320509561919  -12.820382
16754092    2.169  -20.208  -99.054 -0.896152424730  -35.846097
16754098    8.506  -23.611  -98.884 -1.405598460646  -56.223938
16754109   17.863  -15.390  -99.569  1.406020149252   56.240806
16754118   -2.850   -8.410  -99.420  1.362198964393   54.487959
16754143  -23.295    5.749  -99.156  1.268768741480   50.750750
16754153   -0.822    8.710  -99.710  0.243160440028    9.726418
16754178  -22.014   20.385  -98.373  0.668155498769   26.726220
16754179    8.759   19.947  -99.965 -0.314804261870  -12.592170
16754192   -5.018   29.324  -99.992  0.140870755637    5.634830
16754206  -18.389   36.704  -99.720 -1.416582826823  -56.663313
16754228   -3.037   48.130 -100.187  0.459269164923   18.370767
16754238  -18.546   53.667  -99.698 -0.424294701084  -16.971788
16754258   -4.016   63.164  -99.283  1.754410194828   70.176408
16854113   24.012   -9.341  -99.421  0.098442373059    3.937695
16854145   41.833    2.482  -99.349 -0.878735303581  -35.149412
16854155   20.816    7.289  -99.849  0.269581239582   10.783250
16854167   35.158   16.498  -99.949 -1.359760610219  -54.390424
16854192   30.939   31.214 -100.202 -1.065400209986  -42.616008
16854201   14.557   33.711 -100.326  0.375363068197   15.014523
16854229   38.572   44.336 -100.245  0.098022488202    3.920900
16854235   16.640   51.287 -100.385 -2.746941542728 -109.877662
16854244   43.487   58.305  -99.792  1.107565880551   44.302635
16854259   28.794   61.752 -100.541  0.244988486238    9.799539
```

如圖，由左到右分別是點號、U(mm)、V(mm)、W(mm)、F(mm^2)、vol(mm^3)，F為用F=vl*wr-vr*wl公式計算得出、而vol則是透過此公式:vol=F*B計算得出。U、V 、W分別表示模型中三維空間點的X、Y、Z坐標，而F則是表示兩個射線向量的偏差，越小代表兩張影像的相對定位越理想，我計算出來的F在6999053、7997681達到3(mm^2)以上，表示這兩個點，相較其他點都在+/-3(mm^2)之間，稍微不準一些，但由於上述算出來的單位權標準差是1.41089234，用三倍單位權標準差當偵錯門檻為4.23267702，上述

的F值都沒有超過4，所以沒有做剔錯。接下來是F*B基線長為vol/改正數，由基線向量和兩個同名射線定義的體積，可以用來修正模型座標。

(2)數據分析及比較

1.五個旋轉角的值及其中誤差

	matlab(deg)	fortran(deg)	中誤差matlab(deg)	中誤差fortran(deg)
Phie_L	-0.674575	0.674575	0.004335	0.004335
Kapa_L	-2.078596	-2.078596	0.009487	0.009487
Omega_R	-0.549328	-0.549300	0.003293	0.003293
Phie_R	-0.575121	0.575148	0.003606	0.003606
Kapa_R	-0.138761	-0.133246	0.009536	0.009500

從上表格可得五個角度在我自己撰寫的matlab和老師寫的fortran上，幾乎都是一樣的，但是值得探討的是，雖然就值而言是一樣的，但Phie_L和Phie_R兩個角度，在fortran是正的，在matlab是負的，若老師的程式碼是正確無誤，那就表示我的程式碼在計算Phie角度的時候，可能有些沒考慮到的部分。這個地方我有詢問其他同學做出來的結果，但每位同學(有使用matlab、python的)，都得出了Phie角是負值的結果，所以檢查程式碼和算式無誤後，暫且沒有想到其他解決方法或是誤差可能。

2.UVW最大最小值

	matlab(mm)	fortran(mm)
Umax	75.010	75.024
Umin	-23.295	-23.312
Vmax	63.164	63.168
Vmin	-69.887	-69.887

Wmax	-94.657	-94.657
Wmin	-100.541	-100.540

從上表格可得，在matlab(我的程式碼)的UVW的max和min以及在fortran(老師程式碼)比較，可以看出幾乎已經完全一樣了，在U值的max和min值相差較大，其他的最大最小值幾乎沒有差異。

3.F值(以前20筆數據為例)

點號	matlab(mm^2)	fortran(mm^2)
6999053	-3.174845640279	-3.174560546875
6999060	1.796772102305	1.796875000000
6999707	0.424722409185	0.424804687500
7555193	3.042874850182	3.041992187500
7755235	-1.323019159055	-1.322753906250
7997851	0.397456429258	0.397460937500
7997854	-0.580930932410	-0.581542968750
7997855	1.303923396567	1.303222656250
7997856	-0.369388262166	-0.368164062500
7997857	1.043735275645	1.043945312500
7997858	-2.264449206856	-2.264648437500
7997859	0.705102427546	0.704101562500
7997860	1.533692345585	1.534179687500

7997861	-3.455885244506	-3.456054687500
7997862	-0.443896212899	-0.444335937500
7997877	2.143591842165	2.143554687500
7997982	0.051102544557	0.051025390625
7997985	-0.116684767872	-0.117187500000
7998270	-0.602634472627	-0.602539062500
7998373	1.733931915164	1.734008789062

F的計算公式=> F = vl*wr - vr*wl，由左影像和右影像的視線向量計算出來，(ul,vl,wl)和(ur,vr,wr)分別是左影像和右影像的三維射線向量，通常可以用來評估兩個射線的偏差。數值整體看起來偏小，和老師的fortran檔比起來幾乎沒有差異，因此可以先排除計算錯誤的可能，表示這組測量成果，在左片和右片的兩張偏差很小，但在6999053、7555193這兩點可能需要注意一下，但還是小於3倍標準差，因此不用剔除。

4.RMS、單位權中誤差

	matlab	fortran
RMS(mm^3)	55.999921	55.996278
單位權中誤差(mm^2)	1.41089234	1.45706934

RMS值為體積的均方根，RMS值越小，表模型的相對定向精度越高。但因為乘上基線長(40mm)，所以看起來比較大一些，模型的整體精度還是很準。

接著是單位權中誤差。我自己計算出的單位權中誤差為1.41089234(mm^2)，所以三倍單位權中誤差(偵錯門檻)約等於4.2；老師的fortran檔計算結果為1.45706934(mm^2)，對照我自己的F值(mm^2)和老師的F值(mm^2)，都沒有要剔錯的數據，所以我的計算程式沒有針對這部分做偵錯，老師的也沒有剔除的數值。

十、回答第26頁的8個問題

1.共有65 個物點參與相對方位之計算。

2.有 65 個觀測值，有 5 個未知數，所以自由度 = 60 。 其中，每一種觀測值的名稱及其數量分別為何? 每一種未知數的名稱及其數量分別為何?

1.觀測值

xl、yl、xr、yr為一組，共65個

2.未知數

phie_L(左片phie角)、kapa_L(左片kapa角)、omega_R(左片omega角)、phie_R(右片phie角)、kapa_R(右片kapa角)->5個

3.精度最佳者為那一個RO元素?Omega_R 其後驗中誤差為+/- 0.003293 deg。

4.另外的4個姿態角的後驗中誤差為前者(精度最佳者)的幾倍?

Phie_L->0.004335/0.003293=1.316倍；Kapa_L->0.009487/0.003293=2.881倍；Phie_R->0.003606/0.003293=1.095倍；Kapa_R->0.009536/0.003293=2.885倍。

5.全部物點的模型坐標(U, V, W)之值域: U : Umin ~ Umax = -23.295~75.010 (單位:mm) V : Vmin ~ Vmax =-69.887~63.164(單位: mm) W : Wmin ~ Wmax = -100.541~-94.657 (單位: mm)

6.全部點的虛擬體積觀測值之均方根值為 55.999921 (單位: mm^3)。

7.後驗單位權中誤差為 1.41089234 (單位:mm^2)。

8.前述的後驗單位權中誤差的意義為何?

後驗中誤差，顧名思義就是在進行一系列實際測量之後，計算出來的誤差。基於實際數據，反映測量過程中實際發生的誤差，可以用來評估測量過程的準確性。

十一、結語

這次的作業時間因為延長一周，所以時間較充裕，也剛好遇到了颱風天的放假，所以可以好好的完成作業。一開始我因為覺得程式和公式十分複雜，原本打算直接使用老師的程式碼，而先著手去安裝fortran相關套件，並讀懂fortran程式碼和重新進行了些微的更動，讓他變成自己的程式。但是後來發現時間足夠，又花了很多天研究matlab的程式碼，學習如何自己迭代及計算誤差，過程非常辛苦，但最後趕在作業繳交前幾天完成了，由於中間經過反覆的修改，所以也比較沒有多餘的時間善用matlab新穎的套件做繪圖等，這個部分有點可惜。

由於是最後才決定要自己撰寫程式碼，所以我的程式碼是分兩部分進行:一是用python整理資料，python是我比較熟悉的語言，也知道這個語言在處理資料不會有甚麼問題，所以先使用python做資料處理；二是使用matlab做計算，在以前大一大二的

課程，雖然沒有正式教過如何使用matlab，但課堂上一些平差或矩陣的計算，老師的作法都是用matalb操作，因此計算的部分我也選擇使用matlab；接下來是比較fortran的結果，原先我用老師的程式碼已經有跑完一些結果，也理解了大部分，因此我的成果分析著重於兩項成果的資料對比。

而針對這兩項成果的比較，我看到大部分的成果都非常相近，包含迭代、改正數及計算U、V、W等，兩者的相差都不太大，但會一些小數點的不同，我很納悶是否是兩者不同程式在運算時取的小數點位數不同，而導致的計算偏差；還是兩者程式碼邏輯不同，所以在運算時會有一些不一樣?

在這份作業，從處理觀測值到所有的計算，都是一項不簡單的任務，從來沒有自己嘗試，在沒有老師詳細的指導下，要完成一份大型的程式碼，這次的作業算是有真正感受到出去工作後的任務之大以及要注意的細節之多，是一個非常特別的體驗，也從中了解了雙像旋轉法，如何使用左片右片的觀測值，用共面式計算出模型點，在和其他同學討論中，也有看到其他同學畫出的左右片光束和模型點的對應三維圖，看起來十分好看，自己又往更理解攝影測量邁進一步了。

十二、參考文獻

1.MinGW-w64

https://www.mingw-w64.org/

2.ChatGPT

3.Fortran官網 快速入門教程

https://fortran-lang.org/zh_CN/learn/quickstart/

4.課本

5.https://news1.get.com.tw/Html/News/70756.pdf

6.Matlab簡易教學

https://hackmd.io/@FbUJsF5qTbyirb8qlvu2Fw/Sktejk7hc

附錄、程式碼

1.資料處理(python)

```
###第一步驟:讀取檔案、分類內容###
def read_file(file_path):
    data = {}
    with open(file_path, ’r’) as f:
        for line in f:
            line = line.strip()
            if line == '-99':  # 停止讀取的標誌
                break
            parts = line.split()
            if len(parts) >= 3:
                n = int(parts[0])  # 第一個欄位是 n
                x = float(parts[1])  # 第二個欄位是 x
                y = float(parts[2])  # 第三個欄位是 y
                data[n] = (x, y)
    return data
###第二步驟:合併同名像點###
def merge_data(file1, file2, output_file):
    data1 = read_file(file1)
    data2 = read_file(file2)
    with open(output_file, 'w') as f_out:
        for n in sorted(data1.keys() & data2.keys()):  # 找到兩個檔案中都有
的 n
            x1, y1 = data1[n]
            x2, y2 = data2[n]
            f_out.write(f"{n:10} {x1*0.001:15.6f} {y1*0.001:15.6f} {x2*0.00
1:15.6f} {y2*0.001:15.6f}\n")
###第三步驟:運行###
# 檔案路徑
file1 = 'data1.txt'
file2 = 'data2.txt'
output_file = 'output.txt'
# 執行合併
merge_data(file1, file2, output_file)
```

2.計算(matlab)

```
% 讀取數據
data_read = readmatrix('output.txt'); % 將 'your_file.txt' 替換為你的檔案名
% 將數據存入表格
data = array2table(data_read, 'VariableNames', {'no', 'x1', 'y1', 'x2', 'y2'});
%輸入基本資料
phie_L = input(' > initial value of phie_L  = ? deg. (e.g. 0.0): ');
kapa_L = input(' > initial value of kapa_L  = ? deg. (e.g. 0.0): ');
omega_R = input(' > initial value of omega_R = ? deg. (e.g. 0.0): ');
phie_R = input(' > initial value of phie_R = ? deg. (e.g. 0.0): ');
kapa_R = input(' > initial value of kapa_R  = ? deg. (e.g. 0.0): ');
B = input(' > A reference baseline B = ? mm. (e.g. 40)');
threshold = input(' > Threshold value adopted in convergence condition = ? (e.g. 0.00000001): ');
f = input('Principal distance = ? mm. (e.g. 152.818)');
%自由度
n=height(data);
u=5;
freedom=n-u;
% 顯示初始值
fprintf('\nInitial values of 5 R.O. parameters (units: degree):\n');
fprintf('phie_L =  %.6f : phie  angle of the left  photo\n', phie_L);
fprintf('kapa_L =  %.6f : kapa  angle of the left  photo\n', kapa_L);
fprintf('omega_R = %.6f : omega angle of the right photo\n', omega_R);
fprintf('phie_R =  %.6f : phie  angle of the right photo\n', phie_R);
fprintf('kapa_R =  %.6f : kapa  angle of the right photo\n', kapa_R);
% 迭代過程
fprintf('\nComputation converges if max. |X| is less than %.8f\n\n', threshold);
fprintf('Iteration Results: (units: radians)\n');
[phie_L, kapa_L, omega_R, phie_R, kapa_R, sum,std_errors] = deda(data, phie_L, kapa_L, omega_R, phie_R, kapa_R, f, B, threshold, freedom);
%自由度
fprintf('\nN = %.i  ; n = %.i ; u = %.i ;\n', height(data),n,u);
fprintf('\nDegree of freedom = %.i\n', freedom);
fprintf('\nPhie_L、Kapa_L、Omega_R、Phie_R、Kapa_R= \n');
% 以度數輸出結果
fprintf('Phie_L =  %.6f +/- %.6f(deg)\n', rad2deg(phie_L), rad2deg(std_errors(1)));
fprintf('Kapa_L =  %.6f +/- %.6f(deg)\n', rad2deg(kapa_L), rad2deg(std_errors(2)));
fprintf('Omega_R = %.6f +/- %.6f(deg)\n', rad2deg(omega_R), rad2deg(std_errors(3)));
fprintf('Phie_R =  %.6f +/- %.6f(deg)\n', rad2deg(phie_R), rad2deg(std_errors(4)));
fprintf('Kapa_R =  %.6f +/- %.6f(deg)\n\n', rad2deg(kapa_R), rad2deg(std_errors(5)));
%輸出data值
fprintf('Input data:\nNO.、xl(mm)、yl(mm)、xr(mm)、yr(mm)\n');
% 遍歷每一行，並格式化輸出
for i = 1:height(data)
    fprintf('%8.0f %11.6f %11.6f %11.6f %11.6f\n', data.no(i), data.x1(i), data.y1(i), data.x2(i), data.y2(i));
end
%fprintf('Correction value:\nNO.、xl correction(mm)、yl correction(mm)、xr correction(mm)、yr correction(mm)\n');
%for i = 1:height(change)
    %fprintf('%8.0f %11.6f %11.6f %11.6f %11.6f\n', data.no(i),change(1,i), change(2,i), change(3,i), change(4,i));
```

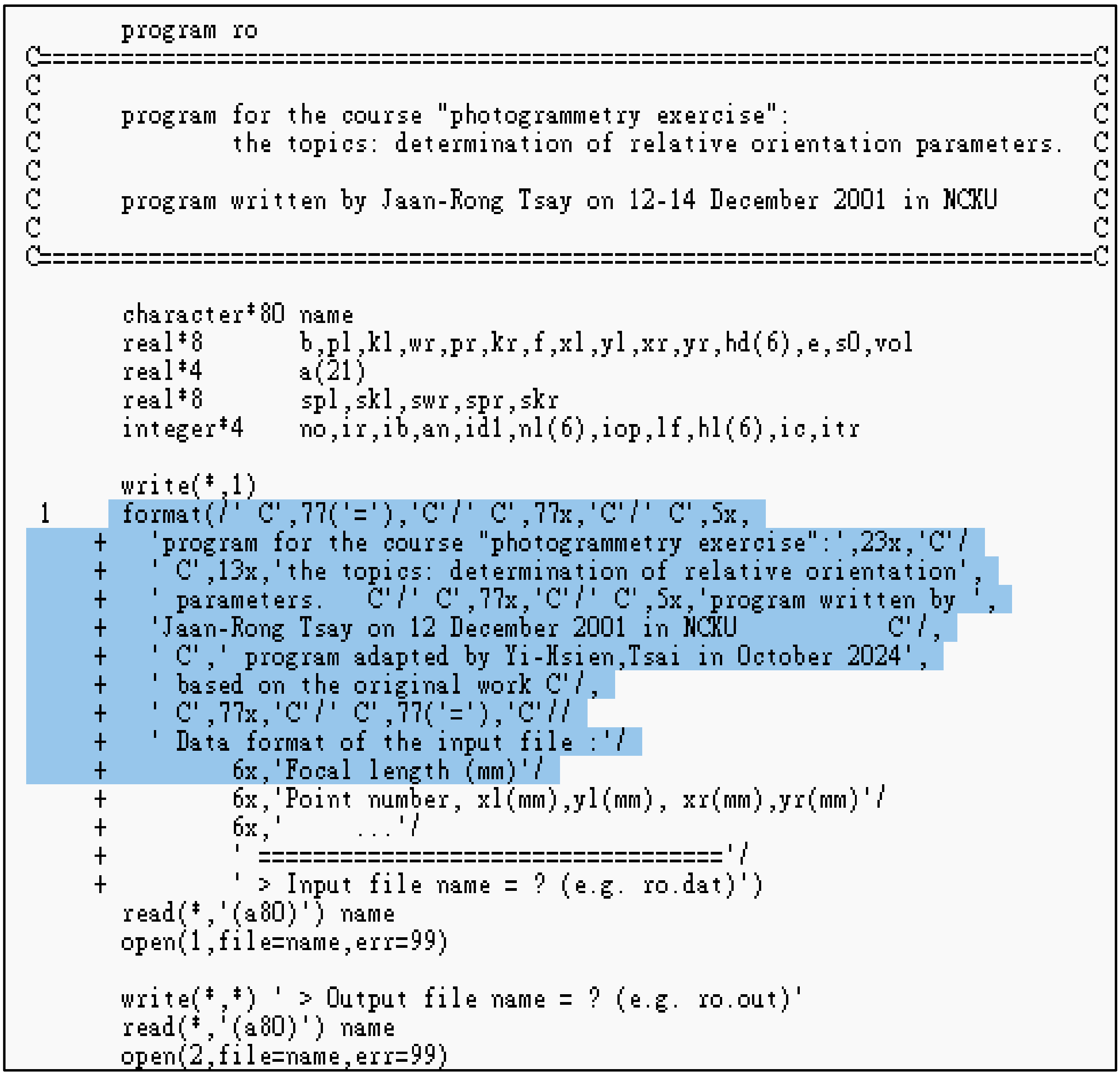
```
%end
fprintf('\n前四項:點號、U、V、W = model coordinates (units: mm)\n');
fprintf('第五項:F (units: mm^2)；第六項:改正數(F*B) (units: mm^3)\n');
% 輸出 uvw，並顯示到小數點後三位
[uvw,rms,std_fmm] = calculate_uvw(data, phie_L, kapa_L, omega_R, phie_R, kapa_R, f, B );
for i = 1:size(uvw, 1)
    fprintf('%8.0f %8.3f %8.3f %8.3f %15.12f %11.6f\n', uvw(i, 1), uvw(i, 2), uvw(i, 3), uvw(i, 4),uvw(i, 5),uvw(i,6));
end
%其他值
fprintf('\nRMS value of all pseudo-observations of "volumes" = %.6f (mm^3)\n', rms);
fprintf('\nstandard deviation of unit weight = %.8f (mm^2) \n', std_fmm);
% 旋轉矩陣
function R_M = Rot_M(omega, phie, kapa)
    m11 = cos(phie)*cos(kapa);
    m12 = sin(omega)*sin(phie)*cos(kapa) + cos(omega)*sin(kapa);
    m13 = -cos(omega)*sin(phie)*cos(kapa) + sin(omega)*sin(kapa);
    m21 = -cos(phie)*sin(kapa);
    m22 = -sin(omega)*sin(phie)*sin(kapa) + cos(omega)*cos(kapa);
    m23 = cos(omega)*sin(phie)*sin(kapa) + sin(omega)*cos(kapa);
    m31 = sin(phie);
    m32 = -sin(omega)*cos(phie);
    m33 = cos(omega)*cos(phie);
    R_M = [m11, m12, m13; m21, m22, m23; m31, m32, m33];
end
%計算detla
function [V1, V2, V3, det_val] = Detla(x1, y1, x2, y2, f, B, p_1, k_1, o_2, p_2, k_2)
```

```
    M1 = Rot_M(0, p_1, k_1)’;
    M2 = Rot_M(o_2, p_2, k_2)’;
    V1 = [B; 0; 0];
    p1 = [x1; y1; -f];
    p2 = [x2; y2; -f];
    V2 = M1 * p1;
    V3 = M2 * p2;
    det_val = dot(V1, cross(V2, V3));
end
% 計算 b2
function b2_value = b2(x1, y1, f, omega_1, phi_1, kappa_1, V1, V3)
    m21 = -x1*sin(phi_1)*cos(kappa_1) + y1*sin(phi_1)*sin(kappa_1) - f*cos(phi_1);
    m22 = x1*sin(omega_1)*cos(phi_1)*cos(kappa_1) - y1*sin(omega_1)*cos(phi_1)*sin(kappa_1) - f*sin(omega_1)*sin(phi_1);
    m23 = -x1*cos(omega_1)*cos(phi_1)*cos(kappa_1) + y1*cos(omega_1)*cos(phi_1)*sin(kappa_1) + f*cos(omega_1)*sin(phi_1);
    V2 = [m21; m22; m23];
    b2_value = dot(V1, cross(V2, V3));
end
% 計算 b3
function b3_value = b3(x1, y1, V1, V3, M1)
    m21 = x1*M1(2,1) - y1*M1(1,1);
    m22 = x1*M1(2,2) - y1*M1(1,2);
    m23 = x1*M1(2,3) - y1*M1(1,3);
    V2 = [m21; m22; m23];
    b3_value = dot(V1, cross(V2, V3));
end
% 計算 b7
function b7_value = b7(V1, V2, V3)
```

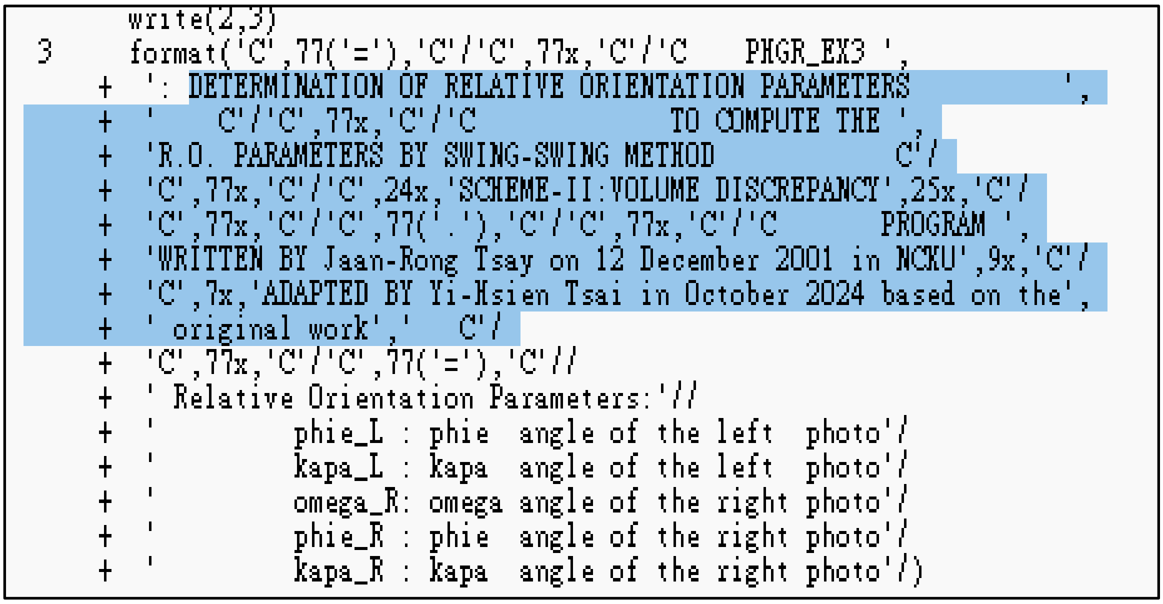
```
    V3 = [0; -V3(3); V3(2)];
    b7_value = dot(V1, cross(V2, V3));
end
% 計算 b8
function b8_value = b8(x2, y2, f, omega_2, phi_2, kappa_2, V1, V2)
    m31 = -x2*sin(phi_2)*cos(kappa_2) + y2*sin(phi_2)*sin(kappa_2) - f*cos(phi_2);
    m32 = x2*sin(omega_2)*cos(phi_2)*cos(kappa_2) - y2*sin(omega_2)*cos(phi_2)*sin(kappa_2) - f*sin(omega_2)*sin(phi_2);
    m33 = -x2*cos(omega_2)*cos(phi_2)*cos(kappa_2) + y2*cos(omega_2)*cos(phi_2)*sin(kappa_2) + f*cos(omega_2)*sin(phi_2);
    V3 = [m31; m32; m33];
    b8_value = dot(V1, cross(V2, V3));
end
% 計算 b9
function b9_value = b9(x2, y2, V1, V2, M2)
    m31 = x2*M2(2,1) - y2*M2(1,1);
    m32 = x2*M2(2,2) - y2*M2(1,2);
    m33 = x2*M2(2,3) - y2*M2(1,3);
    V3 = [m31; m32; m33];
    b9_value = dot(V1, cross(V2, V3));
end
% 構建設計矩陣的行
function row = A_r(x1, y1, x2, y2, f, p_1, k_1, o_2, p_2, k_2, V1, V2, V3, M1, M2)
    b2_val = b2(x1, y1, f, 0, p_1, k_1, V1, V3);
    b3_val = b3(x1, y1, V1, V3, M1);
    b7_val = b7(V1, V2, V3);
    b8_val = b8(x2, y2, f, o_2, p_2, k_2, V1, V2);
    b9_val = b9(x2, y2, V1, V2, M2);
    row = [b2_val, b3_val, b7_val, b8_val, b9_val];
end
% 構建設計矩陣 A 和偏差向量 h0
function [A, h0] = cmatrix(data, p_1, k_1, o_2, p_2, k_2, f, B)
    A = [];
    h0 = [];
    for i = 1:height(data)
        x1 = data.x1(i);
        y1 = data.y1(i);
        x2 = data.x2(i);
        y2 = data.y2(i);
        M1 = Rot_M(0, p_1, k_1);
        M2 = Rot_M(o_2, p_2, k_2);
        [V1, V2, V3, det_val] = Detla(x1, y1, x2, y2, f, B, p_1, k_1, o_2,
p_2, k_2);
        arow = A_r(x1, y1, x2, y2, f, p_1, k_1, o_2, p_2, k_2, V1, V2, V3,
M1, M2);
        A = [A; arow];
        h0 = [h0; -det_val];
    end
end
% 計算 delta 值
function [del_phi_1, del_kappa_1, del_omega_2, del_phi_2, del_kappa_2, sum]
= new(A, h0)
    delta = A \ h0; % 解最小二乘問題
    del_phi_1 = delta(1);
    del_kappa_1 = delta(2);
    del_omega_2 = delta(3);
    del_phi_2 = delta(4);
    del_kappa_2 = delta(5);
    sum = del_phi_1^2 + del_kappa_1^2 + del_omega_2^2 + del_phi_2^2 + del_kappa_2^2;
end
% 迭代過程
function [p_1, k_1, o_2, p_2, k_2, sum,std_errors] = deda(data, p_1, k_1, o_2, p_2, k_2, f, B, threshold,freedom)
    max = 20;
    while max > 0
        [A, h0] = cmatrix(data, p_1, k_1, o_2, p_2, k_2, f, B);
        [delta_phi_1, delta_kappa_1, delta_omega_2, delta_phi_2, delta_kappa_2, sum] = new(A, h0);
        p_1 = p_1 + delta_phi_1;
        k_1 = k_1 + delta_kappa_1;
        o_2 = o_2 + delta_omega_2;
        p_2 = p_2 + delta_phi_2;
        k_2 = k_2 + delta_kappa_2;
        deltax=[delta_phi_1,delta_kappa_1,delta_omega_2,delta_phi_2,delta_kappa_2]’;
        % 印每次迭代的信息
        fprintf(’Iteration %d: d_phie(L)=%-12.9f, d_kapa(L)=%-12.9f, d_omega(R)=%-12.9f, d_phie(R)=%-12.9f, d_kapa(R)=%-12.9f\n’, ...
            21 - max, delta_phi_1, delta_kappa_1, delta_omega_2, delta_phi_2, delta_kappa_2);
        if sum < threshold^2
            v=A*deltax-h0;
            sigma=(v’*v)/freedom;
            cov=sigma*(inv(A’*A));
            std_errors=sqrt(diag(cov));
            return;
        end
        max = max - 1;
    end
    v=A*deltax-h0;
    sigma=(v'*v)/freedom;
    cov=sigma*(inv(A'*A));
    std_errors=sqrt(diag(cov));
end
function [uvw,rms,std_fmm] = calculate_uvw(data, phie_L, kapa_L, omega_R, p
hie_R, kapa_R, f, B )
    % 初始化模型點矩陣
    squaresum=0;
    uvw = zeros(height(data), 6);
    for i = 1:height(data)
        x1 = data.x1(i);
        y1 = data.y1(i);
        x2 = data.x2(i);
        y2 = data.y2(i);
        no=data.no(i);
        % 計算方向向量
        V1 = [x1; y1; -f];
        V2 = [x2; y2; -f];
        M1 = Rot_M(0, phie_L, kapa_L)';
        M2 = Rot_M(omega_R, phie_R, kapa_R)';
        or1=M1*V1;
        or2=M2*V2;
        A = [or1,-or2];
        B_vec = [B; 0;0];
        ab = (A' * A) \ (A' * B_vec);
        a = ab(1);
        b = ab(2);
        % 計算 u, v, w
        uvw_vec1 = a * or1;
        uvw_vec2=b*or2;
        u = uvw_vec1(1);
        v = uvw_vec1(2);
        w = uvw_vec1(3);
        fmm=or1(2)*or2(3)-or2(2)*or1(3);
        vol=fmm*B;
        squaresum=squaresum+vol*vol;
        % 存入結果
        uvw(i, 1) = no;           % 存放點號
        uvw(i, 2:4) = [u,v,w];
        uvw(i,5)=fmm;
        uvw(i,6)=vol;
    end
    rms=sqrt(squaresum/height(data));
    std_fmm = std(uvw(:, 5));
end
```

3.修改老師的檔案、比較用的(fortran)

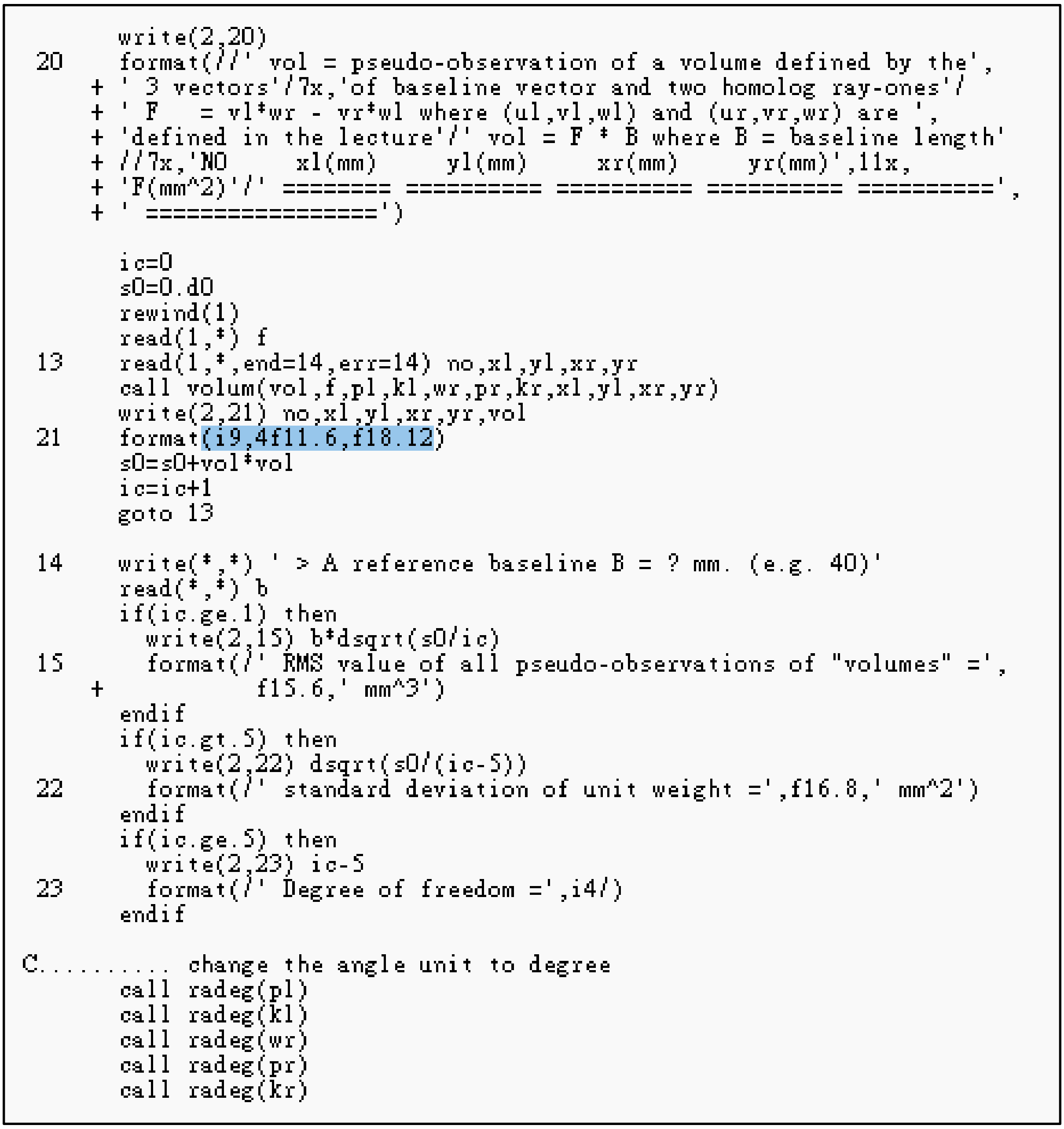
只放上修改部分，修改部分反白



```
      program ro
C=============================================================================C
C                                                                             C
C     program for the course "photogrammetry exercise":                       C
C             the topics: determination of relative orientation parameters.  C
C                                                                             C
C     program written by Jaan-Rong Tsay on 12-14 December 2001 in NCKU        C
C                                                                             C
C=============================================================================C

      character*80 name
      real*8       b,pl,kl,wr,pr,kr,f,xl,yl,xr,yr,hd(6),e,s0,vol
      real*4       a(21)
      real*8       spl,skl,swr,spr,skr
      integer*4    no,ir,ib,an,id1,nl(6),iop,lf,hl(6),ic,itr

      write(*,1)
 1    format(/' C',77('='),'C'/' C',77x,'C'/' C',5x,
     +   'program for the course "photogrammetry exercise":',23x,'C'/
     +   ' C',13x,'the topics: determination of relative orientation',
     +   ' parameters.   C'/' C',77x,'C'/' C',5x,'program written by ',
     +   'Jaan-Rong Tsay on 12 December 2001 in NCKU           C'/,
     +   ' C',' program adapted by Yi-Hsien,Tsai in October 2024',
     +   ' based on the original work C'/,
     +   ' C',77x,'C'/' C',77('='),'C'//
     +   ' Data format of the input file :'/
     +         6x,'Focal length (mm)'/
     +         6x,'Point number, xl(mm),yl(mm), xr(mm),yr(mm)'/
     +         6x,'      ...'/
     +         ' ==================================='/
     +         ' > Input file name = ? (e.g. ro.dat)')
      read(*,'(a80)') name
      open(1,file=name,err=99)

      write(*,*) ' > Output file name = ? (e.g. ro.out)'
      read(*,'(a80)') name
      open(2,file=name,err=99)
```



```
      write(2,3)
 3    format('C',77('='),'C'/'C',77x,'C'/'C    PHGR_EX3 ',
     +  ': DETERMINATION OF RELATIVE ORIENTATION PARAMETERS          ',
     +  '     C'/'C',77x,'C'/'C               TO COMPUTE THE ',
     +  'R.O. PARAMETERS BY SWING-SWING METHOD            C'/
     +  'C',77x,'C'/'C',24x,'SCHEME-II:VOLUME DISCREPANCY',25x,'C'/
     +  'C',77x,'C'/'C',77('.'),'C'/'C',77x,'C'/'C       PROGRAM ',
     +  'WRITTEN BY Jaan-Rong Tsay on 12 December 2001 in NCKU',9x,'C'/
     +  'C',7x,'ADAPTED BY Yi-Hsien Tsai in October 2024 based on the',
     +  ' original work','    C'/
     +  'C',77x,'C'/'C',77('='),'C'//
     +  ' Relative Orientation Parameters:'//
     +  '         phie_L : phie  angle of the left  photo'/
     +  '         kapa_L : kapa  angle of the left  photo'/
     +  '         omega_R: omega angle of the right photo'/
     +  '         phie_R : phie  angle of the right photo'/
     +  '         kapa_R : kapa  angle of the right photo'/)
```



```
      write(2,20)
 20   format(//' vol = pseudo-observation of a volume defined by the',
     + ' 3 vectors'/7x,'of baseline vector and two homolog ray-ones'/
     + ' F   = vl*wr - vr*wl where (ul,vl,wl) and (ur,vr,wr) are ',
     + 'defined in the lecture'/' vol = F * B where B = baseline length'
     + //7x,'NO      xl(mm)      yl(mm)      xr(mm)      yr(mm)',11x,
     + 'F(mm^2)'/' ======== ========== ========== ========== ==========',
     + ' ==================')

      ic=0
      s0=0.d0
      rewind(1)
      read(1,*) f
 13   read(1,*,end=14,err=14) no,xl,yl,xr,yr
      call volum(vol,f,pl,kl,wr,pr,kr,xl,yl,xr,yr)
      write(2,21) no,xl,yl,xr,yr,vol
 21   format(i9,4f11.6,f18.12)
      s0=s0+vol*vol
      ic=ic+1
      goto 13

 14   write(*,*) ' > A reference baseline B = ? mm. (e.g. 40)'
      read(*,*) b
      if(ic.ge.1) then
        write(2,15) b*dsqrt(s0/ic)
 15     format(/' RMS value of all pseudo-observations of "volumes" =',
     +          f15.6,' mm^3')
      endif
      if(ic.gt.5) then
        write(2,22) dsqrt(s0/(ic-5))
 22     format(/' standard deviation of unit weight =',f16.8,' mm^2')
      endif
      if(ic.ge.5) then
        write(2,23) ic-5
 23     format(/' Degree of freedom =',i4/)
      endif

C.........  change the angle unit to degree
      call radeg(pl)
      call radeg(kl)
      call radeg(wr)
      call radeg(pr)
      call radeg(kr)
```

報表、你的程式輸出報表

若需要的是報表.txt，在程式碼加入"fileID"就可以寫入成一個新的檔案，或是複製matlab的輸出資料至新的txt檔。

```
>> HW2_1

 > initial value of phie_L  = ? deg. (e.g. 0.0): 0.0

 > initial value of kapa_L  = ? deg. (e.g. 0.0): 0.0

 > initial value of omega_R = ? deg. (e.g. 0.0): 0.0

 > initial value of phie_R = ? deg. (e.g. 0.0): 0.0

 > initial value of kapa_R  = ? deg. (e.g. 0.0): 0.0
 > A reference baseline B = ? mm. (e.g. 40)40
 > Threshold value adopted in convergence condition = ? (e.g. 0.00000001):
0.00000001
Principal distance = ? mm. (e.g. 152.818)152.818
Initial values of 5 R.O. parameters (units: degree):
phie_L =  0.000000 : phie  angle of the left  photo
kapa_L =  0.000000 : kapa  angle of the left  photo
omega_R = 0.000000 : omega angle of the right photo
phie_R =  0.000000 : phie  angle of the right photo
kapa_R =  0.000000 : kapa  angle of the right photo
Computation converges if max. |X| is less than 0.00000001
Iteration Results: (units: radians)
Iteration 1: d_phie(L)=-0.009663755, d_kapa(L)=-0.037222807, d_omega(R)=-0.
009922448, d_phie(R)=-0.008423786, d_kapa(R)=-0.003140444
Iteration 2: d_phie(L)=-0.002111431, d_kapa(L)=0.000946962 , d_omega(R)=0.0
00335599 , d_phie(R)=-0.001616201, d_kapa(R)=0.000719813
Iteration 3: d_phie(L)=0.000001624 , d_kapa(L)=-0.000002503, d_omega(R)=-0.
000000726, d_phie(R)=0.000002229 , d_kapa(R)=-0.000001197
Iteration 4: d_phie(L)=0.000000000 , d_kapa(L)=0.000000000 , d_omega(R)=0.0
00000000 , d_phie(R)=-0.000000000, d_kapa(R)=0.000000000
N = 65  ; n = 65 ; u = 5 ;
Degree of freedom = 60
Phie_L、Kapa_L、Omega_R、Phie_R、Kapa_R=
Phie_L =  -0.674575 +/- 0.004335(deg)
Kapa_L =  -2.078596 +/- 0.009487(deg)
Omega_R = -0.549328 +/- 0.003293(deg)
Phie_R =  -0.575121 +/- 0.003606(deg)
Kapa_R =  -0.138761 +/- 0.009536(deg)
Input data:
NO.、xl(mm)、yl(mm)、xr(mm)、yr(mm)
```

6999053	88.836688	-18.107858	27.672375	-19.889888
6999060	85.770215	-39.614244	23.699888	-41.268945
6999707	57.694416	-29.128153	-4.794466	-29.833533
7555193	110.773504	-55.915937	48.427801	-58.392820
7755235	-3.063147	-29.842293	-65.532602	-28.473868
7997851	41.278228	-39.657740	-21.400911	-39.793678
7997854	65.542154	-38.272501	2.757922	-39.251674
7997855	-21.163046	-49.830330	-86.218003	-47.802706
7997856	103.880353	-103.087017	39.407210	-105.077768
7997857	78.560723	-88.087350	14.744364	-89.305281
7997858	100.094794	-63.636071	37.307040	-65.754040
7997859	75.119770	-61.743874	10.883005	-62.995887
7997860	70.304837	-66.381606	5.836470	-67.443718
7997861	39.848122	-63.413157	-23.968397	-63.463752
7997862	41.740052	-39.994242	-20.960147	-40.151223
7997877	-12.200509	-101.489930	-78.037792	-99.452249
7997982	-29.511560	-15.122372	-92.396974	-12.833292
7997985	-29.593745	-101.264037	-97.258490	-98.655064
7998270	35.781875	-97.983778	-29.047125	-97.654768
7998373	67.395898	-10.841585	5.770606	-11.843155
7998448	104.018897	-89.979987	40.177540	-92.076414
7998531	-2.746656	-78.540013	-67.173801	-77.009102
7998532	49.301622	-84.867931	-15.862071	-85.128441
7998565	112.968864	-22.708941	51.866788	-25.292364
7999708	11.595188	-40.428896	-51.377882	-39.552441
7999718	96.137928	-93.648694	32.044114	-95.449773
7999947	106.773563	-87.041788	43.016574	-89.243679
7999948	28.448018	-100.665746	-36.612312	-100.062483
7999949	25.943283	-83.838413	-38.610833	-83.277331
7999950	66.222810	-73.456592	1.354468	-74.349732

7999951	91.876568	-24.475159	30.606373	-26.362511
7999952	66.698573	-15.146986	5.072732	-16.127469
16754014	20.344254	-95.659161	-45.596534	-94.823476
16754028	-24.159802	-86.334391	-90.398246	-84.024652
16754042	19.277443	-68.897746	-44.743290	-68.178788
16754055	-4.662367	-60.831005	-68.380072	-59.325062
16754057	-0.509940	-68.039521	-64.328105	-66.643421
16754061	20.086234	-57.849823	-43.561334	-57.223759
16754064	4.583468	-57.751682	-58.780351	-56.597371
16754065	19.393158	-57.702156	-44.209749	-57.055968
16754085	-10.668285	-36.753739	-73.983095	-35.113358
16754089	17.077501	-40.853991	-45.724104	-40.161169
16754092	2.676296	-31.094528	-59.945378	-29.920336
16754098	12.649725	-36.018795	-50.172241	-35.187491
16754109	26.400805	-22.630093	-35.486137	-22.250270
16754118	-5.709216	-13.148571	-67.612493	-11.658278
16754143	-38.102668	7.507900	-99.806520	10.164496
16754153	-3.541376	13.231807	-64.454615	14.734471
16754178	-37.219535	30.423613	-98.772743	33.177656
16754179	10.465831	30.863297	-49.619865	31.999596
16754192	-11.094947	44.473207	-70.967188	46.455311
16754206	-32.070245	55.246901	-92.015670	58.031671
16754228	-9.092954	73.164291	-67.962572	75.396117
16754238	-33.263776	81.294259	-92.494768	84.439926
16754258	-11.507050	96.923210	-70.259782	99.587957
16854113	35.505807	-13.039075	-26.098487	-12.965281
16854145	62.060107	6.053967	1.272598	5.280679
16854155	29.561925	12.209923	-31.007521	12.582025
16854167	50.795716	26.983247	-9.018722	26.716585
16854192	43.471686	49.044619	-15.527596	49.219816

16854201	18.466977	51.968423	-40.648052	53.026868
16854229	54.266487	69.301160	-3.888841	69.351596
16854235	20.644171	78.729047	-37.551794	80.015197
16854244	61.199370	91.116957	3.613415	91.252108
16854259	38.404510	95.007355	-18.922184	95.959690

前四項:點號、U、V、W = model coordinates （units: mm）

第五項:F （units: mm^2）；第六項:改正數(F*B) （units: mm^3）

6999053	59.090	-14.010	-99.740	-3.174845640279	-126.993826
6999060	56.443	-28.001	-99.558	1.796772102305	71.870884
6999707	37.846	-20.222	-98.603	0.424722409185	16.988896
7555193	72.770	-39.458	-99.821	3.042874850182	121.714994
7755235	-1.523	-19.300	-99.290	-1.323019159055	-52.920766
7997851	27.049	-26.736	-99.030	0.397456429258	15.898257
7997854	42.714	-26.329	-98.546	-0.580930932410	-23.237237
7997855	-13.313	-30.853	-96.327	1.303923396567	52.156936
7997856	66.666	-69.887	-99.234	-0.369388262166	-14.775530
7997857	50.535	-59.560	-99.565	1.043735275645	41.749411
7997858	65.393	-44.175	-99.658	-2.264449206856	-90.577968
7997859	47.858	-41.318	-97.447	0.705102427546	28.204097
7997860	44.614	-44.129	-97.375	1.533692345585	61.347694
7997861	25.410	-41.889	-98.470	-3.455885244506	-138.235410
7997862	27.336	-26.961	-99.010	-0.443896212899	-17.755849
7997877	-8.957	-64.268	-97.371	2.143591842165	85.743674
7997982	-18.107	-9.004	-98.205	0.051102544557	2.044102
7997985	-19.429	-61.863	-94.657	-0.116684767872	-4.667391
7998270	22.008	-64.220	-98.663	-0.602634472627	-24.105379
7998373	44.713	-8.636	-98.864	1.733931915164	69.357277
7998448	67.239	-61.474	-99.482	-0.115497869212	-4.619915
7998531	-2.444	-50.503	-98.491	1.150060200485	46.002408
7998532	30.697	-55.397	-97.401	-2.664318283454	-106.572731

7998565	75.010	-17.650	-99.797	1.844835728018	73.793429
7999708	7.718	-26.434	-98.870	-0.827575551042	-33.103022
7999718	61.853	-63.558	-99.334	-0.831038341903	-33.241534
7999947	69.092	-59.594	-99.428	1.530625969302	61.225039
7999948	17.156	-65.607	-98.455	-1.032148628210	-41.285945
7999949	15.926	-54.666	-98.421	-0.542484083129	-21.699363
7999950	41.730	-48.440	-97.160	0.812181048117	32.487242
7999951	61.047	-18.298	-99.903	-2.012160218332	-80.486409
7999952	44.265	-11.445	-99.111	2.791018199339	111.640728
16754014	11.843	-61.143	-96.860	-0.599146511029	-23.965860
16754028	-15.992	-53.613	-96.130	0.545547848191	21.821914
16754042	11.988	-44.916	-98.554	1.710744155627	68.429766
16754055	-3.273	-39.170	-98.788	1.190254015915	47.610161
16754057	-0.763	-44.007	-98.947	1.156218857182	46.248754
16754061	12.775	-37.824	-98.594	-1.014911052202	-40.596442
16754064	2.781	-37.563	-99.150	-2.223216605307	-88.928664
16754065	12.339	-37.737	-98.663	-1.551638632764	-62.065545
16754085	-6.555	-23.367	-98.341	0.073634437282	2.945377
16754089	11.289	-26.916	-99.117	-0.320509561919	-12.820382
16754092	2.169	-20.208	-99.054	-0.896152424730	-35.846097
16754098	8.506	-23.611	-98.884	-1.405598460646	-56.223938
16754109	17.863	-15.390	-99.569	1.406020149252	56.240806
16754118	-2.850	-8.410	-99.420	1.362198964393	54.487959
16754143	-23.295	5.749	-99.156	1.268768741480	50.750750
16754153	-0.822	8.710	-99.710	0.243160440028	9.726418
16754178	-22.014	20.385	-98.373	0.668155498769	26.726220
16754179	8.759	19.947	-99.965	-0.314804261870	-12.592170
16754192	-5.018	29.324	-99.992	0.140870755637	5.634830
16754206	-18.389	36.704	-99.720	-1.416582826823	-56.663313
16754228	-3.037	48.130	-100.187	0.459269164923	18.370767

```
16754238  -18.546   53.667  -99.698 -0.424294701084  -16.971788
16754258   -4.016   63.164  -99.283  1.754410194828   70.176408
16854113   24.012   -9.341  -99.421  0.098442373059    3.937695
16854145   41.833    2.482  -99.349 -0.878735303581  -35.149412
16854155   20.816    7.289  -99.849  0.269581239582   10.783250
16854167   35.158   16.498  -99.949 -1.359760610219  -54.390424
16854192   30.939   31.214 -100.202 -1.065400209986  -42.616008
16854201   14.557   33.711 -100.326  0.375363068197   15.014523
16854229   38.572   44.336 -100.245  0.098022488202    3.920900
16854235   16.640   51.287 -100.385 -2.746941542728 -109.877662
16854244   43.487   58.305  -99.792  1.107565880551   44.302635
16854259   28.794   61.752 -100.541  0.244988486238    9.799539
RMS value of all pseudo-observations of "volumes" = 55.999921 (mm^3)
standard deviation of unit weight = 1.41089234 (mm^2)
```